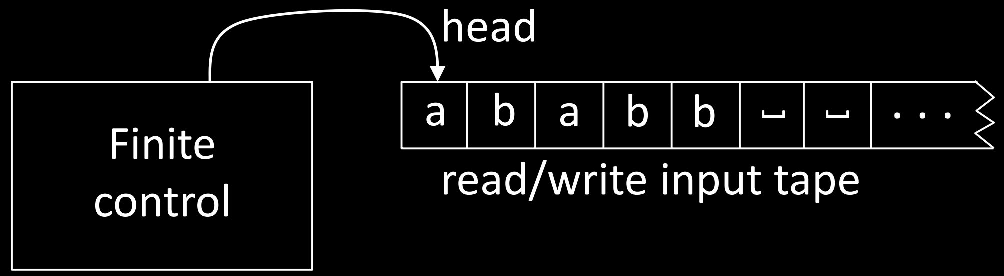
18.404/6.840 Lecture 6

Last time:
- Proving languages not Context Free
- Turing machines
- Recognizers and deciders
- T-recognizable and T-decidable languages

Today:
- Equivalence of variants of the Turing machine model
    a. Multi-tape TMs
    b. Nondeterministic TMs
    c. Enumerators
- Church-Turing Thesis
- Notation for encodings and TMs

Turing machine model – review



On input w a TM M may halt (enter q_{acc} or q_{rej}) or loop (run forever).

So M has 3 possible outcomes for each input w:

1. *Accept* w (enter q_{acc})
2. *Reject* w by halting (enter q_{rej})
3. *Reject* w by looping (running forever)

A is T-recognizable if $A = L(M)$ for some TM M.

A is T-decidable if $A = L(M)$ for some TM decider M.

halts on all inputs

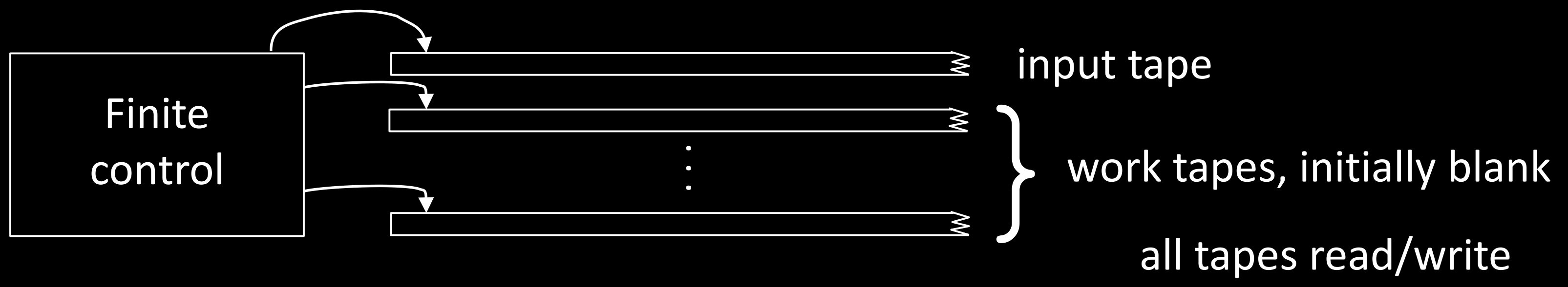
Turing machines model general-purpose computation.

Q: Why pick this model?

A: Choice of model doesn't matter.
All reasonable models are equivalent in power.

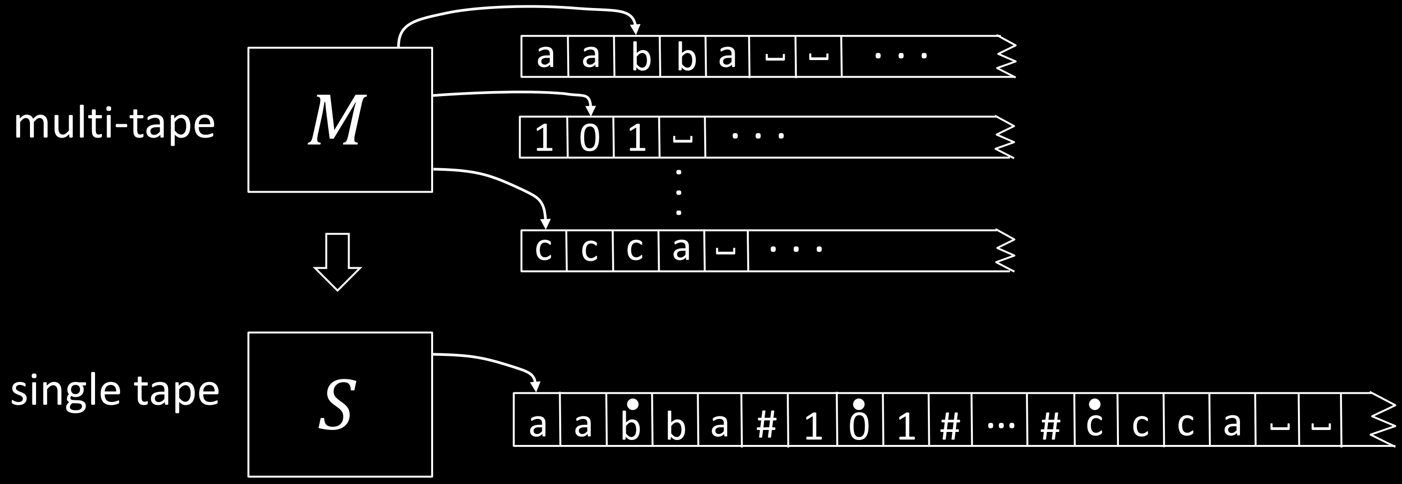
Virtues of TMs: simplicity, familiarity.

Multi-tape Turing machines



Theorem: A is T-recognizable iff some multi-tape TM recognizes A

Proof: $(\rightarrow)$ immediate. $(\leftarrow)$ convert multi-tape to single tape:



S simulates M by storing the contents of multiple tapes on a single tape in "blocks". Record head positions with dotted symbols.

Some details of S:

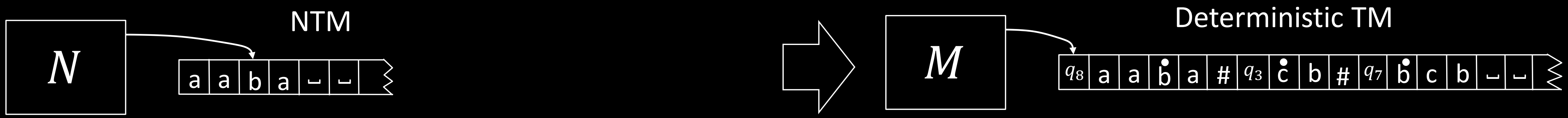
1) To simulate each of M's steps
   a. Scan entire tape to find dotted symbols.
   b. Scan again to update according to M's δ.
   c. Shift to add room as needed.
2) Accept/reject if M does.

Nondeterministic Turing machines

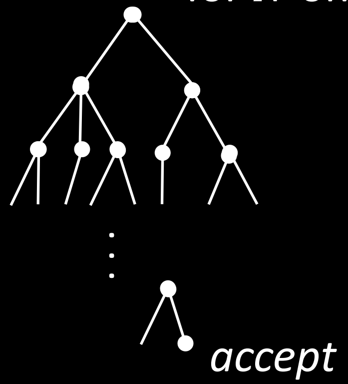
A <u>Nondeterministic TM</u> (NTM) is similar to a Deterministic TM except for its transition function $\delta\colon \mathrm{Q} \times \Gamma \to \mathcal{P}(\, Q \times \Gamma \times \{\mathrm{L, R}\}\,)$.

Theorem: A is T-recognizable iff some NTM recognizes A

Proof: $(\rightarrow)$ immediate. $(\leftarrow)$ convert NTM to Deterministic TM.



Nondeterministic computation tree for N on input w.

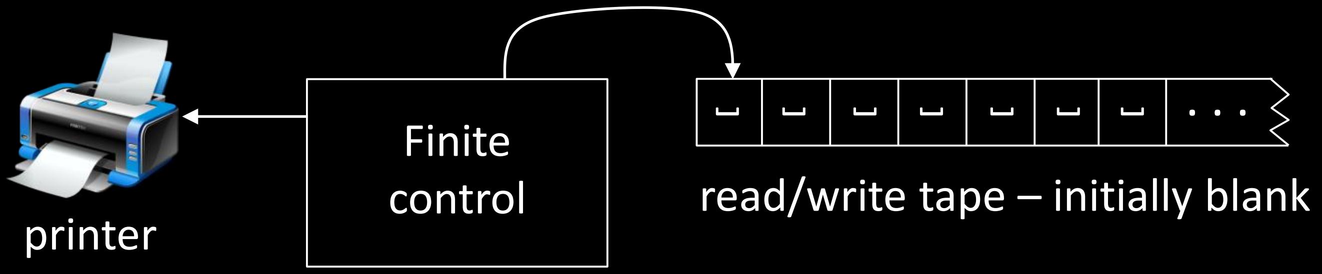


M simulates N by storing each thread's tape in a separate "block" on its tape.
Also need to store the head location, and the state for each thread, in the block.

If a thread forks, then M copies the block.

If a thread accepts then M accepts.

Turing Enumerators



Defn: A Turing Enumerator is a deterministic TM with a printer.

It starts on a blank tape and it can print strings $w_1, w_2, w_3, \ldots$ possibly going forever.

Its language is the set of all strings it prints. It is a generator, not a recognizer.

For enumerator E we say $L(E) = \{w \mid E \text{ prints } w\}$.

Theorem: A is T-recognizable iff $A = L(E)$ for some T-enumerator E.

Check-in 6.1

When converting TM M to enumerator E, does E always print the strings in ***string order***?

a) Yes.

b) No.

Proof: ($\rightarrow$) Convert TM M to equivalent enumerator E.

$E =$ Simulate M on each w_i in $\Sigma^* = \{\varepsilon, 0,1,00,01,10, \ldots\}$

If M accepts w_i then print w_i .

Continue with next w_i .

Problem: What if M on w_i loops?

Fix: Simulate M on $w_1, w_2, \ldots, w_i$ for i steps, for $i = 1,2, \ldots$

Print those w_i which are accepted.

Check-in 6.1


Coffee Break

Teach at Splash!

esp.mit.edu/splash20

Splash is an annual teaching and learning extravaganza, brought to you by MIT ESP!

When?	November 14-15
Where?	Virtual
What?	Teach anything! Any topic, length, or class size!
Who?	Teach thousands of curious and motivated high schoolers

Church-Turing Thesis ~1936

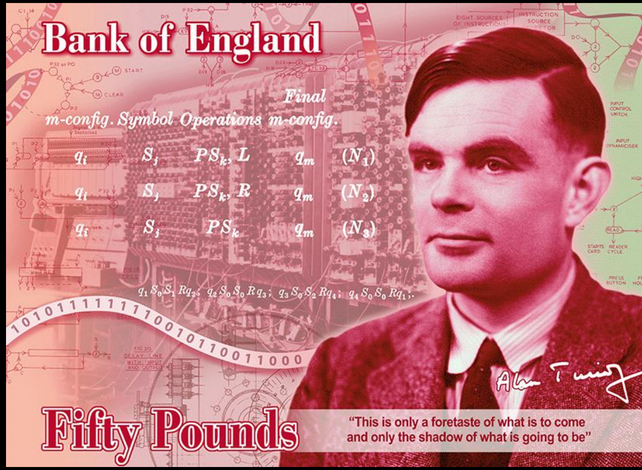
Alonzo Church
1903–1995

Algorithm	=	Turing machine
Intuitive		Formal

Instead of Turing machines,
can use any other “reasonable” model

Alan Turing
1912–1954

Will appear in 2021



Check-in 6.2

Which is the following is true about Alan Turing? Check all that apply.

a) Broke codes for England during WW2.
b) Worked in AI.
c) Worked in Biology.
d) Was imprisoned for being gay.
e) Appears on a British banknote.

Check-in 6.2

Hilbert's 10th Problem

In 1900 David Hilbert posed 23 problems

#1) Problem of the continuum (Does set A exist where $|\mathbb{N}| < |A| < |\mathbb{R}|$?).

#2) Prove that the axioms of mathematics are consistent.

#10) Give an algorithm for solving *Diophantine equations.*

Diophantine equations:

Equations of polynomials where solutions must be integers.

Example: $3x^2 - 2xy - y^2z = 7$ solution: $x = 1,\ y = 2,\ z = -2$

Let $D = \{p|$ polynomial $p(x_1, x_2, \ldots, x_k) = 0$ has a solution in integers$)$

Hilbert's 10th problem: Give an algorithm to decide D.

Matiyasevich proved in 1970: D is not decidable.

Note: D is T-recognizable.

David Hilbert
1862—1943

Notation for encodings and TMs

Notation for encoding objects into strings

- If O is some object (e.g., polynomial, automaton, graph, etc.), we write $\langle O \rangle$ to be an encoding of that object into a string.

- If $O_1, O_2, \ldots, O_k$ is a list of objects then we write $\langle O_1, O_2, \ldots, O_k \rangle$ to be an encoding of them together into a single string.

Notation for writi

We will use high-level ,
knowing that we coul
transition function, et

$M =$ "On input w

[English description of the algorithm]"

Check-in 6.3

If x and y are strings, would xy be a good choice for their encoding $\langle x, y \rangle$ into a single string?

a) Yes.
b) No.

Check-in 6.3

TM – example revisited

TM M recognizing $B = \{\mathrm{a}^k\mathrm{b}^k\mathrm{c}^k \mid k \geq 0\}$

$M =$ "On input w

1. Check if $w \in \mathrm{a}^*\mathrm{b}^*\mathrm{c}^*$, *reject* if not.
2. Count the number of a's, b's, and c's in w.
3. *Accept* if all counts are equal; *reject* if not."

High-level description is ok.
You do not need to manage tapes, states, etc…

Problem Set 2

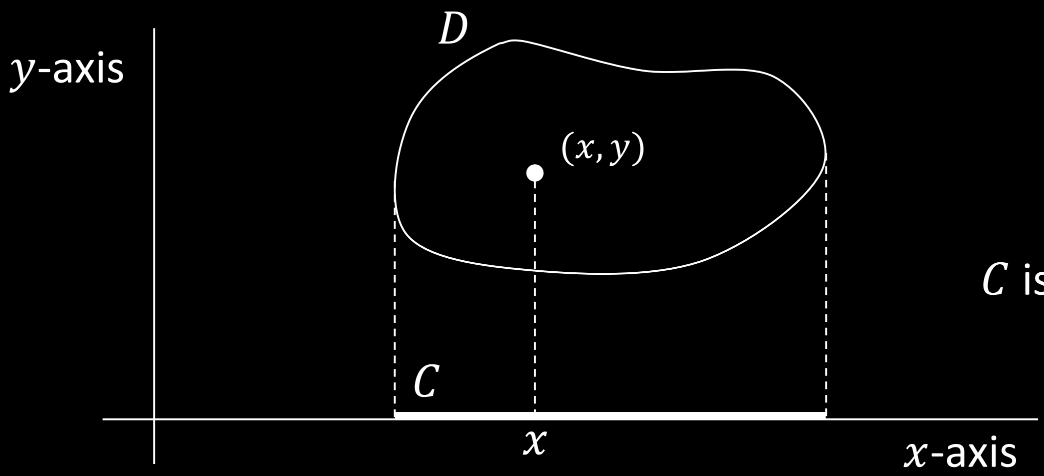
$\longleftarrow$

#5) Show C is T-recognizable iff there is a decidable D where

$$C = \{\, x \mid \exists y \ \ \langle x, y \rangle \in D \,\} \qquad x, y \in \Sigma^*$$

$\langle x, y \rangle$ is an encoding of the pair of strings x and y into a single string.

Think of D as a collection of pairs of strings.



C is a “projection” of D

Quick review of today

1. We showed that various TM variants (multi-tape, nondeterministic, enumerator) are all equivalent to the single-tape model.
2. Concluded that all “reasonable” models with unrestricted memory access are equivalent.
3. Discussed the Church-Turing Thesis:
   Turing machines are equivalent to “algorithms”.
4. Notation for encoding objects and describing TMs.
5. Discussed Pset 2 Problem 5.