Announcements

- checkup 3 reopened (checkup 4: next week)
- HW2 out: due Mon night
- Handout 2 out soon
- Quiz 1: will email when ready
  - Full grading by teaching staff ← what counts
    + separately by Grady ← for feedback/experiment
  - we'll ask your feedback too via anonymous form

Last time:

- Quiz 1! ⇝ Note: bonus problem not meant as regular part of quiz ok if you didn't get to it. (also no IDK/a bit stricter grading for the 5 point bonus)
- Wrapped up regular languages
- Discussion/motivation for where we are in class + next topic: Computability

Computability

Our next model of computation: Turing Machine (TM)

What was missing in DFA that your laptop has?

- memory: DFA has a fixed amount of memory, can't, e.g., implement a counter
- read left to right (can't go back & forth)
- read-only (no writing allowed)

→ TM model will address all these

- unbounded memory
- read+write access
- move right+left

Other things brought up in class discussions:

- longer output (rather than accept/reject)?
- efficiency/time

→ • will be addressed by input output TM but used mostly as "helper function" to describe algorithm more modularly. We'll still focus on decision problems (= yes/no questions = language membership)

TM model will not directly address these.

However, all these turn out to not matter if you only care about what's possible/impossible to solve.

i.e: doesn't change the computational power of the model

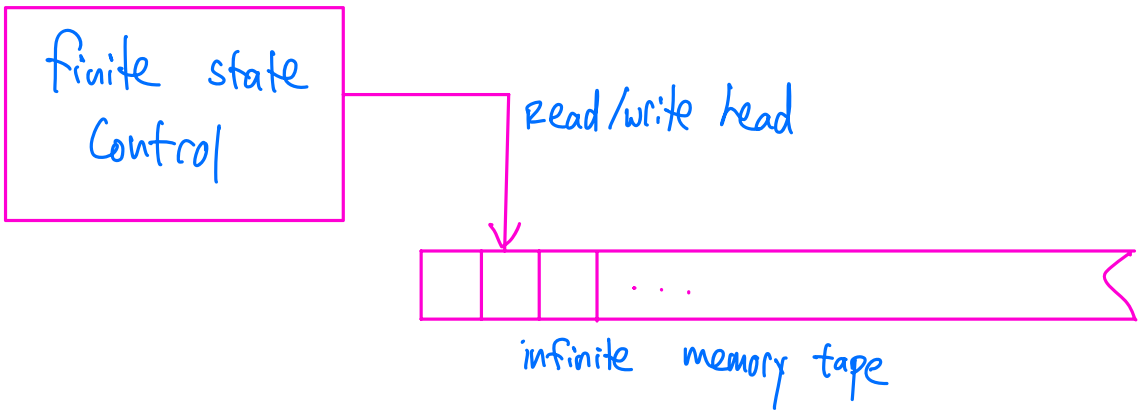
important question we are focusing on!

- Last unit of class will focus on efficiency ("easy vs hard") where these can be addressed!

Today: TM definition, examples

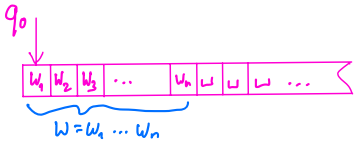
TM model will address all these
- unbounded memory
- read+write access
- move right+left

Turing Machine, informally:



Computation on input $w \in \Sigma^*$:

(initial) starting configuration:

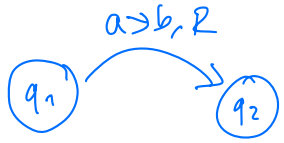


- start in start state q_0
- head points to leftmost cell on tape
- tape contains w followed by infinitely many 'blank' symbols '␣'

Each Computation step:

- Depending on
  - current state
  - symbol read from current cell on tape

Do:
- switch to state
- write on tape (overwriting current symbol)
- move head right or left



..... if you're in q_1, see 'a' on tape
then overwrite 'a' with 'b', move right,
go to state q_2

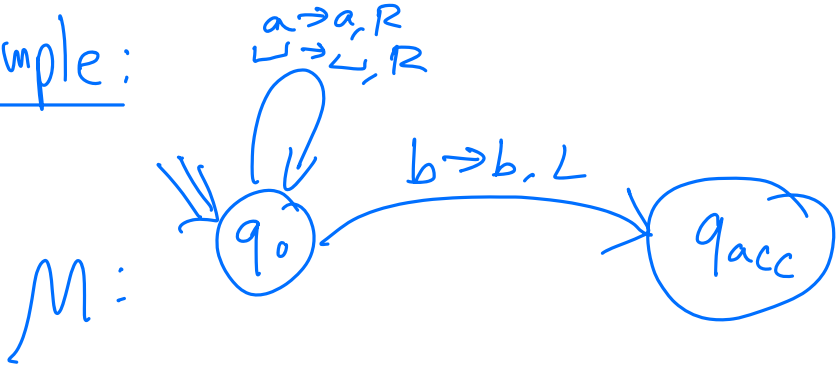
- two special states q_{acc}, q_{rej}:

  if you ever switch into one of them, immediatly halt.

  ↙ different than DFA/NFA!

Note: if head points to leftmost cell and instructed to move left - stays on leftmost cell.

Example:



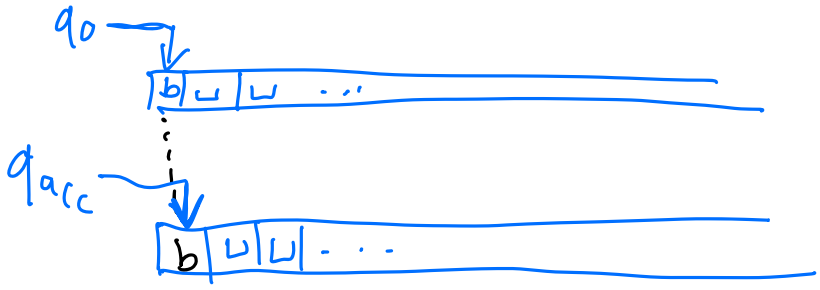
M:

$\Sigma = \{a, b\}$

$L(M) = ?$

Co

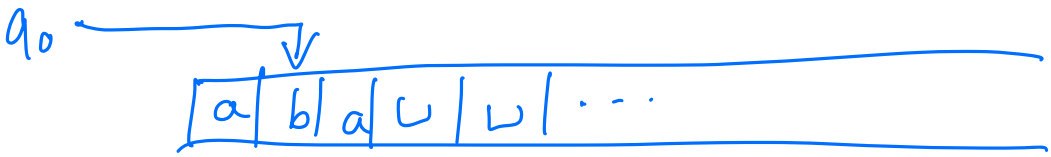
Computation of M on 'b':

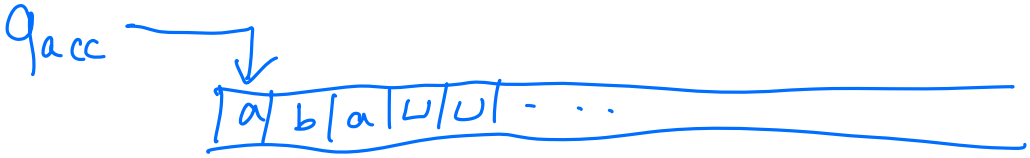


accepted! $b \in L(M)$

Computation of M on 'aba':







accepted! $aba \in L(M)$

Configurations:

$q_0 aba$

$\vdash$

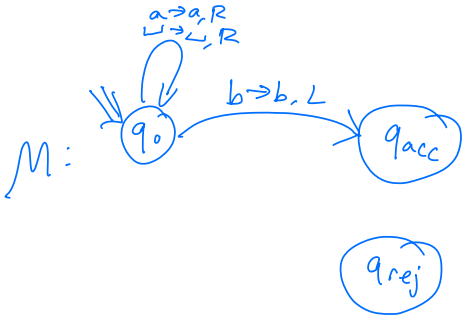
$a q_0 ba$

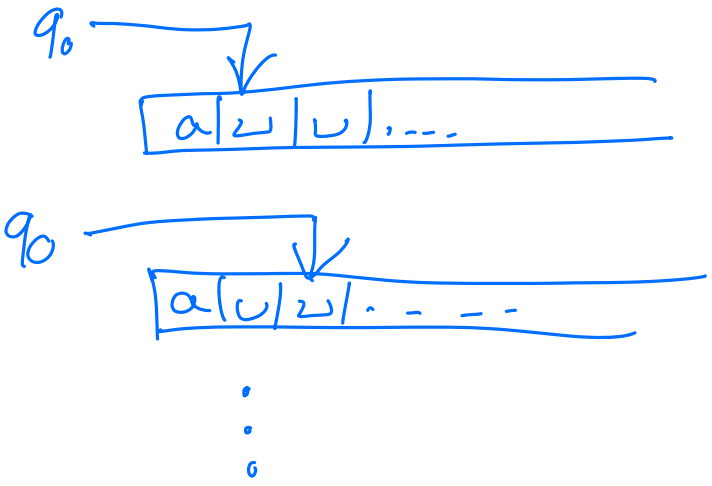
$\vdash$

$q_{acc} aba$

Computation of M on 'a':







never halts! infinite computation (keeps moving right, forever): $a \notin L(M)$

This M corresponds to the algorithm:

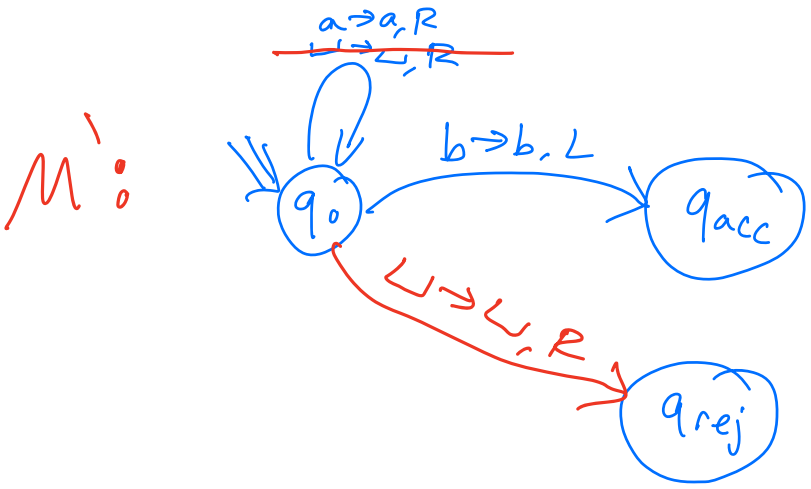
"On input w:
- if w contains 'b', accept
- else, run forever "
  ↳ (keep going right)

$L(M) = \{ w \in \{a,b\}^* \mid M \text{ accepts } w \}$

$= \{ w \in \{a,b\}^* \mid w \text{ contains } b \}$

Another M' for same language:

M':



$L(M') = \{w \mid w \text{ contains 'b'}\}$

This M' corresponds to the alg:

"On input w:
- if w contains 'b', accept
- else, reject. "

Or at a more detailed level:

"On input w:
- go over tape left to right.
  - if you encounter 'b': accept
  - if you encounter '␣': reject"

Def: A **Turing Machine (TM)** is a 7-tuple

$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$ where:

- Q: finite set of states
- Σ: (finite) alphabet
- Γ: (finite) alphabet called "tape alphabet"
  s.t. $\Sigma \subseteq \Gamma$
  $\sqcup \in \Gamma$
  $\sqcup \notin \Sigma$

- $q_0, q_{acc}, q_{rej} \in Q$, called start state, accept state, reject state, respectively

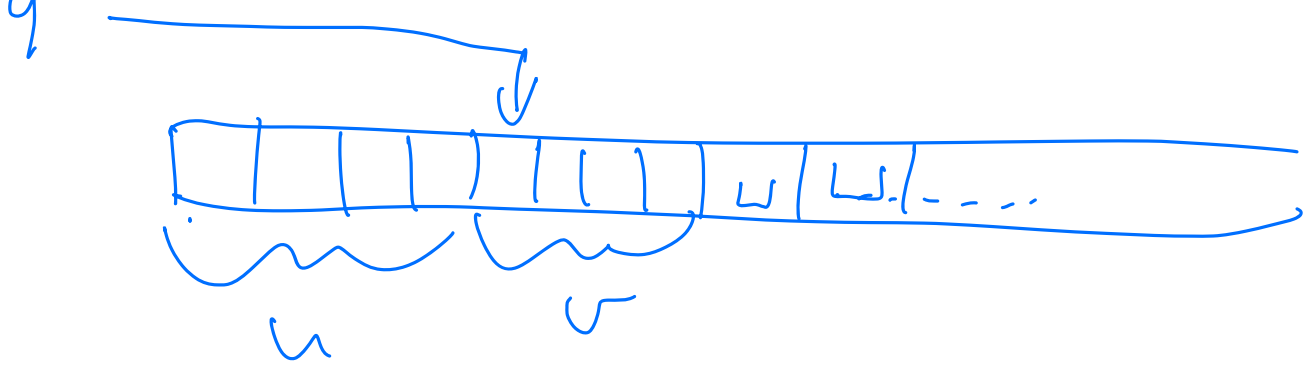
- $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$

δ: transition function
Q: current state
Γ: current symbol read
Q: state to go to
Γ: symbol to write
$\{L, R\}$: direction to move

Def: Let $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$ be a TM. A configuration of M is of the form $\boxed{uqv}$ where $u, v \in \Gamma^*$, $q \in Q$

This represents the current status of computation:

- Content of tape is uv followed by ∞ many $\sqcup$.
- state is q
- head is pointing to first character after u



- For two configurations C_1, C_2 we say that $C_1 \vdash C_2$ ("C_1 yields C_2") if applying δ to C_1 results in C_2.

- For a string $w \in \Sigma^*$, the initial configuration of M on w is $\boxed{q_0 w}$

- For any $u, v \in \Gamma^*$:
  $u\, q_{acc}\, v$ is an accepting configuration
  $u\, q_{rej}\, v$ is a rejecting configuration

- A TM M accepts $w \in \Sigma^*$ if there exists a sequence of configurations $C_1, \dots, C_n$ s.t.:
  - $C_1 = q_0 w$ ⇝ start in initial config for w
  - $C_i \vdash C_{i+1}$ $\forall i \in \{0, \dots, n-1\}$ ⇝ compute δ correctly
  - C_n is an accepting configuratio ⇝ end in accept

Def: For a TM M, the language recognized by M is $L(M) = \{w \in \Sigma^* \mid M \text{ accepts } w\}$

Def: A language L is TM-recognizable if $\exists$ TM M s.t. $L(M) = L$

(aka "recursively enumerable" or r.e.)

Note: For any TM M and string w, there are three possible outcomes:

"M halts on w":
- M accepts w $\longrightarrow$ $w \in L(M)$
- M rejects w $\longrightarrow$ $w \notin L(M)$

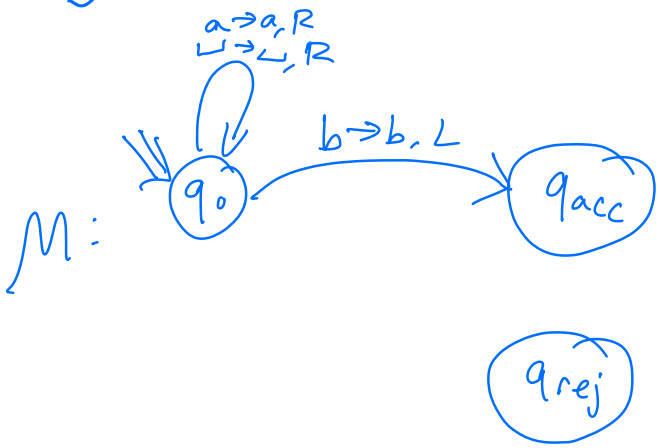
"M doesn't halt on w":
- M runs forever on w $\longrightarrow$ $w \notin L(M)$

Def: If M is a TM s.t. M halts (acc/rej) on all inputs, then M is called a **decider**.

Def: A language L is **TM-decidable** if $\exists$ TM M s.t. $L(M) = L$ and, moreover, M is a decider.

(aka "recursive")

Going back to example from beginning:



$L(M) =$
$\{w \mid w$ contains a 'b'$\}$

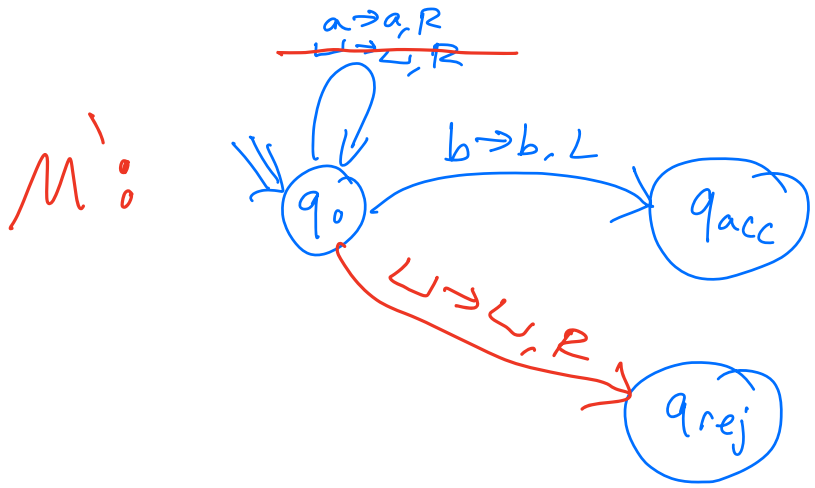
This TM M is <u>not a decider</u>

$L(M) = \{w \mid w \text{ contains a 'b'}\}$

is obviously recognizable.

is it decidable? yes!

because there exists another TM M' that is a decider ($L(M') = L$)



actually this language is even regular, and we will see.

$\{\text{regular languages}\} \subsetneq \{\text{TM-decidable}\} \subsetneq \{\text{TM-recognizable}\} \subsetneq \{\text{all } L\}$

Think for next time:

- How to show $\subseteq$ above? ?
- How to construct a TM-decider for $\{a^n b^n \mid n \geq 0\}$