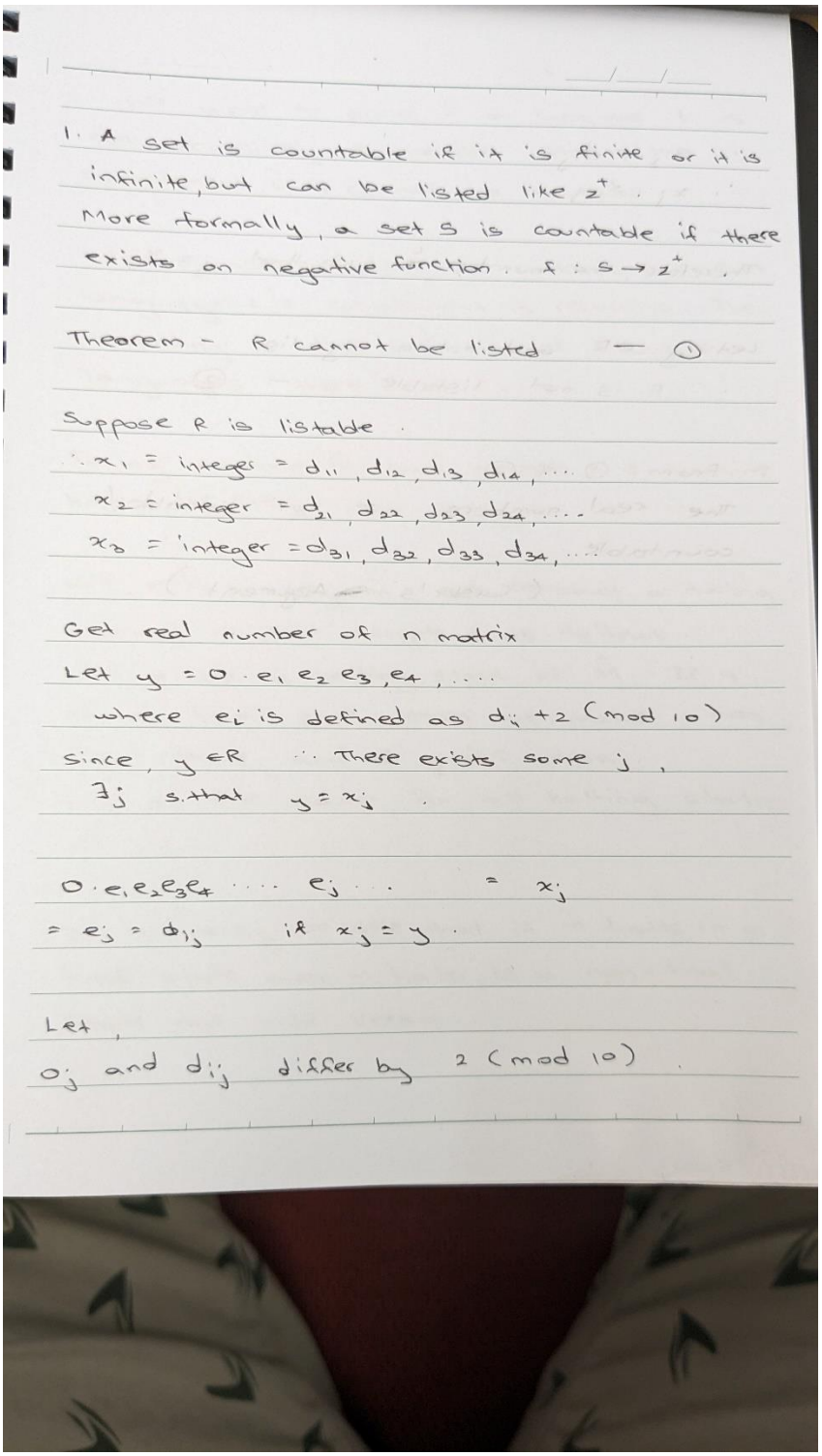
1. Prove that the set of all real numbers is not countable.



1. A set is countable if it is finite or it is infinite, but can be listed like z^+.
More formally, a set S is countable if there exists an negative function. $f : S \to z^+$.

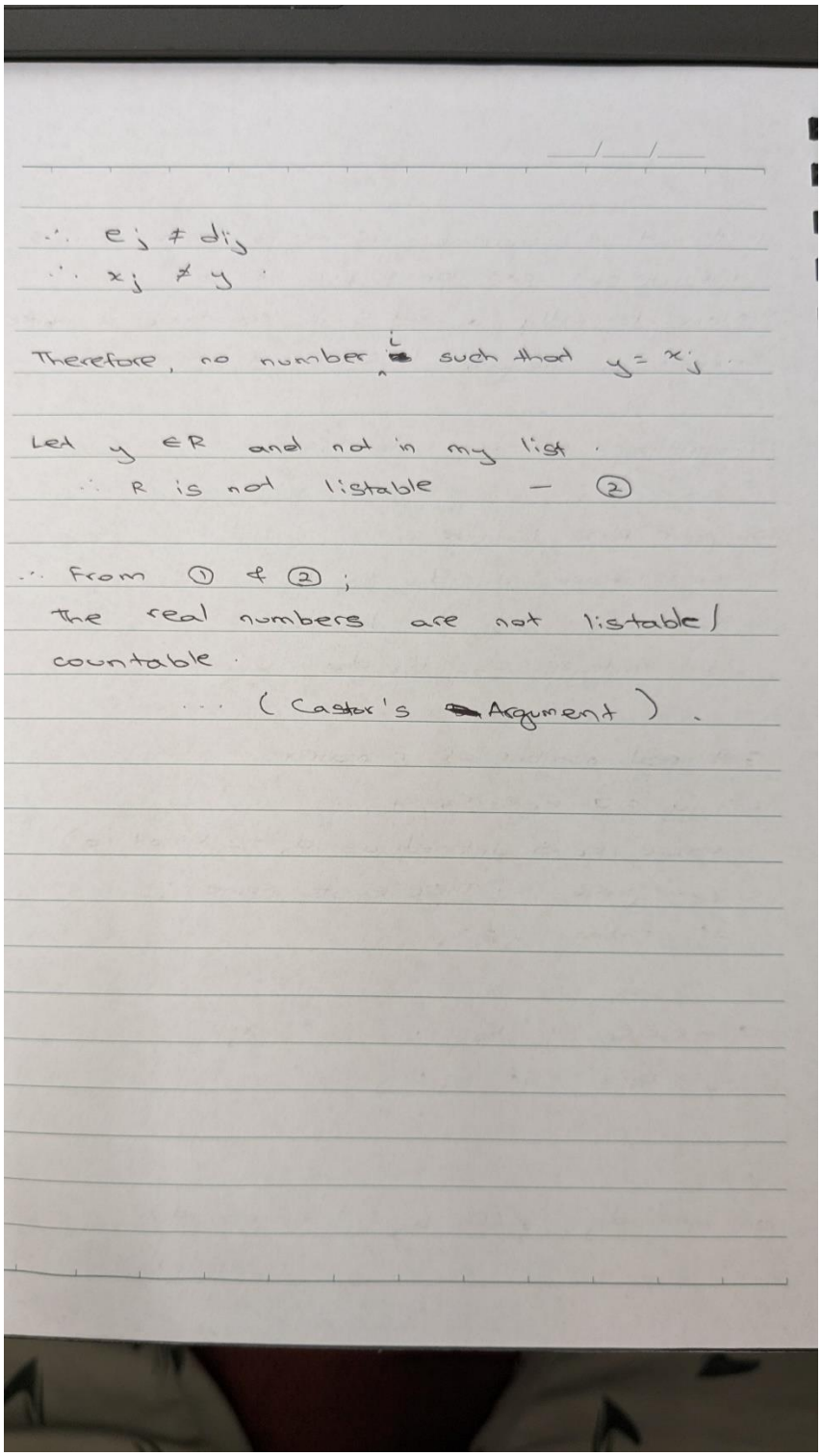
Theorem - R cannot be listed — (1)

Suppose R is listable.
$\therefore x_1$ = integer = $d_{11}, d_{12}, d_{13}, d_{14}, \ldots$
x_2 = integer = $d_{21}, d_{22}, d_{23}, d_{24}, \ldots$
x_3 = integer = $d_{31}, d_{32}, d_{33}, d_{34}, \ldots$

Get real number of n matrix
Let $y = 0 \cdot e_1 e_2 e_3 e_4 \ldots$
where e_i is defined as $d_i + 2 \pmod{10}$
Since, $y \in R$ ∴ There exists some j,
$\exists j$ s.t that $y = x_j$.

$0 \cdot e_1 e_2 e_3 e_4 \ldots e_j \ldots = x_j$
$= e_j = d_{jj}$ if $x_j = y$.

Let,
o_j and d_{jj} differ by 2 (mod 10).



$\therefore e_j \neq d_{ij}$

$\therefore x_j \neq y$.

Therefore, no number i such that $y = x_j$.

Let $y \in R$ and not in my list.

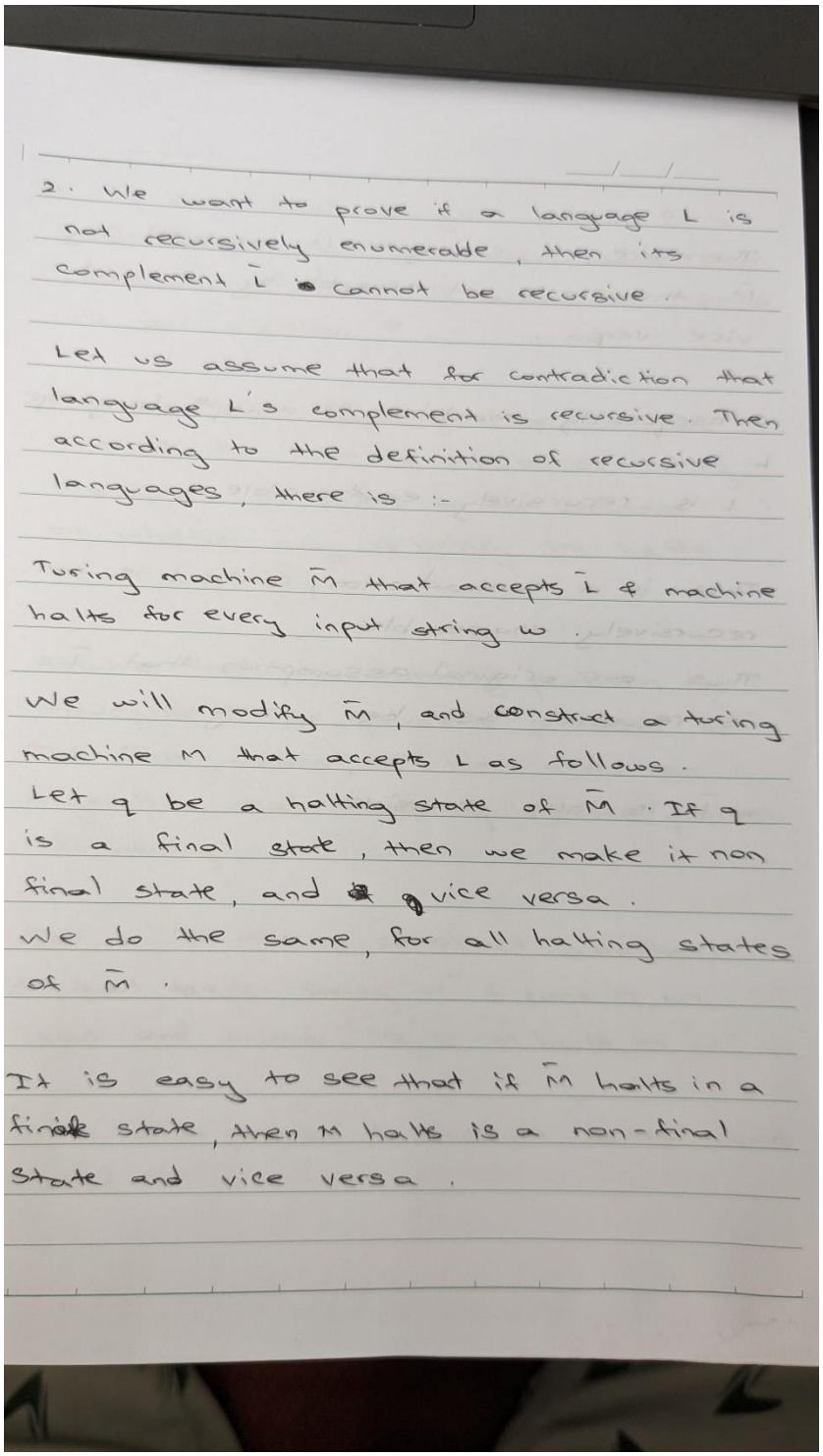
$\therefore$ R is not listable — (2)

$\therefore$ From (1) & (2);

the real numbers are not listable/ countable.

... (Cantor's Argument).

2. Show that if a language is not recursively enumerable, its complement cannot be recursive.



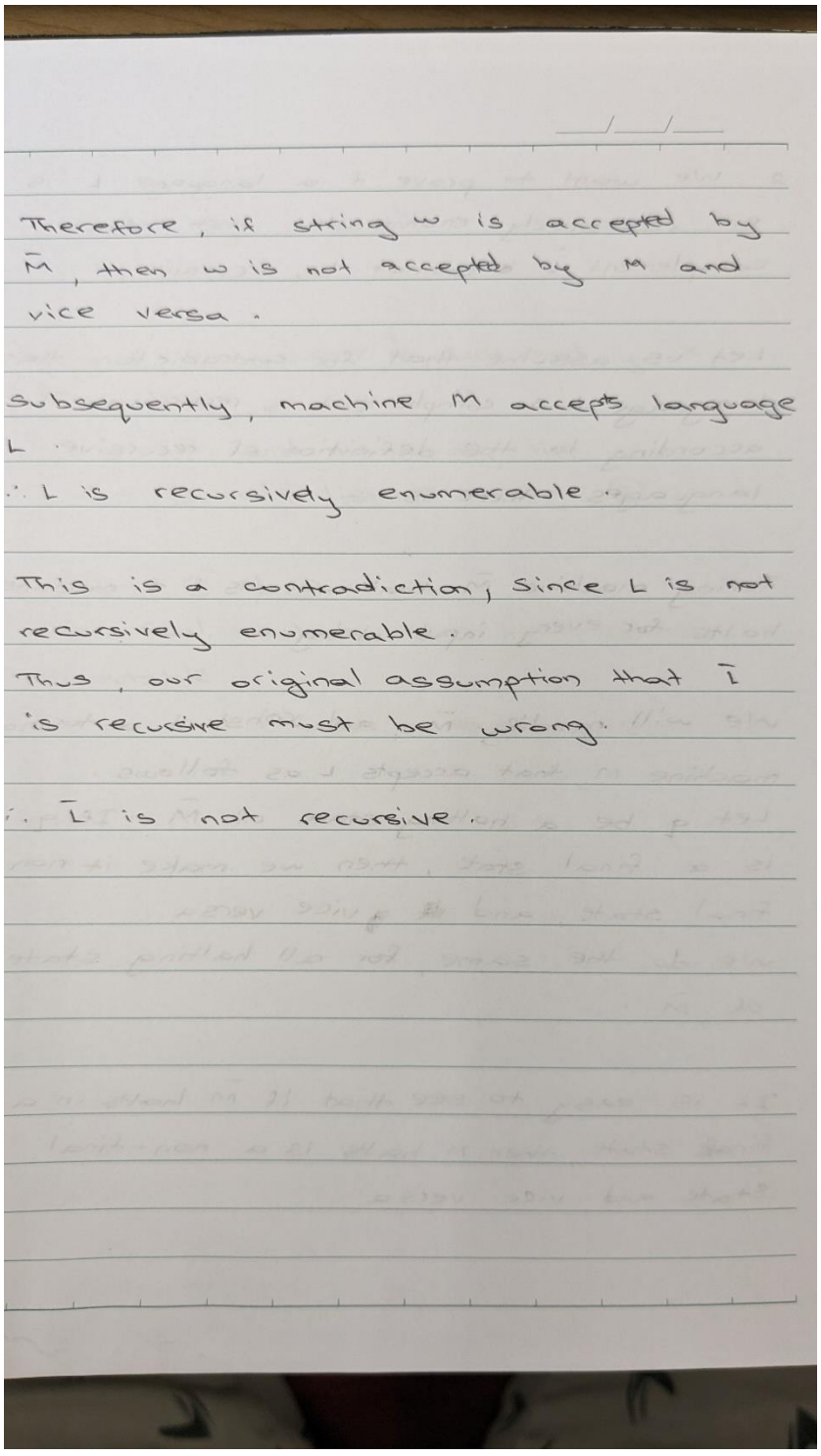
2. We want to prove if a language L is not recursively enumerable, then its complement $\bar{L}$ cannot be recursive.

Let us assume that for contradiction that language L's complement is recursive. Then according to the definition of recursive languages, there is :-

Turing machine $\bar{M}$ that accepts $\bar{L}$ & machine halts for every input string w.

We will modify $\bar{M}$, and construct a turing machine M that accepts L as follows.
Let q be a halting state of $\bar{M}$. If q is a final state, then we make it non final state, and vice versa.
We do the same, for all halting states of $\bar{M}$.

It is easy to see that if $\bar{M}$ halts in a final state, then M halts is a non-final state and vice versa.



Therefore, if string w is accepted by $\bar{M}$, then w is not accepted by M and vice versa.

Subsequently, machine M accepts language L.

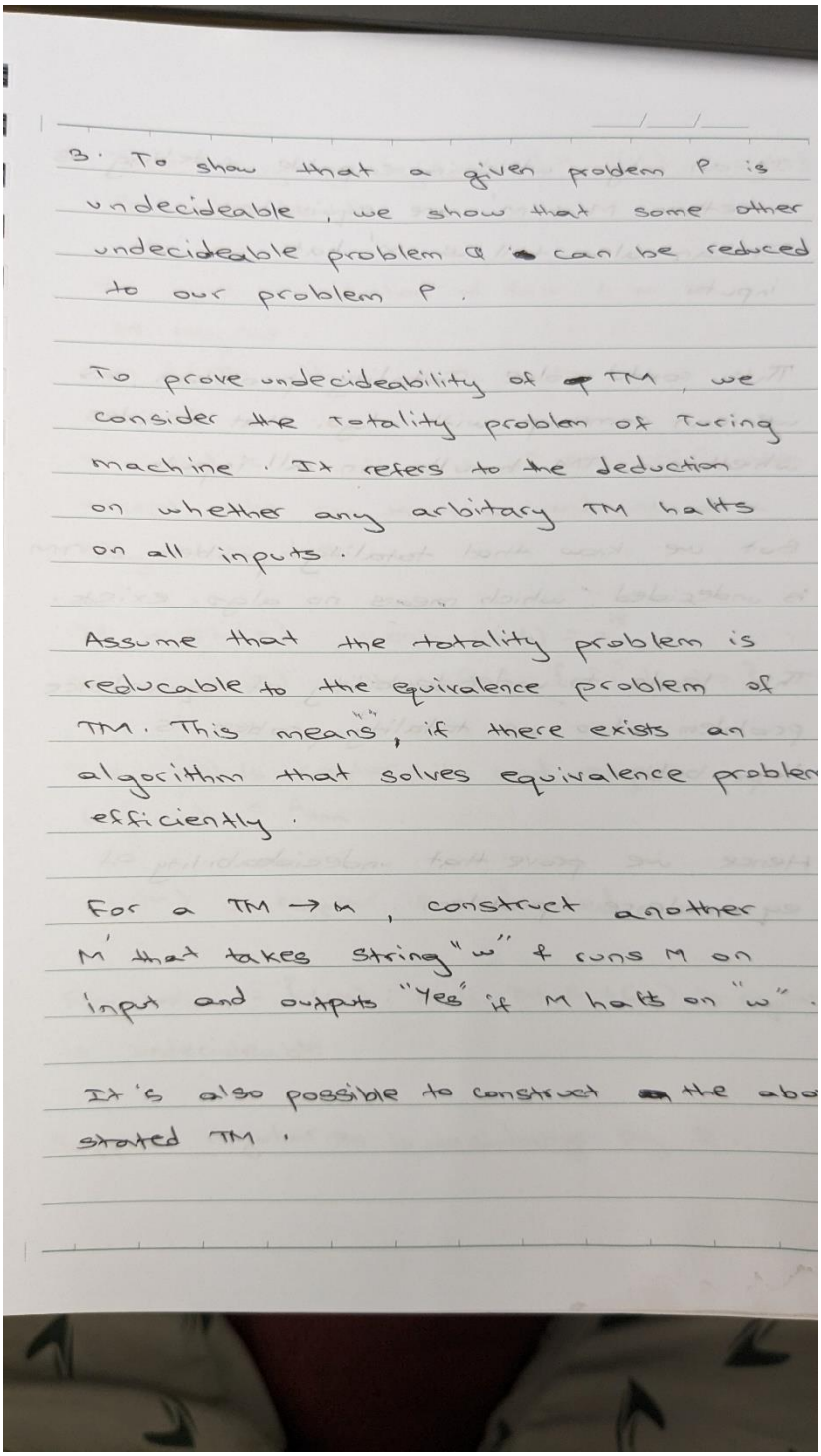
∴ L is recursively enumerable.

This is a contradiction, since L is not recursively enumerable.

Thus, our original assumption that $\bar{L}$ is recursive must be wrong.

∴ $\bar{L}$ is not recursive.

3. Prove by reduction that the problem of determining whether any two given Turing machines accept the same language is undecidable.



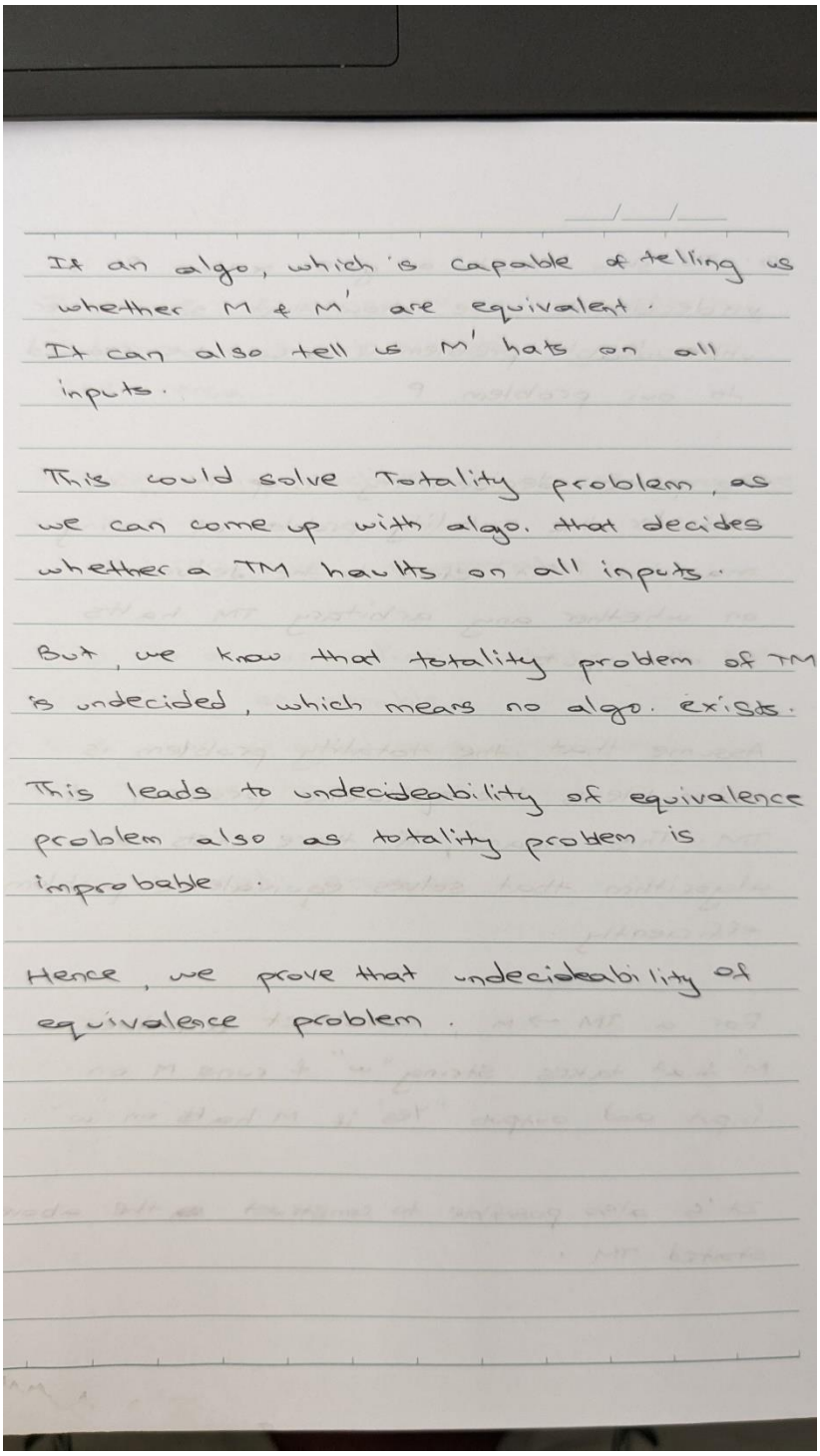
3. To show that a given problem P is undecideable, we show that some other undecideable problem Q can be reduced to our problem P.

To prove undecideability of TM, we consider the totality problem of Turing machine. It refers to the deduction on whether any arbitary TM halts on all inputs.

Assume that the totality problem is reducable to the equivalence problem of TM. This means, if there exists an algorithm that solves equivalence problem efficiently.

For a TM → M, construct another M' that takes string "w" & runs M on input and outputs "Yes" if M halts on "w".

It's also possible to construct the above stated TM.



__/__/__

If an algo, which is capable of telling us whether M & M' are equivalent.
It can also tell us M' halts on all inputs.

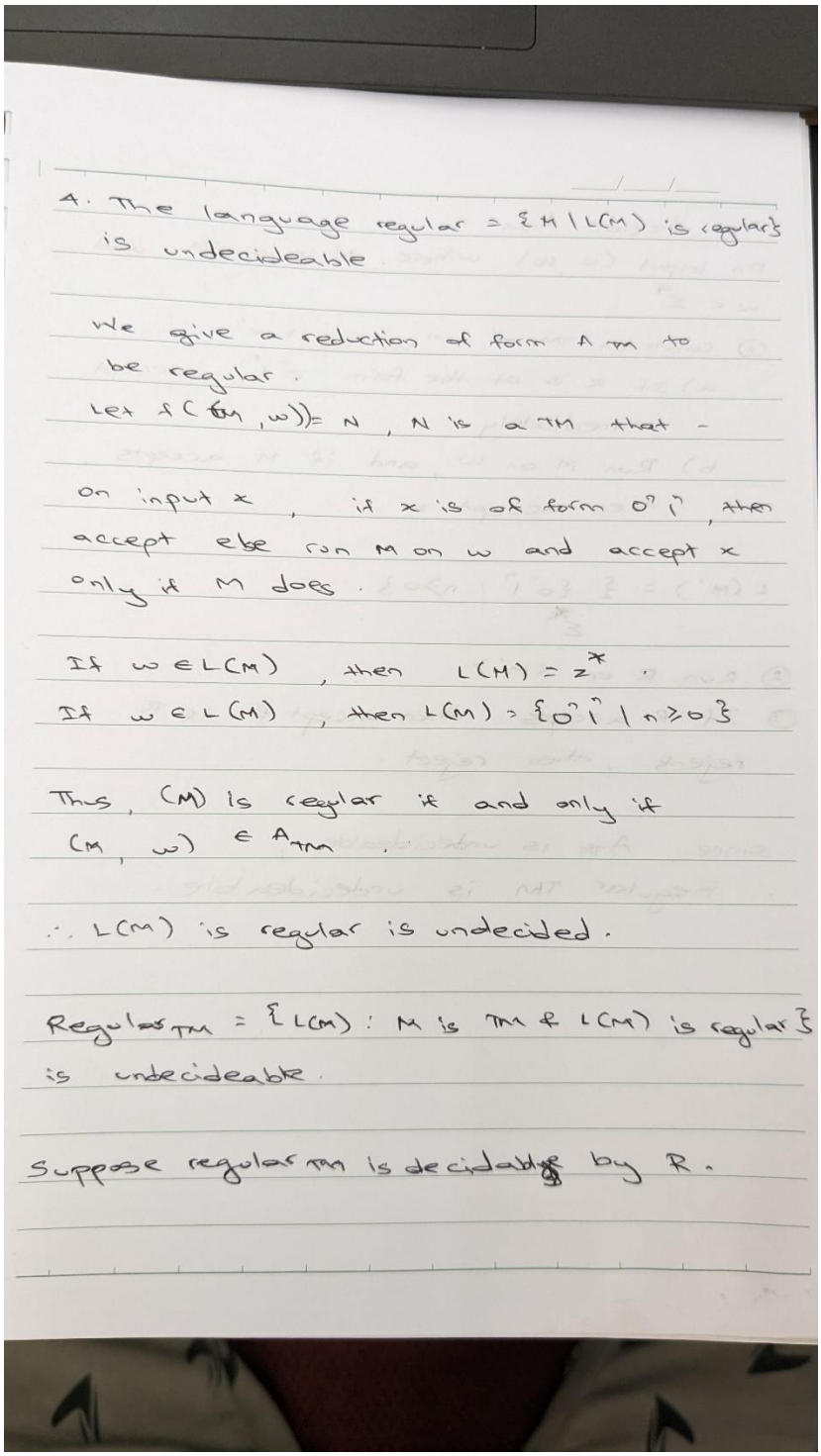
This could solve Totality problem, as we can come up with algo. that decides whether a TM halts on all inputs.

But, we know that totality problem of TM is undecided, which means no algo. exists.

This leads to undecideability of equivalence problem also as totality problem is improbable.

Hence, we prove that undecideability of equivalence problem.

4. Prove by reduction (not by Rice's Theorem) that the problem "L(M) is regular" is undecidable.



4. The language regular = $\{M \mid L(M)$ is regular$\}$ is undecidable

We give a reduction of form A_{TM} to be regular.

Let $f(\langle M, w\rangle) = N$, N is a TM that -

On input x, if x is of form $0^n 1^n$, then accept else run M on w and accept x only if M does.

If $w \in L(M)$, then $L(M) = \Sigma^*$.

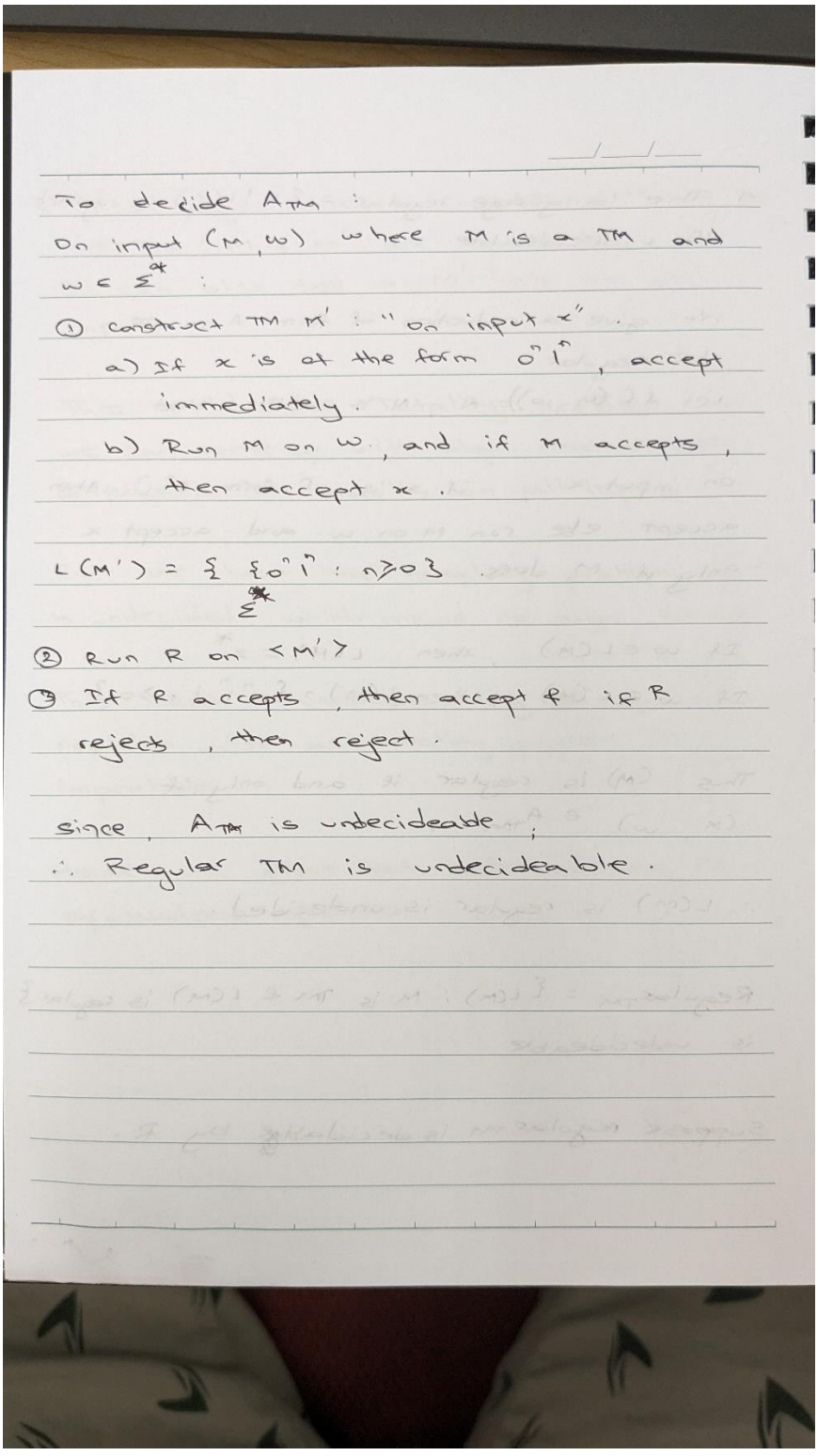
If $w \in L(M)$, then $L(M) = \{0^n 1^n \mid n \geq 0\}$

Thus, (M) is regular if and only if $(M, w) \in A_{TM}$.

$\therefore$ L(M) is regular is undecided.

$Regular_{TM} = \{L(M) : M$ is TM & L(M) is regular$\}$ is undecideable.

Suppose regular TM is decidable by R.



To decide A_{TM} :

On input (M, w) where M is a TM and $w \in \Sigma^*$:

① construct TM M' : "on input x"

a) If x is of the form 0^n1^n, accept immediately.

b) Run M on w, and if M accepts, then accept x.

$L(M') = \{0^n1^n : n \geq 0\}$

Σ^*

② Run R on $\langle M' \rangle$

③ If R accepts, then accept & if R rejects, then reject.

since, A_{TM} is undecideable ;

∴ Regular TM is undecideable.