A specific language that is T-A but not T-D:

11/26/18

Let H ("the halting language") be the language

$$H = \{ w_M \square w_I \mid w_M \text{ is our encoding for TM } M, \ w_I \text{ is our encoding for input } I, \ M \text{ halts on } I \}$$

Thm1: H is T-A

Pf: We can have a TM U′ (it's a modification of the universal TM) that first checks if we have valid input (i.e. the ~~string~~ input actually encodes a TM and its input) and then runs the universal TM on the input, saying "yes" if the simulation halts.

U′ is an acceptor for H.

Thm 2: H is not T-D.

Pf 1: Suppose H is T-D for contradiction.

~~Then, given an acceptor M for L, we can construct a decider M' for L by having~~

Let L be a T-A language, so that M is an acceptor for L. Under our assumption that H is T-D, we could construct a decider M' for L as follows:

Given a string w, M'

1. feed encoding for M + encoding for w into our decider for H
2. If our decider for H says "yes," feed w into M and return the answer that M gives.
3. If our decider for H says "no," return "no."

M' always halts and gives the right answer, so M' is a decider for L.

Therefore, under our assumption that H is T-D, every T-A language L would also be T-D. But we know there is a T-A language that is not T-D, so our assumption can't be true and H is not T-D.

Pf 2: (More direct but longer contradiction proof based on the one I gave on first day)

Suppose H is T-D, so there is a ~~machine~~ TM M that decides H.

To be precise M

1. ~~always~~ halts on every input
2. ends in a state q_y if the answer is yes and q_n if the answer is no.

If M exists, then we can construct a machine M' based on M that

When M halts at q_y, M' will instead go into an infinite loop:

(i.e. every transition

$$(q_y, \ell) \longrightarrow \mathcal{H}$$

is replaced by a transition

$$(q_y, \ell) \longrightarrow (q_e, \square, R)$$

and we add a transition

$$(q_e, a) \longrightarrow (q_e, \square, R)$$

for every letter a in our tape alphabet.)

(and everything else in M is unchanged)

Given M', we can construct $\hat{M}$ as follows:

~~Given~~

① $\hat{M}$ first takes its input w then converts w into w followed by the encoding for w.

② $\hat{M}$ then feeds the result into M', (so M' gets as input w (assumed to be the encoding of a TM) and ~~input~~ and the encoding of w (as an input string))

book forgot about this

Now feed the encoding of $\hat{M}$ to $\hat{M}$.

If $\hat{M}$ halts on this input, then when we do ① + ②, $\hat{M}$ feeds

$\hat{M}$ as the machine (encoded)
encoding of $\hat{M}$ as the input (encoded)

to M', so M^* would say "yes", so M' goes into an infinite loop, so $\hat{M}$

doesn't actually halt.

Hence we know $\hat{M}$ can't halt on this input.

On the other hand, if $\hat{M}$ doesn't halt on this input, then $\hat{M}$ feeds

$$\left\{ \begin{array}{l} \hat{M} \text{ as machine} \\ \text{encoding of } \hat{M} \text{ as input} \end{array} \right\}$$

to M', so M would say "no", so M' halts, and so does $\hat{M}$.

Hence $\hat{M}$ can't not halt on this input either.

This is a contradiction, so $\hat{M}$ can't exist, so M' and M also can't exist. $\square$