Chapter 6

Advanced Topics in Computability Theory

The Recursion Theorem

A self-reproducing machine $SELF$ is a machine that disregards its input and produces its description on the input.

We will contruct such a machine. For that matter, we need to modify the Turing machine and the Turing machine description so as to embrace the concept of concatenation.

Concatenating Turing Machines

For two Turing machines A and B, $A \cdot B$ is the Turing machine M that on input x behaves as follows:

- M acts as A on x;
- if A rejects so does A;
- if A accepts M acts as B, where the computation with respect to B's code starts with the tape contents and the head location at the moment of A's termination.
- if B accepts so does M; if B rejects so does M.

Concatenating Turing Machine Descriptions

For two Turing machine descriptions $a = \langle A \rangle$ and $b = \langle B \rangle$, the string ab (that is, a followed by b) is the description of $A \cdot B$.

Fixed Output Turing Machines

For each fixed string w, there exists a machine that, for all inputs x, writes w on its tape, moves the head to the first character of w, and then accepts.

Fix one strategy for constructing such a machine. The machine for w is a machine that has the characters of w encoded in the state and produces those encoded characters on the tape.

This strategy can be implemented on a Turing machine. Fix such a machine and then for all w, let P_w denote the output of the machine on input w.

Constructing $SELF$

$SELF$ is the concatenation, $A \cdot B$, of two machines A and B. Thus, for all inputs w, $SELF$ outputs $\langle A \cdot B \rangle$.

On input x, the machine B, behaves as follows:

- B computes P_x and inserts it in front of x.
- B erases other parts of the tape and accepts.

Suppose x is the description of a Turing machine C, that is, $\langle C \rangle$. Then B produces $\langle P_x \cdot C \rangle$, that is, the description of the machine that executes P_x and then executes C.

Final Step

The property B has: for all Turing machines C, B on input $\langle C \rangle$ produces $\langle P_{\langle C \rangle} \cdot C \rangle$.

In particular, if $C = B$, then B on input $\langle B \rangle$ produces $\langle P_{\langle B \rangle} \cdot B \rangle$.

Let A be the machine $P_{\langle B \rangle}$ and let $SELF = A \cdot B$. For all inputs x, during the execution of A-part, $SELF$ produces $\langle B \rangle$ and then during the execution of B-part, it produces $SELF = A \cdot B$.

Recursion Theorem

Theorem 6.3. **Let T be a Turing machine that computes a function $t : \Sigma^* \times \Sigma^* \to \Sigma^*$, where the input to T is specified in the form k Then there exists a Turing machine R that computes a function $r : \Sigma^* \to \Sigma^*$ such that for all w**

$$r(w) = t(\langle R \rangle, w).$$

Encoding an Input to T

Assume that ',' is represented by a special character $\#$ not in Σ. The two inputs x and y to T, $x, y \in \Sigma^*$, are given as the word $x\#y$.

Proof The machine R we'll design is $A \cdot B \cdot T$ for some machines A and B

The role of A is to insert in front of its input $x \in \Sigma^*$ $\langle B \cdot T \rangle \#$, thereby creating $\langle B \rangle \langle T \rangle \# x$.

The role of B is to insert in front of its input $\langle A \rangle$.

Thus, on input x, $A \cdot B$ produces $\langle A \rangle \langle B \rangle \langle T \rangle \# x$. This is equal to $\langle R \rangle \# x$.

Now, T produces $t(\langle R \rangle, x)$ as desired.

B is now set to be a machine that divides its input into the form $\langle C \rangle u$ and inserts the description of a machine D defined by: on input w, D inserts $\langle C \rangle \langle T \rangle$ in front of w.

∎

Using the Recursion Theorem to Construct $SELF$

Set T to be a machine that on input $\langle M, w\rangle$ outputs $\langle M\rangle$.

Set R to be a machine from the theorem with respect to this T.

On input w, R executes T on $\langle R, w\rangle$ and so outputs $\langle R\rangle$.

Simpler Proof That A_{TM} Is Undecidable

Theorem 6.5. **A_{TM} Is Undecidable.**

Proof Assume A_{TM} is decidable. Let H be a Turing machine that decides the complement of A_{TM}. By Recursion Theorem, there is a machine B that, on input w, executes H on $\langle B, w\rangle$.

For all w, B accepts $w \Leftrightarrow H$ accepts $\langle B, w\rangle \Leftrightarrow \langle B, w\rangle \in A_{\mathrm{TM}} \Leftrightarrow B$ does not accept w. ∎

Minimum Description

Define MIN_{TM} be the set of all $\langle M \rangle$ with the following property: there is no machine N such that $|\langle N \rangle| < |\langle M \rangle|$ and $L(M) = L(N)$.

Theorem 1. ***6.7.*** **MIN_{TM} is not Turing-recognizable.**

Proof Assume, to the contrary, that MIN_{TM} is Turing-recognizable. Then there is an enumerator E of all members of MIN_{TM}. Let T be a machine that on input $\langle M, w\rangle$ behaves as follows: (i) T simulates E until a Turing machine that is longer than $\langle M\rangle$ is produced, and then, (ii) T simulates that machine on input w.

According to Recursion Theorem, there is a machine R that on input w executes T on input $\langle R, w\rangle$. Let D be the machine that R finds.

Then D and R recognize the same language and $\langle D\rangle$ is longer than $\langle R\rangle$, which contradicts the assumption that $\langle D\rangle$ appears in E's enumeration.

∎