CS 208

HW4 - Q1

SAKSHAM RATHI

22B1003

1) (a) $L_1 = \{n \in \mathbb{N} \mid \exists m \in \mathbb{N}$ s.t. M_n halts on $w_m\}$

So, L_1 is basically the set of all turing machines M_n which halt on atleast one input string ($H(M) \neq \phi$)

Claim: L_1 is undecidable

Proof: It is known that the Halting Problem is undecidable. So, if we reduce halting problem to L_1, then we can show that L_1 is undecidable.

(Halting problem = set of pairs (M, w) such that w is in $H(M)$ i.e. M halts on w.)

We will describe an algorithm that transforms (M, w) into an output M', the code for another turing machine, such that w is in $H(M)$ iff $H(M') \neq \phi$.

We can make M' ignore its input and instead simulate M on input w. If M halts, then M' accepts its own input.

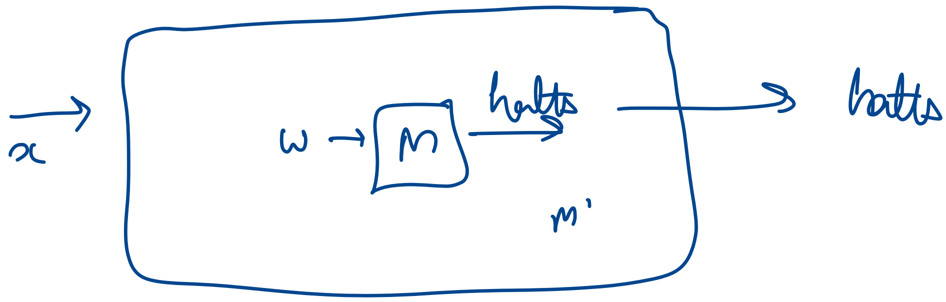
As we can see, if M does not halt on w then M' accepts none of its inputs i.e. $H(M') = \phi$. However, if M halts on w then M' accepts every input and thus $H(M') \neq \phi$

{By ignoring its input x, we mean M' replaces (x) by (M, w), this can be accomplished by some extra q_n states where n = length of the pair (M, w)}

Now using these additional states, M' simulates the turing machine for the halting problem.

Therefore, we have reduced the halting problem to L_1

Now, since halting problem is not recursive, and L_1 is as hard as halting problem, we can deduce that L_1 is undecidable.



<u>Proof of halting problem being undecidable:</u>

Let us assume that we have a turing machine which moves to an accepting state if M halts on w. Call this machine X.

$X(M, w)$

- Yes: if M halts on w
- No: if M does not halt on w

Consider Y such that:

$Y(M)$

- if $X(M, M) = \text{Yes}$ → loop forever
- if $X(M, M) = \text{No}$ → halt

$Y(Y)$

$X(Y, Y) = \text{Yes} \rightarrow$ Loop forever

$X(Y, Y) = \text{No} \rightarrow$ Halt

It will never halt because of the above condition

Therefore, the halting problem is undecidable.

* Another smaller proof for this will be using Rice theorem.

We consider the property that $H(M) \neq \phi$. (Clearly a non-trivial property, because we can create machines that never halt and also machines which always halt.)

Now, using Rice theorem, we can deduce that this property is undecidable so is our language L_1.