Rice's Theorem

Video Lecture "Rice's Theorems", related practice problems and their solutions are on http://cse.iitkgp.ac.in/ abhij/course/theory/FLAT/Spring20/

1 Nontrivial Properties

For a fixed finite alphabet Σ, a *property* of the recursively enumerable sets is a map

$$P : \{ \text{ r.e subsets of } \Sigma^* \} \rightarrow \{T, F\}$$

Here T represents truth and F represents falsity.

Eg: The property of emptyness has the following map:

$P(A) = T$ if $A = \emptyset$ and F for any other set A.

We can assume that an r.e subset is represented in a finite manner using the finite representation of a Turing machine that accepts it. However, there could be multiple Turing machines accepting the same r.e subset. The property of the r.e subset should be independent of the description of a Turing machine accepting it or being chosen to represent it. Some examples of properties: finiteness, whether the set is regular, whether the set contains ϵ etc.

A property P is decidable if we can design a total Turing machine that takes a Turing machine M as input and accepts the r.e set $L(M)$ if and only if $P(L(M)) = T$.

A property is said to be *nontrivial* if it is neither universally true nor universally false. Note that there are only two trivial properties and any other property is non-trivial.

2 Rice's Theorem

Statement: Every nontrivial property of the r.e sets is undecidable. In other words, if P is a nontrivial property of the r.e sets, then the set $T_P = \{M | P(L(M)) = T\}$ is not recursive.

Proof: Let P be a nontrivial property of r.e sets. By definition of nontriviality, there is a r.e set A such that $P(A) = T$ and a r.e set B such that $P(B) = F$. Without loss of generality let $P(\emptyset) = F$ (If $P(\emptyset) = T$ then a symmetric argument can be given).

Let K be a Turing machine that accepts B.

We give a reduction from HP to the set $\{M|P(L(M)) = T\}$. Since HP is undecidable, this would show that T_P is not recursive. So for an input $M\#x$ of HP we construct a machine $M' = \sigma(M\#x)$ that does the following:

1. M' takes a string y as input.
2. It simulates M on x.
3. If M halts on x, then M' simulates K on input y and accepts y if K accepts y.

Note that:
M halts on $x \implies L(M') = B \implies P(L(M')) = P(B) = T$.
M does not halt on $x \implies L(M') = \emptyset \implies P(L(M')) = P(\emptyset) = F$.
So, $M\#x$ is a "yes" instance of HP if and only if M' is such that $P(L(M')) = T$. This completes the description of the reduction from HP to T_P.

3 Monotone properties

A property on r.e sets is said to be *monotone* if for all r.e sets A, B such that $A \subseteq B$, $P(A) \leq P(B)$ (we are assuming the $F \leq T$). In words, If a r.e subset has the property P then all r.e sets that are its supersets also have the property P. Eg: An r.e set is infinite.

When a property on r.e sets is non-monotone this means that there is a pair of r.e sets A, B such that $A \subseteq B$ but $P(A) = T$ and $P(B) = F$.

4 Rice's Theorem Part II

Statement: No non-monotone property of the r.e sets is semidecidable. In other words, if P is a non-monotone property of the r.e sets, then the set $T_P = \{M|P(L(M)) = T\}$ is not r.e.

Proof: Since P is non-monotone there is a pair of r.e sets A, B such that $A \subseteq B$ but $P(A) = T$ and $P(B) = F$. Let M_A be a Turing machine for A and M_B be a Turing machine for B.

In order to show that T_P is not recursively enumerable we give a reduction from $\overline{HP}$ (We know that $\overline{HP}$ is not recursively enumerable). By definition of a reduction, we can equivalently reduce HP to $\overline{T_P} = \{M|P(L(M)) = F\}$. Thus, given an instance $M\#x$ for HP we construct a machine M' as follows:

1. M' takes a string y as input.
2. On separate tracks, M' simulates M_A on y, M_B on Y and M on x step by step. So, the first step is a step of simulation of M_A on y, the second step is a step of simulation of M_B on y, the third step is a step of simulation of M on x and so on.
3. M' accepts y if (a) M_A accepts y, or (b) M_B accepts y and M halts on x.

Now let us see what the language of M' is:

If M does not halt on x then event (b) of step 3 of the description of M' will never occur. This, $L(M') = L(M_A) = A$.

If M halts on x then the strings that are accepts by M' are those in $L(M_A) \cup L(M_B) = L(M_B)$ (as $A \subseteq B$). Thus, $L(M') = L(M_B) = B$.

Thus, M halts on $x \implies L(M') = L(M_B) \implies P(L(M')) = P(B) = F$.

M does not halt on $x \implies L(M') = L(M_A) \implies P(L(M')) = P(A) = T$.

This completes the reduction from HP to $\overline{T_P}$ and therefore a reduction from $\overline{HP}$ to T_P. Thus, T_P is not recursively enumerable.