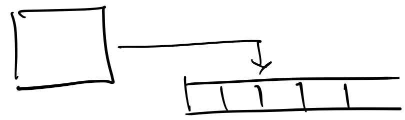
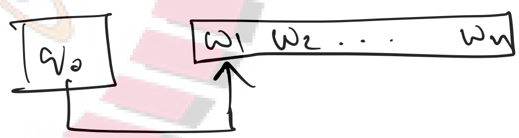
Turing Machines.



The TM **accepts w** if there is a sequence of configurations $C_1, C_2, \ldots C_k$ such that

1. C_1 is the start configuration : $q_0 w$
2. $C_i \rightarrow C_{i+1}$ for all i
3. C_k is an accepting configuration (the state in C_k is q_{accept}).

$q_0 w_1 w_2 \ldots w_n$



Other possibilities are
- → TM rejects w.
- → TM loops on w.

The language **recognized** by M is denoted **$L(M)$**.

$$L(M) = \{ w \mid w \text{ is accepted by } M \}.$$

Def 3.5 L is **Turing recognizable** (**recursively enumerable**) if some TM recognizes it.

Given a string w, there are 3 outcomes possible.

Accept | reject | loop.

TM M **decides** L (M is a decider for L) if **$L(M) = L$** and M always halts. → Strings not in L are rejected.

Def 3.6: L is **decidable** (**recursive**) if some TM decides it.

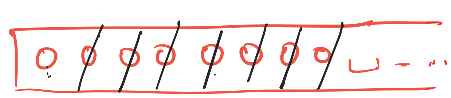
Intuitively a decider = algorithm

✓ 0
✓ 00
✓ 0000

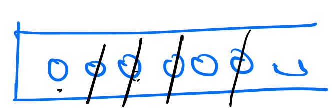
Example 3.7: $A = \{0^{2^n} \mid n \geq 0\}$.

X 000000

* Move the tape from left to right, crossing off every other 0.
* If only one 0, accept.
* If number of 0's is an odd number greater than 1, reject.
* Return head to the left most position
* Repeat



— This can be implemented in detail.



Reject

For example, see the TM described below.

Goal: Shift input to the right and add a # at the beginning

$\boxed{0001101}$

$\downarrow$

$\boxed{\#0001101}$

$M = (Q, \{0,1\}, \Gamma, \delta, q_s, q_a, q_r)$

$\delta(q_s, 0) = (q_0, \#, R)$ $\quad$ $\delta(q_s, 1) = (q_1, \#, R)$

$\delta(q_0, 0) = (q_0, 0, R)$ $\quad$ $\delta(q_0, 1) = (q_1, 0, R)$

$\delta(q_1, 0) = (q_0, 1, R)$ $\quad$ $\delta(q_1, 1) = (q_1, 1, R)$

$\delta(q_0, \sqcup) = (q_2, 0, L)$ $\quad$ $\delta(q_1, \sqcup) = (q_2, 1, L)$

$\delta(q_2, 0) = (q_2, 0, L)$ $\quad$ $\delta(q_2, 1) = (q_2, 1, L)$

$\delta(q_2, \#) = (q_a, \#, L)$

$\boxed{\#\ 0\ 0\ 1\ 0\ 1\ 1\ \sqcup \ldots}$

$\uparrow$

q_s

q_0 $\quad$ q_1

q_0 $\quad$ q_1

q_1 $\quad$ q_2

q_0

TM's can be used to compute functions

$w = w_1 w_2 \dots w_n$: A binary number.

\# helps to indicate left end

Assume the tape contains $\# w_1 w_2 \dots w_n$.

GOAL : To increment w.

1. Move to the right most end.
2. Repeat
   - If the current symbol is 0, make it 1. Move to the left end. Accept.
   - If the current symbol is 1, make it 0. Move one step left.
   - If the current symbol is #.
     - Change to 1
     - Shift every symbol one step to the right.
     - Move to left end.
     - Write #
     - Halt. Accept.

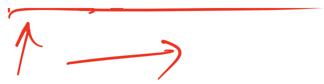
1000

1000

$\# 0 0 1 1 0 \sqcup \sqcup \dots$

$1 0 0 0 \sqcup \sqcup \sqcup$

$\# 1 0 0 0 \dots$



$C = \{ a^i b^j c^k \mid i * j = k \ , \ i, j, k \geq 1 \}$

1. Scan from left to right. Check if member of $a^+ b^+ c^+$. } DFA

2. Return head to left. } Need a special symbol like # to find left end.

3. Cross a, move till first b.
   - Cross one b, Cross one c
   - Repeat till b's are over
   - Reject if #b > #c.

4. Restore crossed b's. Repeat stage 3 for each a. When all a's are crossed, check if all c's are crossed. If yes, accept. If not, reject.

$\# \ \cancel{a} \ \cancel{a} \ \cancel{a} \ \cancel{b} \ \cancel{b} \ \cancel{c} \ \cancel{c} \ \cancel{c} \ \cancel{c} \ \cancel{c} \ \cancel{c} \ c$

$\# \ \cancel{a} \ \cancel{a} \ \cancel{a} \ \cancel{b} \ b \ \cancel{c} \ \cancel{c} \ \cancel{c} \ \cancel{c}$?