Example: Blank-tape halting problem:

Given a (suitably-encoded) TM as input, determine if that TM halts when started with a blank-tape.

Prop: This problem is undecidable.

Pf: Suppose there was a TM B that solves this problem. We will use B to build a TM H that solves the original halting problem.

H takes 2 inputs, an encoding of a TM M and an encoding of a string w.

H will take the 2 inputs M + w, and ~~create a new~~ figure out the encoding of a new TM M_w that first writes w on the tape, then runs M.

Then H feeds M_w to B.

This H solves the Halting Problem. (assuming B solves the blank-tape halting problem)

If M_w halts on a blank tape as input, M halts on w, and if M_w doesn't halt on a blank tape as input, M doesn't halt on w.

~~So H answers the Halting~~

So the output of B answers the Halting Problem, so H answers the Halting Problem.

So B can't exist.

<u>Terminology</u>: We proved the blank-tape HP is undecidable by reducing the normal HP to the blank-tape HP.

Problem:

Input: 1) A TM M

2) An instantaneous desc x for M

3) " y " M

Output: "Yes", if, when you run M starting at the instant desc x, you end up at y at some point, and "No" otherwise.

Call this problem $\mathcal{P}$.

Prop: $\mathcal{P}$ is undecidable.

Pf: Suppose P is a TM that solves $\mathcal{P}$. We Create a TM H using P to solve the HP as follows:

H takes in as input ~~M~~ a TM M and a string w, H creates a new (encoding of a) TM M' that differs from M as follows

1) M' writes $\hat{\square}$'s instead of writing $\square$'s (and, when it reads, treats $\hat{\square}$'s + $\square$'s the same)

(This is so that we know when we get to the edge of the portion of tape that M has touched)

2) M', instead of halting, always completely blanks the tape, and then goes into a special state ~~$q_{...}$~~ q_i (for some i we can figure out on the fly, but it should be different from all the states of M)

H gives M', (λ, q_0, w), (λ, q_i, λ) as input to P. Then P's answer will be the answer to the HP on

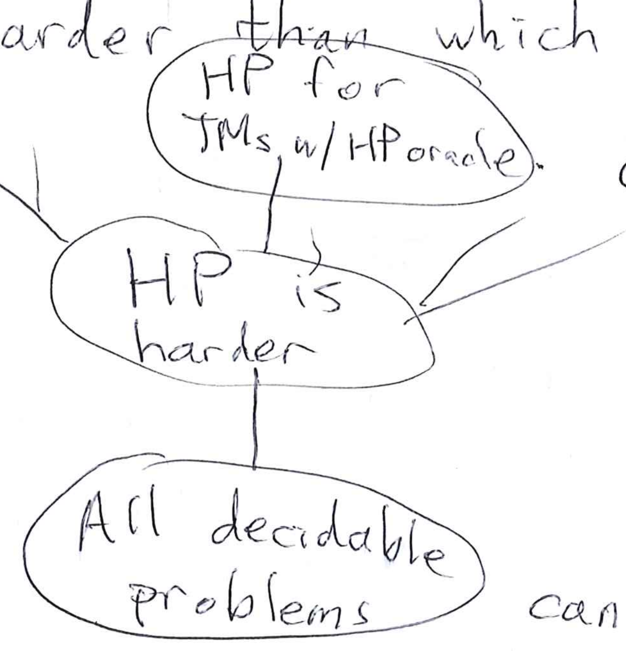
M of w.

So, if P exists, then the HP is decidable, so P doesn't exist and P is undecidable.

<u>Very brief intro to recursion theory:</u>

We have "problems" - and we have "oracles" for the problem - hypothetical functions that solve these (usually undecidable) problems.

<u>Main question:</u> Which problems are strictly harder ~~than~~ which other problems?



all kinds of other harder problems also - ~~[illegible]~~ can't be described in nice way.