Thm: Every Turing-enumerable ~~mach~~ language is Turing-acceptable.

11/14/18

Pf: Let M be a TM that enumerates our language L. ~~A ma~~ We can construct a machine M′ that accepts L by having M′ ~~run M~~ simulate M and halting and saying yes when M "prints" ~~our~~ the input string. (Note – M′ will never halt if the input string is not in L)

So Turing-acceptable is the same as Turing-enumerable.

Thm: If L and $\overline{L}$ are both Turing-~~enumerable~~ acceptable, then L is Turing-decidable.

(equivalent, but easier to think about in terms of acceptable)

Pf: Since L & $\overline{L}$ are T-A, We have ~~m~~TMs M & $\overline{M}$ that accept L & $\overline{L}$ respectively.

Construct a TM ~~that~~ M' that runs M + $\overline{M}$ "in parallel". ~~Either our~~ ~~Inp~~ Our input string w is either in L or $\overline{L}$, so one of M + $\overline{M}$ will halt and say "yes" on w. When one of M + $\overline{M}$ halts, M' will halt and give the appropriate answer (depending on whether M or $\overline{M}$ halted w/ a yes answer).

<u>Thm</u>: There exists a language L that is Turing-acceptable but for which $\overline{L}$ is not Turing-acceptable.

<u>Pf</u>: Let our alphabet be $\Sigma = \{a\}$. We had a way of encoding all our TMs as strings. We have an ordering of all strings, so we can get from this a list of all TMs (i.e. the set of

valid encodings of TMs is Turing-enumerable and hence countable)

Let

$$L = \{ a^i \mid a^i \text{ is accepted by the } i\text{-th TM} \}$$

L is T-A - we can build a TM M that accepts L by having M take as input a^i, compute what the i-th TM is (by enumerating all TMs and stopping when it gets to the i-th one). Then it uses the universal TM to run the i-th TM on a^i, and halts if this computation halts.

However, $\overline{L}$ is not T-A, because $\overline{L}$ is not accepted by any of the machines on our list of TMs. The argument is as follows. ~~So~~ $\overline{L}$ is not accepted by the i-th TM because, if $a^i \in \overline{L}$, then $a^i \notin L$, so a^i is <u>not</u> accepted by the i-th TM by our def'n of L.

Visually:

a^s

TM # 0, 1, 2, 3 — table of N_s + Y_s saying whether TM # i accepts a^j.

We construct a sequence not on our list by reversing the diagonal

If our table looks like

a's

TM #	0	1	2	3			
0	Y	N	Y	N	Y	...	
1	N	N	N	Y	N	...	
2	Y	N	N	Y	Y	N	...
3	Y	Y	Y	N	Y	Y	...

our sequence is the opposite of this one

Cor: L is T-A but not T-D.

Pf: ~~?~~ If L was T-D, then ~~?~~ $\overline{L}$ would be T-D ~~a~~ also, but we know $\overline{L}$ is not T-A and hence not T-D.

Very brief intro to "recursion theory"

Imagine we have a magical device (usually called an "oracle") that decides ~~$\overline{L}$~~ L (or $\overline{L}$).

We can talk about what languages TMs can accept.

We can talk about what languages TMs w/ L-oracles can accept.

Construct a language L′ that behaves for L-oracular TMs just like L does for normal TMs.

The same argument shows L′ is not L-oracular-TM-decidable.