Parul® University

NAAC A++

The Universal and Diagonalization Languages

Chapter 5: Undecidability

Prof. Riddhi Atulkumar Mehta
Assistant Professor
Department of Computer Science and Engineering

Parul® University

NAAC A++

Content

INDEX

Learning Objectives

By the end of this lecture, you should be able to:

- Define the universal language LuL u
- Define the diagonalization language LdL d
- Understand the concepts of encoding, simulation, and self-reference
- Use diagonalization to prove undecidability

The Universal Language *Lu*

Definition:

- Lu={⟨M,w⟩∣M is a TM and M accepts w}
- It contains all encodings of TMs and inputs such that the machine accepts the input.
- Captures the behavior of any TM on any input.

Properties of Lu

- Recursively Enumerable (RE):

  There exists a TM (UTM) that accepts all strings in Lu

- Not Recursive (Decidable):

  There is no TM that can decide for *every* input whether it's in Lu.

🧠 The Halting Problem is reducible to Lu

Diagonalization Language Ld

Definition:

- Ld={⟨M⟩|M is a TM and M does not accept ⟨M⟩}
- Think of M being run on its own description.
- Ld contains all TMs that do NOT accept themselves.

Diagonalization: The Idea

- Inspired by Cantor’s diagonal argument
- Show that Ld is not recursively enumerable (not RE)
- Suppose Ld is RE
  - There is a TM D accepting it
  - Ask: Does D accept ⟨D⟩?
  - Leads to contradiction

Summary of Languages

Language	Definition	RE?	Recursive?
L_u	$\langle M, w\rangle$ where $M(w)$ accepts	✅ Yes	❌ No
L_d	$\langle M\rangle$ where M does **not** accept $\langle M\rangle$	❌ No	❌ No

Applications

Forms the basis for:

- The Halting Problem
- Rice’s Theorem
- Proving undecidability of other decision problems
- Foundational for complexity theory and logic

Parul® University | NAAC GRADE A++

https://paruluniversity.ac.in/