Turing Machine Examples

Define a Turing Machine that takes a string of 1's and 0's as an input and appends y at the end of the string if it actually contains some 1's, otherwise it appends n

[Δ][1][1][1][0][**U**]

Δ = Start of tape

U = End of tape

A, Δ => A, Δ, R

A, 1 => B, 1, R; *Found a 1, go to B state*

A, 0 => A, 0, R; *No 1 found, continue on A state*

A,U => C,N,L; *Go to final state and append N, no 1s found*

B,1 => B,1,R;

B,0 => B,0,R;

B,U => C,Y,L; *Go to final state and append Y, 1s were found*

C,0 => C,0,L

C,1 => C,1,L

C,Δ => Accept

Define a Turing Machine that takes a string of 1's and 0's as an input and and accepts it if the first and the last digit of the string are identical

[Δ][0][0][0][1][1][0][**U**]

Δ = Start of tape

U = End of tape

A, Δ => A, Δ, R

A, 1 => B, 1, R *First digit is 1, go to b state*

A, 0 => C, 0, R *First digit is 0, go to c state*

B, 1 => B,1,R

B, 0 => B,0,R

B, U => D,U,L *At end, go to d state and move to last digit*

D,1 => Accept *Accept if last digit is 1, first digit must be 1(came from B state)*

D,0=> Reject

C, 1 => C,1,R

C,0 => C,0,R

C,U => E,U,L *At end, go to E state and move to last digit*

E,0 => Accept *Accept if last digit is 0, first digit must be 0(came from C state)*

E,1 => Reject

Define a Turing Machine that takes a string of 1's and 0's as an input and accepts only if there is no 1 followed by 0 (i.e. 0 can not occur after 1)

[Δ][1][1][1][0][**U**]

Δ = Start of tape

U = End of tape

A, Δ => A, Δ, R

A, 0 => A, 0, R

A, 1 => B, 1, R *Found 1, go to B state (0s cannot occur now)*

A,U => Accept

B,0 => C,0,R *0 occured, go to C state and fail*

B,1 => B,1,R

B,U => Accept

C,0 => C,0,R

C,1 => C,1,R

C,U => Reject

Define a Turing Machine that takes a string of 1's and 0's as an a binary number and increments it by 1

[Δ][0][0][1][**U**]

Δ = Start of tape

U = End of tape

A, Δ => A, Δ ,R

A,0 => A,0,R

A,1 => A,1,R

A,U => B,U,L *Go to B state at the end*

B,1 => B,0,L *Change 1's to 0's*

B,0 => C,1,L *Change 0's to 1's, and end the state*

B, Δ =>C,1,L *Change* Δ to 1, and end the state

C => Accept/Final state

Example Tests:

[0][0][1]

[0][0][0] *Change 1's to 0's*

[0][1][0] *Change 0's to 1's, and end the state; completed*

[Δ][1][1][1]

[Δ][0][0][0] Change 1's to 0's

[1][0][0][0]Change Δ to 1, and end the state; completed