For a TM M, write $L(M)$ for the language accepted by M

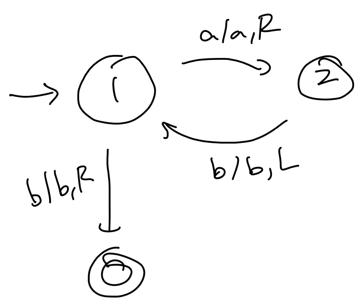
all strings accepted by M.

Any language accepted by a TM is called recursively enumerable (RE)

2 types of non-accepted strings:

A string is rejected if the TM stops (no appropriate arrow) in a non-accepting state.

Some strings result in an infinite loops



ab:

$1ab \mapsto a2b \mapsto 1ab \mapsto \dots$

loops forever.

Loops forever on any abx

$1aa \mapsto a2a$ rejected!

Rejects any string aax

Accepts any bx

Accepted: $L(M) = \{ bx \mid x \in \{a,b\}^* \}$

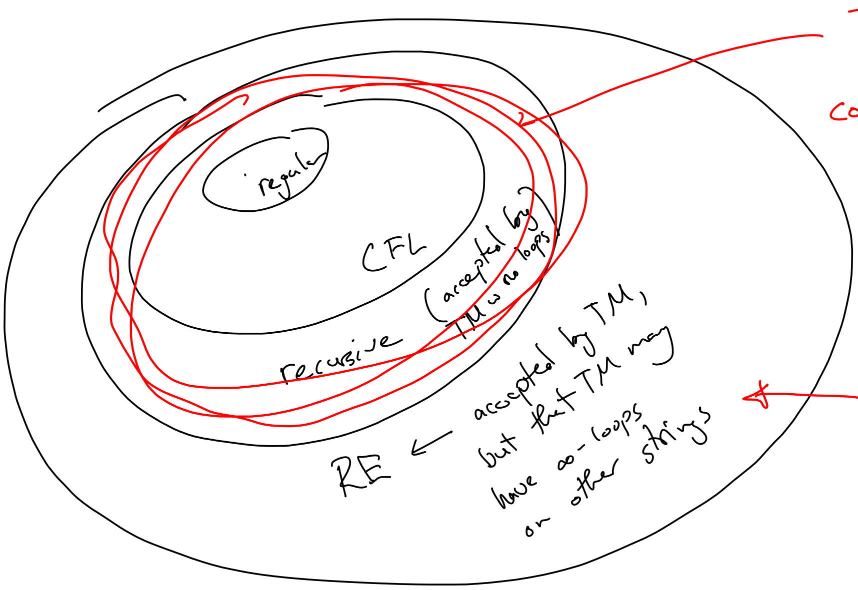
rejected: $R(M) = \{ aax \mid x \in \{a,b\}^* \} \cup \{\varepsilon, a\}$

Loop forever: $F(M) = \{ abx \mid x \in \{a,b\}^* \}$

in these cases we say the TM <u>halts</u>

<u>Def</u> If M is a TM with $F(M) = \emptyset$, then M is called a <u>total</u> <u>TM</u>.

Any language accepted by a total TM is called recursive.



Truly practical for computing answers to yes/no questions

Sometimes give good answers, but sometimes not.

L is RE means $\exists$ a TM with $L = L(M)$

but sometimes this TM may have an ∞-loop for some inputs.

Practically, it is hard to tell if a string is looping forever, or if it's taking a long time to be accepted.

L is recursive means the TM will always halt, which means we can always just wait for the answer.

TM variations

Can we add features to make a TM stronger?

How about 2 tapes?

like

a	b	a	B	B	...
B	B	B	B	B	...

2 tapes may be convenient, but they don't allow for more languages.

Any 2-tape machine can be <u>simulated</u> by a 1-tape machine

(a,B)	(b,B)	(a,B)	(B,B)	...

Allowing for 2 tapes doesn't actually make a stronger machine.

<u>Thm</u> An "n-tape TM" can be simulated by a TM.

i.e. If M_n is an n-tape machine, then $\exists$ a TM M with

$$L(M) = L(M_n)$$
$$R(M) = R(M_n)$$
$$F(M) = F(M_n)$$

Also any head movement with jumps of several locations is equiv. to an ordinary TM.

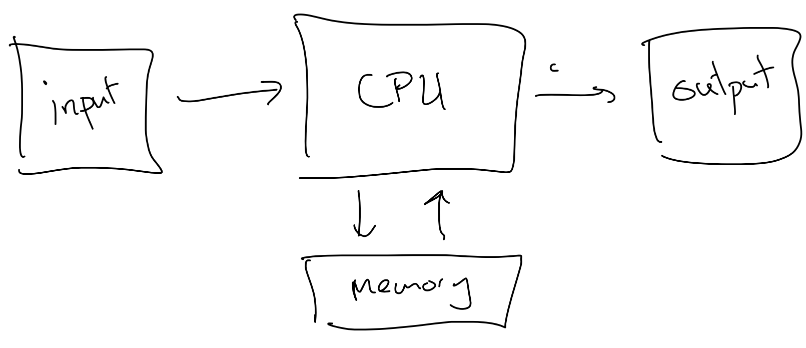
A real computer is basically a TM with many tapes, and a jumping head.

Any reasonable feature you can imagine will result in an equivalent type of machine.

The TM is "a universal machine"

Real computers have more complicated architectures

Standard model is the Von Neumann Architecture:



A V.N. machine is equivalent to a TM.

VN machines can simulate each other.

Can run a SNES on Windows,
Mac on Windows,
Linux on Linux