Last time:
- Midterm

Today:
- Turing Machines
- Turing Machine Variants
TM versus PDA

- TM can both write to and read from the tape
- The head can move left and right
- The input does not have to be read entirely
- Accept and Reject take immediate effect
- Infinite tape on the right, stick on the left

**TM is deterministic (NTM is nondeterministic)**
CONFIGURATIONS

11010q₇001110
MUL = \{1^i1^j1^k \mid ij = k \text{ and } i, j, k \geq 1\}
LUP = \{1^i#x_1#...#x_n \mid n \geq i \text{ and } x_i = x_1\}
A **TM** is a 7-tuple $P = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$

- $Q$ is a finite set of states
- $\Sigma$ is the input alphabet, where $\square \notin \Sigma$
- $\Gamma$ is the stack alphabet, where $\square \in \Gamma$ and $\Sigma \subseteq \Gamma$
- $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$ is the transition function
- $q_0, q_{\text{accept}}, q_{\text{reject}} \in Q$ are the start, accept and reject states
Accepting and rejecting

A TM on input string $w$ may

either halt (enter $q_{\text{accept}}$ or $q_{\text{reject}}$)
or never halt (loop)

A TM is a decider if it halts on every input.
A TM recognizes a language $L$ if it accepts all strings in $L$ and no other strings.

- A language is called recognizable (or enumerable) if some TM recognizes it.

A TM decides a language $L$ if it accepts all strings in $L$ and rejects all strings not in $L$.

- A language is called decidable (or recursive) if some TM decides it.
Recognizable vs. decidable languages

• A language $L$ is **recognizable** (enumerable) if some TM
  1. accepts strings in $L$ and
  2. rejects strings not in $L$ by entering $q_{\text{reject}}$ or looping.

• A language $L$ is **decidable** (recursive) if some TM
  1. accepts strings in $L$ and
  2. rejects strings not in $L$ by entering $q_{\text{reject}}$.
Finite State Control

δ : Q × Γ^k → Q × Γ^k × \{L,R,S\}^k
Theorem: Every Multitape Turing Machine can be transformed into a single-tape Turing Machine.
SIMULATING MULTIPLE TAPES

1. “Format” tape.

2. For each move of the k-tape TM:
   - Scan left-to-right, finding current symbols
   - Scan left-to-right, writing new symbols
   - Scan left-to-right, moving each tape head.

3. If a tape head goes off right end, insert blank.
   - If tape head goes off left end, move back right.
Exercise

Which of these statements are valid descriptions of nondeterministic steps (in a PDA)?

A. Nondeterministically read the input and push it onto the stack.
B. Nondeterministically either read $a$ and push it onto the stack or read $b$ and pop $b$ from the stack.
C. Nondeterministically read the input character $a$ and either push it onto the stack or pop $b$ from the stack.
D. Nondeterministically push one of positive integers onto the stack.
E. None of the above.
F. More than one choice above works.
Theorem. A deterministic TM can simulate a nondeterministic TM.

Proof idea: Consider an NTM $N$. Use a 3-tape TM.

- Let $b$ be the largest number of nondeterministic choices $N$ has in a step. Use alphabet $\{1, \ldots, b\}$ for addresses.
- Do a BFS of the computation tree.
TMs are equivalent to multitape TMs
(proof on the board)
TMs are equivalent to nondeterministic TMs
(proof on the board)