Giorgi Japaridze

Theory of Computability

Reducibility

Chapter 5

The undecidability of the halting problem

Let $HALT_{TM} = \{\langle M,w\rangle \mid M \text{ is a TM and } M \text{ halts on input } w\}$

$HALT_{TM}$ is called the *halting problem*.

Theorem 5.1: $HALT_{TM}$ is undecidable.

Proof idea: Assume, for a contradiction, that $HALT_{TM}$ is decidable. I.e. there is a TM R that decides $HALT_{TM}$. Construct the following TM S:

S = "On input $\langle M,w\rangle$, an encoding of a TM M and a string w:

1. Run R on input $\langle M,w\rangle$.
2. If R rejects, *reject*.
3. If R accepts, simulate M on w until it halts.
4. If M has accepted, *accept*; if M has rejected, *reject*."

- If M works forever on w, what will S do on $\langle M,w\rangle$?
- If M accepts w, what will S do on input $\langle M,w\rangle$?
- If M explicitly rejects w, what will S do on $\langle M,w\rangle$?

Thus, S decides the language But this is impossible (Theorem 4.11)

Definition of mapping reducibility

Let A and B be languages over an alphabet Σ. We say that A is *mapping reducible* to B, written $A \leq_m B$, if there is a computable function $f: \Sigma^* \to \Sigma^*$ such that, for every $w \in \Sigma^*$,

$$w \in A \text{ iff } f(w) \in B.$$

The function f is called a *mapping reduction* of A to B.