Today’s lecture

- Countable and uncountable sets
- Diagonalization
- There are languages not in RE or coRE
- An interesting undecidable language
Countable Sets

• You can make an infinite list of all natural numbers \( N: 0,1,2,3,4,\ldots \).

• You can make an infinite list containing all integers \( Z: 0,1,-1,2,-2,3,-3,4,-4,\ldots \).

• Definition: a set \( S \) is countable if there is a bijection \( f: N \to S \)
  
  - Informally: you can list the elements of \( S \): \( f(0), f(1), f(2), \ldots \).
  
  - \( N \) and \( Z \) are countable.
Countable Sets

- You can make an infinite list of all natural numbers $\mathbb{N}$: 0, 1, 2, 3, …
- You can make an infinite list containing all integers $\mathbb{Z}$: 0, 1, -1, 2, -2, 3, -3, 4, -4, …
- Definition: a set $S$ is countable if there is a bijection $f: \mathbb{N} \rightarrow S$
  - Informally: you can list the elements of $S$: $f(0), f(1), f(2), \ldots$
  - $\mathbb{N}$ and $\mathbb{Z}$ are countable

Question: Is any of these sets countable? $\mathbb{Q}$ (rational numbers), $\mathbb{R}$ (real numbers), $\mathbb{C}$ (complex numbers)

A) None of them is countable
B) $\mathbb{Q}$ is countable, but $\mathbb{R}$ and $\mathbb{C}$ are not
C) $\mathbb{Q}$ and $\mathbb{R}$ are countable, but $\mathbb{C}$ is not because
D) They are all countable
Diagonalization

- Theorem: R is not countable. Even \([0,1)\) is uncountable!
- Proof: assume for contradiction it is countable
- List its element \([0,1)\)
  (1) 0.0100010...
  (2) 0.1101000...
  (3) 0.0110001...
  (4) 0.1010101...
  (5) 0.1111000...
- Flip the diagonal: \(r=0.10011...\)
- The real number \(r\) is not on the list: Contradiction!
Some countable sets

- The set of all strings \( \{0,1 \}^* \) is countable
  - Just list them in lexicographic order: \( \varepsilon, 0, 1, 00, 01, 10, 11, 000, \ldots \)
  - Lexicographic order: first sort by length, then alphabetically.

- The set \( TM = \{ <M> \mid M \text{ is a Turing Machine} \} \)

- RE: The set of all Turing Recognizable languages
  - \( L(M_1), L(M_2), \ldots L(M_k), \ldots \)

- coRE: \( L(M_1), L(M_2), \ldots L(M_k), \ldots \)

- RE U coRE
Some uncountable sets

- Recall: \{0,1\}^* = \{w[i] | i in N\} is countable
- \mathbb{R} is uncountable
- \[0,1\] is uncountable
- The set of all languages \text{P}\{\{0,1\}^*\} is not countable
  - Languages can be represented by real numbers:
  - \(r=0.10010101011100.... \leftrightarrow \{w[i] | \text{the } i^{\text{th}} \text{ digit of } r \text{ is } 1\}\)
  - Different reals are mapped to different languages
  - There are at least as many languages as \(r\) in \[0,1\]
Summary

- $\text{RE U coRE}$ is countable
- $\mathcal{P}(\{0,1\}^*)$ is uncountable
- There is a language $L$ in $\mathcal{P}(\{0,1\}^*) - (\text{RE U coRE})$!
Undecidable Languages

- There are uncountably many undecidable languages!
- In fact, there are uncountably many languages that are not even in RE (or coRE)!
- Questions:
  - Can we find a specific language not in RE or coRE?
  - Can we find interesting languages not in RE or coRE?
  - Is $\text{HALT}_\text{TM}$ undecidable?
Diagonalization

- $[0,1)$ is uncountable because given any list of $r$ in $[0,1)$
  - $0.0100010...$
  - $0.1101000...$
  - $0.0110001...$
  - $0.1010101...$
  - $0.1111000...$

  we can build an $r=0.10011...$ that is not in the list

- We can make a list of all $L$ in RE: $L(M_1), L(M_2), L(M_3)...$

- Can we build a language not in this list?
A language not in RE

• We want a language L that is
  – different from $L(M_1)$
  – different from $L(M_2)$
  – different from $L(M_3)$
  – .....  
  – different from $L(M_k)$
  – .....
A language not in RE

- We want a language $L$ that is
  - different from $L(M_1)$
  - different from $L(M_2)$
  - different from $L(M_3)$
  - .....  
  - different from $L(M_k)$
  - .....  

Two numbers are different if they differ at some digit.

Two languages are different if they differ at some string $w$:
Either $w$ in $L(M)$ but not in $L$, Or $w$ in $L$ but not in $L(M)$.
A language not in RE

• We want a language $L$ that is
  - different from $L(M_1)$ at $<M_1>$
  - different from $L(M_2)$ at $<M_2>$
  - different from $L(M_3)$ at $<M_3>$
  - .....  
  - different from $L(M_k)$ at $<M_k>$
  - ..... 

• $L = \{<M> \mid M \text{ it a TM such that } <M> \text{ is not is } L(M) \}$
A “diagonal” language

- Diag = \{ <M> | M is a TM s.t. <M> is not in L(M) \}
- Why is Diag different from L(M_k)?
  - We need a string \( w \) that belongs to one but not the other
  - Let \( w = <M_k> \)
  - If \( <M_k> \) is in L(M_k), then
    - Not (Not “\( <M_k> \) is in L(M_k)”)
    - “\( <M_k> \) is not in L(M_k)” is false
    - \( <M_k> \) is not in Diag
  - If \( w \) is not in Diag, then \( w \) is in L(M_k)
A “diagonal” language

- \( \text{Diag} = \{ <M> \mid M \text{ is a TM s.t. } <M> \text{ is not in } L(M) \} \)

- Why is \( \text{Diag} \) different from \( L(M_k) \)?
  - We need a string \( w \) that belongs to one but not the other
  - Let \( w = <M_k> \)
  - If \( <M_k> \) is in \( L(M_k) \),
    - \( \text{Not (Not } "<M_k> \text{ is in } L(M_k)\)"
    - \( "<M_k> \text{ is not in } L(M_k)\)" is false
    - \( <M_k> \text{ is not in } \text{Diag} \)
  - If \( w \) is not in \( \text{Diag} \), then \( w \) is in \( L(M_k) \)

Question: What can you tell about \( \text{Diag} \)?

- A) \( \text{Diag} \) is in RE, but not in coRE
- B) \( \text{Diag} \) is coRE, but not in RE
- C) \( \text{Diag} \) is neither in RE nor in coRE
- D) \( \text{Diag} \) is decidable
- E) I don’t know
Diag is in coRE

- Diag=$\{<M> \mid M \text{ is a TM s.t. } <M> \text{ is not in } L(M) \}$
- Here is a recognizer for Diag

\[M_{\text{diag}}(w) =\]

1) Check if $w=<M>$ for some TM $M$. If not, accept.
2) Parse $w$ as $<M>$ for some TM $M$
3) Run $M$ on input $w$
4) If $M(w)$ accepts, then accept, else reject.
Diag is in coRE

- **Diag** = \{ <M> | M is a TM s.t. <M> is not in L(M) \}
- Here is a recognizer for **Diag**: 
  
  \[ M_{\text{diag}}(w) = \]
  
  1) Check if \( w = <M> \) for some TM M. If not, accept.
  2) Parse \( w \) as \( <M> \) for some TM M
  3) Run M on input \( w \)
  4) If M(w) accepts, then accept, else reject.

Is \( M_{\text{diag}} \) a decider?

A) Yes, because \( L(M_{\text{diag}}) = \text{Diag} \)
B) No, because it can loop at step 1
C) No, because it can loop at step 2
D) No, because it can loop at step 3
E) I don’t know
Diag is in coRE

- Diag = \{ <M> \mid M \text{ is a TM s.t. } <M> \text{ is not in } L(M) \}
- Here is a recognizer for Diag:
  
  \[ M_{\text{diag}}(w) = \]
  
  1) Check if \( w = <M> \) for some TM \( M \). If not, accept.
  2) Parse \( w \) as \( <M> \) for some TM \( M \)
  3) Run \( M \) on input \( w \)
  4) If \( M(w) \) accepts, then accept, else reject.

What can you say about Diag?

A) Diag is decidable
B) Diag is in RE, but not coRE
C) Diag is in coRE, but not RE
D) Diag is neither in RE nor coRE
E) I don't know
Summary

- Diag is in coRE because $L(M_{\text{diag}}) = \text{Diag}$
- Diag is not in RE because $\text{Diag} \neq L(M_k)$ for all TM $M_k$
- Diag is undecidable
- Diag is in coRE, but not RE
- So far: We have found specific languages not in RE or coRE.
- Can we find more interesting examples?
  - What about $\text{HALT}_{\text{TM}} = \{<M,w> | M(w) \text{ terminates}\}$?
  - What about $\text{A}_{\text{TM}} = \{<M,w> | M(w) \text{ accepts}\}$?
$A_{TM}$ is undecidable

- Assume for contradiction $A_{TM}$ is decidable
- $A_{TM}$ is also decidable
- $MM = \{<M,<M>> \mid M \text{ is a TM}\}$ is decidable
- $DD = MM \cap A_{TM}$ is decidable
- Notice:
  - $DD = \{<M,<M>> \mid M \text{ is a TM such that } M \text{ is not in } L(M)\}$
A_{TM} is undecidable

- Assume for contradiction A_{TM} is decidable
- A_{TM} is also decidable
- MM = \{<M,<M>> | M is a TM\} is decidable
- DD = MM \cap A_{TM} is decidable
- Notice:
  - DD = \{ <M,<M>> | M is a TM such that M is not in L(M) \}

What can you say about DD?

A) DD is decidable
B) DD is undecidable because DD = Diag
C) DD is undecidable but I am not sure why
D) DD is both decidable and undecidable
E) I don’t know
\( A_{TM} \) is undecidable

- Assume for contradiction \( A_{TM} \) is decidable
- \( DD = \{ <M,<M>> \mid M \text{ is a TM such that } M \text{ is not in } L(M) \} \) is decidable. Let \( M_{DD} \) be a decider for \( DD \).
- \( M'_{\text{diag}}(<M>) = \text{Run } M_{DD}(<M,<M>>) \)
  - \( L(M'_{\text{diag}}) = \text{Diag} \)
  - \( M'_{\text{diag}} \) is a decider (\( M'_{\text{diag}} \) always terminates)
- This proves that Diag is decidable: contradiction!
For next Time

- Try to prove that $\text{HALT}^\text{TM}$ is undecidable
- Reading: Sipser \textit{Chapters 3 and 4}
- \textbf{HW6} due on Nov 15
- \textbf{Haskell 4} due Nov 21
- \textbf{Exam 3}: Nov 18