BU CS 332 – Theory of Computation

https://forms.gle/z3CYEiw9CpKv6ghK6

Lecture 13:

- More decidable languages
- Universal Turing Machine
- Countability

Reading:

Sipser Ch 4.1, 4.2

Mark Bun

October 25, 2022

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)

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

Today: More decidable languages

What languages are undecidable? How can we prove so?

A "universal" algorithm for recognizing regular languages

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

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

Other decidable languages

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

$A_{\text{NFA}} = \{\langle N, w \rangle \mid \text{NFA } N \text{ accepts } w\}$

$A_{\text{REX}} = \{\langle R, w \rangle \mid \text{regular expression } R \text{ generates } w\}$

NFA Acceptance

Which of the following describes a **decider** for $A_{\mathrm{NFA}} = \{\langle N, w\rangle \mid \text{NFA } N \text{ accepts } w\}$?

a) Using a deterministic TM, simulate N on w, always making the first nondeterministic choice at each step. Accept if it accepts, and reject otherwise.

b) Using a deterministic TM, simulate all possible choices of N on w for 1 step of computation, 2 steps of computation, etc. Accept whenever some simulation accepts.

c) Use the subset construction to convert N to an equivalent DFA M. Simulate M on w, accept if it accepts, and reject otherwise.

Regular Languages are Decidable

Theorem: Every regular language L is decidable

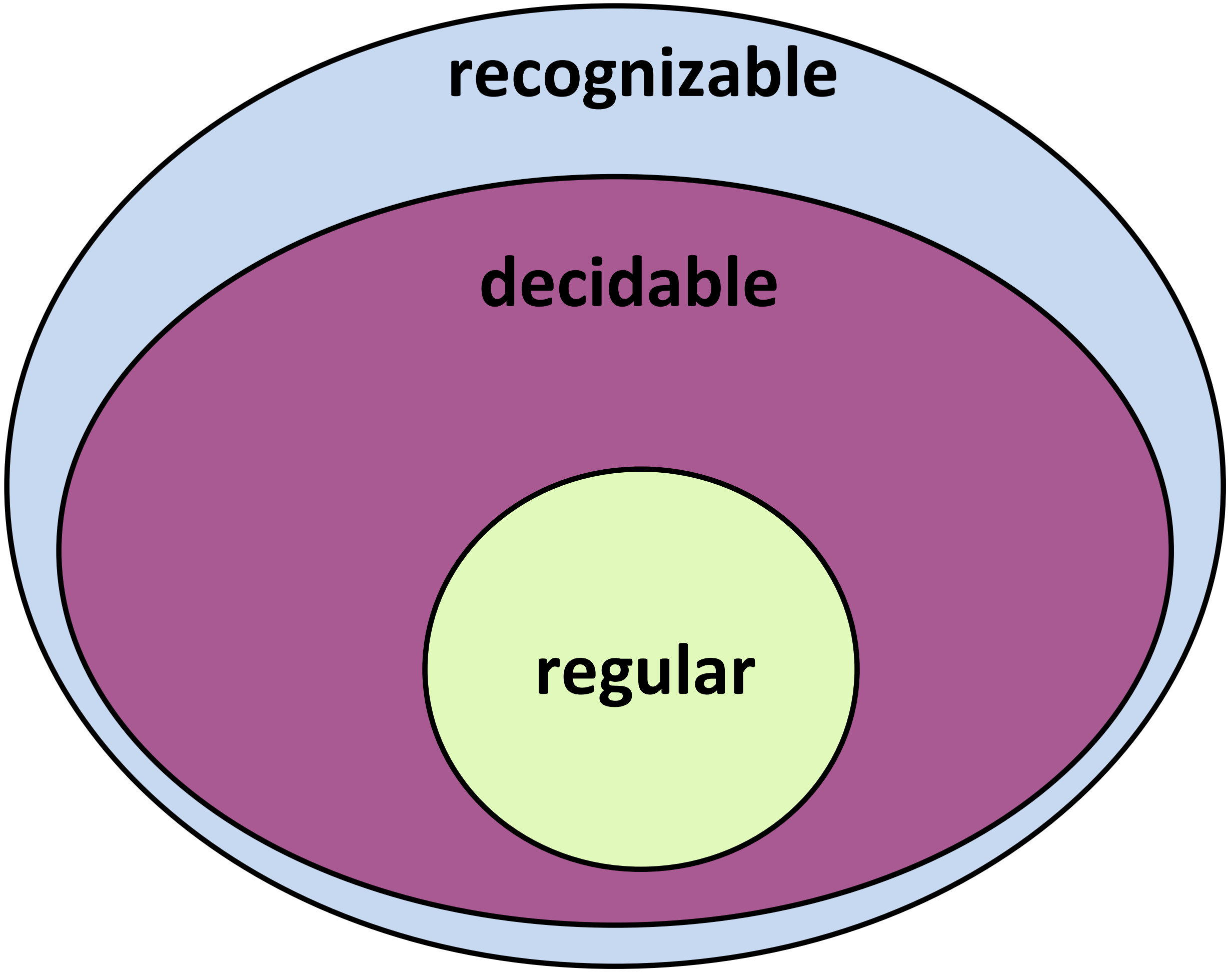
Proof 1: If L is regular, it is recognized by a DFA D. Convert this DFA to a TM M. Then M decides L.

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

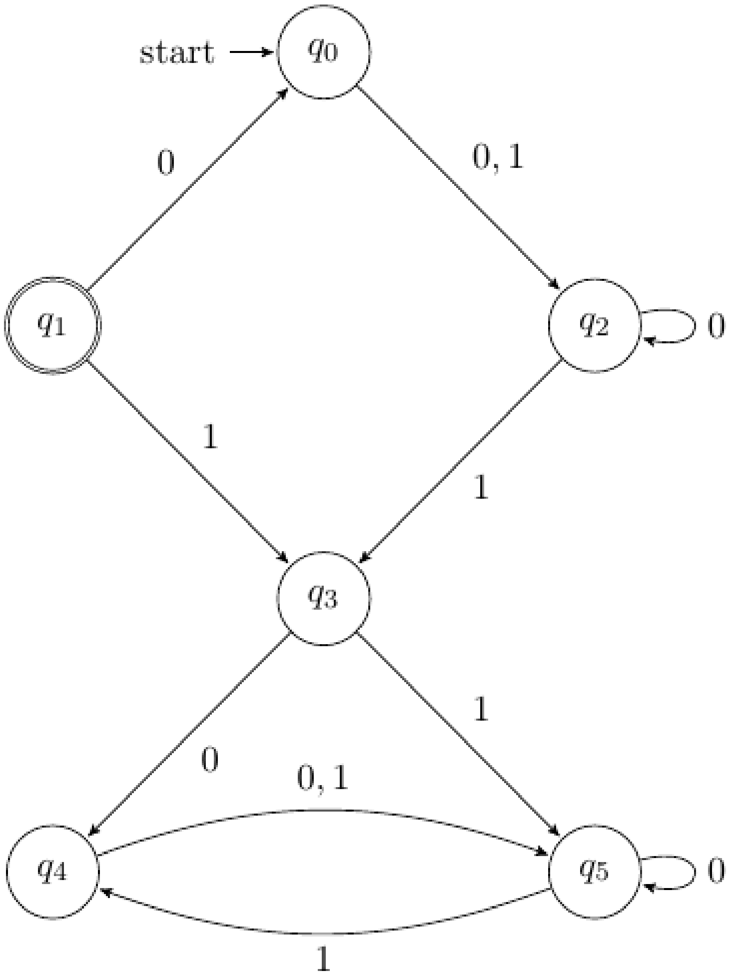
Theorem: $E_{\mathrm{DFA}} = \{\langle D \rangle \mid D \text{ is a DFA such that } L(D) = \emptyset\}$ is decidable

Proof: The following TM decides E_{DFA}

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

1. Perform 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 $\langle D_1, D_2 \rangle$ are DFAs:

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

Symmetric Difference

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

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

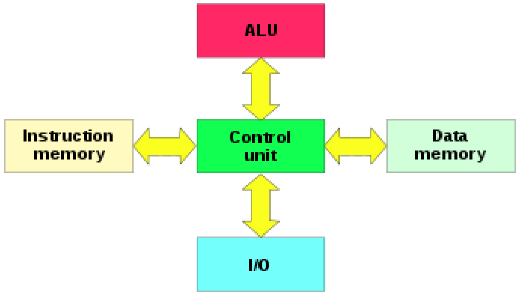
d) It may loop on inputs $\langle M, w \rangle$ where M accepts w

More on the Universal TM

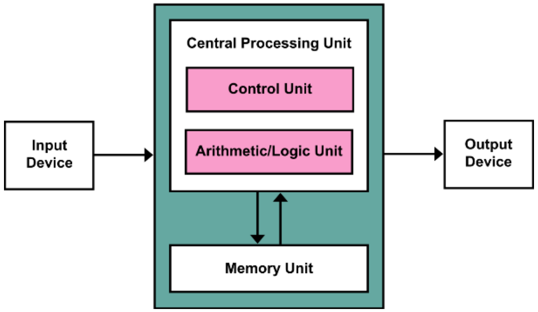
"It is possible to invent a single machine which can be used to compute any computable sequence. If this machine **U** is supplied with a tape on the beginning of which is written the S.D ["standard description"] of some computing machine **M**, then **U** will compute the same sequence as **M**."

- 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 $E_{\mathrm{TM}}, EQ_{\mathrm{TM}}$) is **undecidable**

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

How can we prove this?

First, a mathematical interlude…

Countability and Diagonalizaiton

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: $E = \{2, 4, 6, \dots\}$

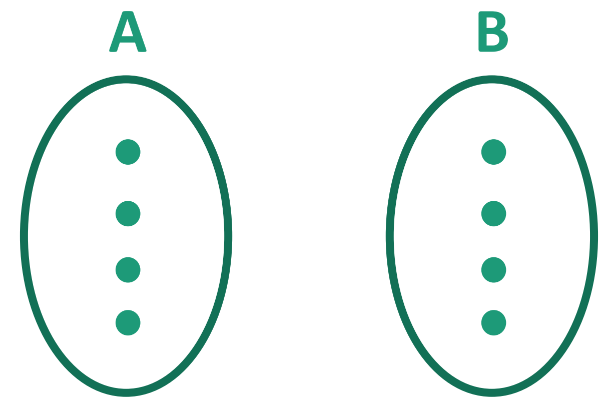
c) The positive powers of 2: $POW2 = \{2, 4, 8, 16, \dots\}$

d) They all have the same size

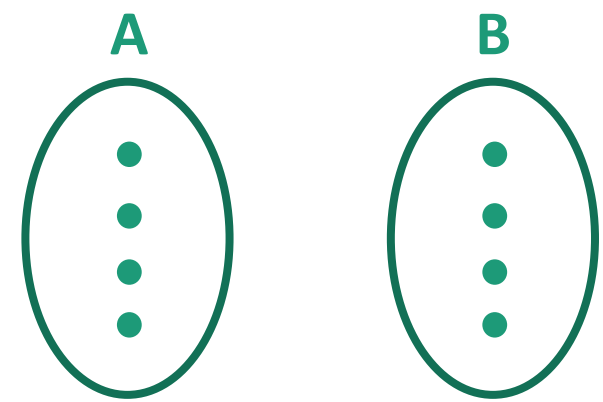
Set Theory Review

A function $f: A \rightarrow B$ is

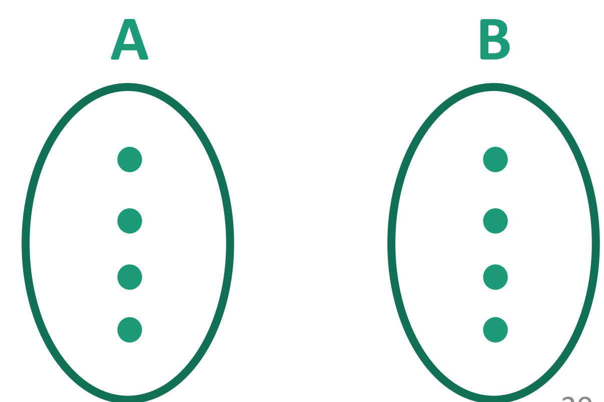
- 1-to-1 (injective) if $f(a) \neq f(a')$ for all $a \neq a'$



- 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

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

Examples of countable sets

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

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

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

How to show that $\mathbb{N} \times \mathbb{N}$ is countable?

$(1,1)$	$(2,1)$	$(3,1)$	$(4,1)$	$(5,1)$	$\ldots$
$(1,2)$	$(2,2)$	$(3,2)$	$(4,2)$	$(5,2)$	$\ldots$
$(1,3)$	$(2,3)$	$(3,3)$	$(4,3)$	$(5,3)$	$\ldots$
$(1,4)$	$(2,4)$	$(3,4)$	$(4,4)$	$(5,4)$	$\ldots$
$(1,5)$	$(2,5)$	$(3,5)$	$(4,5)$	$(5,5)$	
					$\ddots$

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

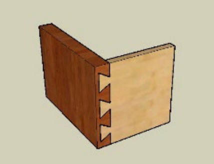
More examples of countable sets

- $\{0,1\}^*$
- $\{\langle M \rangle \mid M \text{ is a Turing machine}\}$
- $\mathbb{Q} = \{\text{rational numbers}\}$

- If $A \subseteq B$ and B is countable, then A is countable
- If A and B are countable, then $A \times B$ is countable

- S is countable if and only if there exists a surjection (an onto function) $f : \mathbb{N} \to S$

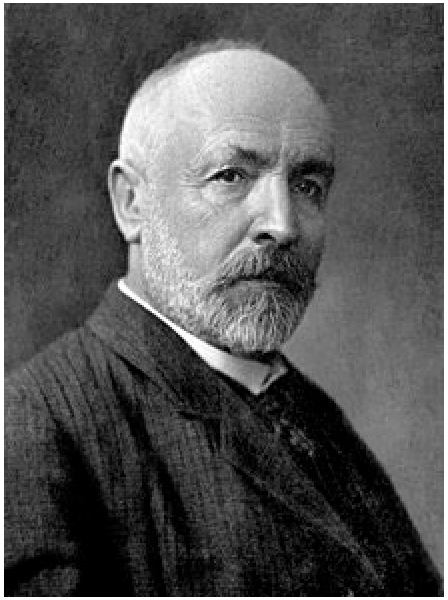
Another version of the dovetailing trick



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

So what *isn't* countable?

Cantor’s Diagonalization Method



Georg Cantor 1845-1918

- Invented set theory
- Defined countability, uncountability, cardinal and ordinal numbers, …

Some praise for his work:

“Scientific charlatan…renegade…corruptor of youth”
–L. Kronecker

“Set theory is wrong…utter nonsense…laughable”
–L. Wittgenstein

Uncountability of the reals

Theorem: The real interval $[0,1]$ is uncountable.

Proof: Assume for the sake of contradiction it were countable, and let $f: \mathbb{N} \to [0,1]$ be onto (surjective)

n	$f(n)$
1	$0\,.\,d_1^1\ d_2^1\ d_3^1\ d_4^1\ d_5^1\ \ldots$
2	$0\,.\,d_1^2\ d_2^2\ d_3^2\ d_4^2\ d_5^2\ \ldots$
3	$0\,.\,d_1^3\ d_2^3\ d_3^3\ d_4^3\ d_5^3\ \ldots$
4	$0\,.\,d_1^4\ d_2^4\ d_3^4\ d_4^4\ d_5^4\ \ldots$
5	$0\,.\,d_1^5\ d_2^5\ d_3^5\ d_4^5\ d_5^5\ \ldots$

Construct $b \in [0,1]$ which does not appear in this table

– contradiction!

$b = 0.b_1 b_2 b_3 \ldots$ where $b_i \neq d_i^i$ (digit i of $f(i)$)

Uncountability of the reals

A concrete example of the contradiction construction:

n	$f(n)$
1	$0.8675309\ldots$
2	$0.1415926\ldots$
3	$0.7182818\ldots$
4	$0.4444444\ldots$
5	$0.1337133\ldots$

Construct $b \in [0,1]$ which does not appear in this table

– contradiction!

$b = 0.b_1 b_2 b_3 \ldots$ where $b_i \neq d_i^i$ (digit i of $f(i)$)

Diagonalization

This process of constructing a counterexample by "contradicting the diagonal" is called diagonalization