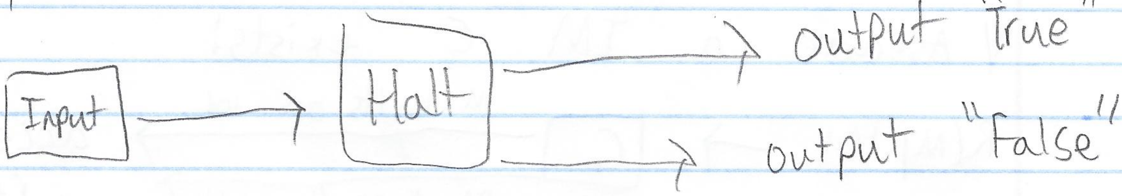
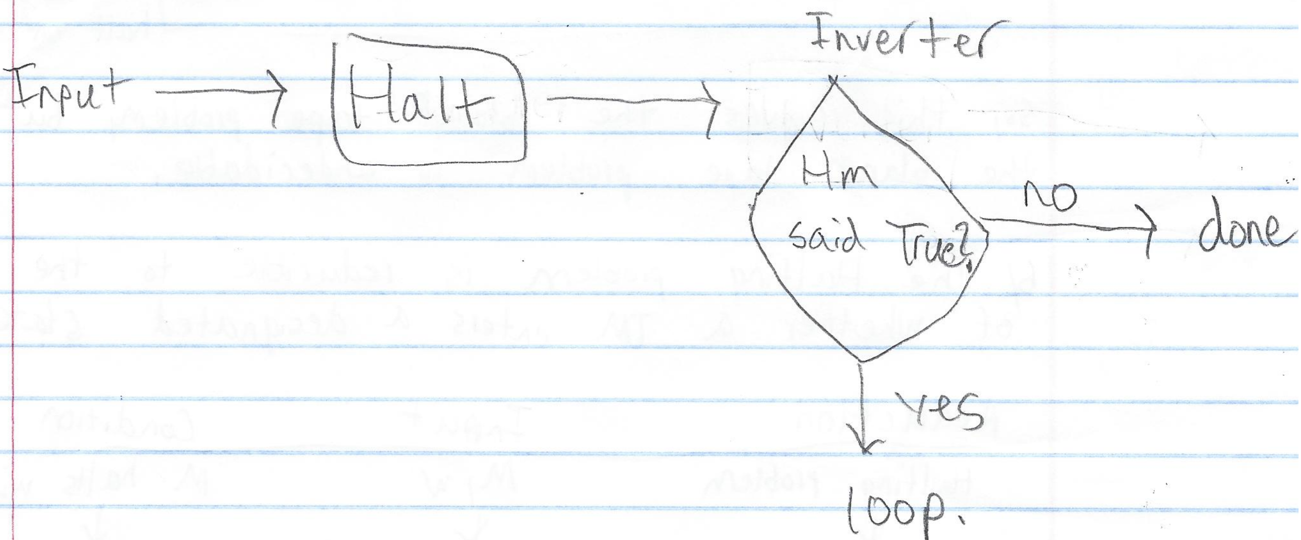
Chapter 12

1)



Assume we have a TM that is capable of solving the halting problem (Halt). If a tm input that will halt is fed into halt, then it outputs True and halts, otherwise it outputs false and halts.

Now, we construct a second TM which inverts the answer of Halt machine.

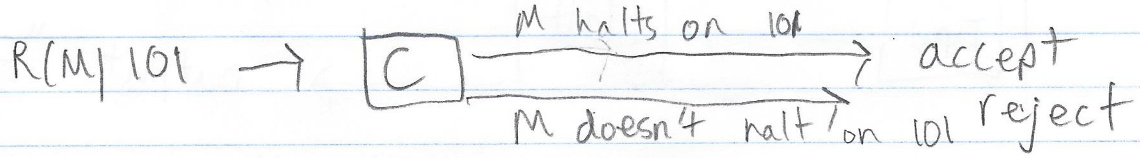


So, If the input does halt, it goes through the inverter and loops forever (doesn't halt).
If the input doesn't halt, it goes through the inverter and halts.

So, Halt gave the wrong answer for both cases.
Therefore, it is possible to construct a program that will

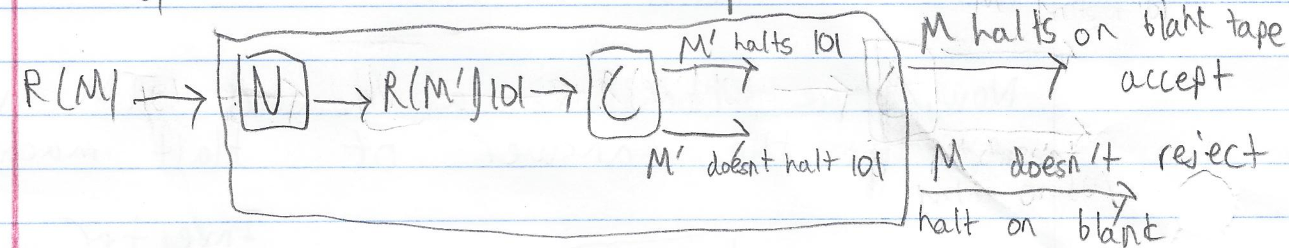
fail for any TM that claims to solve the halting problem.

4/ Assume a TM C exists:



Now consider a TM N, which produces a representation of a Machine: M':

a) erase any input
b) Runs M on blank Tape.



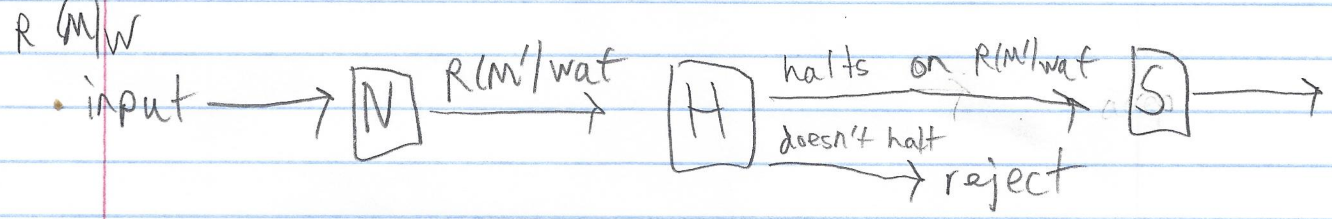
So, this solves the blank tape problem, but the blank tape problem is undecidable.

6/ The Halting problem is reducible to the question of whether a TM enters a designated state.

Reduction	Input	Condition
Halting problem	M, w	M halts with w
↓	↓	↓
State problem	M', w, qf	M' enters state qf when run with w

The string w is the same for both problems. M' is constructed from M by adding an additional state qf. M' enters qf whenever M halts. So, M halting is equivalent to M' entering qf.

Now Assume there is a machine S that solves the state problem. S accepts the input of M, w, q_i



10)

12) a) The language L consisting solely of the string w is recursive, since every finite language is recursive. The complement of L is also recursive and clearly does not contain w. Consequently the property of containing the string W is a nontrivial property of recursively enumeable languages. Thus, the question of w being in L(M) is undecidable.