Lecture 33

Finishing up from last time

Thm. The following problems about Turing machine are undecidable:

1) Given a Turing machine M and an input string w, does M halt on input w?
2) For a certain fixed machine, given an input string w, does M halt on input w?
3) Given a Turing machine M, does M halt on the empty tape?
4) Given a Turing machine M, is there any string at all on which M halts?
5) Given a Turing machine M, does M halt on every input string?
6) Given 2 machines M_1 and M_2, do they halt on the same input strings?
7) Given a Turing machine M, is the language M accepts regular? context free? decidable?

So what can you do?

From the homework

5.4.1

(a) Let $M = (K, \Sigma, \delta, s, H)$ be a Turing machine.

Let $q = |K|$ i.e., the total number of possible states.

Let $c = |\Sigma|$ i.e., the total number of tape symbols, including the blank symbol and >

Now consider the total number of configurations that are possible using k tape squares:

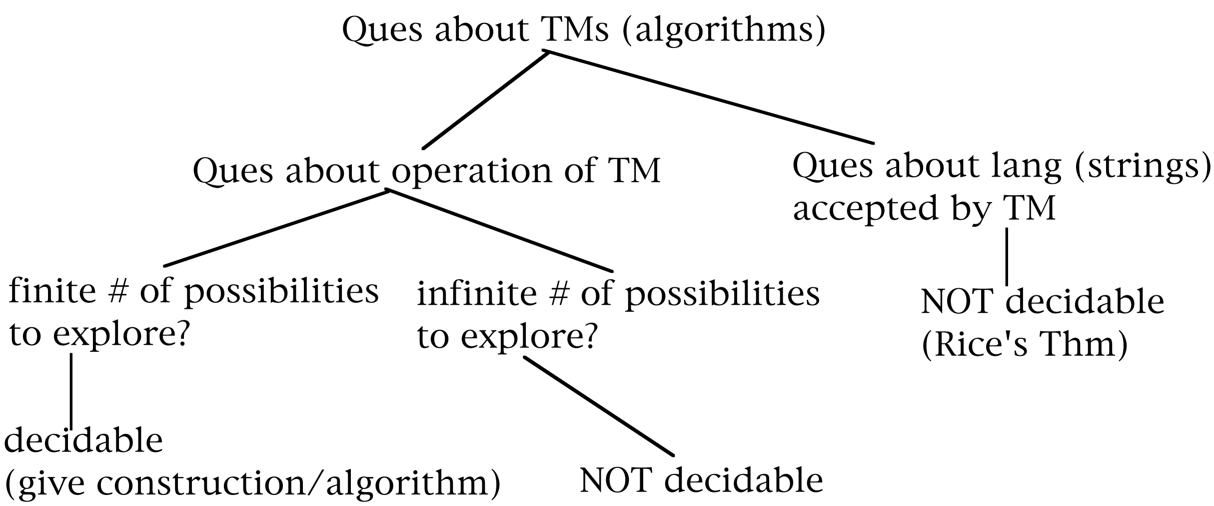
$t \leq (q)(c^k)(k)$, where the final k in the multiplication refers to the possible head positions. ($\leq$ due to the fact that > should never appear in more than the leftmost tape square.)

To check whether the Turing machine M uses k tape squares on a given input w:

- simulate M on w for at most t steps.
- if M has used k tape squares in that time, answer **Y**;
- if not, answer **N** [M must be in a loop and will continue to cycle without reaching k tape squares]

How can you decide whether a problem is decidable or not? Here are some heuristics:

decidable vs undecidable problems:



<u>Rice's Theorem</u>. Any non-trivial property of recursively enumerable languages is undecidable.

ex. emptiness
finiteness
regularity
context free-dom

Def. a <u>trivial property</u> is one that is either true for all r.e. languages or false for all r.e. languages.