Reductions, Oracles and a Hierarchy of Undecidable Problems

We defined a language A to be reducible to a language B iff there is a total recursive map f such that $\forall x, x \in A \quad \text{iff} \quad f(x) \in B$. In words, this is equivalent to saying that if I am given an "oracle" (a subroutine) (we will always assume that an "oracle" halts in finite time on all inputs) which solves (membership in) B, then with **just one call** to this oracle, I can answer membership in A.

This notion of reduction is called **many-one** reduction and we write $A \leq_m B$ to denote that A is "many-one" reducible to B.

Note that **every** r.e. set is many-one redicible to L_u. We say that an r.e. set is **r.e. complete** if every r.e. set is many-one reducible to it. Thus, L_u is r.e. complete. We showed a many-one reduction of L_u to NONEMPTY= $\{M : L(M) \neq \emptyset\}$; so also NONEMPTY is r.e. complete. (Why is that what we showed is not a many-one reduction of L_u to EMPTY ?)

Another natural notion of reduction is when we allow any finite number of oracle calls. This is called a Turing reduction.

Definition A set A is Turing reducible to a set B denoted $A \leq_T B$ if there is a halting TM M with access to an oracle for B which accepts A.

If B is recursive and $A \leq_T B$, then A is recursive too. If B is recursive, then the oracle can be "implemented" to run in finite time. But, even if B is not recursive, we still can define TM 's as above with an oracle for B; it is just a hypothetical machine we use to study the relative hardness of A and B. A TM with an oracle for B is often written M^B. We will say that two sets A and B are equivalent (in hardness) - written - $A \equiv_T B$, if each of them is reducible to the other. All recursive sets are equivalent (Why?)

Are there sets harder than all r.e. sets, i.e., harder than L_u ? Indeed, there is an infinite hierachy of harder and harder sets of which the r.e. sets form only the first level. Consider TM's with L_u as an oracle and suppose $L_u^{(2)}$ is the universal language for them; i.e.,

$$L_u^{(2)} = \{< w, M^{L_u} >: M^{L_u} \text{ aceepts } w\}.$$

Theorem $L_u^{(2)} \not\leq_T L_u$.

The proof is word for word almost the same as the proof that L_u is not recursive. Now we can define the universal language for TM's with $L_u^{(2)}$ as oracle etc...