Last time:
• Midterm

Today:
• Turing Machines
• Turing Machine Variants
TURING MACHINE (TM)

$q_1$

INPUT

UNBOUNDED (on the right) TAPE
A TM can loop forever
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

11010\text{q}_7001110
{ \( 0^{2^n} \mid n \geq 0 \) }
\{ 0^{2n} \mid n \geq 0 \}
MUL = \{1^i \#1^j \#1^k \mid ij = k \text{ and } i, j, k \geq 1\}
LP = \{1^i#x_1#...#x_n \mid n \geq i \text{ and } x_i = x_1\}

111#101#11#101

x11#101#11#101

xx1#101#11#101

xxx#101#11#101
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 $\square \not\in \Sigma$
- $\Gamma$ is the tape 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
Implementation-level Description of a TM

• Describe (in English) the instructions for a TM
  – How to move the head
  – What to write on the tape

• Example

1. Scan the tape from left to right and, for every 1 read until non-1 symbol is encountered
   • replace 1 with x,
   • find the next # on the right and replace it with $\overline{\#}$.
   • If no matching # found, reject.
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.
A language $L$ is **recognizable** (enumerable) if some TM
1. accepts strings in $L$ and
2. does not accept 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 $q_{\text{reject}}$. 

Sofya Raskhodnikova; based on slides by Nick Hopper
Fixed number of tapes, $k$
(can’t change during computation)

$$\delta: Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R, S\}^k$$
Theorem. Every multitape TM can be transformed into an equivalent single-tape TM.
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.
Theorem. Every multitape TM can be transformed into an equivalent single-tape TM.
To show one type of machine can simulate another...

1. Explain how to initialize the new machine.
2. Explain how the new machine simulates each step of the old machine.
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.
NTMs are equivalent to TMs

**Theorem.** Every nondeterministic TM can be transformed into an equivalent deterministic TM.

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

- Let $b$ be the largest # 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
TMs are equivalent to nondeterministic TMs
TMs are equivalent to doubly unbounded TMs
Doubly unbounded TMs

A TM with doubly unbounded tape is like an ordinary TM but

- Its tape is infinite on the left and on the right.

Initially, only the input is written on the tape and the head is on the first nonblack symbol.