UMB CS 420
Undecidability
Wednesday, March 23, 2022

Diagram:
- Turing-recognizable
- Decidable
- Context-free
- Regular

??
Announcements

• HW 7 due Sun 3/27 11:59pm EST
Recap: Decidability of Regular and CFLs

- $A_{\text{DFA}} = \{ \langle B, w \rangle | B \text{ is a DFA that accepts input string } w \}$ Decidable
- $A_{\text{NFA}} = \{ \langle B, w \rangle | B \text{ is an NFA that accepts input string } w \}$ Decidable
- $A_{\text{REX}} = \{ \langle R, w \rangle | R \text{ is a regular expression that generates string } w \}$ Decidable
- $E_{\text{DFA}} = \{ \langle A \rangle | A \text{ is a DFA and } L(A) = \emptyset \}$ Decidable
- $EQ_{\text{DFA}} = \{ \langle A, B \rangle | A \text{ and } B \text{ are DFAs and } L(A) = L(B) \}$ Decidable
- $A_{\text{CFG}} = \{ \langle G, w \rangle | G \text{ is a CFG that generates string } w \}$ Decidable
- $E_{\text{CFG}} = \{ \langle G \rangle | G \text{ is a CFG and } L(G) = \emptyset \}$ Decidable
- $EQ_{\text{CFG}} = \{ \langle G, H \rangle | G \text{ and } H \text{ are CFGs and } L(G) = L(H) \}$ Undecidable?
- $A_{\text{TM}} = \{ \langle M, w \rangle | M \text{ is a TM and } M \text{ accepts } w \}$ Undecidable?
Thm: $A_{TM}$ is Turing-recognizable

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

$U = \text{ "On input } \langle M, w \rangle, \text{ where } M \text{ is a TM and } w \text{ is a string:}
1. \text{ Simulate } M \text{ on input } w.
2. \text{ If } M \text{ ever enters its accept state, } accept; \text{ if } M \text{ ever enters its reject state, } reject.\text{"} \$

$U = \text{ Extended } \delta \text{ "run" function for TMs}
\bullet \text{ Computer that can simulate other computers}
\bullet \text{ i.e., "The Universal Turing Machine"}
\bullet \text{ Problem: } U \text{ loops when } M \text{ loops}

So it's a recognizer, not a decider
Thm: $A_{TM}$ is undecidable

$$A_{TM} = \{ \langle M, w \rangle | M \text{ is a TM and } M \text{ accepts } w \}$$

- ???

It’s hard to prove that something is not true!

Typically need complicated proof techniques

e.g., pumping lemma and proof by contradiction for proving non-regularness

next
Kinds of Functions (a fn maps Domain → Range)

- **Injective**, a.k.a., “one-to-one”
  - Every element in Domain has a unique mapping
  - How to remember:
    - Entire Domain is mapped “in” to the Range

- **Surjective**, a.k.a., “onto”
  - Every element in Range is mapped to
  - How to remember:
    - “Sur” = “over” (eg, survey); Domain is mapped “over” the Range

- **Bijective**, a.k.a., “correspondence” or “one-to-one correspondence”
  - Is both injective and surjective
  - Unique pairing of every element in Domain and Range
Countability

• A set is “countable” if it is:
  • Finite
  • Or, there exists a bijection between the set and the natural numbers
    • In this case, the set has the same size as the set of natural numbers
    • This is called “countably infinite”
Exercise: Which set is larger?

- The set of:
  - Natural numbers, or
  - Even numbers?
- They are the **same size**! Both are **countably infinite**
  - Proof: **Bijection:**

<table>
<thead>
<tr>
<th>( n )</th>
<th>( f(n) = 2n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>\vdots</td>
<td>\vdots</td>
</tr>
</tbody>
</table>
Exercise: Which set is larger?

- The set of:
  - Natural numbers $\mathcal{N}$, or
  - Positive rational numbers? $\mathcal{Q} = \{\frac{m}{n} \mid m, n \in \mathcal{N}\}$

- They are the same size! Both are countably infinite

One possible mapping?

But these don’t get mapped to: (not a bijection)
Exercise: Which set is larger?

• The set of:
  • Natural numbers $\mathbb{N}$, or
  • Positive rational numbers? $\mathbb{Q} = \left\{ \frac{m}{n} \mid m, n \in \mathbb{N} \right\}$

• They are the same size! Both are countably infinite

Another mapping: This is a bijection bc every natural number maps to a unique fraction, and vice versa
Exercise: Which set is larger?

- The set of:
  - Natural numbers, or $\mathbb{N}$
  - Real numbers, $\mathbb{R}$
- There are **more** real numbers. It is **uncountably infinite**.

**Proof, by contradiction:**
- **Assume** a bijection between natural and real numbers exists.
  - This means: every nat num maps to a unique real, and vice versa.
  
  But we show that in any given mapping,
  - Some real number is **not** mapped to ... 
  - E.g., a number that has different digits at each position:
    
    $$ x = 0.4641 \ldots $$

    - This number **cannot** be included in mapping ...
    - ... So we have a **contradiction**!
Georg Cantor

• Invented set theory

• Came up with countable infinity (1873)

• And uncountability:
  • Also: how to show uncountability with “diagonalization” technique
### Diagonalization with Turing Machines

#### Diagonal: Result of Giving a TM its own Encoding as Input

<table>
<thead>
<tr>
<th>(M_1)</th>
<th>(M_2)</th>
<th>(M_3)</th>
<th>(M_4)</th>
<th>(D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>accept</td>
<td>reject</td>
<td>accept</td>
<td>reject</td>
<td>accept</td>
</tr>
<tr>
<td>accept</td>
<td>accept</td>
<td>accept</td>
<td>accept</td>
<td>accept</td>
</tr>
<tr>
<td>reject</td>
<td>reject</td>
<td>reject</td>
<td>reject</td>
<td>reject</td>
</tr>
<tr>
<td>accept</td>
<td>accept</td>
<td>accept</td>
<td>accept</td>
<td>accept</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>reject</td>
<td>reject</td>
<td>accept</td>
<td>accept</td>
<td>...</td>
</tr>
</tbody>
</table>

**Opposites**

**All TMs**

**Try to construct “opposite” TM \(D\)**

**TM \(D\) can’t exist!**

**What should happen here?**

**It must both accept and reject!**
Thm: $A_{TM}$ is undecidable

$A_{TM} = \{\langle M, w \rangle | M$ is a TM and $M$ accepts $w\}$

Proof by contradiction:

1. Assume $A_{TM}$ is decidable. Then there exists a decider $H$:

   $$H(\langle M, w \rangle) = \begin{cases} 
   accept & \text{if } M \text{ accepts } w \\
   reject & \text{if } M \text{ does not accept } w 
   \end{cases}$$

2. If $H$ exists, then we can create the “opposite” machine:

   $D =$ “On input $\langle M \rangle$, where $M$ is a TM:
   1. Run $H$ on input $\langle M, \langle M \rangle \rangle$.
   2. Output the opposite of what $H$ outputs. That is, if $H$ accepts, $\text{reject}$; and if $H$ rejects, $\text{accept}$.”

From the previous slide

Result of giving a TM itself as input

Do the opposite
Thm: $A_{TM}$ is undecidable

$A_{TM} = \{\langle M, w \rangle | M \text{ is a TM and } M \text{ accepts } w \}$

Proof by contradiction:

1. **Assume** $A_{TM}$ is decidable. Then there exists a decider $H$:

   $$H(\langle M, w \rangle) = \begin{cases} \text{accept} & \text{if } M \text{ accepts } w \\ \text{reject} & \text{if } M \text{ does not accept } w \end{cases}$$

2. If $H$ exists, then we can create an “opposite” machine:

   
   
   $D = \text{“On input } \langle M \rangle, \text{ where } M \text{ is a TM:}
   
   1. \text{Run } H \text{ on input } \langle M, \langle M \rangle \rangle.
   
   2. \text{Output the opposite of what } H \text{ outputs. That is, if } H \text{ accepts, reject; and if } H \text{ rejects, accept.”} $

3. But $D$ does not exist! **Contradiction!** So assumption is false.
Easier Undecidability Proofs

• We proved \( A_{TM} = \{ \langle M, w \rangle | M \text{ is a TM and } M \text{ accepts } w \} \) undecidable ...

• ... by contradiction:

• By showing its decider can help create impossible decider “D”!

• Coming up with “D” was hard (needed to invent diagonalization)

• But then we more easily reduced \( A_{TM} \) to “D”

• Easier: reduce problems to \( A_{TM} \)!
The Halting Problem

\[ \text{HALT}_{TM} = \{ \langle M, w \rangle \mid M \text{ is a TM and } M \text{ halts on input } w \} \]

Thm: \( \text{HALT}_{TM} \) is undecidable

Proof, by contradiction:

- **Assume** \( \text{HALT}_{TM} \) has decider \( R \); use it to create decider for \( A_{TM} \):

- ... 

- **But** \( A_{TM} \) is undecidable and has no decider!

I.e., "Algorithm to determine if a TM is an decider"?
The Halting Problem

\[ HALT_{\text{TM}} = \{ \langle M, w \rangle | \text{M is a TM and M halts on input } w \} \]

Thm: \( HALT_{\text{TM}} \) is undecidable

Proof, by contradiction:

- Assume \( HALT_{\text{TM}} \) has decider \( R \); use it to create decider for \( A_{\text{TM}} \):

  \[
  S = \text{"On input } \langle M, w \rangle, \text{ an encoding of a TM } M \text{ and a string } w:\n  \]

  1. Run TM \( R \) on input \( \langle M, w \rangle \).
  2. If \( R \) rejects, reject. \( \text{This means } M \text{ loops on input } w \)
  3. If \( R \) accepts, simulate \( M \) on \( w \) until it halts. \( \text{This step always halts} \)
  4. If \( M \) has accepted, accept; if \( M \) has rejected, reject."

Termination argument:
Step 1: \( R \) is a decider so always halts
Step 3: \( M \) always halts bc \( R \) said so
The Halting Problem

\[ \text{HALT}_\text{TM} = \{ \langle M, w \rangle \mid M \text{ is a TM and } M \text{ halts on input } w \} \]

**Thm:** \( \text{HALT}_\text{TM} \) is undecidable

Proof, by contradiction:

- **Assume** \( \text{HALT}_\text{TM} \) has decider \( R \); use it to create decider for \( A_\text{TM} \):

  \[
  S = \text{“On input } \langle M, w \rangle, \text{ an encoding of a TM } M \text{ and a string } w:\n  
  1. \text{ Run TM } R \text{ on input } \langle M, w \rangle. 
  2. \text{ If } R \text{ rejects, reject. } 
  3. \text{ If } R \text{ accepts, simulate } M \text{ on } w \text{ until it halts.} 
  4. \text{ If } M \text{ has accepted, accept; if } M \text{ has rejected, reject.”}
  \]

- **But** \( A_\text{TM} \) is undecidable!
  - i.e., this decider that we just created cannot exist! So \( \text{HALT}_\text{TM} \) is undecidable
Easier Undecidability Proofs

In general, to prove the undecidability of a language:

• Use proof by contradiction:

1. Assume the language is decidable,

2. Show that its decider can be used to create a decider for ...

• ... a known undecidable language ...

3. ... which doesn’t have a decider! Contradiction!
Summary: The Limits of Algorithms

- $A_{DFA} = \{ \langle B, w \rangle \mid B$ is a DFA that accepts input string $w \}$
- $A_{CFG} = \{ \langle G, w \rangle \mid G$ is a CFG that generates string $w \}$
- $A_{TM} = \{ \langle M, w \rangle \mid M$ is a TM and $M$ accepts $w \}$
- $E_{DFA} = \{ \langle A \rangle \mid A$ is a DFA and $L(A) = \emptyset \}$
- $E_{CFG} = \{ \langle G \rangle \mid G$ is a CFG and $L(G) = \emptyset \}$
- $E_{TM} = \{ \langle M \rangle \mid M$ is a TM and $L(M) = \emptyset \}$

Decidable
Decidable
Undecidable
Decidable
Decidable
Undecidable
Check-in Quiz 3/23

On gradescope