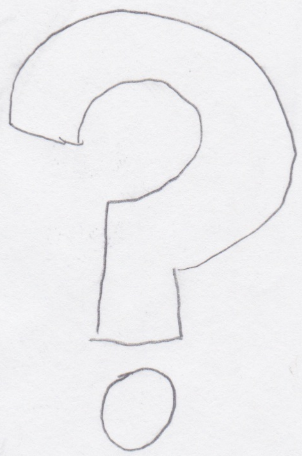
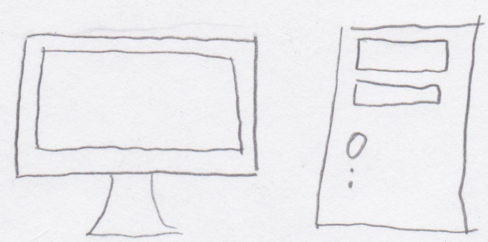
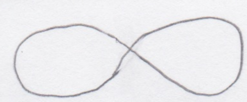


What is a Turing Machine?

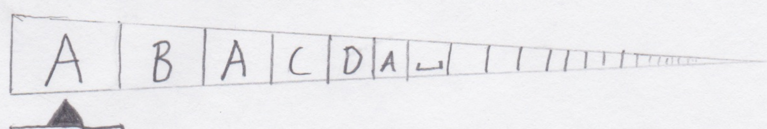
- It is a model of a general purpose computer.



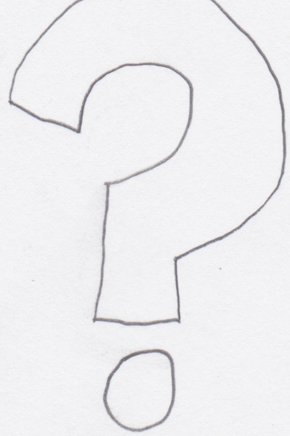
- It has unlimited and unrestricted memory.



- This memory is stored in an infinitely long tape.

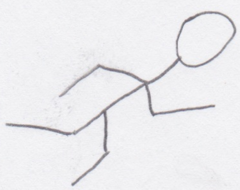


- A tape head points to the current position in memory of the Turing machine. The Turing machine can read or write at the position of the tape head.
- The tape head can move to the left or the right one symbol at a time. It must STOP moving to the left if the beginning of the tape is reached.
- A Turing machine will continue to run until either an accept ✓ or reject X state is reached.



Why should we care?

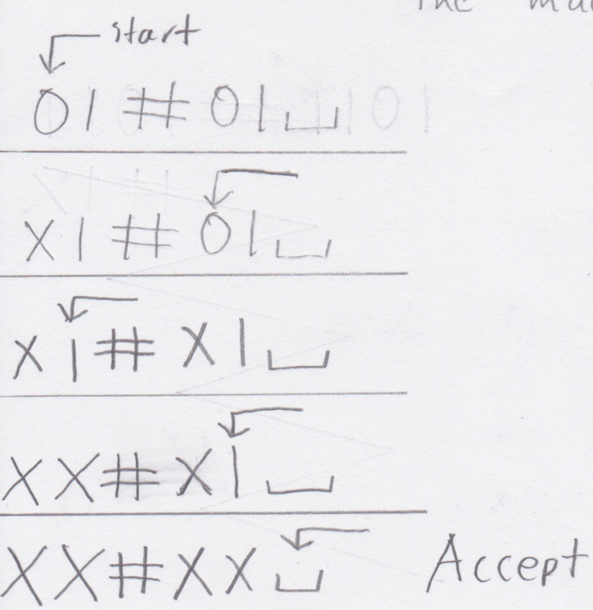
- A Turing machine can be created to solve any problem that is within the theoretical limits of computation.
- Turing machines are able to solve problems that cannot be solved using finite automata.



Running a Turing Machine

This example will show the strategy that a Turing machine would use to accept a language where all symbols before a # sign are the same as those after. Otherwise, the machine will reject.

Tape Configurations



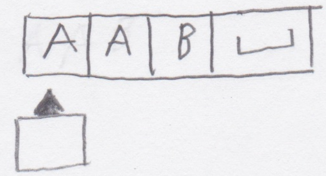
Strategy

- Start at the leftmost element, and mark it off with an X.
- Move right until a # symbol is found.
- Move to the right again. If this symbol is the same as the leftmost symbol was, mark it off as an X.
- Now move to the left until the # sign is found. Then move left again until a symbol marked with an X is found. Mark off the symbol to the right of this.
- Continue to Zig-Zag until either an unmatched pair is found (REJECT), or all are Xed off (ACCEPT).

What is a Turing Machine's Configuration?

- A setting of the following three items at a given point in time while the Turing machine is being run.

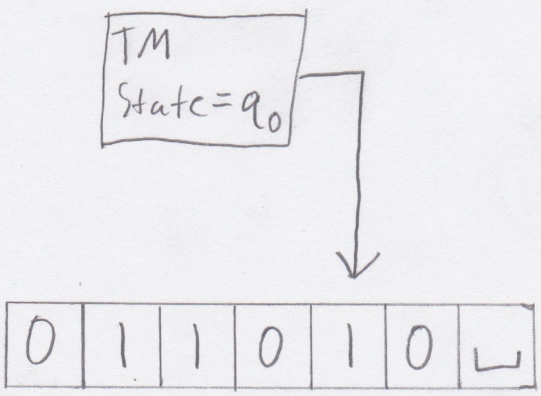
  - Current state
  - Current tape contents
  - Current head location



How can a configuration be represented?

- Using a string created in the following way:

Configuration ⟶ Representative String



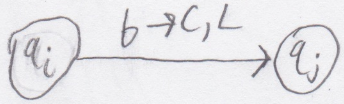
$0110\,q_0\,10$

↑

The current state name is put to the left of the symbol the tape head is pointing at.

What is Yield?

- One configuration **Yields** another if the second configuration can be reached from the first in 1 step.
- Say the following transition exists in the state diagram representing TM M:



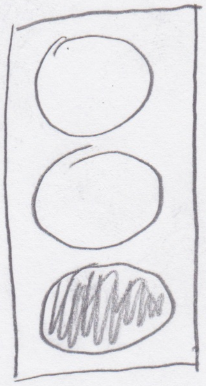
- Then we could say that configuration:

$$aaq_ibv \quad \text{YIELDS} \quad aq_jacv$$

Special Configurations

- Start Configuration: q_0w, where q_0 is the start state, and w is the initial contents of the tape.
- Accept Configuration: $\Sigma^* q_{accept} \Sigma^*$, which is any configuration consisting of valid symbols and the accept state.
- Reject Configuration: $\Sigma^* q_{reject} \Sigma^*$, which is any configuration consisting of valid symbols and the reject state.

These are HALTING configurations, which means that they do not yield any further configurations once reached.



How does a TM accept?

- To accept, a sequence of configurations $C_1, C_2, \ldots, C_{k-1}, C_k$ must exist where:
  1. C_1 is the start configuration.
  2. each configuration C_i yields C_{i+1}
  3. C_k is an accepting configuration.

Turing Recognizeable Vs. Turing Decideable

- First, define recognizeable and decideable:

Recognizeable: A Turing machine can accept, reject, or loop on input. A Turing machine recognizes a language if it accepts when a string in that language is inputted.

Decideable: A Turing machine decides a language if it recognizes that language and does not loop for any input.

Turing-Recognizeable

A language is Turing-Recognizeable if some Turing machine recognizes it.

Turing-Decideable

A language is Turing-Decideable if some Turing machine decides it. Notice that all Turing-Decideable languages are also Turing-Recognizeable.