- Visualizing a "mathematical model" of a computer :
- It should not be limited by the limitations (memory, speed etc) of the "actual" machine.

- So we may assume
  "infinite memory" is available.
- We do not bother about the "speed,"
  instead we just count the
  "number of steps"
  needed to make the computation.

- i.e, we need a "mathematical" machine such that

(1) any problem that can be solved using a computer (whichever computer it may be) can be solved using this machine.

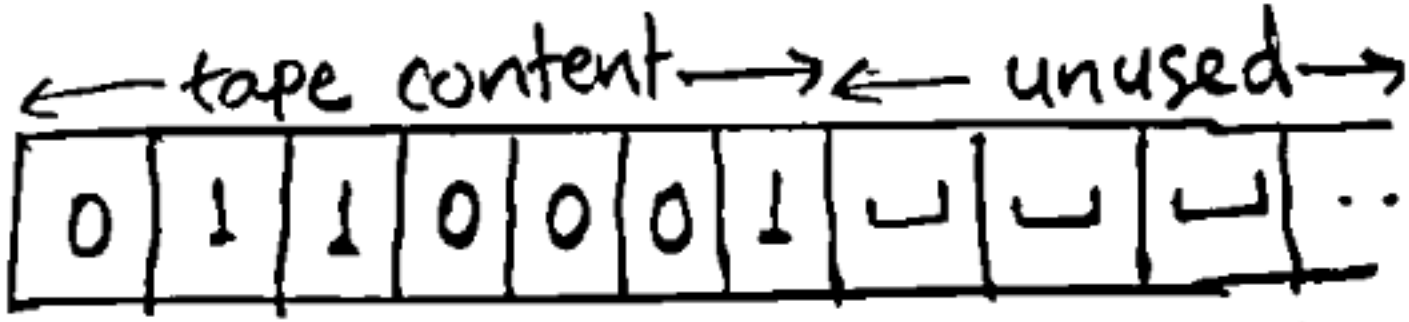
(2) Any problem that can be solved using this machine can be solved with a computer (if it has enough memory).

- We will try to understand this with a problem (and an algorithm to solve it) before getting to the "formal" definition of this mathematical model.
  - Problem: Given a string of 0's and 1's, is it of the form 0^n1^n?
    eg: 0011, 000111 → Yes. 0101 → No.

- In general, input can be
  - Numbers
  - Strings
  - Images
  - Graphs...

But eventually, when it goes to computer, they are all "strings" (say, of 0's & 1's).

* To keep the model simple,
we will assume there is a single long
"tape" that initially contains the
input, and it is also used for "rough"
work, and at the end the "result"
if any, is also written on it.



- Initially, it contains the input.
- Machine can "read" a cell, and then move left or right.

- To estimate how long the algorithm takes to finish, we count each move (towards left or right) on this tape as one move in the algorithm.

- How can the machine "remember" things?

There are 2 types of "memory".

One. when we have to remember only a limited (finite) options.

eg: Is the length of the string odd?

0 1 0 0 1 0 1 0 0 1

- We can use just "two states"!

• But when we have to remember a number, or a count, that can be arbitrarily large (depending on the input), we can not use it.

- Then, we have to "write down" on the tape.

Algorithm: (without "writing down" the count of zero's/1's)

000001111 0 ⇒ NO.

000011111

formalizing:

A 'turing machine' M consists of

- An infinite tape (initially it contains input string, w)
- A finite set of states
- 'Moves': Depending on the state and "current" tape symbol

- To keep track : A "tape head", initially at the left most cell
- The input alphabet
  and
  Tape alphabet (symbols that can appear on tape) also should be clear.

Algorithm: (without "writing down" the count of zero's/1's)

0 0 0 0 0 1 1 1 1 0 → No.

0 0 0 0 0 1 1 1 1 1