Recursive Inseparability

Definition 1. *Two languages are called* **recursively inseparable** *if there is no recursive language R such that $L_1 \cap R = \phi$ and $L_2 \subset R$.*

Theorem 1. *Let $L_1 = \{M : M(M) = yes\}$ and $L_2 = \{M : M(M) = no\}$. Then L_1 and L_2 are recursively inseparable.*

Proof: Suppose not. Let recursive language R separate them:

$L_1 \cap R = \phi$ and $L_2 \subset R$.

Consider the TM M_R deciding R.

What is $M_R(M_R)$?

If it is "yes" then $M_R \in L_1$ and this implies $M_R(M_R) = no$.

If it is "no" then $M_R(M_R) \in L_2$ and this implies $M_R(M_R) = yes$.

Contradiction.

Corollary 1. *Let* $L'_1 = \{M : M(\epsilon) = yes\}$ *and* $L'_2 = \{M : M(\epsilon) = no\}$*. Then* L'_1 *and* L'_2 *are recursively inseparable.*

Proof: Suppose not. Let R' separate the two. Then we will show that we can separate L_1 and L_2 of the Theorem.

For any TM M, let M' be a TM which, on any input, generates M's description and simulates M on it.

Consider the following TM N:

On input M:

0. If M is not a TM description accept.

1. Construct M' and test whether $M' \in R'$.

2. If yes, then accept else reject.

The recursive language decided by N separates L_1 from L_2.

Recursion Theorem

"Machines can reproduce themselves."

Lemma 1. *There is a computable function* $q : \Sigma^* \rightarrow \Sigma^*$*, where, for any string* w*,* $q(w)$ *is the description of a TM machine* P_w *that prints out* w *and then halts.*

Proof: The following TM computes $q(w)$:

On input w:

1. Construct the following TM P_w:

   On any input:

   1. erase input

   2. Write w on the tape (string).

   3. Halt.

2. Output $< P_w >$.

$$\mathbf{M(x)} = < \mathbf{M} >$$

We will construct a TM S that ignores its input and outputs its own description.

Print out this sentence.

Print out two copies of the following, the second one in quotes:

"Print out two copies of the following, the second one in quotes:"

Theorem 2. *There exists a TM that, on any input, outputs a copy of its own description.*

Proof

We will construct such a TM S. It has two parts A and B, i.e., $< S > = < AB >$, defined as follows.

1. $A = P_{<B>}$

2. B:

On input $< M >$, where M is a portion of a TM:

2.1 Compute $q(< M >)$.

2.2 Combine the result with $< M >$ to make a complete TM description.

2.3 Output this description and halt.

Recursion Theorem

Theorem 3. *Let T be a TM that computes a function $t : \Sigma^* \times \Sigma^* \to \Sigma^*$.*

There is a TM R that computes a function $r : \Sigma^* \to \Sigma^*$, where for every w,

$$r(w) = t(< R >, w).$$

To make a TM R that can obtain its own description **and** then compute with it, we need only to make a machine T that takes an extra input that receives the description of the machine.

Proof

TM R has three parts A, B, and T.

1. $A = P_{<BT>}$. (After A runs the tape contains $< BT >$.)

2. $B = q(< BT >)$ (Applies q to the output of A to get $< A >$).

B then combines A, B, T into a single machine, writes its description on the tape.

3. T takes the output of B and the input w and computes with it.