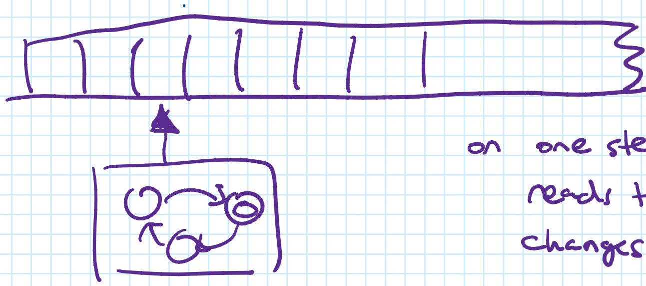
Lecture 9

Tuesday, 23 February, 2021 10:47

Today: A closer look at TMs

- Some basic subroutines
- the precise model doesn't matter b/c simulation
- TMs are code
- TMs can run code (universal TMs)

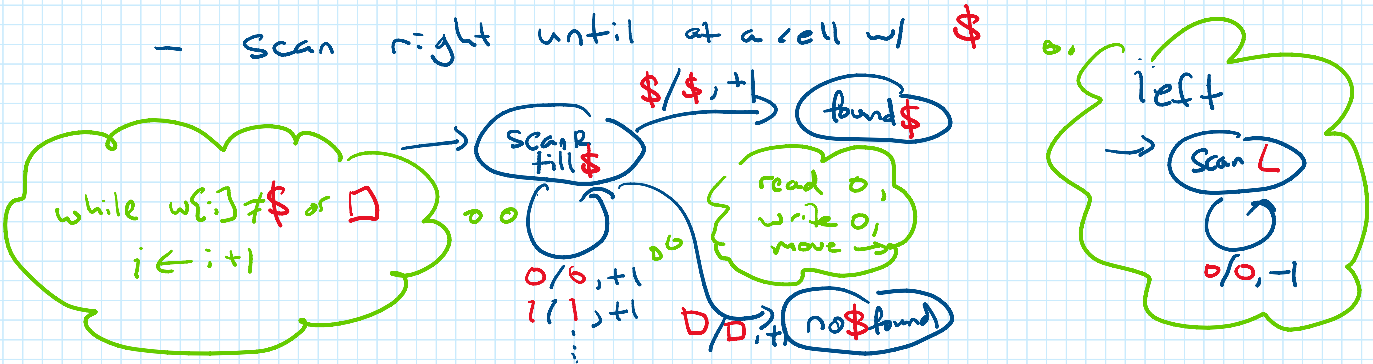
Recall:



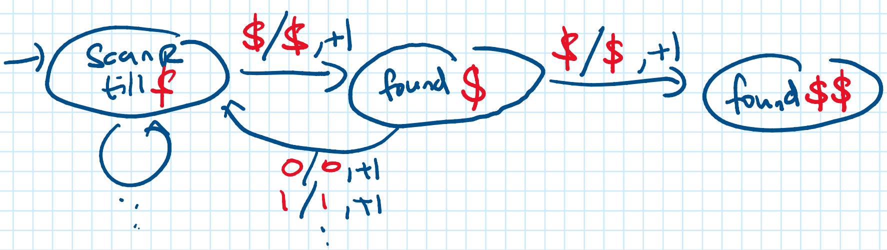
$\Gamma = \{\underbrace{0, 1}_{\Sigma}, \overset{\text{blank}}{\square}, \underbrace{x, \$}_{\text{extra TM symbols}}\}$

on one step:
reads tape, writes to tape
changes state, moves ←/→.

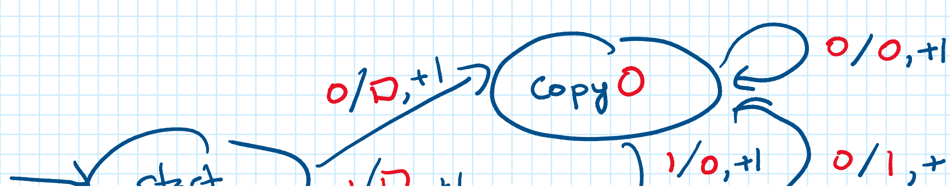
- scan right until at a cell w/ \$

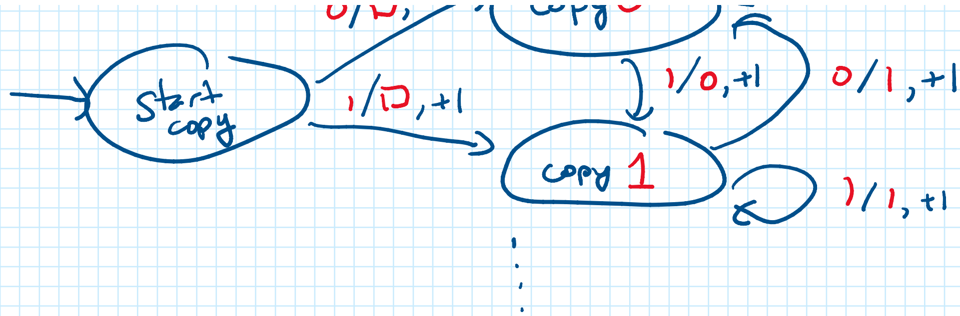


until \$\$



- shift everything right.





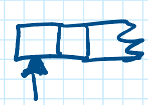
- keep a binary counter

  idea: - find the least significant bit
  - flip it &
    if 1, move to "carry" state.
  - or if carrying, move on.

→ "assume" that I can say things like
the TM will find the first 0
↳ failsafe: on □
move to no 0 found state.

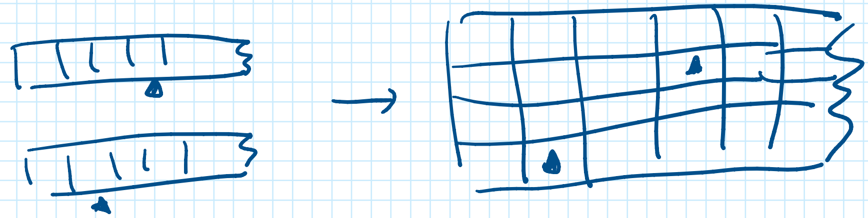
Many variations on TMs:

- in this class: defined TM to have 1 tape,
  infinite to the right.

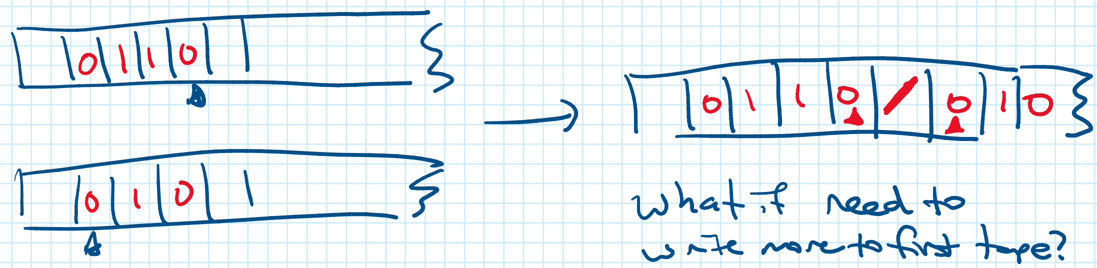
  Q: what if you move ← from  ?
  crash!
  But... don't worry: always first shift everything →.
  so doesn't really matter.

- some people define TMs to have multiple tapes.

  one workaround: expand Γ to have "columns" one workaround: expand Γ to have "columns" of symbols

- 

to simulate multiple tapes:
for each tape, scan for ▲ for that tape & execute 1 step.

- 

what if need to write more to first tape?

shift everything else $\rightarrow$.

a lot of this is bookkeeping on how your data is being stored

---

Aside: $L = \{ w \mid w \text{ is a regex} \}$ is not regular.

but it is context-free. $\rightarrow$ exercise

Q: is $L = \{ w \mid w \text{ is a TM} \}$ decidable by a TM?

- understand what "w is a TM" means
- if a TM can tell if "w is a TM" or (exec() in python etc.)

— if a TM can tell if "w is a TM" can we execute the TM described by w?

or exec() in python etc.

Universal Turing Machines

once again the precise model doesn't matter, because simulation a lot of bookkeeping

Suppose given

$$M = (Q, \Gamma, start, acc, rej, \delta)$$

$\Gamma = \{0, 1, \square, x, \$\}$

create encoding over some other alphabet

$$\{\varnothing, 1, [,], \bullet, \underset{\blacktriangle}{\varnothing}, \underset{\blacktriangle}{1}, \boxed{\square}\}$$

make some arbitrary encoding decisions

$\langle \square \rangle = \varnothing\varnothing\varnothing$

$\langle 0 \rangle = \varnothing\varnothing 1$

$\langle 1 \rangle = \varnothing 1 \varnothing$

$\langle x \rangle = \varnothing 1 1$

$\langle \$ \rangle = 1 \varnothing \varnothing$

$\langle start \rangle = \underbrace{111}_{\log|Q| \text{ bits}}$

$\vdots$

$\langle acc \rangle = \varnothing\varnothing 1$

$\langle rej \rangle = \varnothing\varnothing\varnothing$

$$\langle 001 \rangle = [\varnothing\varnothing 1 \bullet \varnothing\varnothing 1 \bullet \varnothing 1 \varnothing]$$

$$\langle p \xrightarrow{a/b,\Delta} q \rangle = [[\langle p \rangle \bullet \langle a \rangle \bullet \langle b \rangle \bullet \langle q \rangle \bullet \langle \Delta \rangle]]$$

$\langle \delta \rangle$ = concat of all transitions.

$$\langle M \rangle = [\langle rej \rangle \bullet \langle \square \rangle][\langle \delta \rangle]$$

this allows us to <u>sketch</u> a TM that takes in

$\langle M \rangle \bullet \langle w \rangle$ and then <u>run</u> M on input w.

UTM w/ 3 tapes

- input tape (won't modify it)
- state tape remembers current state of M
- work tape all computations here.

- initialization:

  copy $\langle w \rangle$ to work tape.

  scan on input until start of $\langle w \rangle$.
  copy: read on input, write same thing in work, move $\rightarrow$

  mark first symbol of $\langle w \rangle$

  $0 \rightarrow \underset{\blacktriangle}{0}$, $1 \rightarrow \underset{\blacktriangle}{1}$

  copy $\overset{111}{\langle start \rangle}$ to state tape

- loop:
  - find marked char on work tape.

    $\rightarrow$ gives us $\langle \underset{\blacktriangle}{a} \rangle$

  - Scan input tape for $[[\langle p \rangle \cdot \langle a \rangle \cdot \langle b \rangle \cdot \langle q \rangle \cdot \langle \Delta \rangle]]$

    where $\langle p \rangle$ is on state tape.

    - change $\langle \underset{\blacktriangle}{a} \rangle$ to $\langle b \rangle$ on work tape
    - change $\langle p \rangle$ to $\langle q \rangle$ on state tape
    - mark the beginning of the encoding of next symbol { depends on Δ

  - If state tape is $\langle acc \rangle$ or $\langle rej \rangle$

    accept / reject accordingly.

TM for simulating assembly +RAM?

given: encoding of set of assembly instructions.

- keep track of what line we're at via a binary counter tape.
- Scan until the right line by counting on another tape
- "remember" instruction read on instruction tape
- run whatever instruction on RAM tape

$\boxed{\text{We are all Universal Turing Machines}}$

X

Strictly optional for today for the curious will covered again in April/May

Bonus Halting problem & Friends

Define: L is decidable if $\exists$ TM M s.t.

$\forall w \in L$, M accepts
$w \notin L$, M rejects

never infinite loops

$\text{SELFREJECT} = \{\langle M \rangle \mid M \text{ } \underline{\text{rejects}} \text{ } \langle M \rangle\}$.

Suppose $\exists$ TM M_{SR} that decides SELFREJECT.

so
- accepts $\langle M \rangle \in$ SELFREJECT
- rejects $\langle M \rangle \notin$ SELFREJECT
- never loops

M_{SR} accepts $\langle M \rangle \iff M$ rejects $\langle M \rangle$

plug in $\langle M_{SR} \rangle$ for $\langle M \rangle$

$\downarrow$

M_{SR} accepts $\langle M_{SR} \rangle \iff M_{SR}$ rejects $\langle M_{SR} \rangle$.

contradiction!

$\rightarrow$ no TM deciding SELFREJECT! It's undecidable.

$\text{SELFHALT} = \{\langle M \rangle \mid M \text{ halts on } \langle M \rangle\}$

SELFHALT $= \{ \langle M \rangle \mid M$ halts on $\langle M \rangle \}$

ends in acc or rej state

Suppose $\exists$ TM M_{SH} deciding SELFHALT

$M_{SH} \rightarrow \overline{M_{SH}}$ by changing all transitions to acc to a new loop state

all transitions to rej to acc.

$\overline{M_{SH}}$ accepts $\langle M \rangle \Leftrightarrow M_{SH}$ rejects $\langle M \rangle \Leftrightarrow M$ does not halt on $\langle M \rangle$

plug in $\overline{M_{SH}}$

$\overline{M_{SH}}$ accepts $\langle \overline{M_{SH}} \rangle \Leftrightarrow \overline{M_{SH}}$ does not halt on $\langle \overline{M_{SH}} \rangle$.

$\Downarrow$

halt

contradiction $\therefore$ no TM deciding SELFHALT!