Uncountable and Undecidable

Countable and Uncountable

There are levels of infinity. The first level is the ***countable*** sets. A collection where each object has a finite description is countable. So the set of FAs or strings or integers or TMs etc. is countable. However,

> ***Cantor's Theorem.*** The set of languages is uncountable.

Thus, almost every language is not r.e.

Reductions

We omit the exact definition of reduction. But

Idea. If language A reduces to language B, then B is as least as hard as A.

In CS we usually reduce a problem to a problem that we know how to solve. Here we do the reverse and use the following consequence:

Fact. If language A is not recursive/r.e. and A reduces to B, then B is not recursive/r.e.

Decidable Problems = Recursive Languages

Recall that a yes/no problem P is ***decidable*** if there is an algorithm/TM for it that always halts and gives the correct answer.

If one writes down all the instances where the answer is yes, one gets a language, say Y_P. It follows that:

Connection. Problem P is decidable if and only if language Y_P is recursive.

Halting Problem

We claimed earlier that most problems about TMs are undecidable. We can now justify those claims. For example:

Theorem. The halting problem is undecidable

The idea is that, given any TM, one can re-program it so that it never halts-and-rejects but instead goes off into an infinite loop. So answering the acceptance problem reduces to answering the halting problem.