# *Intro to Theory of Computation*

CS
464

## LECTURE 12

**Last time:**

- Jeopardy!

**Today:**

- Turing Machines
- Turing Machine Variants

**Sofya Raskhodnikova**

# 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 (will consider NTMs later)**

## MUL = $\{1^i\#1^j\#1^k \mid ij = k \text{ and } i, j, k \geq 1\}$

11#111#111111

x1#111#111111

x1#yyy#zzz111

x1#111#zzz111

xx#yyy#zzzzzz

LUP = {$1^i\#x_1\#\ldots\#x_n \mid n \geq i$ and $x_i = x_1$}

111#101#11#101

x11$\overline{\#}$101#11#101

xx1$\overline{\#}$101$\overline{\#}$11#101

xxx$\overline{\#}\dot{1}\dot{0}\dot{1}\overline{\#}11\ddot{\#}\ddot{1}\ddot{0}\ddot{1}$

# CS 464 | Formal Definition of a TM

A ***TM*** is a 7-tuple $\mathbf{P = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})}$

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

# Accepting and rejecting

**A TM on input sting w may**

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

**A TM is a decider if it halts on every input.**

# Language of a TM

A TM **recognizes** a language L if it accepts all strings in L and no other strings.

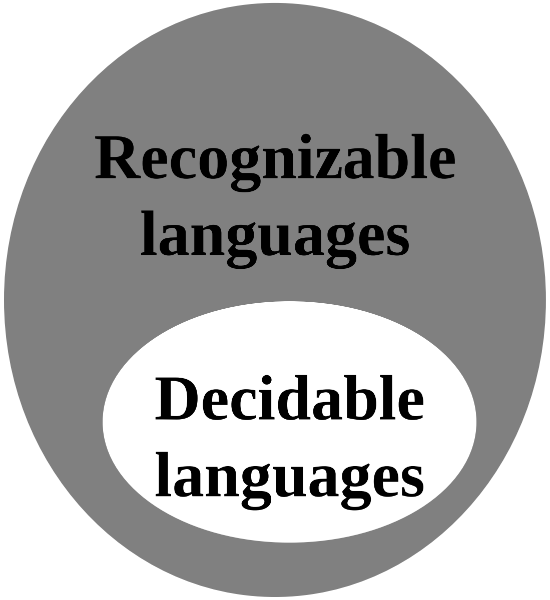
- A language is called **recognizable** (or enumerable) if some TM recognizes it.

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.

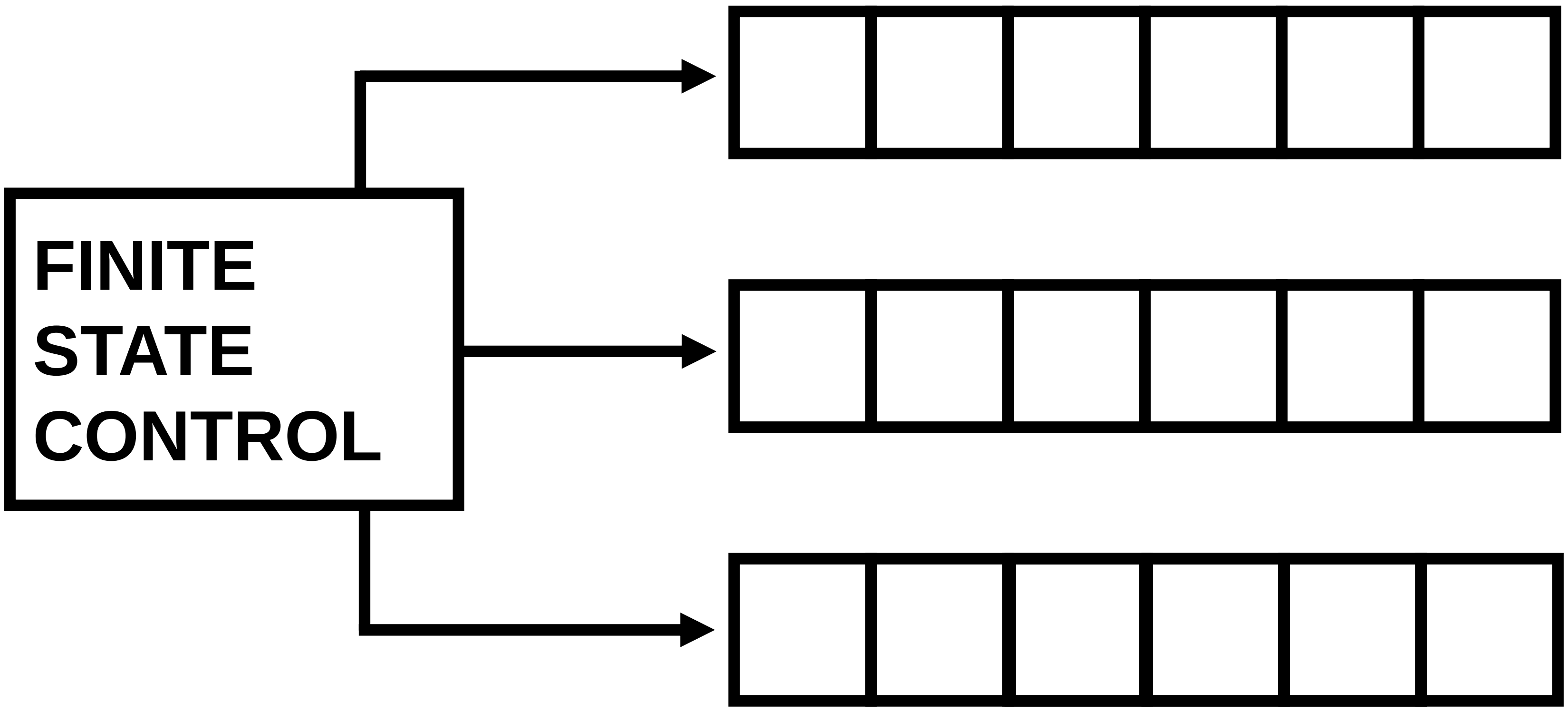
# Recognizable vs. decidable languages

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



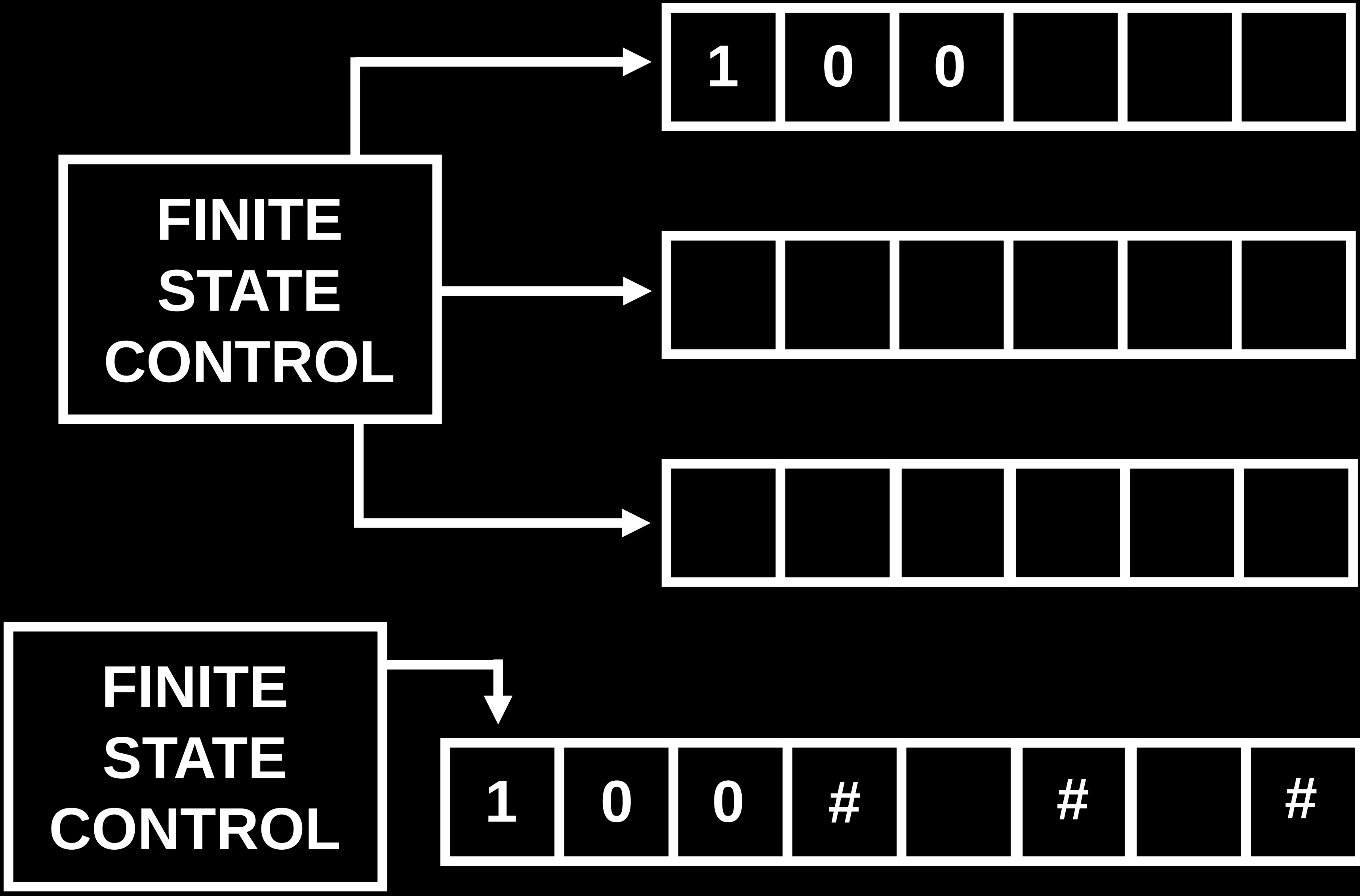

- A language L is **decidable** (*recursive*) if some TM
  1. accepts strings in L and
  2. rejects strings not in L by entering $q_{reject}$ .

# TM variant: multitape TM




$$\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L,R,S\}^k$$

**Theorem:** Every Multitape Turing Machine can be transformed into a single-tape Turing Machine



# SIMULATING MULTIPLE TAPES

●● ● ●

L#100#0#1#R

$q_{jRSS}$ $q_{1}$ $q_{i1\square}$ $q_{i1\square\square}$ $q_{j101RSS}$

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.

# TMs are equivalent to…

**TMs are equivalent to multitape TMs**

(proof on the board)

**TMs are equivalent to nondeterministic TMs**

(proof on the board)

**TMs are equivalent to double unbounded TMs**

(proof on the board)

**TMs are equivalent to FIFO automata.**

(HW problem)

**TMs are equivalent to primitive recursive functions.**

**TMs are equivalent to cellular automata.**