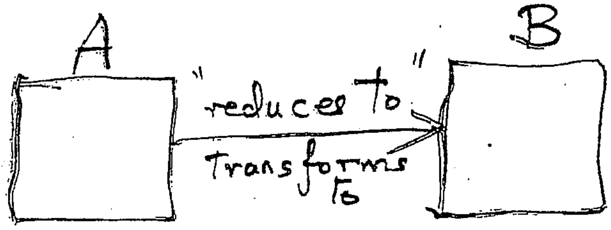
REDUCIBILITY

- Technique to show problems are not solvable.
  // "relating" the new problem to another problem
  that we know is (not) solvable



eg $A_{TM} \xrightarrow{\text{reduce}} HALT_{TM}$

- a soln to B can be used as soln to A
  (//show how)

- solving A cannot be harder than solving B

∴ If B is decidable, A is decidable

&

∴ || If A is undecidable, then B is also undecidable.
   (eg A_{TM})

[contrapositive]

KEY/MAIN BURDEN: show how soln to B [ie. decider of B]

can be used as a soln to A [ie. decider of A].

HALTING PROBLEM. // $A_{TM} = \{ \langle M, w \rangle \mid M \text{ accepts } w \}$

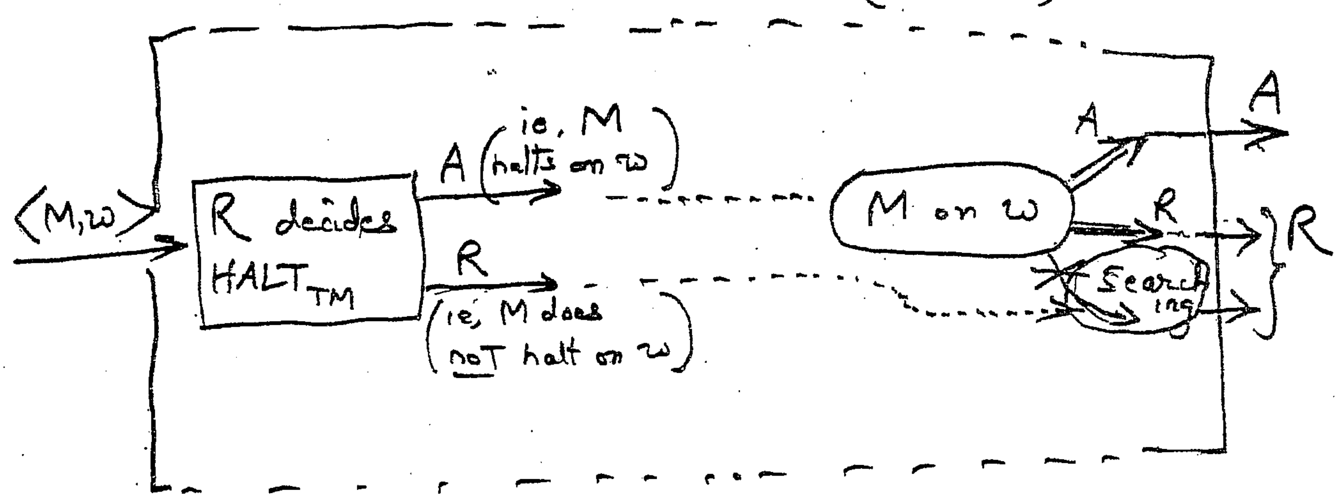
$HALT_{TM} = \{ \langle M, w \rangle \mid M \text{ halts on } w \}$

<u>Th</u>: $HALT_{TM}$ is undecidable.

<u>Proof</u>: Reduce A_{TM} to $HALT_{TM}$.

(Contradiction:) If 'R' decided $HALT_{TM}$, then A_{TM} also be decidable.

SHOW how soln (decider) of $HALT_{TM}$ can be used as a soln (decider) to A_{TM}.



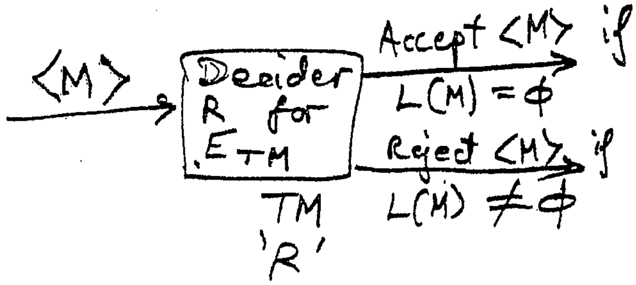
<u>'Decider'</u> for A_{TM}

<u>Proof</u>: (construct a TM S to decide A_{TM}, using TM R that decides $HALT_{TM}$

S: On input $\langle M, w \rangle$:

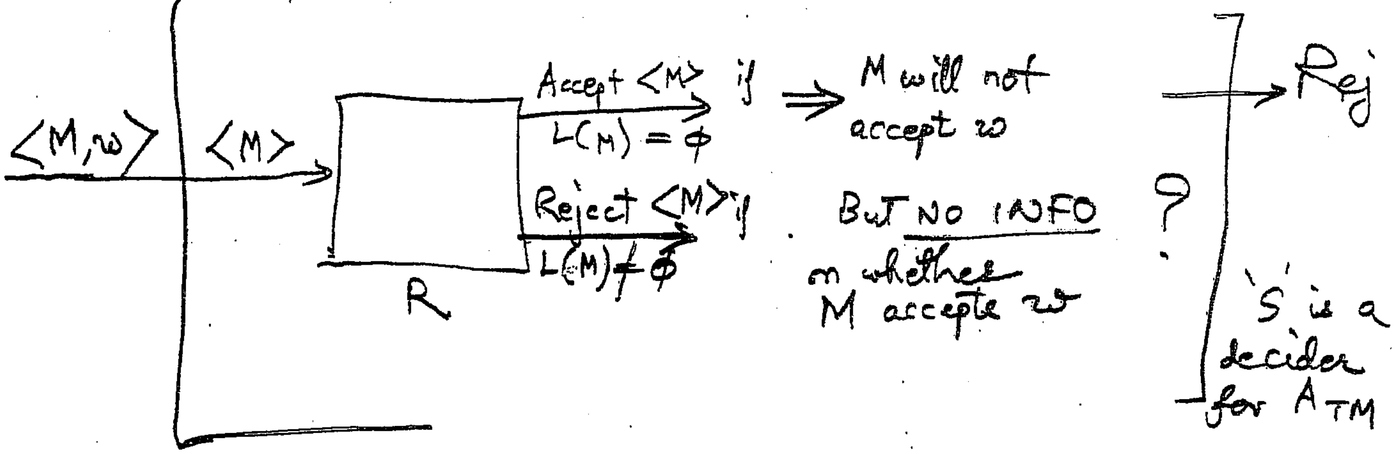
1) Run R on input $\langle M, w \rangle$
2) If R rejects $\langle M, w \rangle$, then REJECT
Else R accepts $\langle M, w \rangle$, then simulate M on w // must halt
— if M accepts w, then ACCEPT
REJECT

$E_{TM} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) = \phi\}$



Is E_{TM} decidable? i.e., is it possible to build such a TM as in box above?

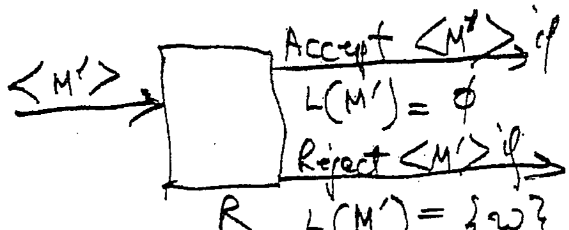
Initial attempt



Show how a decider TM S for A_{TM} can be built using a decider TM R for E_{TM}.

Hint: Modify M to M′ such that $L(M') = \begin{cases} \{w\} & \text{if M accepts } w \\ \phi & \text{if M rejects } w \text{ or keeps searching.} \end{cases}$

Now, if



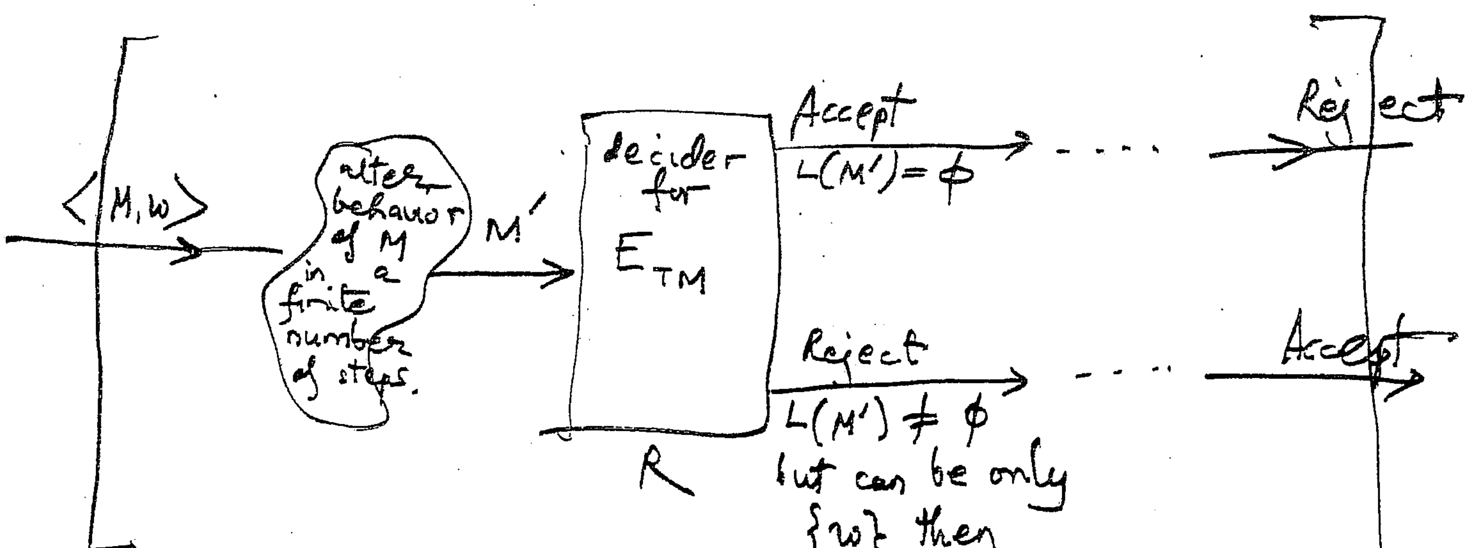
M' (built using input $\langle M, w\rangle$): On any input x:

1) If $x \neq w$, then reject // $x \notin L(M')$
2) Else $x = w$, then ~~run M~~ (M′ behaves as M would) on w and accept if M accepts w.

// $x = w$; Then $x \in L(M')$ if M accepts w

S: On input $\langle M, w\rangle$

1) Construct M' using description of M, and w
2) Run R, the decider for E_{TM}, on $\langle M' \rangle$
3) If R accepts then S rejects
   else R rejects then S accept

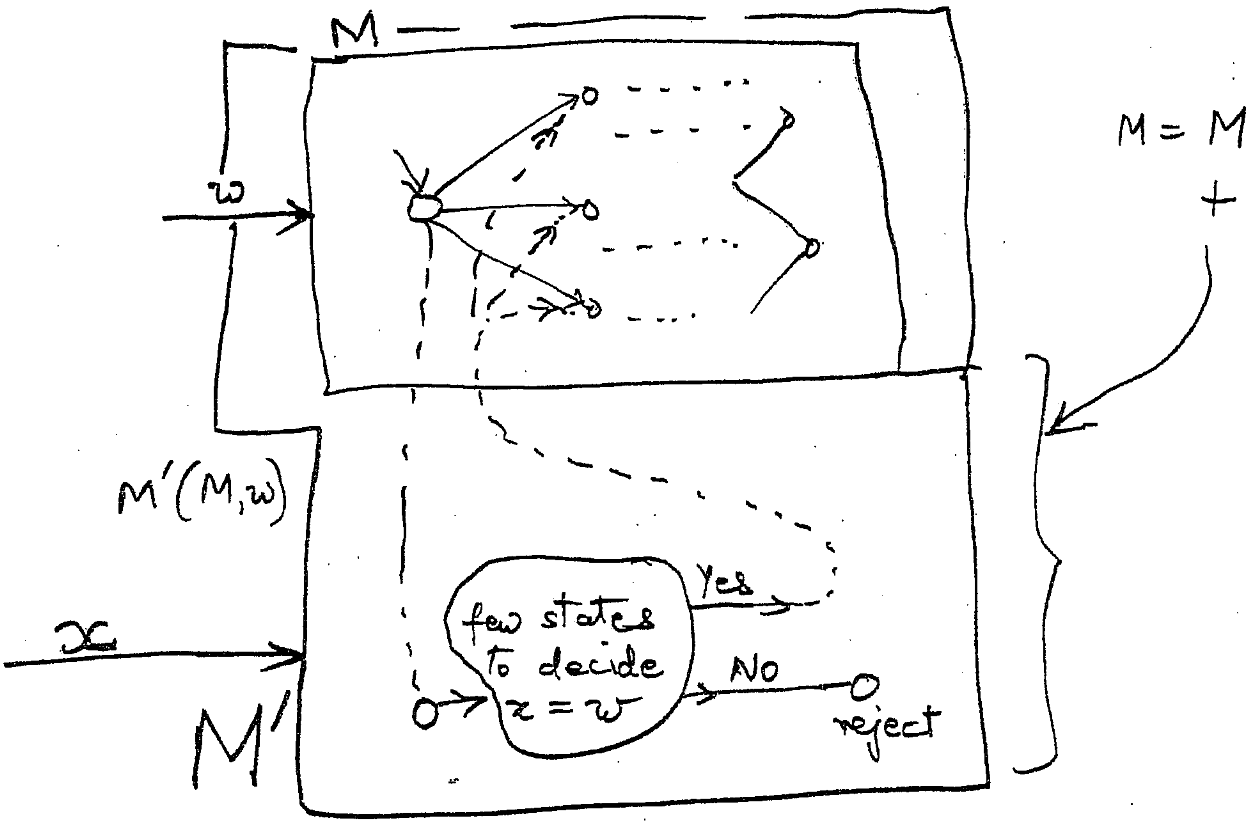


<u>Note</u>: M' need not be a decider!

M' is a recognizer of a single-string language $= \{w\}$

~~M′ behaves~~ like M when input x to M' is equal to w

$M = \langle Q, ---, \delta, -,- \rangle$



eg let $L(M) = \{w, w', w''\}$.
$L(M') = ?$

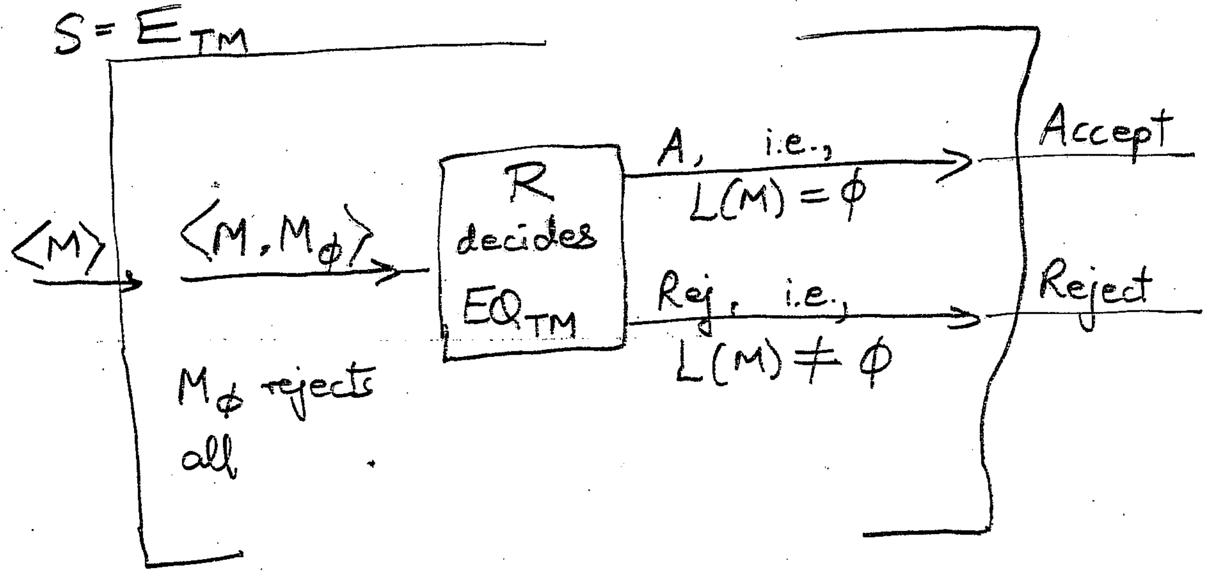
if $x = w'''$, w'', w } reject

$x = w$ } accept if M accepts w

Th: EQ_{TM} is undecidable

$EQ_{TM} = \{\langle M_1, M_2\rangle \mid M_1 \,\&\, M_2$ are TMs and $L(M_1) = L(M_2)\}$

Proof: Reduce E_{TM} to EQ_{TM}



If R decides EQ_{TM}, S decides E_{TM}

But as E_{TM} is undecidable, EQ_{TM} also undecidable.

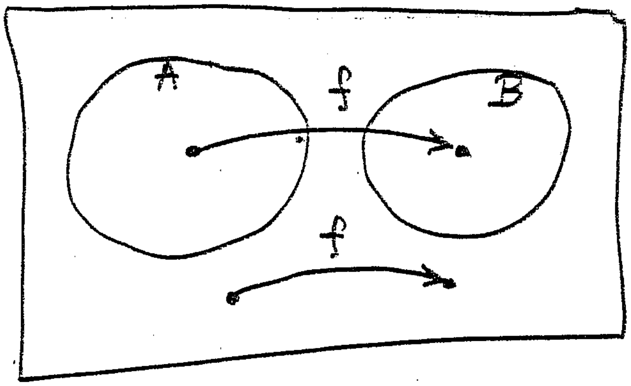
MAPPING REDUCABILITY

Apps: showing langs as not T-R, apps in complexity theory

__Def:__ Function $f : \Sigma^* \to \Sigma^*$ is a computable function if some TM, on every input w, halts with just $f(w)$ on tape.

__Def:__ Language A is mapping reducible to B, i.e., $A \leq_m B$, if there is a computable fn $f : \Sigma^* \to \Sigma^*$, where for every w,

$$w \in A \iff f(w) \in B$$



$w \in A ? \equiv f(w) \in B ?$

$A \leq_m B \equiv \overline{A} \leq_m \overline{B}$

__Th:__ If $A \leq_m B$ and B is decidable, then A is decidable

T-R T-R

__Th:__

__Cor:__ and A is undecidable, then B is undecidable

not T-R not T-R

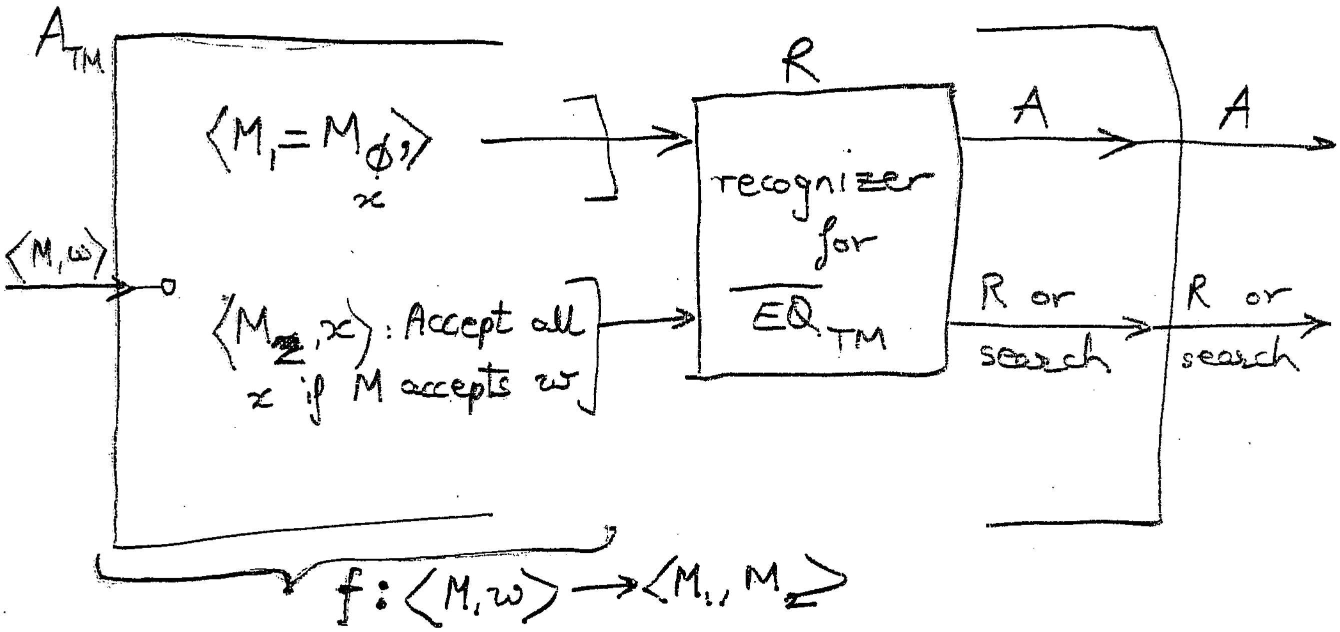
__Cor:__

Exercise: show mapping reducability $A_{TM} \leq_m HALT_{TM}$

$E_{TM} \leq_m EQ_{TM}$

Th: EQ_{TM} is not T-R nor co-T-R.

① $\overline{A}_{TM} \leq_m EQ_{TM} \equiv A_{TM} \leq_m \overline{EQ}_{TM}$



② $\overline{A}_{TM} \leq_m \overline{EQ}_{TM} \equiv A_{TM} \leq_m EQ_{TM}$