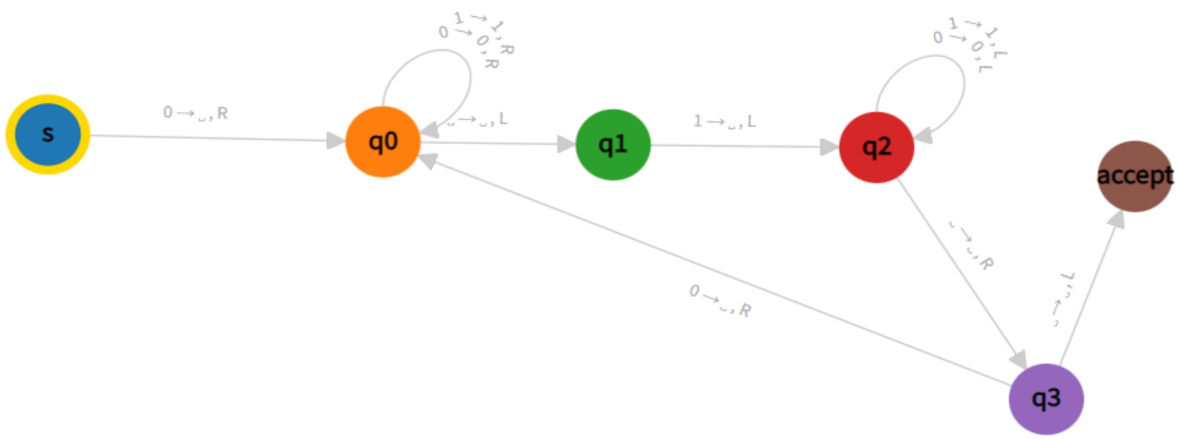
280 HW3 REPORT

Q1)

a) The screenshot of the created machine.



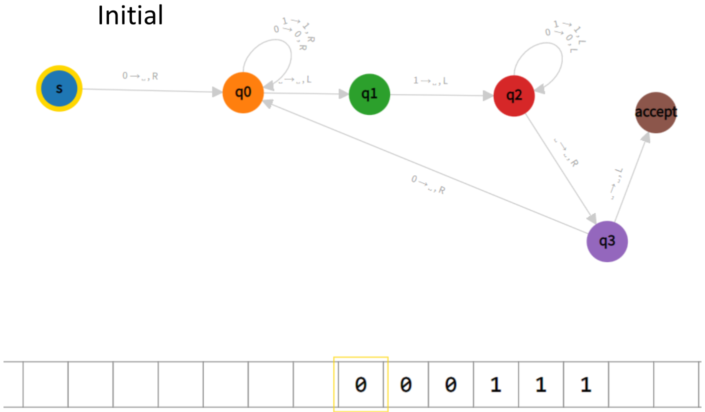
b) A clear description of every state used in the machine.

s is the start state.
Machine deletes the first 0 in the string and changes its state to q0.
q0 moves head to the end of the string.
Machine goes to q1 state if head reaches to the end of the string.
Machine deletes the last 1 in the string and changes its state to q2.
q2 moves head to the beginning of the string.
Machine goes to q3 state if head reaches to the beginning of the string.
If string is empty, machine goes to accept state and machine halts.
If first symbol is 0 then machine goes to q0 state and repeat same process until string is empty.
If the machine crashes with unexpected input, then it means machine goes to reject state.

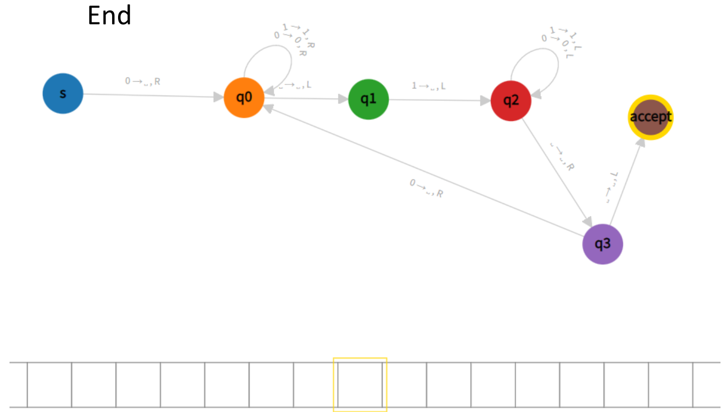
c) Give initial and end state screenshots with a few input samples.

i) 000111 (accepted)

Initial

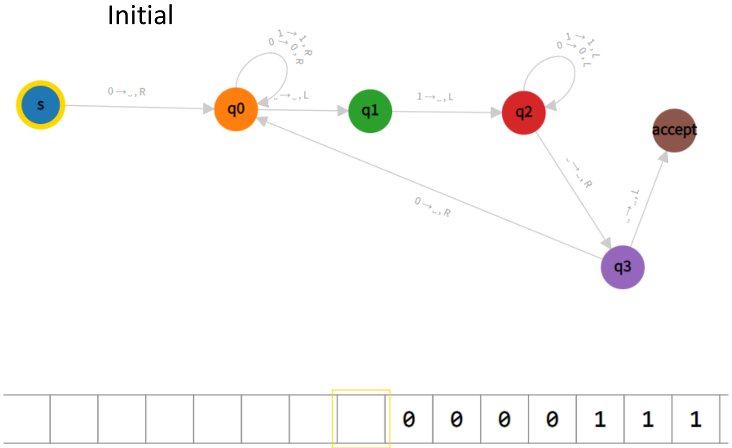


End

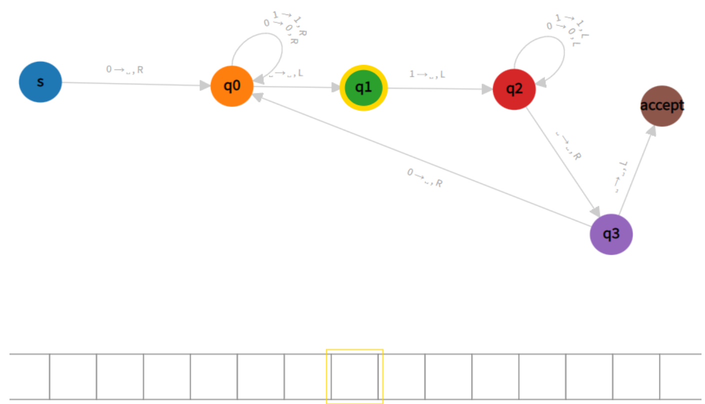


ii) 0000111 (rejected)

Initial

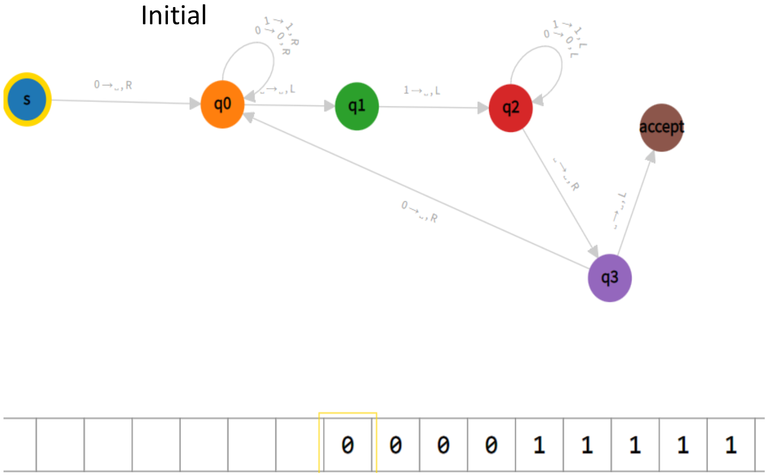


End

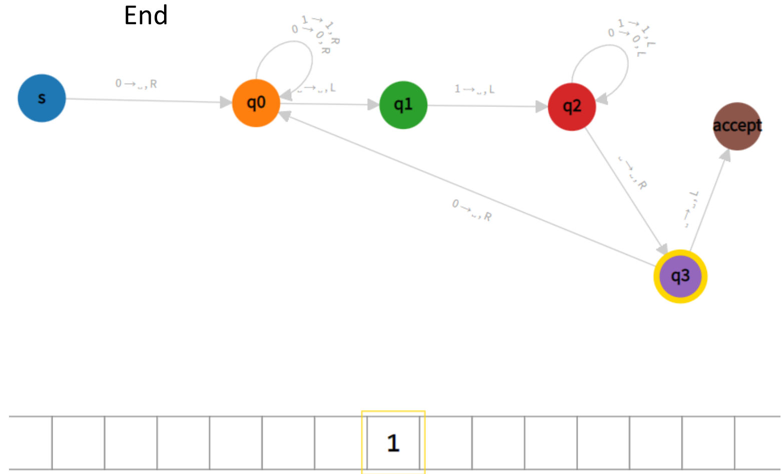


iii) 000011111 (rejected)

Initial

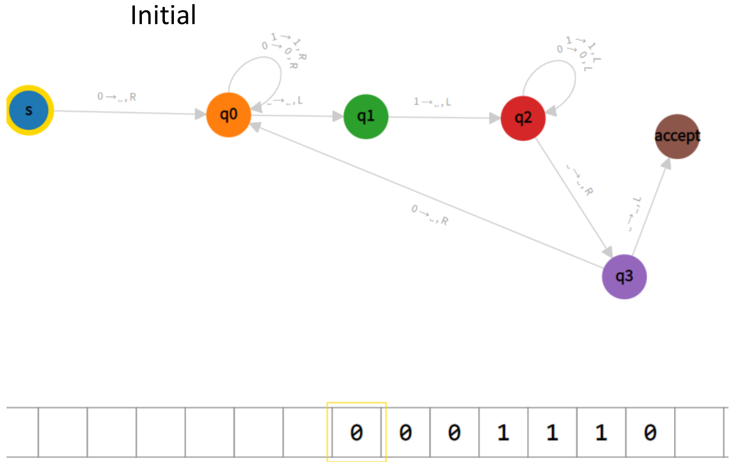


End

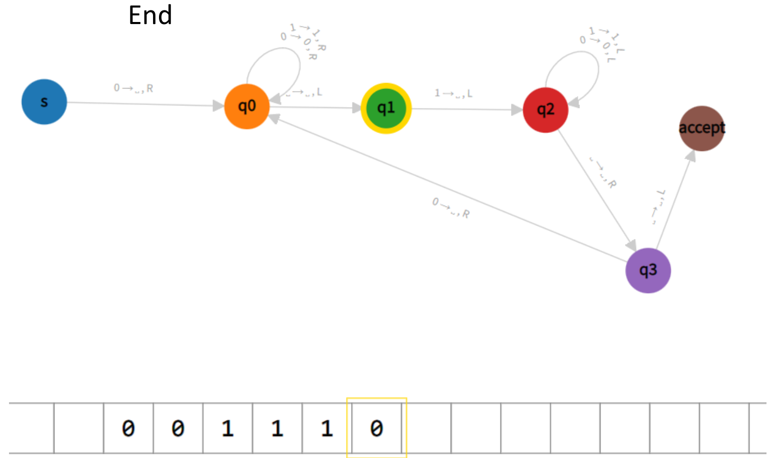


iv) 0001110 (rejected)

Initial

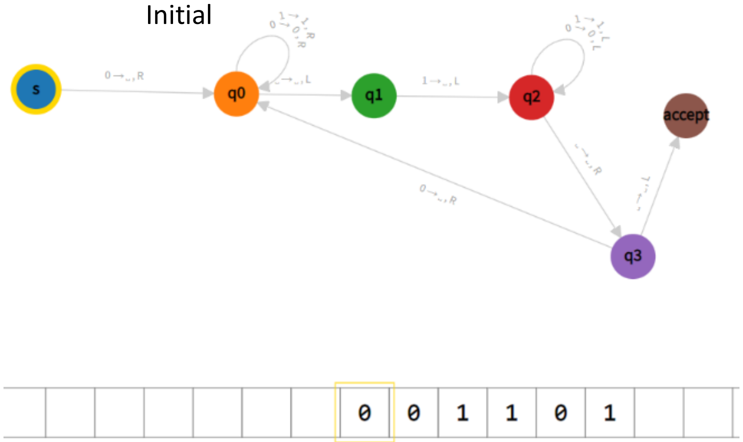


End

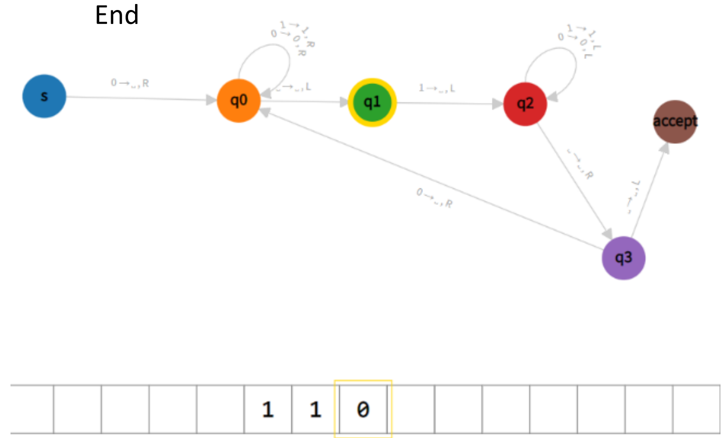


v) 001101 (rejected)

Initial

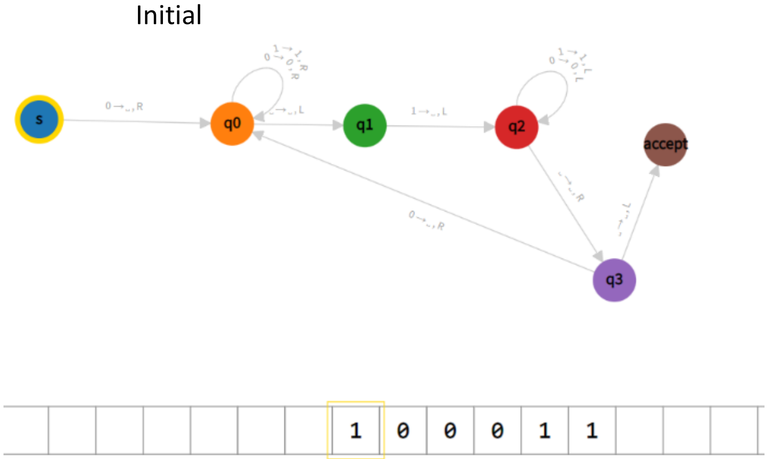


End

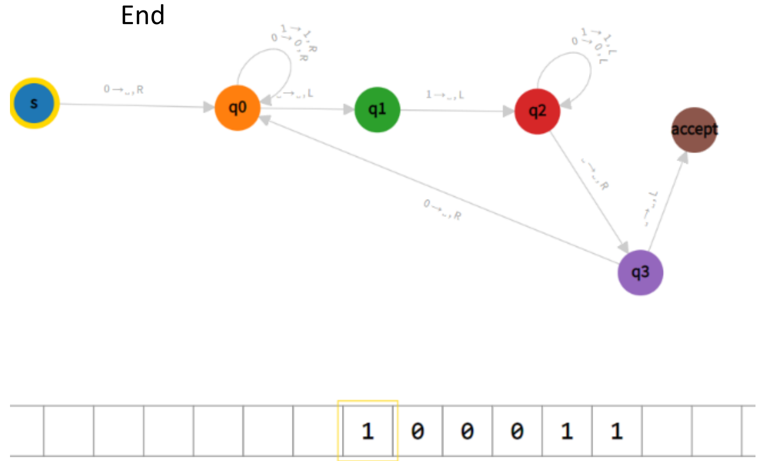


vi) 100011 (rejected)

Initial

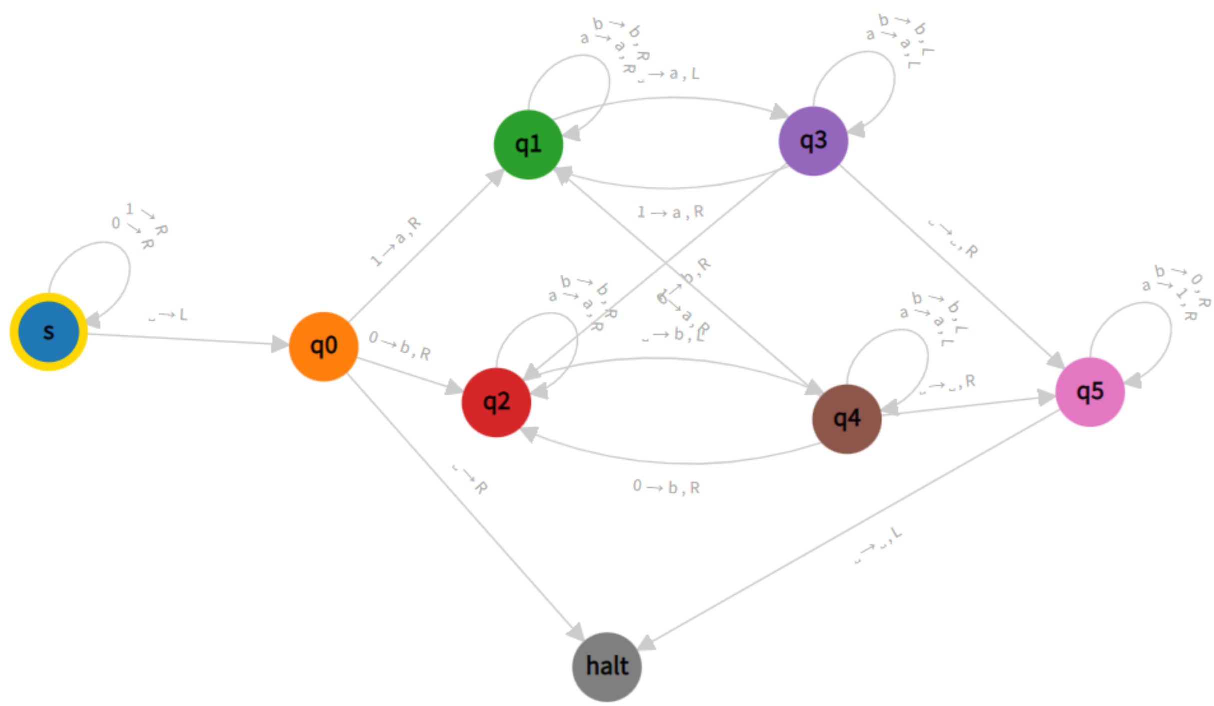


End



Q2)

a) The screenshot of the created machine.



b) A clear description of every state used in the machine.

s is the start state.

s moves head to the end of the string.

Machine goes to q0 state and moves head to the left if head reaches to the end of the string.

Machine halts if string is empty.

If the last character in the string is 1 then machine changes it to a then goes to q1 state.

If the last character in the string is 0 then machine changes it to b then goes to q2 state.

q1 state moves the head to the end of string and when head reaches to the end of the string it writes a.

q2 moves head to the end of the string and when head reaches to the end of the string it writes b.

q3 moves head to the last 0 or 1 on the string.

> If the last character in the string is 1 then machine changes it to a then goes to q1 state.
> If the last character in the string is 0 then machine changes it to b, then goes to q2 state.
> Machine repeats this process until there are not any 1's or 0's.

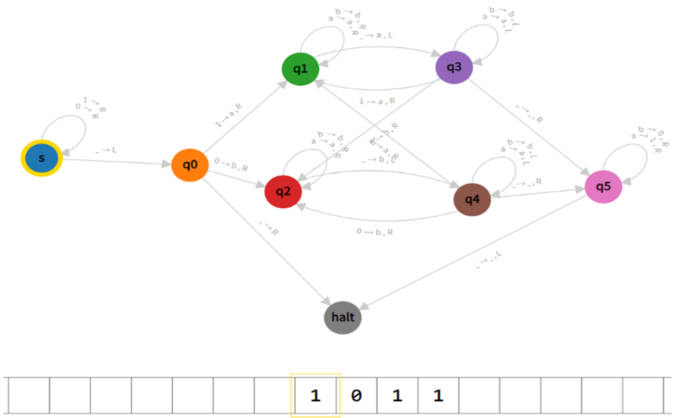
When there are not any 1's or 0's in the string, machine changes its state to q5.

In q5, all a's are written as 1's and all b's written as 0's. Then machine halts.

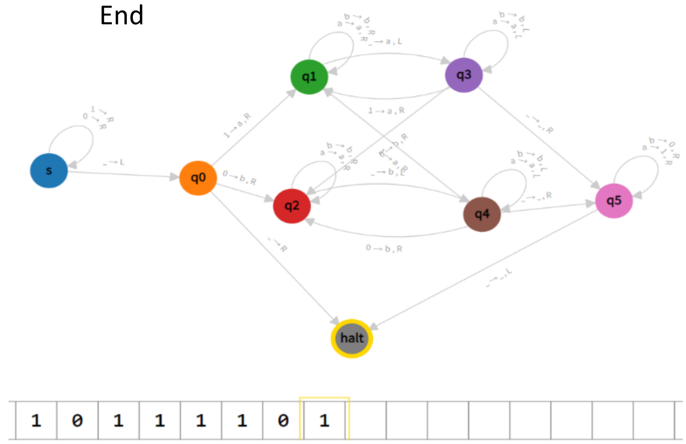
c) Give initial and end state screenshots with a few input samples.

i) 1011

Initial

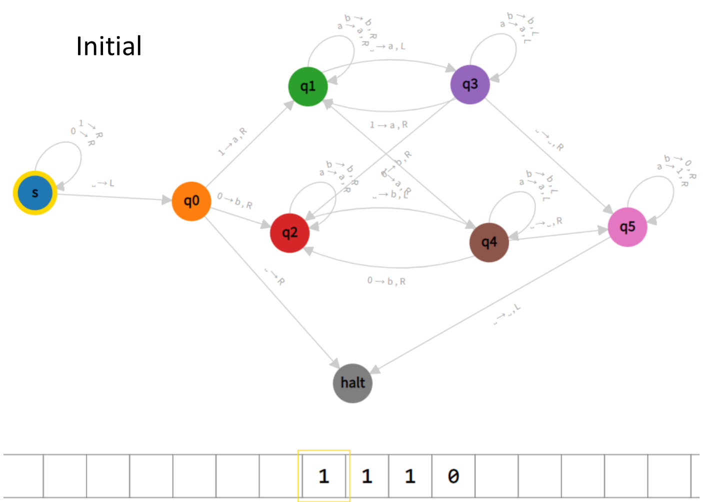


End

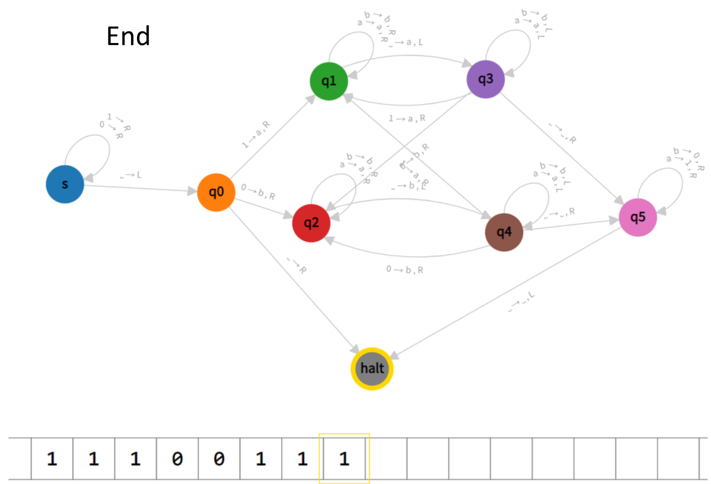


ii) 1110

Initial

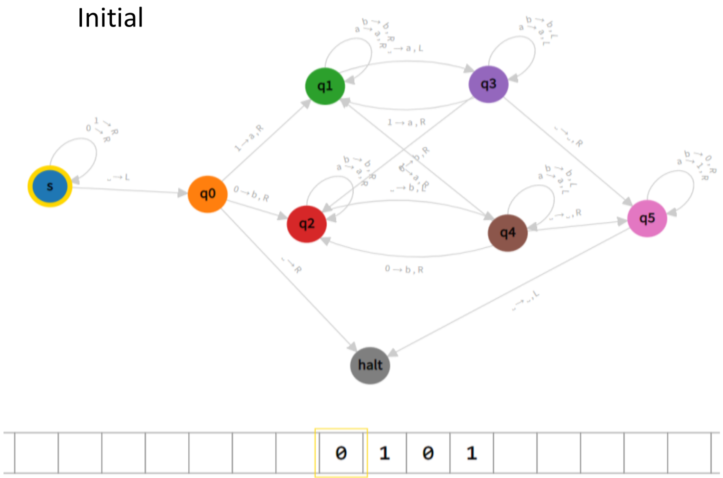


End

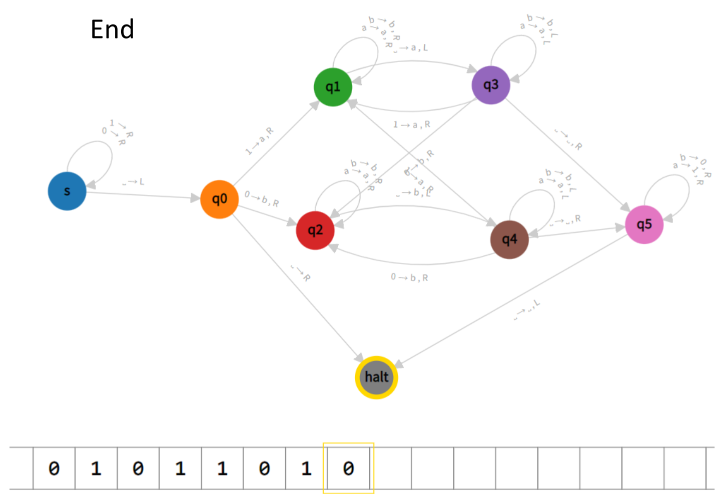


iii) 0101

Initial

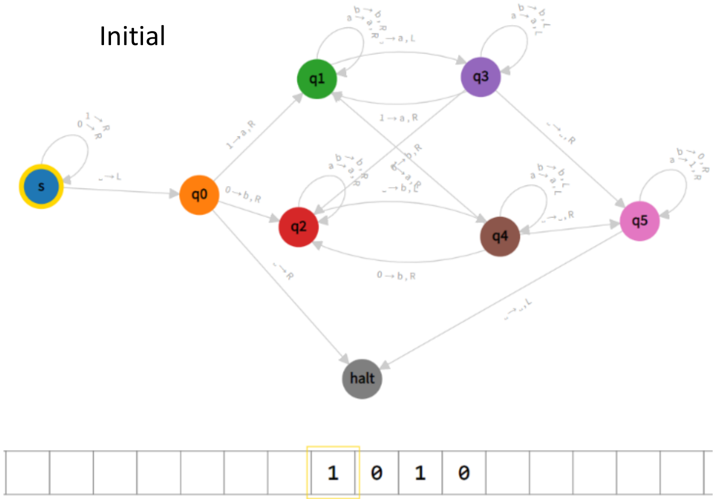


End

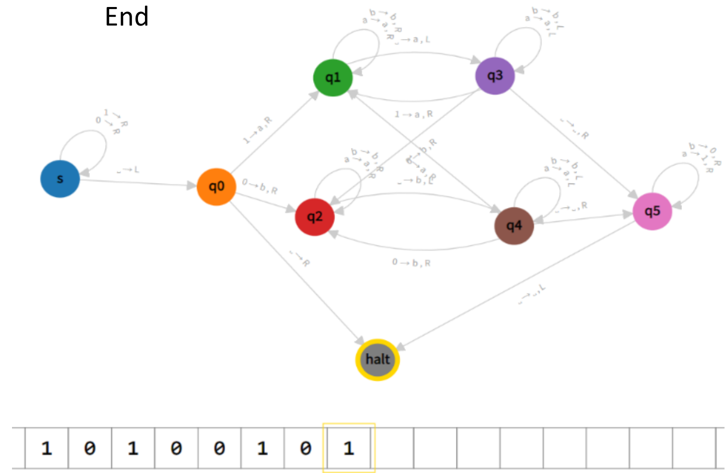


iv) 1010

Initial

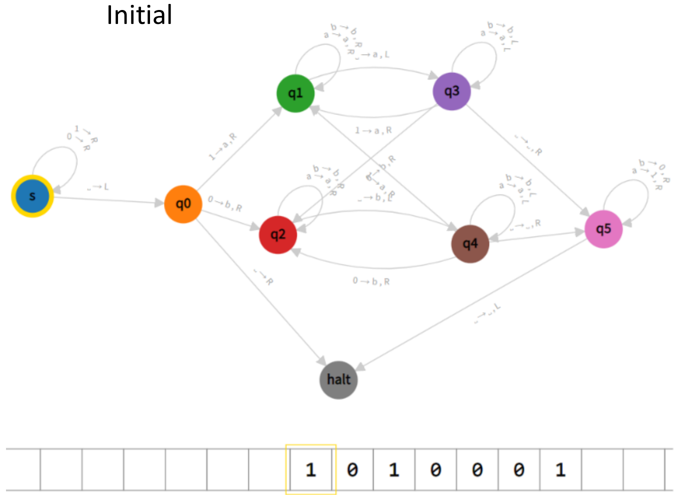


End

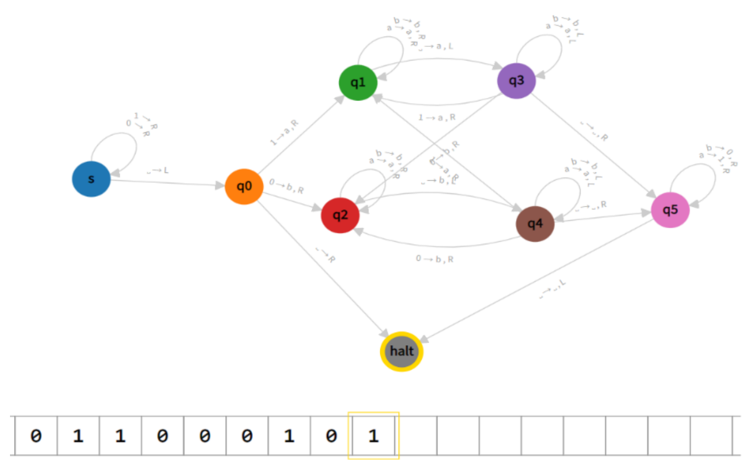


v) 1010001

Initial

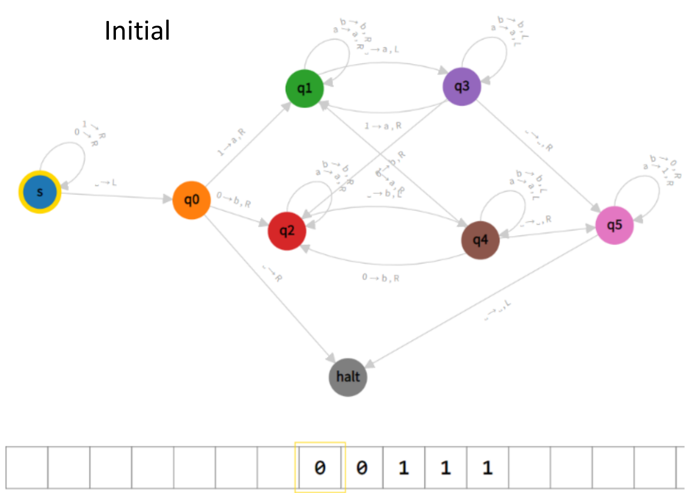


End

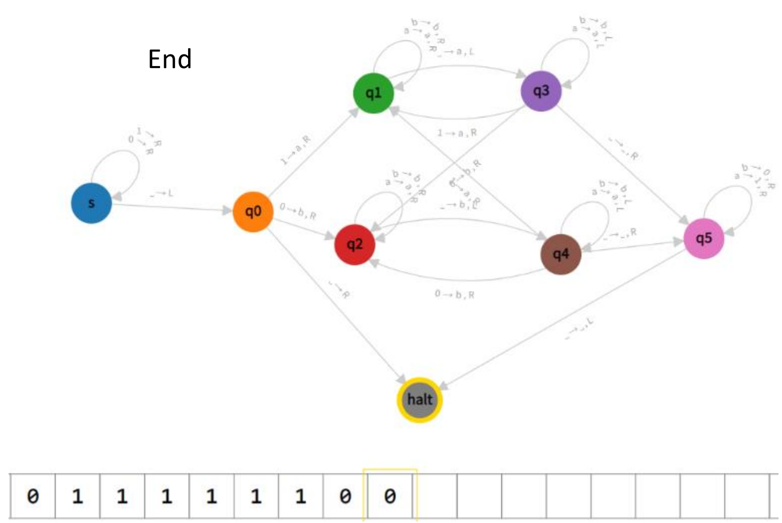


vi) 00111

Initial



End



Q3)

We can define a two-dimensional Turing machine as a pentuple M = (K, Σ, δ, s, H) where,

K is the finite set of states,

s is initial state,

H ⊆ K is the set of halting states,

Σ is tape alphabet which includes the beginning marker △ and blank symbol ␣ ,

δ is transition function (K x Σ) -> (K x (Σ ∪ {↑, ←, →, ↓}))

The transition function δ includes {↑, ↓} symbols because our tape is two dimensional and head can move up or down on that tape. This feature is different than standard Turing machine in which we can move the head only right or left. Also we should note that δ(q0, ▷) = (q1, →) and δ(q0, △) = (q1, ↓) for all s=q0. Which means when we are at the beginning marker and initial state we can only move the head left or down.

We can define a configuration of a two-dimensional Turing machine as K x Z^+ x Z^+ x L.

K is the current state,

Z^+ is a natural number ((Z^+ x Z^+) shows the current position of the head),

L is the list of tuples in which symbols and their locations on the tape are hold. (For example, [(2,3,a),(3,4,b)] means that there is a 'a' on third cell of the second row and there is a 'b' on fourth cell of the third row. Also, we should note that first cell of the first row (1,1) has the beginning marker △.

Briefly, we can represent the configuration by the current state, current head position (Z^+ x Z^+) and a list of all non-blank cells on the tape.

We can show a (yields in one step) operation (q2, x1, y1, l1) ⊢ (q3, x2, y2, l2) if δ(q2, y) = (q3, σ) where y is the current input at (x1,y1). If σ= →, then we don't change the row the head is at this means x1=x2 but we change the column the head is at by moving the head to one cell right which means y2 = y1+1. If σ= ←, then we don't change the row the head is at this means x1=x2 but we change the column the head is at by moving the head to one cell left which means y2 = y1-1. If σ= ↑, then we change the row the head is at by moving the head to one cell upwards this means x2=x1+1 but we don't change the column the head is at which means y2 = y1. Finally, if σ= ↓, then we change the row the head is at by moving the head to one cell downwards this means x2=x1-1 but we don't change the column the head is at which means y2 = y1.

We can decide a language L by using two different halting states namely, yes and no. When given a string w to our machine M if our machine halts at yes state than we can say that w is in the language w ϵ L(M). If our machine halts at no state than we can say that w is not in the language w ∉ L(M).