Computability Theory: Decidability and Recognizability

- Encoding Turing Machines and the Universal TM
- Computability
- Halt: Undecidable Problems using Diagnolization
- Accept: Undecidable Problems using Diagnolization
- Turing Reductions
- Mapping Reductions
- Undecidable and Unrecognizable Problems
- Rice Theorem

IMDAD ULLAH KHAN

Reduction

Turing Reduction

Reduction is a general method to prove undecidability

Reduce an undecidable problem to X to prove undecidability of X

We want to prove that a language L is undecidable. We prove that if L is decidable, then so is A_{TM}

$$A_{TM} \leq L$$

pronounced as A_{TM} is reducible to L

This is impossible as we know A_{TM} is undecidable

$A \leq B$ means B is at least as hard as A

▷ technically called Turing reduction

Note that the other direction, does not prove that L is undecidable

Reduction: $A_{TM} \leq$ HALT

Using reduction we show that halting problem is undecidable

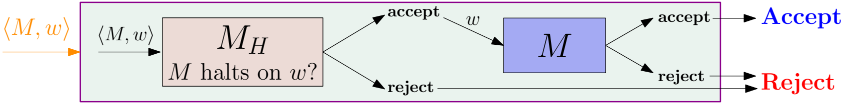
HALT $= \{\langle M, w\rangle :\ M$ halts on input $w\}$ is undecidable

Proof: Suppose HALT is decidable and M_H is the decider TM

Using M_H (the decider) we construct a TM M_A to decide A_{TM}

$M_A(\langle M, w\rangle)$: Run M_H on $\langle M, w\rangle$ and

- If M_H accepts, run M on w until it halts and output as M does
- If M_H rejects, then reject



M_A on input $\langle M, w\rangle$, runs M_H and (possibly) M to decide whether M accepts w