Today's learning goals

• Use counting arguments to prove the existence of unrecognizable (undecidable) languages.
• Determine and prove whether sets are countable.
• Use diagonalization in a proof of uncountability.
• Use diagonalization in a proof of undecidability.

Exam 2?
A. Easier than exam 1.
B. Harder than exam 1.
C. About what you expected.
Regular

Context-Free

Decidable

Turing-Recognizable

\{a^n b^n | n \geq 0\}

\{a^n b^n a^n | n \geq 0\}
Counting arguments

Before we proved the Pumping Lemma …

We proved there was a set that was not regular because
Counting arguments

Recall: sets A and B have the same size, \(|A| = |B|\) means there is a one-to-one and onto function between them.

A set is countable iff it is either

• finite (has the same size as \(\{0, 1, \ldots, n\}\) for some nonnegative integer \(n\)), or

• has the same size as \(\mathbb{N}\) (can list all and only the elements of the list in a sequence)

OTHERWISE: i.e. uncountable, infinite \& \(|\mathbb{N}|\neq\mathbb{N}|\)
Counting arguments

Which of the following is true?

A. Any two infinite sets have the same size.
B. If A is a strict subset of B and then A and B do not have the same size.
C. If A is a subset of B and B is countable, then A is countable.
D. If A is countable then AxA is not countable.
E. I don't know.
Countable sets

Some examples:

\(\mathbb{N}\)

\(\mathbb{Z}\)

\(\mathbb{Q}\)

\(\{0,1\}^*\)

\(\Sigma^*\) for any alphabet \(\Sigma\)

Corollary: The set of all TMs is countable.

Proof Idea: \(|\{M: M \text{ is a TM}\}| = |\{<M>: M \text{ is a TM}\}|\) and \(<M>\) is a string over the alphabet \(\{0,1,\_,(,),\ldots\}\).

Cor: The set of all TM-recognizable langs is countable.
Uncountable sets

Some examples:

- \( \mathbb{R} \)
- \([0,1]\)
- \( \{ \text{infinite sequences of 0s and 1s} \} \)
- \( P(\{0,1\}^*) \).

Diagonalization Proof: Assume towards a contradiction that the set is countable. This gives a correspondence with \( \mathbb{N} \), but we can derive a contradiction.
Proof that $P(\{0,1\}^*)$ not countable

Diagonalization Proof: Assume *towards a contradiction that the set is countable*. This gives a correspondence with $\mathbb{N}$, but we can derive a contradiction.

<table>
<thead>
<tr>
<th>n</th>
<th>f(n)</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$A_1$</td>
<td>0 in $A$ iff 0 is not in $A_1$</td>
</tr>
<tr>
<td>2</td>
<td>$A_2$</td>
<td>00 in $A$ iff 00 is not in $A_2$</td>
</tr>
<tr>
<td>3</td>
<td>$A_3$</td>
<td>$0^3$ in $A$ iff $0^3$ is not in $A_3$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Given the function $f$

Define $A$ so it couldn't be in the image of $f$
Proof that $P(\{0,1\}^*)$ not countable

Given the function $f$

<table>
<thead>
<tr>
<th>$n$</th>
<th>$f(n)$</th>
<th>$A_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$A_1$</td>
<td>$0 \in A \iff 0 \not\in A_1$</td>
</tr>
<tr>
<td>2</td>
<td>$A_2$</td>
<td>$00 \in A \iff 00 \not\in A_2$</td>
</tr>
<tr>
<td>3</td>
<td>$A_3$</td>
<td>$0^3 \in A \iff 0^3 \not\in A_3$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

A is defined by "$0^n \in A \iff 0^n \not\in A_n$"

BUT since $A$ is a set of strings, it is the image of some int $c$.

Is $0^c \in A$?
Proof that $P(\{0,1\}^*)$ not countable

A is defined by "$0^n$ in $A$ iff $0^n$ is not in $A_n$"

BUT since $A$ is a set of strings, it is the image of some int $c$.

Is $0^c$ in $A$?

Diagonalization???

Self-reference

"Is $0^c$ an element of $f(c)$?"
Why is the set of Turing-recognizable languages countable?

A. It's equal to the set of all TMs, which we showed is countable.
B. It's a subset of the set of all TMs, which we showed in countable.
C. Each Turing-recognizable language is associated with a TM, so there can be no more Turing-recognizable languages than TMs.
D. More than one of the above.
E. I don't know.
What's the "size" of the set of Turing-decidable languages over fixed $\Sigma$?

A. It is finite.
B. It must be countable.
C. It must be uncountable.
D. We haven't proved anything about it yet.
E. I don't know.
Satisfied?

• Maybe not …

• What's a specific example of a language that is not Turing-recognizable? or not Turing-decidable?

• Idea: consider set that, were it to be Turing-decidable, would have to "talk" about itself.
A_{TM}

Recall A_{DFA} = \{<B,w>|B \text{ is a DFA and } w \text{ is in } L(B)\}

A_{TM} = \{<M,w>|M \text{ is a TM and } w \text{ is in } L(M)\}

What is A_{TM}?
A. A Turing machine whose input is codes of TMs and strings.
B. A set of pairs of TMs and strings.
C. A set of strings that encode TMs and strings.
D. Not well defined.
E. I don't know.
Define the TM \( N \):

1. Simulate \( M \) on \( w \).
2. If \( M \) accepts, accept. If \( M \) rejects, reject.
Define the TM \( N = \) "On input \( <M,w> \):
1. Simulate \( M \) on \( w \).
2. If \( M \) accepts, accept. If \( M \) rejects, reject."

What is \( L(N) \)?
A. \( A_{TM} \)
B. Some superset of \( A_{TM} \)
C. \( \{<M,w> \mid M \text{ is a TM and } w \text{ is a string}\} \)
D. I don't know.
\( A_{TM} \)

\[ A_{TM} = \{<M,w> \mid M \text{ is a TM and } w \text{ is in } L(M) \} \]

Define the TM \( N = \) "On input \( <M,w> \):
1. Simulate \( M \) on \( w \).
2. If \( M \) accepts, accept. If \( M \) rejects, reject."

Which statement is true?
A. \( N \) decides \( A_{TM} \)
B. \( N \) recognizes \( A_{TM} \)
C. \( N \) always halts
D. I don't know.
$A_{TM}$

$A_{TM} = \{ <M,w> \mid M \text{ is a TM and } w \text{ is in } L(M) \}$

Define the TM $N = "\text{On input } <M,w>:\n1. Simulate } M \text{ on } w.\n2. If } M \text{ accepts, accept. If } M \text{ rejects, reject."}$

**Conclude:** $A_{TM}$ is Turing-recognizable.

**Is it decidable?**
Diagonalization proof: $A_{TM}$ not decidable \textit{Sipser 4.11}

Assume, towards a contradiction, that it is.

I.e. let $M_{ATM}$ be a Turing machine such that for every TM $M$ and every string $w$,

- Computation of $M_{ATM}$ on $<M,w>$ halts and accepts if $w$ is in $L(M)$.
- Computation of $M_{ATM}$ on $<M,w>$ halts and rejects if $w$ is not in $L(M)$. 

Diagonalization proof: $A_{TM}$ not decidable \textit{Sipser 4.11}

Assume, towards a contradiction, that it is.

I.e. let $M_{ATM}$ be a Turing machine such that for every TM $M$ and every string $w$,

- Computation of $M_{ATM}$ on $<M,w>$ halts and accepts if $w$ is in $L(M)$.
- Computation of $M_{ATM}$ on $<M,w>$ halts and rejects if $w$ is not in $L(M)$.

If $N$ is TM with $L(N) = \{ w \mid w \text{ starts with 0} \}$ and $N$ does not halt on all strings not in $L(N)$, what is result of computation of $M_{ATM}$ on $<N, 11>$?

A. $M_{ATM}$ halts and accepts.
B. $M_{ATM}$ halts and rejects.
C. $M_{ATM}$ loops.
D. I don't know.
Diagonalization proof: $A_{\text{TM}}$ not decidable

Assume, towards a contradiction, that $M_{\text{ATM}}$ decides $A_{\text{TM}}$

Define the TM $D = "\text{On input } <M>:"
1. Run $M_{\text{ATM}}$ on $<M, <M>>$. 
2. If $M_{\text{ATM}}$ accepts, reject; if $M_{\text{ATM}}$ rejects, accept."
Diagonalization proof: $A_{TM}$ not decidable \textit{Sipser 4.11}

Assume, towards a contradiction, that $M_{ATM}$ decides $A_{TM}$

Define the TM $D =$ "On input $<M>$:
1. Run $M_{ATM}$ on $<M, <M>>$.
2. If $M_{ATM}$ accepts, reject; if $M_{ATM}$ rejects, accept."

Is $D$ a decider?
A. Yes: it's a TM that always halts.
B. No: it's a well-defined TM but may loop.
C. No: it's not even a well-defined TM.
D. I don't know.
Diagonalization proof: $A_{TM}$ not decidable  

Assume, towards a contradiction, that $M_{ATM}$ decides $A_{TM}$

Define the TM $D = "On input <M>:"

1. Run $M_{ATM}$ on $<M, <M>$. 
2. If $M_{ATM}$ accepts, reject; if $M_{ATM}$ rejects, accept.

If $M_0$ is a TM with $L(M_0) = \emptyset$, what is result of computation of $D$ with input $<M_0>$?

A. Halt and accept. 
B. Halt and reject. 
C. Loop. 
D. I don't know.
Diagonalization proof: $A_{TM}$ not decidable  

Assume, towards a contradiction, that $M_{ATM}$ decides $A_{TM}$

Define the TM $D = \text{"On input } <M>:\n$
1. Run $M_{ATM}$ on $<M, <M>>$.
2. If $M_{ATM}$ accepts, reject; if $M_{ATM}$ rejects, accept.

If $M_1$ is a TM with $L(M_1) = \Sigma^*$, what is result of computation of $D$ with input $<M_1>$?
A. Halt and accept.
B. Halt and reject.
C. Loop.
D. I don't know.
Diagonalization proof: $A_{TM}$ not decidable

Assume, towards a contradiction, that $M_{ATM}$ decides $A_{TM}$

Define the TM $D = "On input <M>:"

1. Run $M_{ATM}$ on <M, <M>>.
2. If $M_{ATM}$ accepts, reject; if $M_{ATM}$ rejects, accept.

Consider running $D$ on input <D>. Because $D$ is a decider:

- either computation halts and accepts …
- or computation halts and rejects …
{a^n b^n | n ≥ 0}

{a^n b^n a^n | n ≥ 0}

A_{TM}

Regular

Context-Free

Decidable

Turing-Recognizable
Do we have to diagonalize?

• Next time: undecidability proofs without diagonalization (or counting).