February 12, 2019 10:57 AM

# Turing Machine (TM)

**Ex1** $\{0^n 1^n 2^n \mid n \geq 1\}$

**Ex2** $\{ww \mid w \in \{0,1\}^*\}$

**Ex3** $\{0^{n^2} \mid n \geq 0\}$

tricks - can copy, shift, ... (by marking)
- compose functions (subroutine)
- do loops
- keep multiple vars

etc.

**Def** Let $f: \Sigma^* \to \Sigma^* \cup \{\text{undef}\}$.

A TM $M$ computes $f$ iff

$\forall x \in \Sigma^*$, $q_0 x \overset{*}{\underset{M}{\rightsquigarrow}} q_{acc} f(x)$ if $f(x) \neq$ undef

$M$ does not accept $x$ else

Let $f: \mathbb{N} \to \mathbb{N} \cup \{\text{undef}\}$

$M$ computes $f$ iff

$\forall n \in \mathbb{N}$, $q_0 0^n \overset{*}{\underset{M}{\rightsquigarrow}} q_{acc} 0^{f(n)}$ if $f(n) \neq$ undef

not accept else

Let $f: \mathbb{N} \times \mathbb{N} \to \mathbb{N} \cup \{\text{undef}\}$

$n_1$ $n_2$ $*$ $0^{f(n_1,n_2)}$

Let $f: \mathbb{N} \times \mathbb{N}$ …

similar $q_0\, 0^{n_1} \# 0^{n_2} \overset{*}{\underset{M}{\leadsto}} q_{acc}\, 0^{f(n_1,n_2)}$ ..

Ex

$f(n) = 2n$

$f(m,n) = mn$

$f(n) = 2^n$

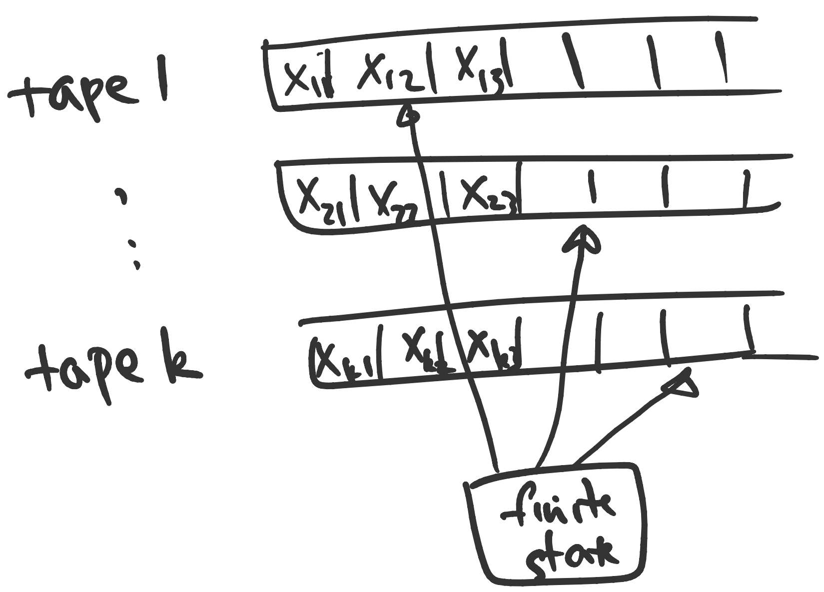
$\lfloor \log_2 n \rfloor$

Fibonacci #

⋮

## Extensions

- multitape TM




e.g. $\{0^n 1^n 2^n \mid n \geq 1\}$

can be simulated by single-tape TM

$$| X_{11} | X_{21} | \dots | X_{k1} | X_{12} | X_{22} \dots | X_{k2} | \dots$$

- nondeterministic TM

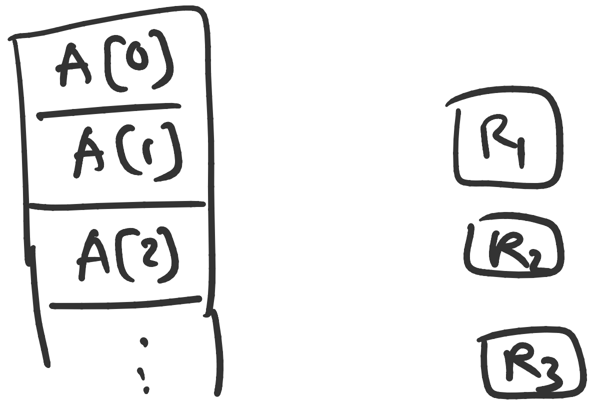
can be simulated by det. TM

idea - try all possible execution paths simultaneously

(exponential slow down)

- Random Access Machine (RAM)

memory as an array + finite # registers




instruction set:

$R_i := c$

$R_i := R_i + R_j$

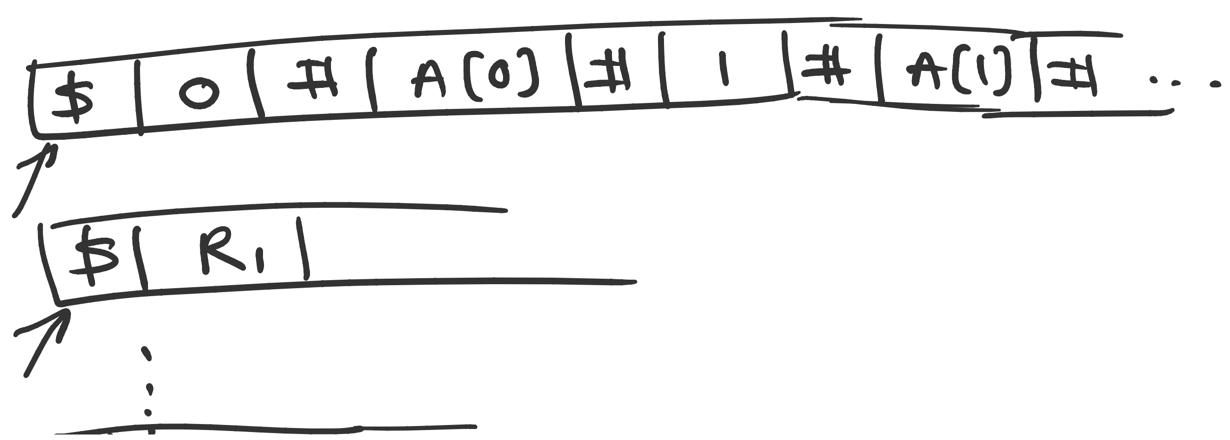
$R_i := A(R_j)$

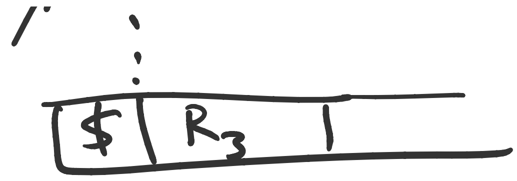
$A(R_i) := R_j$

goto line $\ell$

if $R_i = 0$ goto line $\ell$

can be simulated by TM






## Church-Turing Thesis

Any lang/function can be solved by some systematic procedure, i.e. "algorithm" iff it can be accepted/computed by a TM.

**Rmk:** "thesis" can't be proved mathematically (think of as "defn"/"axiom")

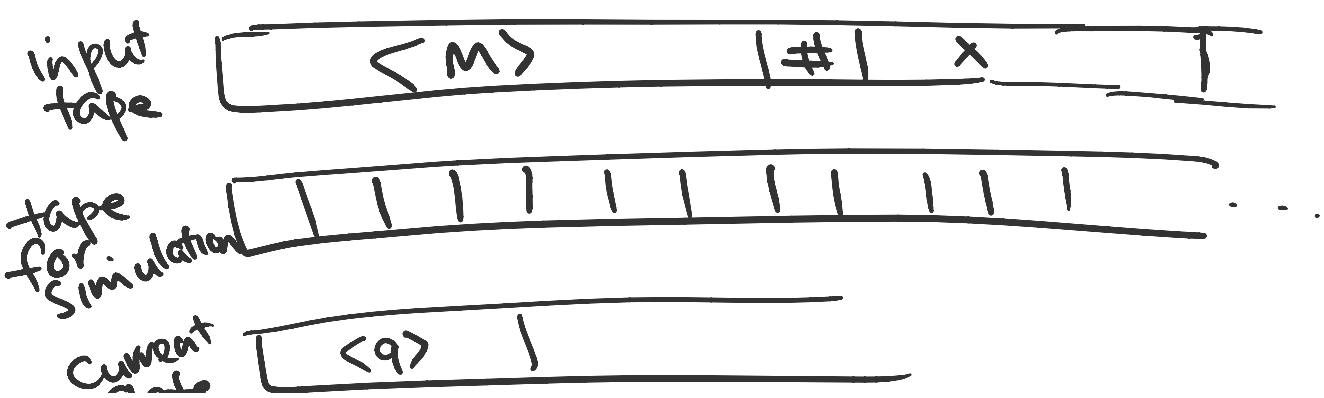
**Caveat:** assume no bounds on time & space

## A Universal TM

a TM that simulates all TMs


"stored-program computer"

let ⟨M⟩ denote code for M



Current State $\boxed{\langle q \rangle \quad}$

## An Undecidable Problem: Halting

$L_{halt} = \{ \langle M \rangle \# x \mid$ TM $M$ halts on input string $x \}$

(it is r.e.)

**Thm** (Turing '36) $L_{halt}$ is not recursive, i.e. undecidable.

**Pf:** By contradiction.

Suppose you claim to have a TM/program $M_0$ that solves $L_{halt}$.

I'll construct a counterexample: $\langle M_{bad} \rangle \# w_{bad}$

$M_{bad}$ is the following program:

On input $\langle \underline{M} \rangle$,
run $\underline{M_0}$ on $\underline{\langle M \rangle \# \langle M \rangle}$
If $M_0$ returns $\underline{yes}$,
then go to infinite loop
else $\underline{halt}$.

$w_{bad} = \langle M_{bad} \rangle$

Case 1. $M_0$ outputs yes on $\langle M_{bad} \rangle \# \langle M_{bad} \rangle$

$M_{bad}$ on input $\langle \underline{M_{bad}} \rangle$ goes to infinite loop

So $M_0$ is wrong!

So $M_0$ is wrong!

Case 2. $M_0$ outputs no on $\langle M_{bad} \rangle \# \langle M_{bad} \rangle$

$M_{bad}$ on input $\langle M_{bad} \rangle$ halts

So $M_0$ is wrong!

$\square$