TURING MACHINE

Turing machine was invented in 1936 by **Alan Turing**. It is an accepting device which accepts Recursive Enumerable Language generated by type 0 grammar.

There are various features of the Turing machine:

1. It has an external memory which remembers arbitrary long sequence of input.
2. It has unlimited memory capability.
3. The model has a facility by which the input at left or right on the tape can be read easily.
4. The machine can produce a certain output based on its input. Sometimes it may be required that the same input has to be used to generate the output. So in this machine, the distinction between input and output has been removed. Thus a common set of alphabets can be used for the Turing machine.

Formal definition of Turing machine

A Turing machine can be defined as a collection of 7 components:

Q: the finite set of states
∑: the finite set of input symbols
T: the tape symbol
q0: the initial state
F: a set of final states
B: a blank symbol used as a end marker for input
δ: a transition or mapping function.

The mapping function shows the mapping from states of finite automata and input symbol on the tape to the next states, external symbols and the direction for moving the tape head. This is known as a triple or a program for turing machine.

1. $(q0, a) \rightarrow (q1, A, R)$

That means in q0 state, if we read symbol 'a' then it will go to state q1, replaced a by X and move ahead right(R stands for right).

Example:

Construct TM for the language L ={0^n1^n} where n>=1.

Solution:

We have already solved this problem by PDA. In PDA, we have a stack to remember the previous symbol. The main advantage of the Turing machine is we have a tape head which can be moved forward or backward, and the input tape can be scanned.

The simple logic which we will apply is read out each '0' mark it by A and then move ahead along with the input tape and find out 1 convert it to B. Now, repeat this process for all a's and b's.

Now we will see how this turing machine work for 0011.

The simulation for 0011 can be shown as below:

0	0	1	1	Δ	-------

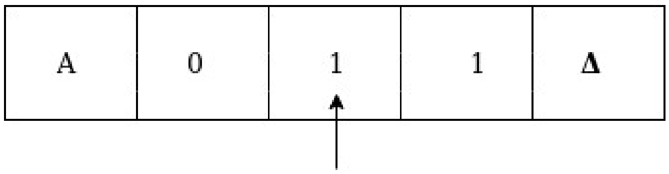
Now, we will see how this turing machine will works for 0011. Initially, state is q0 and head points to 0 as:

0	0	1	1	Δ

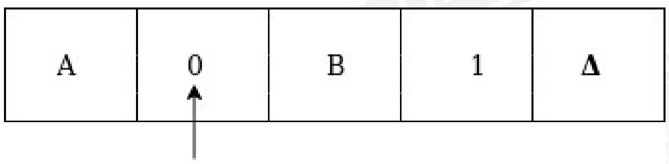
The move will be δ(q0, 0) = δ(q1, A, R) which means it will go to state q1, replaced 0 by A and head will move to the right as:

A	0	1	1	Δ

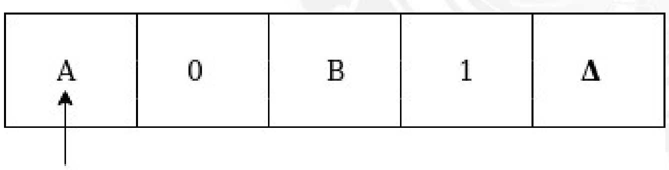
The move will be δ(q1, 0) = δ(q1, 0, R) which means it will not change any symbol, remain in the same state and move to the right as:



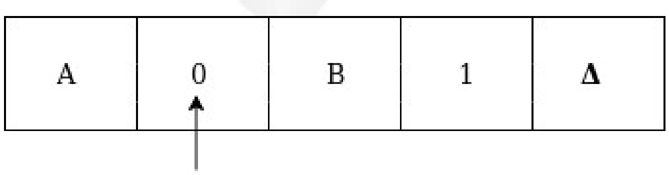
The move will be δ(q1, 1) = δ(q2, B, L) which means it will go to state q2, replaced 1 by B and head will move to left as:



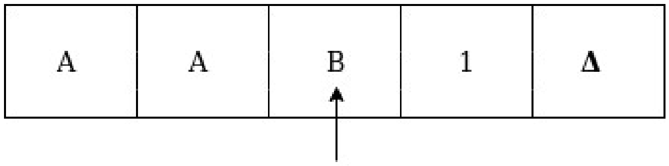
Now move will be δ(q2, 0) = δ(q2, 0, L) which means it will not change any symbol, remain in the same state and move to left as:



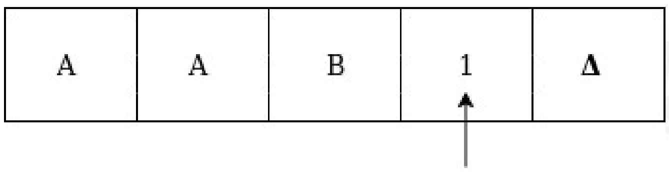
The move will be δ(q2, A) = δ(q0, A, R), it means will go to state q0, replaced A by A and head will move to the right as:



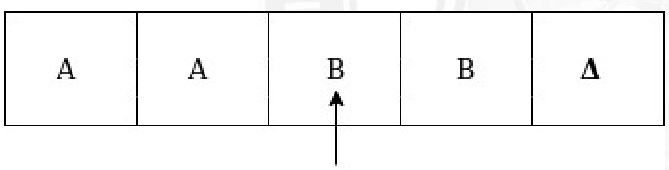
The move will be δ(q0, 0) = δ(q1, A, R) which means it will go to state q1, replaced 0 by A, and head will move to right as:



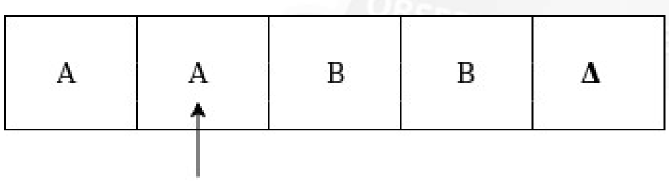
The move will be δ(q1, B) = δ(q1, B, R) which means it will not change any symbol, remain in the same state and move to right as:



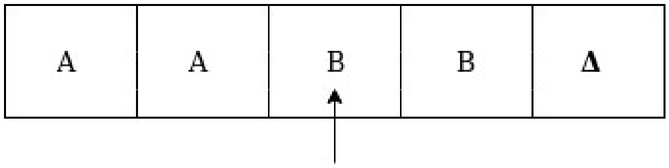
The move will be δ(q1, 1) = δ(q2, B, L) which means it will go to state q2, replaced 1 by B and head will move to left as:



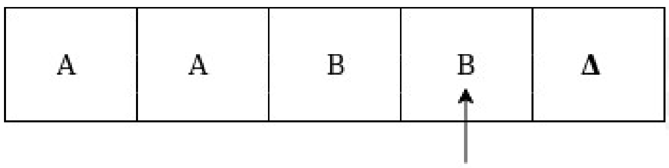
The move δ(q2, B) = (q2, B, L) which means it will not change any symbol, remain in the same state and move to left as:



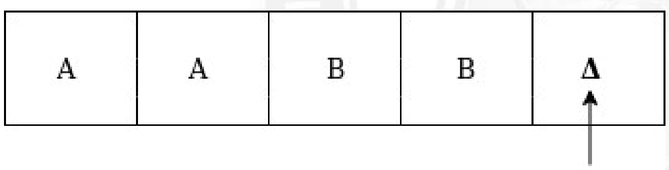
Now immediately before B is A that means all the 0?s are market by A. So we will move right to ensure that no 1 is present. The move will be δ(q2, A) = (q0, A, R) which means it will go to state q0, will not change any symbol, and move to right as:



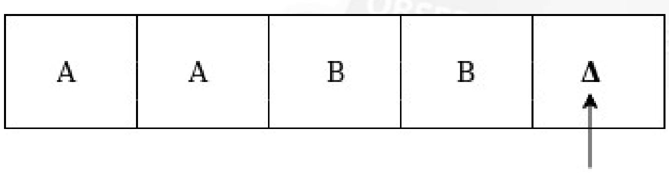
The move δ(q0, B) = (q3, B, R) which means it will go to state q3, will not change any symbol, and move to right as:



The move δ(q3, B) = (q3, B, R) which means it will not change any symbol, remain in the same state and move to right as:



The move δ(q3, Δ) = (q4, Δ, R) which means it will go to state q4 which is the HALT state and HALT state is always an accept state for any TM.

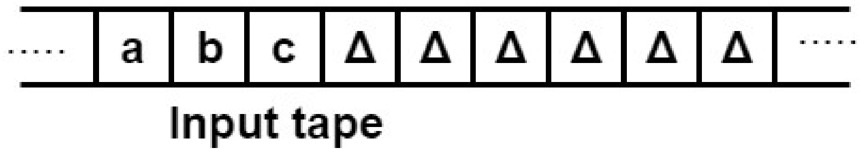


The same TM can be represented by Transition Diagram:

Basic Model of Turing machine

The turning machine can be modelled with the help of the following representation.

1. The input tape is having an infinite number of cells, each cell containing one input symbol and thus the input string can be placed on tape. The empty tape is filled by blank characters.



2. The finite control and the tape head which is responsible for reading the current input symbol. The tape head can move to left to right.

3. A finite set of states through which machine has to undergo.

4. Finite set of symbols called external symbols which are used in building the logic of turing machine.

Language accepted by Turing machine

The turing machine accepts all the language even though they are recursively enumerable. Recursive means repeating the same set of rules for any number of times and enumerable means a list of elements. The TM also accepts the computable functions, such as addition, multiplication, subtraction, division, power function, and many more.

Example:

Construct a turing machine which accepts the language of aba over ∑ = {a, b}.

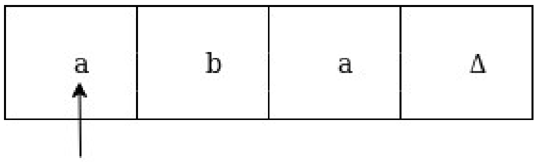
Solution:

We will assume that on input tape the string 'aba' is placed like this:

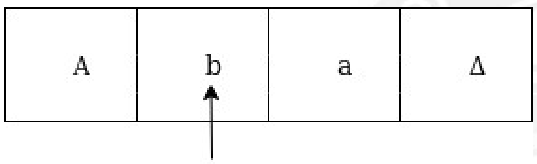
a	b	a	Δ	-------

The tape head will read out the sequence up to the Δ characters. If the tape head is readout 'aba' string then TM will halt after reading Δ.

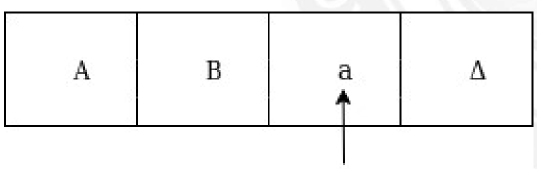
Now, we will see how this turing machine will work for aba. Initially, state is q0 and head points to a as:



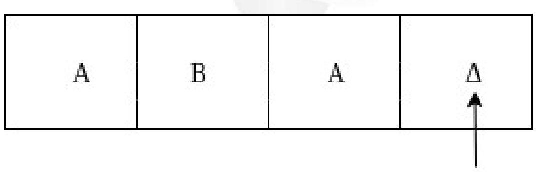
The move will be δ(q0, a) = δ(q1, A, R) which means it will go to state q1, replaced a by A and head will move to right as:



The move will be δ(q1, b) = δ(q2, B, R) which means it will go to state q2, replaced b by B and head will move to right as:



The move will be δ(q2, a) = δ(q3, A, R) which means it will go to state q3, replaced a by A and head will move to right as:



The move δ(q3, Δ) = (q4, Δ, S) which means it will go to state q4 which is the HALT state and HALT state is always an accept state for any TM.

The same TM can be represented by Transition Table:

States	A	b	Δ
q0	(q1, A, R)	–	–
q1	–	(q2, B, R)	–
q2	(q3, A, R)	–	–
q3	–	–	(q4, Δ, S)
q4	–	–	–

The same TM can be represented by Transition Diagram: