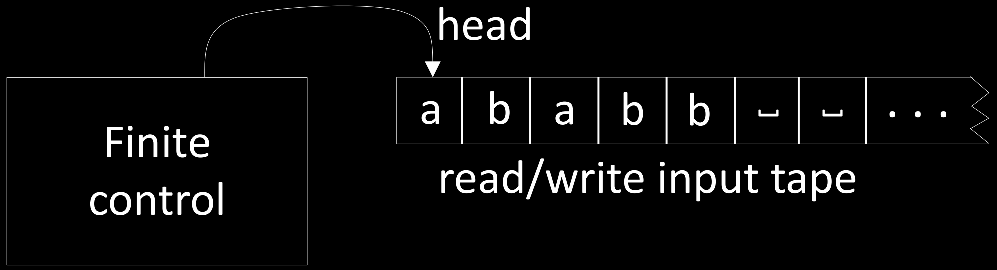
18.404/6.840 Lecture 6

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 a TM may halt (enter or)
or loop (run forever).

So has 3 possible outcomes for each input :

1. *Accept* (enter)
2. *Reject* by halting (enter)
3. *Reject* by looping (running forever)

 is T-recognizable if for some TM .

 is T-decidable if for some TM decider .

halts on all inputs

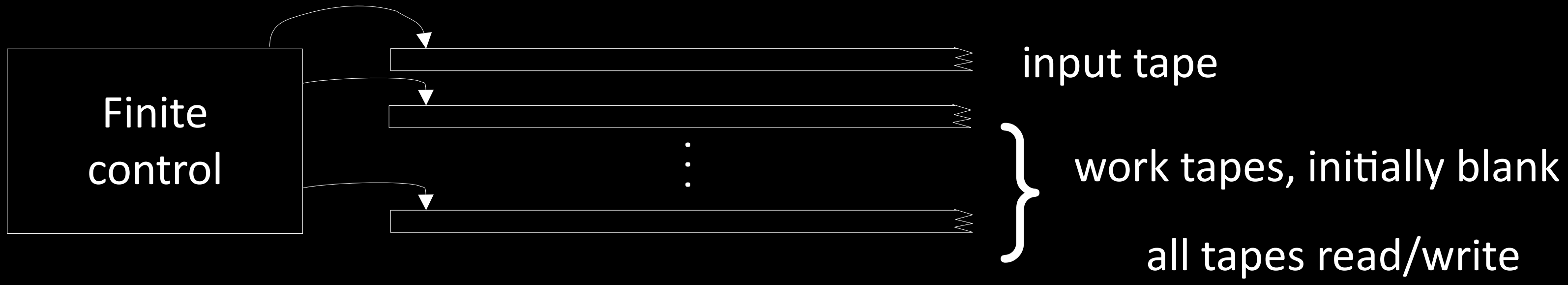
Turing machines model general-purpose computation.

Q: Why pick this model?

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

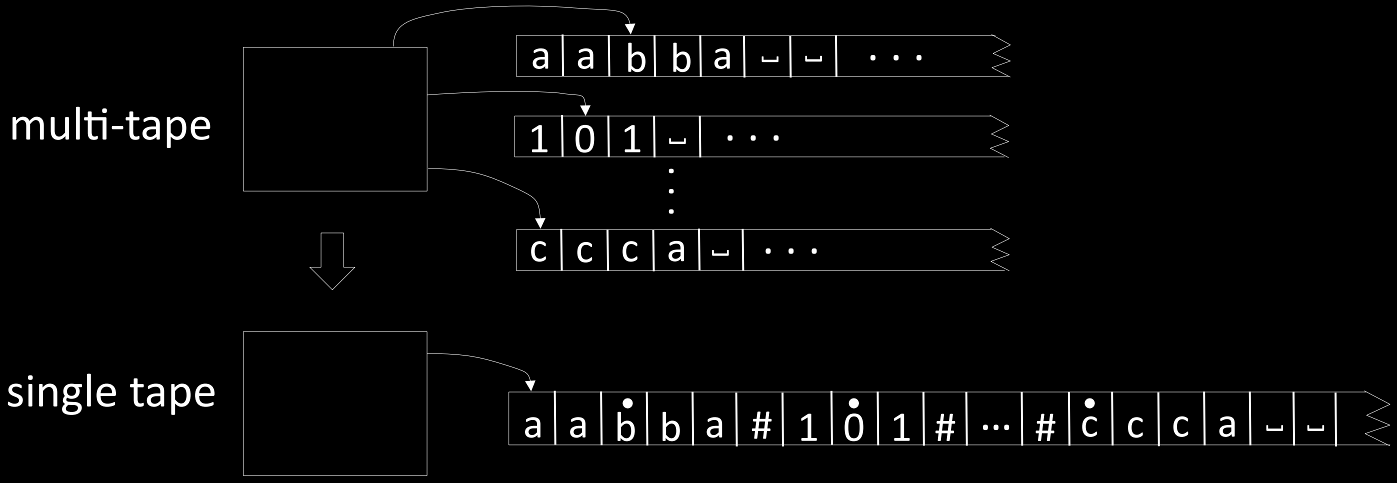
Virtues of TMs: simplicity, familiarity.

Multi-tape Turing machines



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

Proof: immediate. convert multi-tape to single tape:



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

Some details of :

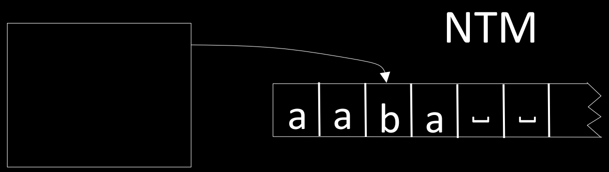
1) To simulate each of 's steps
   a. Scan entire tape to find dotted symbols.
   b. Scan again to update according to 's .
   c. Shift to add room as needed.
2) Accept/reject if does.

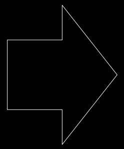
Nondeterministic Turing machines

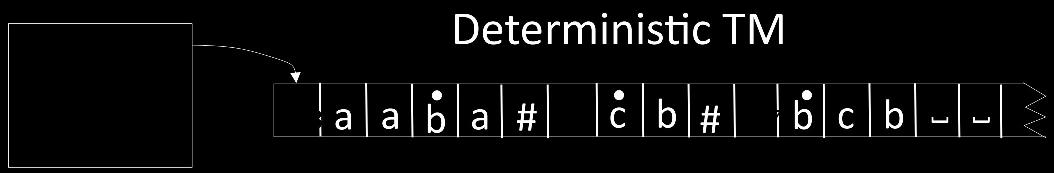
A <u>Nondeterministic TM</u> (NTM) is similar to a Deterministic TM
except for its transition function   {L, R} .

Theorem: is T-recognizable iff some NTM recognizes

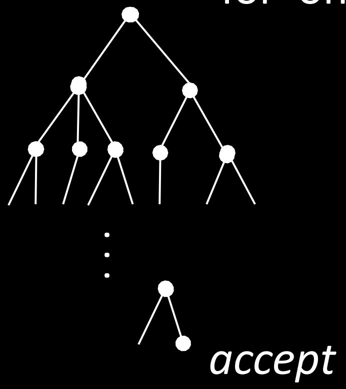
Proof: immediate.   convert NTM to Deterministic TM.







Nondeterministic computation tree
for on input .

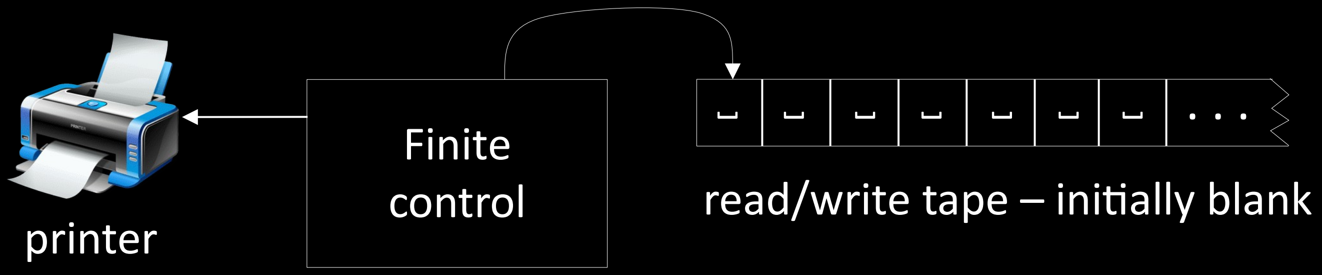


 simulates by storing each thread's tape in a separate "block" on its tape.
Also need to store the head location,
and the state for each thread, in the block.

If a thread forks, then copies the block.

If a thread accepts then accepts.

Turing Enumerators



Defn: A <u>Turing Enumerator</u> is a deterministic TM with a printer.
It starts on a blank tape and it can print strings possibly going forever.
Its language is the set of all strings it prints. It is a generator, not a recognizer.
For enumerator we say prints .

Theorem: A is T-recognizable iff for some T-enumerator .

Check-in 6.1
When converting TM to enumerator ,
does always print the strings in ***string order***?
a) Yes.
b) No.

Proof: () Convert TM to equivalent enumerator .
Simulate on each in
If accepts then print .
Continue with next .
Problem: What if on loops?
Fix: Simulate on for steps, for
Print those 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

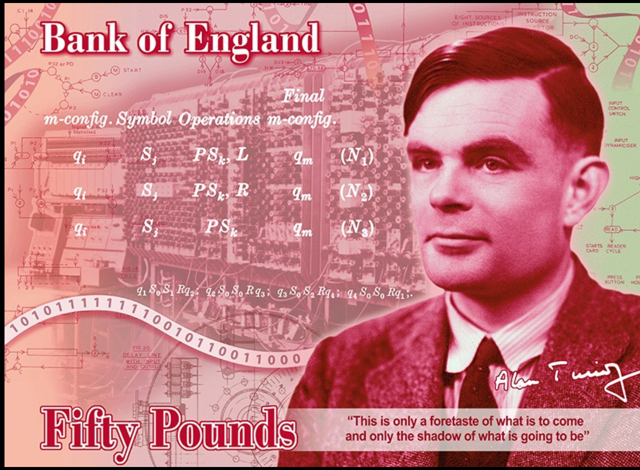
Algorithm = Turing machine

Intuitive — Formal

Alan Turing
1912–1954

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

Will appear in 2021


Bank of England
Fifty Pounds
“This is only a foretaste of what is to come and only the shadow of what is going to be”

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 exist where ?).

#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: solution:

Let polynomial has a solution in integers)

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

Matiyasevich proved in 1970: is not decidable.

Note: is T-recognizable.

David Hilbert
1862—1943

Notation for encodings and TMs

Notation for encoding objects into strings

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

- If is a list of objects then we write
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

“On input

[English description of the algorithm]”

Check-in 6.3

If and are strings, would be a good choice
for their encoding into a single string?

a) Yes.

b) No.

Check-in 6.3

TM – example revisited

TM recognizing

“On input

1. Check if , *reject* if not.
2. Count the number of ’s, b’s, and c’s in .
3. *Accept* if all counts are equal; *reject* if not.”

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

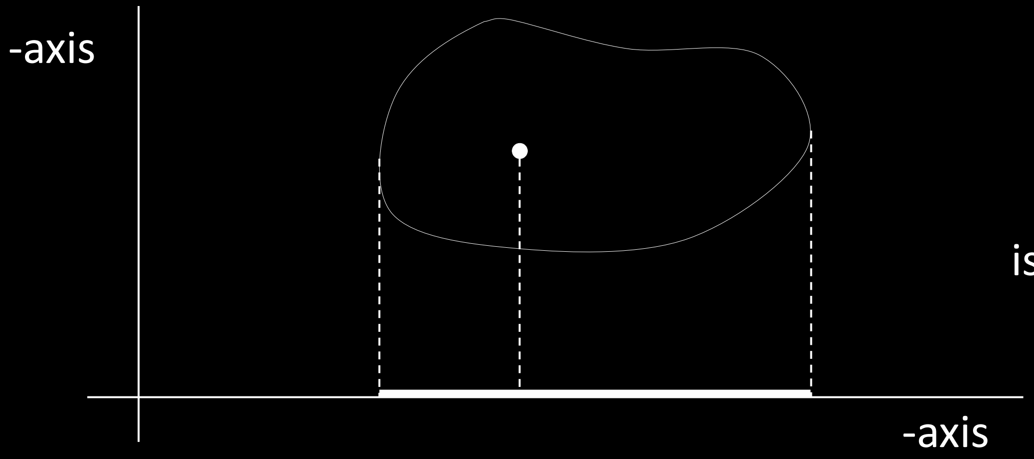
Problem Set 2

$\leftarrow$

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

is an encoding of the pair of strings and into a single string.

Think of as a collection of pairs of strings.



is a “projection” of

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.