Note Title yonchu UNDECIDABILITY 9/16/2015

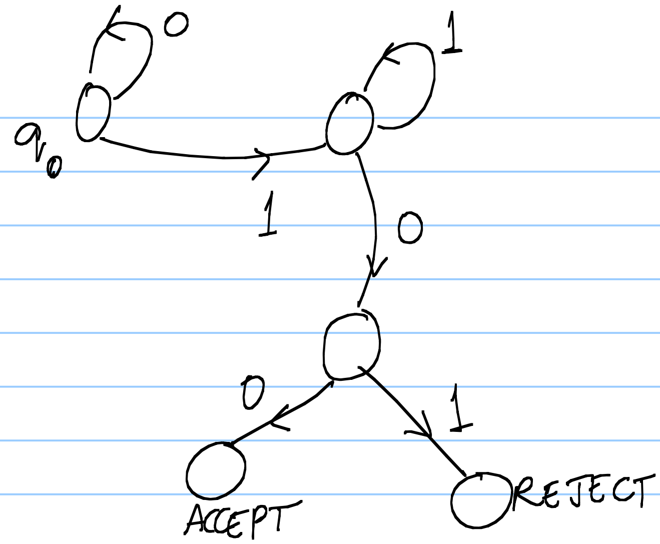
What is computation?

- Calculator, phone, laptop, desktop
  Supercomputer
- cell, body, leaf, tree, life
- atom, molecules, gas, weather, nature ...

Finite state machine (automaton)

set of states Q
input alphabet Γ including a "blank" symbol $\_$.
transition function $\delta: Q \times \Gamma \longrightarrow Q \times \Gamma$
(at state q, when reading symbol a, go to state q' and output a'.)

accept state q_{accept}
reject state q_{reject}
start state q_0



Q. What is the set of languages recognized by finite state machines?

Turing Machine

Q set of states

Σ input alphabet $\not\ni$ _

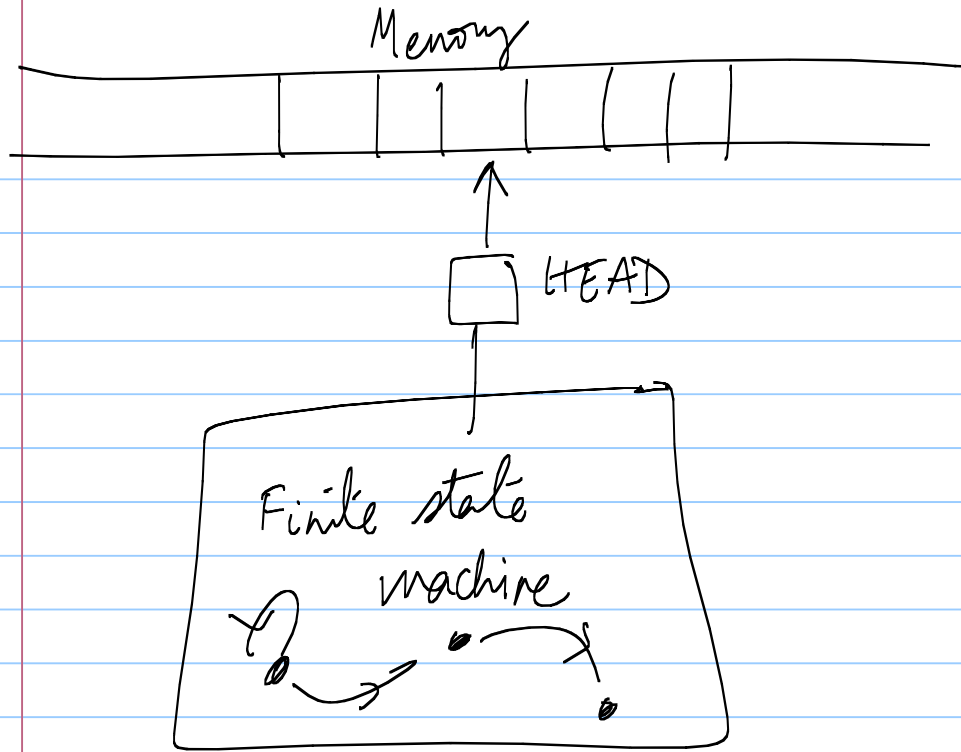
Γ tape alphabet $\ni$ _

δ transition function

$$\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$$

Tape (infinite)

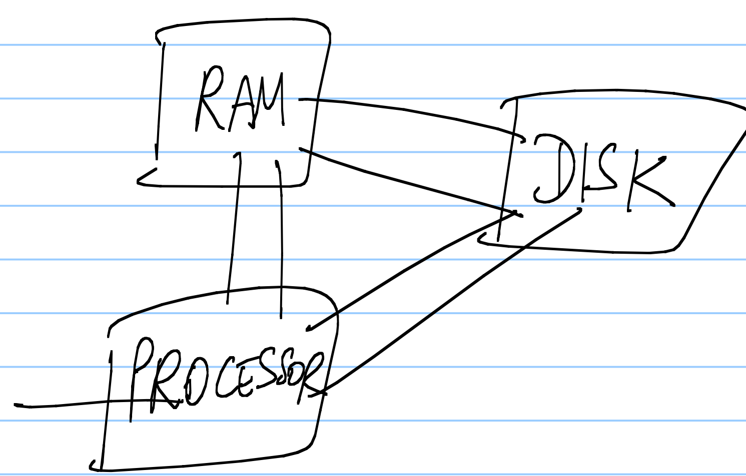
q_0, q_{ACCEPT} q_{REJECT}



Q. What languages over Σ do TMs accept?

All?

A TM can do anything a supercomputer can do.



just a bit slower.

"CHURCH-TURING thesis": Anything computable can be computed by a TM.

M accepts a set of strings
i.e. a language. L_M.

Does M accept w? $\iff$ $w \in L_M$?

— Languages are uncountable.
— What about TMs?
Countable!
$\langle M \rangle$: description of a TM
Q, δ
a string!
TMs $\subseteq$ strings, so countable.

Q. What is a language that cannot be decided by a TM?

$$L_A = \{ (\langle M \rangle, w) : \langle M \rangle \text{ is a valid TM},\ M \text{ accepts } w \}$$

Thm. $\nexists$ TM that decides L_A

Pf. Suppose TM A decides L.

On input $(\langle M\rangle, w)$

A accepts if M accepts w

A rejects if M does not accept w.

Consider TM D on input $\langle M\rangle$

1. Run A on $(\langle M\rangle, \langle M\rangle)$
2. if A accepts then reject
   if A rejects then accept

Q. What happens if we run D on $\langle D\rangle$?

accepts if D rejects $\langle D\rangle$
rejects if D accepts $\langle D\rangle$.

Contradiction! A does not exist.

Thm 2. "Does M halt on w?" undecidable

Thm 3. "Do M_1 and M_2 accept the same L?"

Thm 4. "Is $L(M) = \emptyset$?" — is also undecidable.

Pf (of Thm 2).

$L_{HALT} = \{ (\langle M \rangle, w) : \text{TM } M \text{ halts (accepts or rejects) on input } w \}$

Suppose $\exists$ TM H that decides L_{HALT}.

Then we can decide L_{ACCEPT}

A on input $(\langle M \rangle, w)$

- Runs H on $(\langle M \rangle, w)$
- if H says M does not halt, then reject

  else Run M on w; accept if M accepts, reject if M rejects

Pf. (of Thm 4)

$L_\phi = \{ \langle M \rangle : L_M = \phi \}$

(M does not accept any strings).

Suppose B is a TM that decides L_ϕ.

First consider M_w:

on input x:

- if $x \neq w$, reject

  else if $x = w$, run M on w.

Now to decide L_A,

A on input $(\langle M \rangle, w)$

- Constructs a description $\langle M_w \rangle$
- Runs B on $\langle M_w \rangle$

Reject if B accepts
Accept if B rejects

B accepts if $L_{M_w} = \phi$
M_w does not accept any x
$M \longrightarrow w$.

B rejects if $L_{M_w} \neq \phi$
M_w accepts some x
M_w accepts w
M accepts w.

NOTE: Software verification in full generality is UNDECIDABLE: Does a TM accept exactly a given set of inputs?

Other examples (we'll work out some on Friday)

Kolmogorov Complexity of a string X
min $\langle M \rangle + \langle Y \rangle$ M on input Y produces X.

$L_K = \{ x : K(x) \geq |x| \}$ is UNDECIDABLE

Rice's theorem: For any set of languages $\mathcal{L}$
Consider TMs that accept languages in $\mathcal{L}$
$L_{\mathcal{L}} = \{ \langle M \rangle : L_M \in \mathcal{L} \}$
Then $L_{\mathcal{L}}$ is either empty, all TMs or UNDECIDABLE.