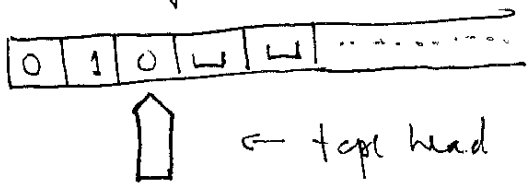
Turing machines

Task Look at a more realistic model of a computer than a finite state acceptor.

Turing machines were first proposed by Alan Turing in 1936 in order to explore the theoretical limits of computation. We shall see that certain problems cannot be solved even by a Turing machine and are thus beyond the limits of computation.

A Turing machine is similar to a finite state acceptor but has unlimited memory given by an infinite tape (we mean countably infinite here). The infinite tape is divided into cells each of which holds character of a tape alphabet. The Turing machine is equipped with a tape head that can read and write symbols on the tape and move left (back) or right (forward) on the tape. Initially, the tape contains only the input string and is blank everywhere else. To store information, the Turing machine can write this information on the tape. To read information that it has written, the Turing machine can move its head back over it. The Turing machine continues computing until it decides to produce an output. The outputs "accepts" and "rejects" are obtained by entering accepting or rejecting states respectively. It is also possible for the Turing machine to go on forever never stopping if it does not enter either an accepting or a rejecting state.

Illustration of a Turing machine



␣ the blank symbol is part of the tape alphabet

Example Let $A = \{0, 1\}$ and let $L = \{0^m 1^m \mid m \in \mathbb{N}, m \geq 1\}$.

We know L is not a regular language, so there is no finite state acceptor that can recognize it, but there is a Turing machine that can.

Initial state of the tape: input string of 0's and 1's, then infinitely many blanks

Idea of this Turing machine change a 0 to an X, and then a 1 to a Y until either
- all 0's and 1's have been matched, hence ACCEPT
- the 0's and 1's do not match or the string does not have the form 0^*1^*, hence REJECT.

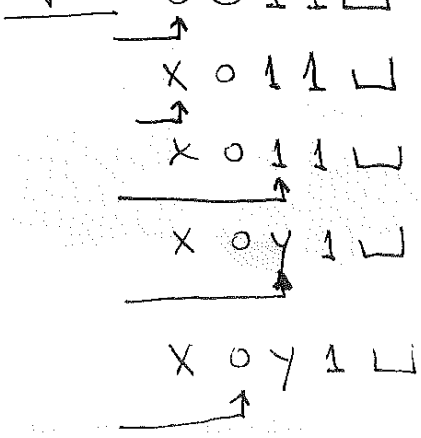
Algorithm

The tape head is initially positioned over the first cell.

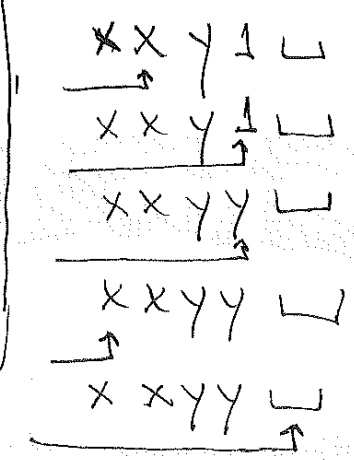
1. If anything other than 0 is in the first cell, then REJECT.
2. If 0 is in the cell, then change 0 to X.
3. Move right to the first 1. If none, then REJECT.
4. Change 1 to Y.
5. Move left to the leftmost 0. If none, move right looking for either a 0 or a 1. If either 0 or 1 is found before the first blank symbol, then REJECT; otherwise, ACCEPT.
6. Go to step 2.

Let's process some strings:

Input 0 0 1 1 ␣

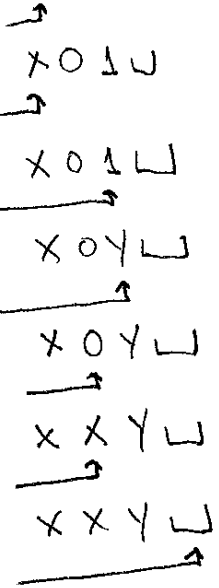


→ we continue here



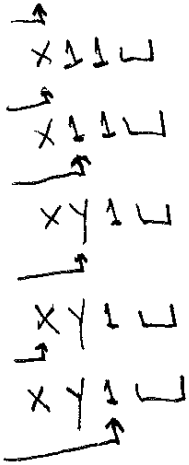
Outcome ACCEPT (step 5)

Input 001␣



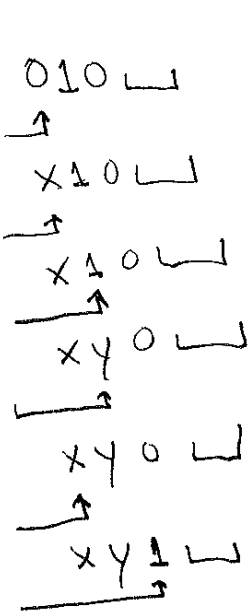
Outcome REJECT (step 3)

Input 011␣



Outcome REJECT (step 5)

Input 010␣



Outcome REJECT (step 5)

Note That we have the following:

$A = \{0,1\}$ the input alphabet

$\sqcup \notin A$, where $\sqcup$ is the blank symbol.

$\tilde{A} = \{0, 1, X, Y, \sqcup\}$ is the tape alphabet

S a set of states.

Note also that the tape head is moving right or left, so we also need to have a set $\{L, R\}$ of L for left and R for right specifying where the tape head goes.