# *Intro to Theory of Computation*

## LECTURE 17

CS 332

**Last time**
- Reductions
- Mapping reductions

**Today**
- Computation history method

**Sofya Raskhodnikova**

# Exercise

**A language L is Turing-recognizable ⇔ $L \leq_m A_{TM}$**

**A.** **Only the ⇒ direction is true.**

**B.** **Only the ⇐ direction is true.**

**C.** **Both directions are true.**

**D.** **Neither direction is true.**

# Problems in language theory

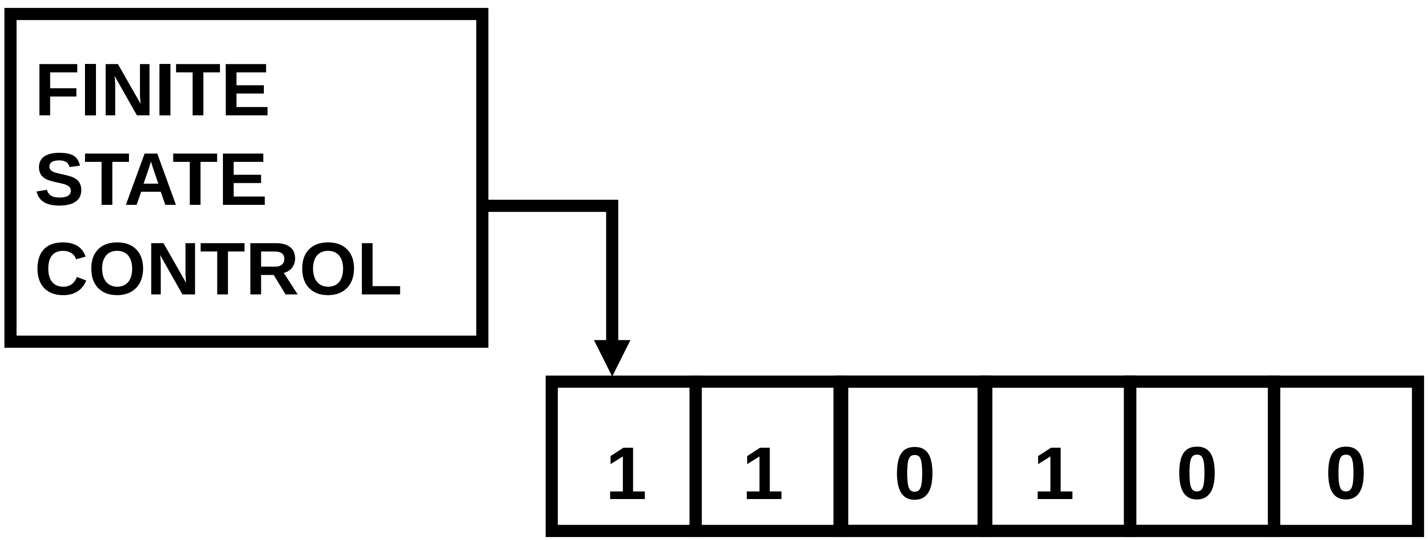
| | | |
|---|---|---|
| $A_{DFA}$ decidable | $A_{CFG}$ decidable | $A_{TM}$ **un**decidable |
| $E_{DFA}$ decidable | $E_{CFG}$ decidable | $E_{TM}$ **un**decidable |
| $EQ_{DFA}$ decidable | $EQ_{CFG}$ ? | $EQ_{TM}$ **un**decidable |

# Proving undecidability for languages that do not involve TM descriptions

## Computation history method

# A linear bounded automaton (LBA)

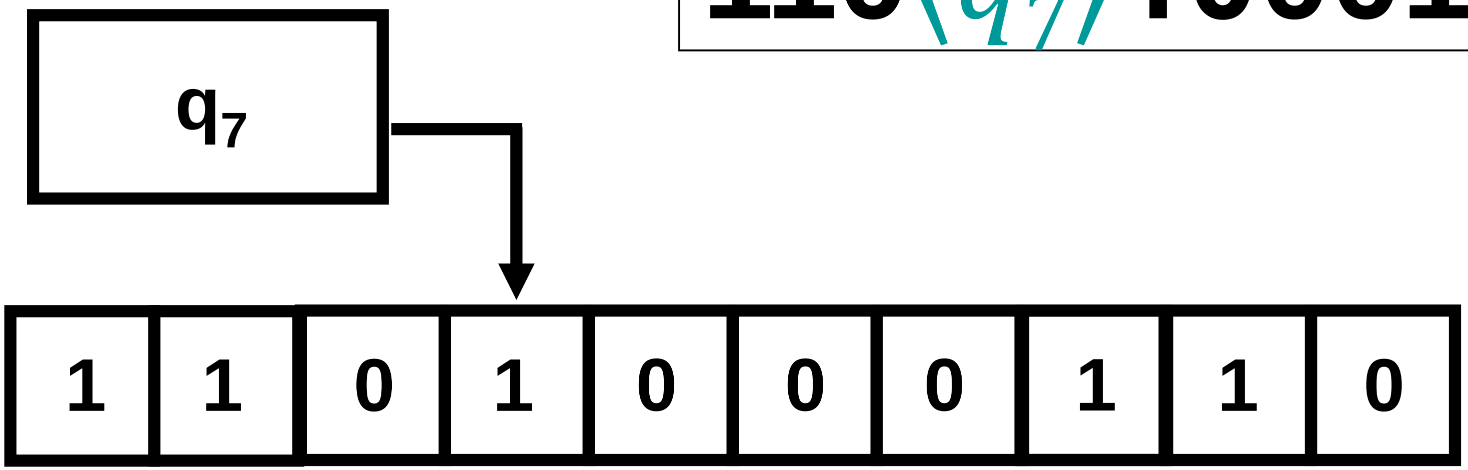
A linear bounded automaton (LBA) is a TM variant that has bounded tape, with the number of tape squares equal to the size of the input.



# Configurations

**A configuration of an LBA is a setting of state, head position and tape contents.**

$$110\langle q_7\rangle 1000110$$



CS 332

# Exercise

**How many distinct configurations does an LBA have if it has $q$ states, $g$ symbols in tape alphabet, and the tape of length $n$?**

A. $qgn$

B. $q+g+n$

C. $qg^n$

D. $qng^n$

E. None of the above.

# Prove that $A_{LBA}$ is decidable

$A_{LBA} = \{\langle B, w\rangle \mid B$ is an LBA that accepts string $w\}$

**Idea:** Given $\langle B, w\rangle$, simulate B on w.

If it halts, we know the answer.

If it loops, we can detect because B repeats a configuration.

**S =** `` On input $\langle B, w\rangle$, where $B$ is an LBA and *w* is a string:

1. Simulate B on w for $qng^n$ steps.
2. If it accepts, accept.
3. If it rejects or does not halt, reject.”

# Computation history

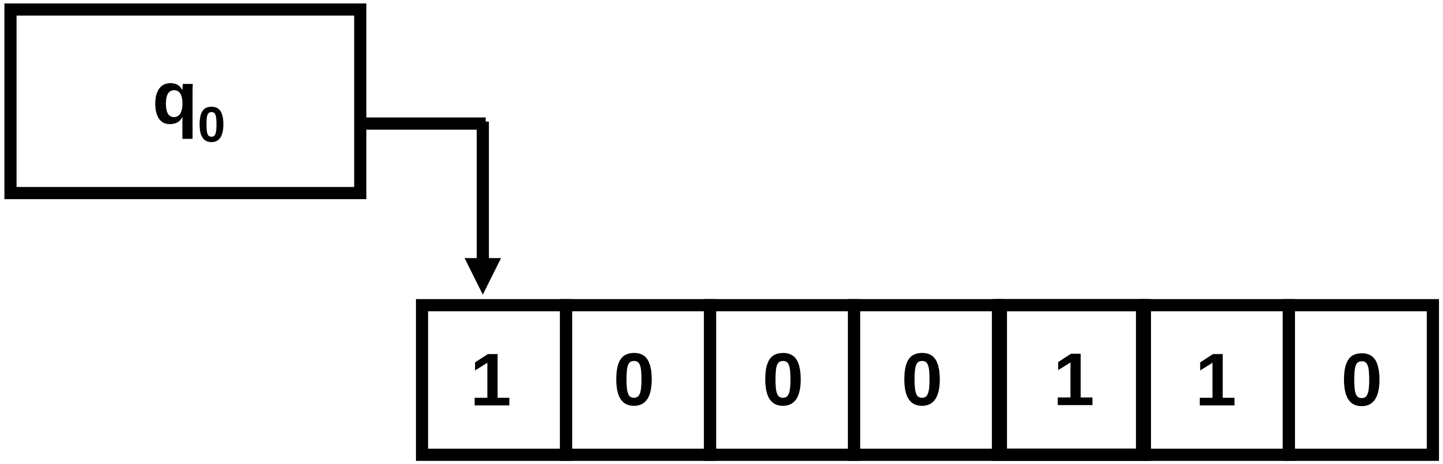
**An accepting computation history for a TM M on input w is a sequence of configurations entered by M on input w:**

$$C_0 \# C_1 \# \ldots \# C_\ell$$

starting configuration

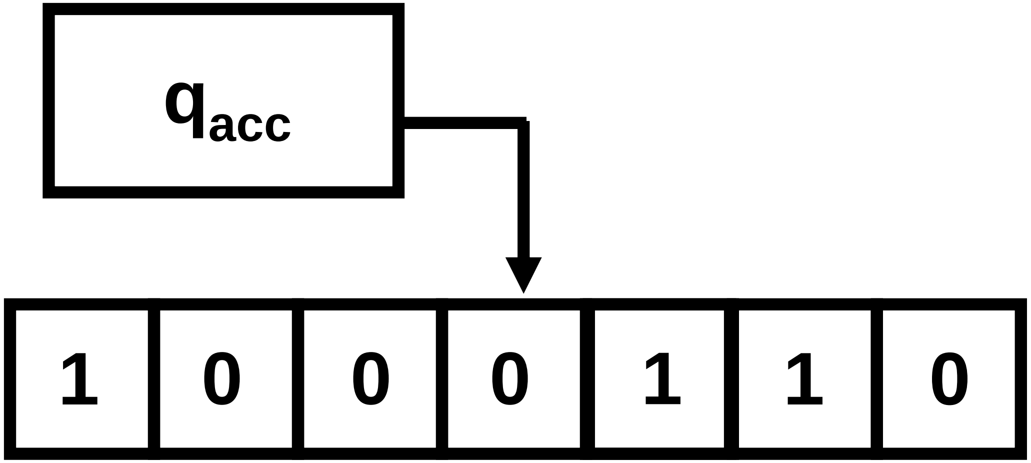
accepting configuration

# Starting configuration



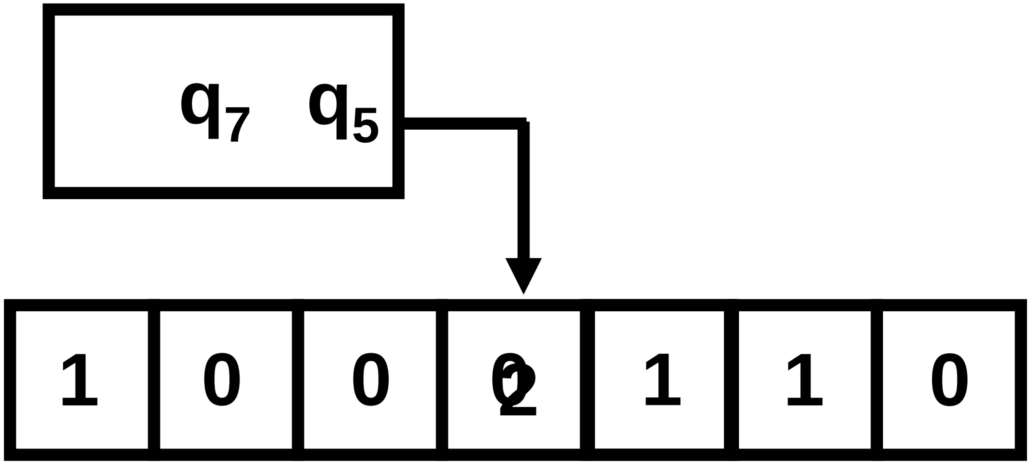

$$C_0 = \langle q_0 \rangle \mathbf{w}$$

# Accepting configuration




$$C_\ell = \ldots \langle q_{acc} \rangle \ldots$$

# Each $C_{i+1}$ legally follows from $C_i$




$$C_i = 1\ 0\ 0\ \langle q_7 \rangle\ 0\ \ 1\ 1\ 0$$

$$C_{i+1} = 1\ 0\ 0\ \ 2\ \langle q_5 \rangle\ 1\ 1\ 0$$

# LBAs can check computation histories of TMs

**Given a TM M and a string w, we can construct an LBA that checks whether its input is the accepting computation history of M on w.**

CS 332

# Prove that $E_{LBA}$ is undecidable

$E_{LBA} = \{ \langle B \rangle \mid B$ is a LBA and $L(B) = \emptyset \}$

**Proof:** Suppose to the contrary that TM R decides $E_{LBA}$.
We construct TM S that decides $A_{TM}$.

**S =** `` On input $\langle M, w \rangle$, where $M$ is a TM and $w$ is a string:

1. Construct an LBA $B$ from $M$ and $w$:

   ***B*** **=** `` On input *x*,
   ***Accept*** ***if x =*** $C_0\# \ldots \# C_\ell$ ***is the accepting computation history of M on w:***
   1. $C_0$ is the starting configuration of $M$ on $w$
   2. Each $C_{i+1}$ legally follows from $C_i$
   3. $C_\ell$ is an accepting configuration for $M$ ”

2. Run TM *R* on input <$B$>.
3. If it rejects, accept. O.w. reject.”

# $ALL_{PDA}$ is undecidable

$ALL_{PDA}$ = { ⟨*P* ⟩ | *P* is a PDA that accepts all strings }

- We can use the computation history method to show that $ALL_{PDA}$ is undecidable.
- It follows that $ALL_{CFG}$ is undecidable.
- It follows that $EQ_{CFG}$ is undecidable.

# Problems in language theory

| | | |
|---|---|---|
| $A_{DFA}$ decidable | $A_{CFG}$ decidable | $A_{TM}$ undecidable |
| $E_{DFA}$ decidable | $E_{CFG}$ decidable | $E_{TM}$ undecidable |
| $EQ_{DFA}$ decidable | $EQ_{CFG}$ undecidable | $EQ_{TM}$ undecidable |

# Post Correspondence Problem

**Domino:** $\left[\frac{a}{ab}\right]$ **. Top and bottom are strings.**

**Input: collection of dominos.**

$$\left[\frac{aa}{aba}\right], \left[\frac{ab}{aba}\right], \left[\frac{ba}{aa}\right], \left[\frac{abab}{b}\right]$$

**Match: list of some of the input dominos (repetitions allowed) with top = bottom**

$$\left[\frac{ab}{aba}\right], \left[\frac{aa}{aba}\right], \left[\frac{ba}{aa}\right], \left[\frac{aa}{aba}\right], \left[\frac{abab}{b}\right]$$

**Problem: determine if a match exists.**

**POST={input with a match} is undecidable.**

# Exercise

A two-dimensional automaton (2DIM-DFA) takes an $m \times n$ rectangle as input, for any $m, n \geq 2$. The boundary squares contain #; internal squares contain symbols from alphabet $\Sigma$. The transition function $\delta: Q \times (\Sigma \cup \{\#\}) \rightarrow Q \times \{L, R, U, D\}$ indicates the next state and head movement (left, right, up, down). How many distinct configurations does a 2DIM-DFA have on a given input?

A. $(m-2)(n-2)|Q|$

B. $mn|Q|$

C. $mn|Q| \cdot |\Sigma|$

D. $mn|Q| \cdot |\Sigma|^{(m-2)(n-2)}$

E. None of the above.

| | | | | | | |
|---|---|---|---|---|---|---|
| # | # | # | # | # | # | # |
| # | 0 | 0 | 1 | 0 | 0 | # |
| # | 0 | 1 | 1 | 1 | 0 | # |
| # | 0 | 1 | 1 | 1 | 1 | # |
| # | 0 | 0 | 1 | 0 | 1 | # |
| # | # | # | # | # | # | # |

# Exercise

Let $A_{\mathrm{2DIM-DFA}} = \{\langle D, x \rangle \mid$ D is a 2DIM-DFA and $D$ accepts $x\}$

1. Can a 2DIM-DFA loop?
2. Is $A_{\mathrm{2DIM-DFA}}$ decidable?

**A.** YES to both.

**B.** NO to both.

**C.** YES to 1, NO to 2.

**D.** NO to 1, YES to 2.