

BU CS 332 – Theory of Computation

<https://forms.gle/xJ1wybwReq4x4dEW6>



Lecture 12:

- More decidable languages
- Universal Turing Machine
- Countability

Reading:

Sipser Ch 4.1, 4.2

Alexander Poremba & Mark Bun

March 17, 2026

Last Time

Church-Turing Thesis

v1: The basic TM (and all equivalent models) capture our intuitive notion of algorithms

v2: Any physically realizable model of computation can be simulated by the basic TM

Decidable languages (from language theory)

$A_{\text{DFA}} = \{\langle D, w \rangle \mid \text{DFA } D \text{ accepts input } w\}$, etc.

Today: More decidable languages

Are there undecidable languages? How can we prove so?

A “Universal” Algorithm for Recognizing Regular Languages

$$A_{\text{DFA}} = \{\langle D, w \rangle \mid \text{DFA } D \text{ accepts } w\}$$

Theorem: A_{DFA} is decidable

Proof: Define a (high-level) 3-tape TM M on input $\langle D, w \rangle$:

1. Check if $\langle D, w \rangle$ is a valid encoding (reject if not)
2. Simulate D on w , i.e.,
 - Tape 2: Maintain w and head location of D
 - Tape 3: Maintain state of D , update according to δ
3. **Accept** if D ends in an accept state, **reject** otherwise

All Regular Languages are Decidable

Theorem: Every regular language L is decidable

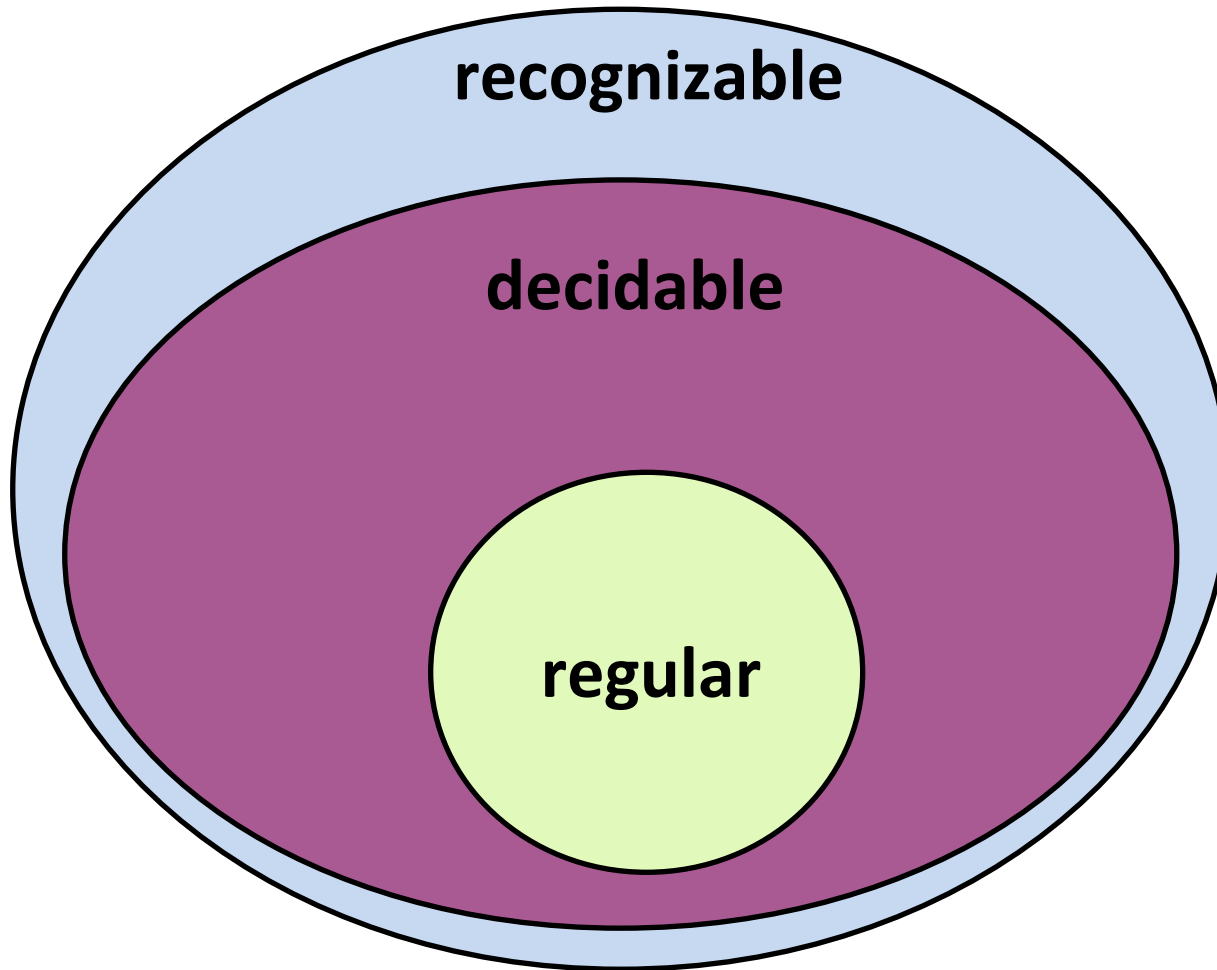
Proof 1: If L is regular, it is recognized by a DFA D . Convert this DFA to a TM M . Then M decides L .

Proof 2: If L is regular, it is recognized by a DFA D . The following TM M_D decides L .

On input w :

1. Run the decider for A_{DFA} on input $\langle D, w \rangle$
2. **Accept** if the decider accepts; **reject** otherwise

Classes of Languages



More Decidable Languages: Emptiness Testing

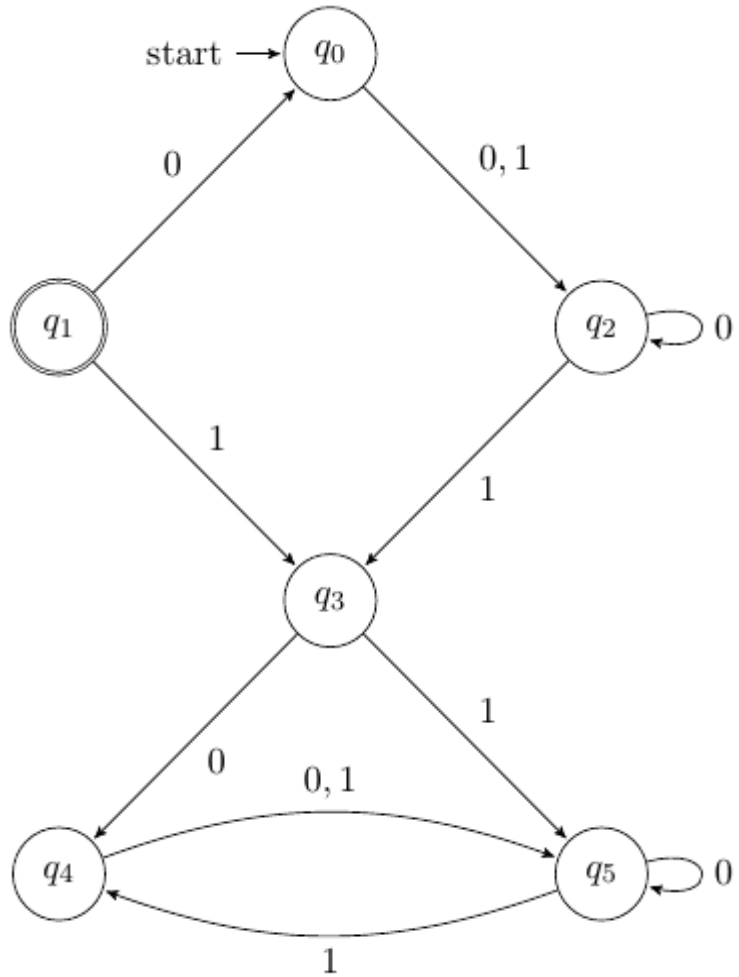
Theorem: $E_{\text{DFA}} = \{\langle D \rangle \mid D \text{ is a DFA such that } L(D) = \emptyset\}$ is decidable

Proof: The following TM decides E_{DFA}

On input $\langle D \rangle$, where D is a DFA with k states:

1. Run k steps of breadth-first search on state diagram of D to determine if an accept state is reachable from the start state
2. **Reject** if a DFA accept state is reachable; **accept** otherwise

E_{DFA} Example



New Deciders from Old: Equality Testing

$EQ_{\text{DFA}} = \{\langle D_1, D_2 \rangle \mid D_1, D_2 \text{ are DFAs and } L(D_1) = L(D_2)\}$

Theorem: EQ_{DFA} is decidable

Proof: The following TM decides EQ_{DFA}

On input $\langle D_1, D_2 \rangle$, where D_1, D_2 are DFAs:

1. Construct DFA D recognizing the **symmetric difference**
 $L(D_1) \Delta L(D_2)$
2. Run the decider for E_{DFA} on $\langle D \rangle$ and return its output

Symmetric Difference

$$A \Delta B = \{w \mid w \in A \text{ or } w \in B \text{ but not both}\}$$

Universal Turing Machine

Meta-Computational Languages

$$A_{\text{DFA}} = \{\langle D, w \rangle \mid \text{DFA } D \text{ accepts } w\}$$

$$A_{\text{TM}} = \{\langle M, w \rangle \mid \text{TM } M \text{ accepts } w\}$$

$$E_{\text{DFA}} = \{\langle D \rangle \mid \text{DFA } D \text{ recognizes the empty language } \emptyset\}$$

$$E_{\text{TM}} = \{\langle M \rangle \mid \text{TM } M \text{ recognizes the empty language } \emptyset\}$$

$$EQ_{\text{DFA}} = \{\langle D_1, D_2 \rangle \mid D_1 \text{ and } D_2 \text{ are DFAs, } L(D_1) = L(D_2)\}$$

$$EQ_{\text{TM}} = \{\langle M_1, M_2 \rangle \mid M_1 \text{ and } M_2 \text{ are TMs, } L(M_1) = L(M_2)\}$$

The Universal Turing Machine



$A_{\text{TM}} = \{\langle M, w \rangle \mid M \text{ is a TM that accepts input } w\}$

Theorem: A_{TM} is Turing-recognizable

The following “Universal TM” U recognizes A_{TM}

On input $\langle M, w \rangle$:

1. Simulate running M on input w
2. If M accepts, **accept**. If M rejects, **reject**.

Universal TM and A_{TM}



Why is the Universal TM **not** a decider for A_{TM} ?

The following “Universal TM” U recognizes A_{TM}

On input $\langle M, w \rangle$:

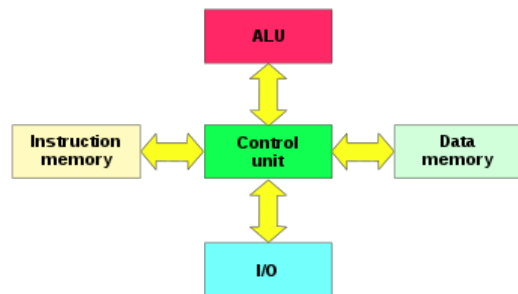
1. Simulate running M on input w
 2. If M accepts, **accept**. If M rejects, **reject**.
-
- a) It may reject inputs $\langle M, w \rangle$ where M accepts w
 - b) It may accept inputs $\langle M, w \rangle$ where M rejects w
 - c) It may loop on inputs $\langle M, w \rangle$ where M loops on w
 - d) It may loop on inputs $\langle M, w \rangle$ where M accepts w

More on the Universal TM

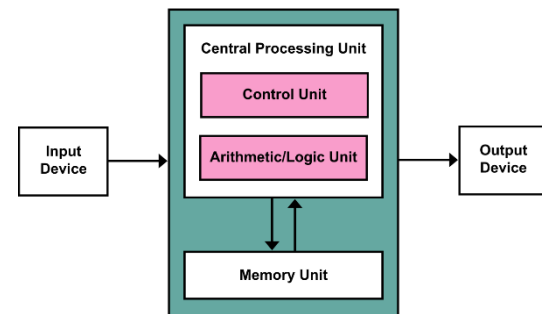
"It is possible to invent a single machine which can be used to compute any computable sequence. If this machine **U** is supplied with a tape on the beginning of which is written the S.D ["standard description"] of some computing machine **M**, then **U** will compute the same sequence as **M**."

- Turing, "On Computable Numbers..." 1936

- Foreshadowed general-purpose programmable computers
- No need for specialized hardware: Virtual machines as software



Harvard architecture:
Separate instruction and data pathways



von Neumann architecture:
Programs can be treated as data

Undecidability

A_{TM} is Turing-recognizable via the Universal TM

...but it turns out A_{TM} (and $E_{\text{TM}}, EQ_{\text{TM}}$) is **undecidable**

i.e., computers cannot solve these problems no matter how much time they are given

How can we prove this?

... but first, a math interlude

Countability and Diagonalization

What's Your Intuition?



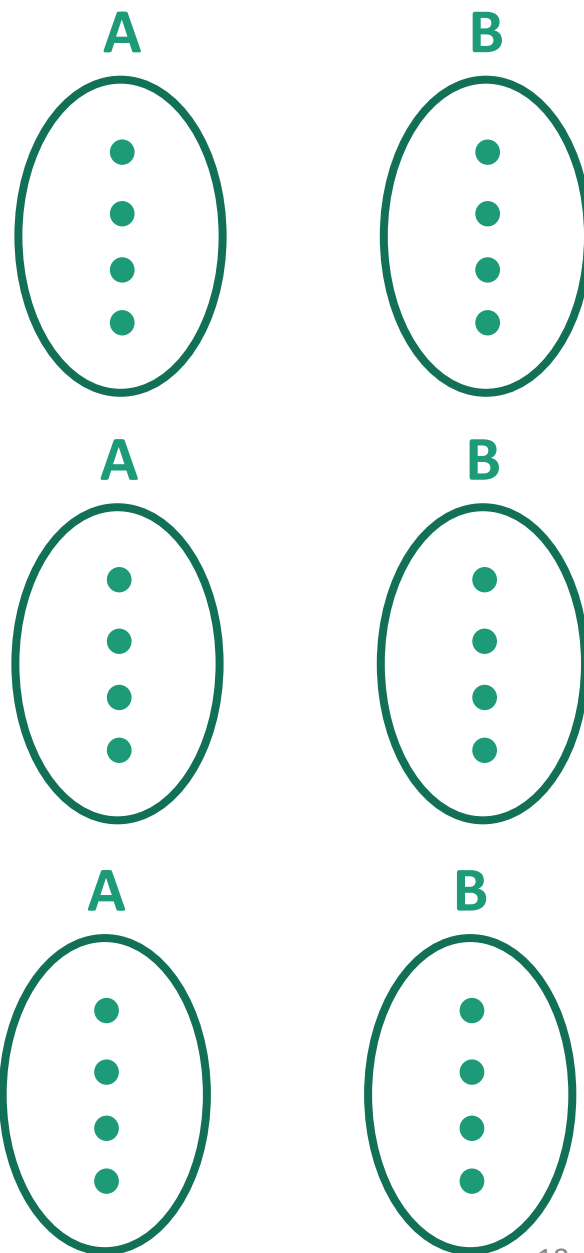
Which of the following sets is the “biggest”?

- a) The natural numbers: $\mathbb{N} = \{1, 2, 3, \dots\}$
- b) The even numbers: $E = \{2, 4, 6, \dots\}$
- c) The positive powers of 2: $POW2 = \{2, 4, 8, 16, \dots\}$
- d) They all have the same size

Set Theory Review

A function $f: A \rightarrow B$ is

- **1-to-1 (injective)** if $f(a) \neq f(a')$ for all $a \neq a'$
- **onto (surjective)** if for all $b \in B$, there exists $a \in A$ such that $f(a) = b$
- **a correspondence (bijective)** if it is 1-to-1 and onto, i.e., every $b \in B$ has a unique $a \in A$ with $f(a) = b$



How Can We Compare Sizes of Infinite Sets?

Definition: Two sets have **the same size** if there is a bijection between them

A set is **countable** if either

- it is a finite set, or
- it has the same size as \mathbb{N} , the set of natural numbers

Examples of Countable Sets

- \emptyset
- $\{0,1\}$
- $\{0, 1, 2, \dots, 8675309\}$

- $E = \{2, 4, 6, 8, \dots\}$
- $SQUARES = \{1, 4, 9, 16, 25, \dots\}$
- $POW2 = \{2, 4, 8, 16, 32, \dots\}$

$$|E| = |SQUARES| = |POW2| = |\mathbb{N}|$$

Why is $\mathbb{N} \times \mathbb{N}$ Countable?

(1, 1)	(2, 1)	(3, 1)	(4, 1)	(5, 1)	...
(1, 2)	(2, 2)	(3, 2)	(4, 2)	(5, 2)	...
(1, 3)	(2, 3)	(3, 3)	(4, 3)	(5, 3)	...
(1, 4)	(2, 4)	(3, 4)	(4, 4)	(5, 4)	...
(1, 5)	(2, 5)	(3, 5)	(4, 5)	(5, 5)	...

How to Argue That a Set S is Countable

- Describe how to “list” the elements of S , usually in stages:

Ex: Stage 1) List all pairs (x, y) such that $x + y = 2$

Stage 2) List all pairs (x, y) such that $x + y = 3$

...

Stage n) List all pairs (x, y) such that $x + y = n + 1$

...

- Explain why every element of S appears in the list

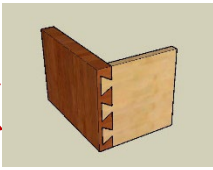
Ex: Any $(x, y) \in \mathbb{N} \times \mathbb{N}$ will be listed in stage $x + y - 1$

- Define the bijection $f: \mathbb{N} \rightarrow S$ by $f(n) =$ the n 'th element in this list (ignoring duplicates if needed)

More Examples of Countable Sets

- $\{0,1\}^*$
- $\{\langle M \rangle \mid M \text{ is a Turing machine}\}$
- $\mathbb{Q} = \{\text{rational numbers}\}$
- If $A \subseteq B$ and B is countable, then A is countable
- If A and B are countable, then $A \times B$ is countable
- Nonempty S is countable if and only if there exists a surjection (an onto function) $f : \mathbb{N} \rightarrow S$

Another Version of the Dovetailing Trick



Ex: Show that $\mathcal{F} = \{L \subseteq \{0, 1\}^* \mid L \text{ is finite}\}$ is countable