Ex B2

- Part 1: Diagonalization
  - Task 1: Power set of natural numbers
  - Task 2: Languages
- Others

Any way to solve undecidable problems?

Systematic way of dealing with infinite cases?

Review

Mathematical Induction

- A widely accepted proof technique
- Let $\mathbf{N} = \{0, 1, 2, 3, ...\}$
- Let $P(n)$ be a unary relation (set) on $n \in \mathbf{N}$
- If both of the following conditions hold:
  - $P(n_0)$, where $n_0 \in \mathbf{N}$ e.g., $n_0 = 0$
  - For every $n \geq n_0$ ($n \in \mathbf{N}$), $P(n)$ implies $P(n + 1)$
- Then, $P(n)$ is true for every $n \geq n_0$ ($n \in \mathbf{N}$)

Review

Math Induction Proof Pattern

- Main hypothesis: None
- Main conclusion: $\Sigma_{1\leq i\leq n}\, i = n \times (n + 1) / 2$
- Base case ($n = 1$): $\Sigma_{1\leq i\leq 1}\, i = 1 = 1 \times (1 + 1) / 2$
- Induction step
  - Induction hypothesis: $\Sigma_{1\leq i\leq n}\, i = n \times (n + 1) / 2$
  - Conclusion: $\Sigma_{1\leq i\leq (n+1)}\, i = (n + 1) \times (n + 2) / 2$
  - Proof (of the induction step)
    1. $\Sigma_{1\leq i\leq (n+1)}\, i = \Sigma_{1\leq i\leq n}\, i + (n + 1)$ [LHS Conclusion]
    2. $\Sigma_{1\leq i\leq n}\, i + (n + 1) = n \times (n + 1) / 2 + (n + 1)$ [Ind. hyp.]
    3. $n \times (n + 1) / 2 + (n + 1) = (n + 1) \times (n + 2) / 2$ [Arithmetic]
    4. $\Sigma_{1\leq i\leq (n+1)}\, i = (n + 1) \times (n + 2) / 2$ [Transitivity eq/ineq: 1.-3.]
- By Math Induction (shaded), main conclusion holds.

Uncountability, Revisited

- Countable
  - E.g., natural numbers, set of TMs, set of strings
  - Existence of the base case (basis) [well-founded]
  - Applicable inductive techniques: math induction, inductive definition of a set, recursion
- Uncountable
  - E.g., real numbers, set of languages
  - No base case [non-well-founded]
  - The above-mentioned inductive techniques are not applicable. Algorithms cannot handle this properly.

Unit B3: Overview

- Analyze the computability classes of the main problems
  - Review some available tools
  - Understand how to analyze the main problems
- Preview Exercise B3 "Halting Problem"
  - Write up proofs
  - Prepare for mini presentations next time

Main Problems

- The universal language is semi-decidable.
  $ACCEPT_{\mathrm{TM}} = \{(M, w) \mid \mathrm{TM}\ M \text{ accepts } w\}$ w for "word" (input)
- The complement of universal language is non-TM-recognizable.
  $NACCEPT_{\mathrm{TM}} = \{(M, w) \mid \mathrm{TM}\ M \text{ does } \textit{not} \text{ accept } w\} = (ACCEPT_{\mathrm{TM}})'$
- The halting problem is semi-decidable.
  $HALT_{\mathrm{TM}} = \{(M, w) \mid \mathrm{TM}\ M \text{ halts on } w\}$
- Infinite loop detection is non-TM-recognizable.
  $LOOP_{\mathrm{TM}} = \{(M, w) \mid \mathrm{TM}\ M \text{ loops on } w\} = (HALT_{\mathrm{TM}})'$

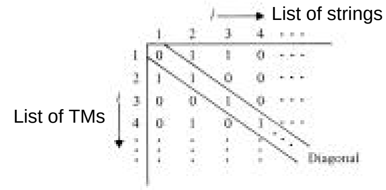
A': set complement of A

Available Tools

- The diagonalization language (L_d) as an example of a non-TM-recognizable set
- Proof by contradiction (also used in the diagonalization technique)
- Theorem: Decidable ⇔ TM-recognizable *and* co-TM-recognizable

Diagonalization Language (L_d)

Review



- Entry 1: M_i accepts w_j
- Entry 0: M_i does not accept w_j
- $L_d = \{w_i \mid M_i$ does not accept $w_i\}$ No TM accepts!

Proof by Contradiction

Review

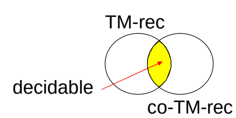
To prove: Statement X

- Assume the negation of X, i.e., *not* X
- Derive a contradiction
- Then, by this proof technique (proof by contradiction) X must be true (because X is either true or false)

Useful Theorems

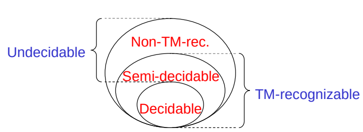
Review

- **Theorem**: Decidable ⇔ TM-recognizable *and* co-TM-recognizable
- **Corollary**: The complement of a decidable problem is decidable.



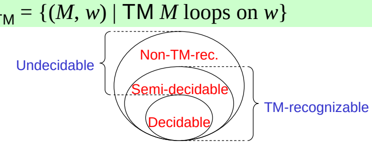
Practice: Co-X

- Co-decidable ⇒
- Co-TM-recognizable ⇒
- Co-semi-decidable ⇒
- Co-non-TM-recognizable ⇒



Group Exercise 1

- Explain (prove) the computability classes for one or more of the following problems:
  - $ACCEPT_{TM} = \{(M, w) \mid$ TM M accepts $w\}$
  - $NACCEPT_{TM} = \{(M, w) \mid$ TM M does *not* accept $w\}$
  - $HALT_{TM} = \{(M, w) \mid$ TM M halts on $w\}$
  - $LOOP_{TM} = \{(M, w) \mid$ TM M loops on $w\}$

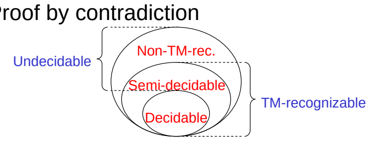


Proof Idea: $ACCEPT_{\text{TM}}$

To show: $ACCEPT_{\text{TM}}$ is semi-decidable.

- $ACCEPT_{\text{TM}}$ is TM-recognizable.
  - Why? Could still be decidable.
- $ACCEPT_{\text{TM}}$ is undecidable.
  - Use: L_d is non-TM-recognizable.
  - Use: Proof by contradiction

Intuition: can't squeeze "uncountable" within "countable



Subproof Idea

To show: $ACCEPT_{\text{TM}}$ is TM-recognizable.

- Existence proof (proof by construction)
  - It is possible to construct the UTM that can simulate the behavior of the input, (M, w), which would relay the acceptability of M on w.

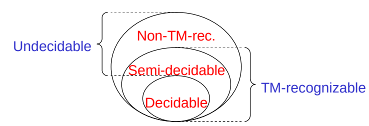
Subproof Idea

To show: $ACCEPT_{\text{TM}}$ is undecidable.

- Proof by contradiction
  - Suppose that $ACCEPT_{\text{TM}}$ is *decidable*.
  - For a string w in L_d, compute its index (decidable). Thus, we have w_i.
  - Compute (M_i, w_i) using the UTM.
  - Reject w *if and only if* (*iff*) M_i accepts w_i.
  - This is an algorithm for L_d.
  - A contradiction (cf. L_d is non-TM-rec.)

Interim Summary

- $ACCEPT_{\text{TM}}$ is TM-recognizable.
- $ACCEPT_{\text{TM}}$ is not decidable.



- $ACCEPT_{\text{TM}}$ is semi-decidable.

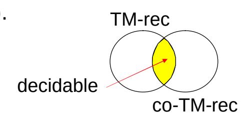
Group Exercise 2

- Explain (prove) the computability classes for each of the following problems:
  - $NACCEPT_{\text{TM}} = \{(M, w) \mid \text{TM } M \text{ does } \textit{not} \text{ accept } w\}$
  - $HALT_{\text{TM}} = \{(M, w) \mid \text{TM } M \text{ halts on } w\}$
  - $LOOP_{\text{TM}} = \{(M, w) \mid \text{TM } M \text{ loops on } w\}$

Proof Idea: $NACCEPT_{\text{TM}}$

To show: $NACCEPT_{\text{TM}}$ is non-TM-recognizable.

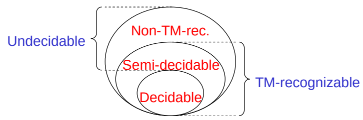
- Proof by contradiction
  - Suppose that $NACCEPT_{\text{TM}}$ is TM-recognizable.
  - $ACCEPT_{\text{TM}}$ becomes decidable (Theorem introduced earlier).
  - A contradiction



Proof Idea: $HALT_{TM}$

To show: $HALT_{TM}$ is semi-decidable.

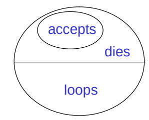
- $HALT_{TM}$ is TM-recognizable. Why?
- $HALT_{TM}$ is undecidable.



Subproof Idea

To show: $HALT_{TM}$ is undecidable.

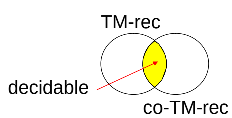
- Proof by contradiction
  - Suppose that $HALT_{TM}$ is *decidable*.
  - If "halt" is detected, accept/reject based on the UTM ($ACCEPT_{TM}$); otherwise, reject.
  - This is an algorithm for $ACCEPT_{TM}$.
  - A contradiction



Proof Idea: $LOOP_{TM}$

To show: $LOOP_{TM}$ is non-TM-recognizable.

- Proof by contradiction
  - Suppose that $LOOP_{TM}$ is TM-recognizable.
  - $HALT_{TM}$ becomes decidable (Theorem).
  - A contradiction



Liar Paradox, Revisited

- "This sentence is false."
- Intuition behind this problem
  - The set of sentences are countable.
  - A sentence in this set can be indexed.
  - The use of "this" can be interpreted as the use of its own index (self reference).
  - The language must have a means for a sentence to refer to an arbitrary sentence (universality) including itself.

Based on Megan's idea

Would this work?

- Suppose that a TM erases all the non-blank symbols when it terminates.
- "This message is left on the tape of the TM when it terminates."
- Reasoning
  - The message itself can be associated with some TM state, e.g., termination.

But where is universality?

Unit Summary

- Tools for the proofs
  - The diagonalization language
  - Proof by construction
  - The theorem on decidability
- Analyze the computability classes of the main problems

Midterm Survey

- To be administered and collected by a volunteer