REDUCIBILITY

Technique to show problems are not solvable.

"relating" the new problem to another problem
that we know is (not) solvable

\[
\begin{array}{c}
A \\
\downarrow \text{reduces to} \quad \downarrow \text{transforms to} \\
\quad \quad B
\end{array}
\]

\[A_{TM} \xrightarrow{\text{reduce}} \text{HALT}_{TM}\]

1. a soln to B can be used as soln to A
   (\[1/1\text{show how}\])
2. solving A cannot be harder than solving B
3. \(\text{if } B \text{ is decidable, } A \text{ is decidable}\)
4. \(\text{if } A \text{ is undecidable, then } B \text{ is also undecidable.}\)
5. \(\text{[contrapositive]}\)

\[
\text{**KEY MAIN BURDEN**: Show how soln to } B \text{ [i.e. } B \text{ [decider of } B]\]
\[
\text{can be used as a soln to } A \text{ [i.e. decider of } A]\]
HALTING PROBLEM. // $A_{TM} = \{ \langle M, w \rangle | M \text{ accepts } w \}$

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

Th: $HALT_{TM}$ is undecidable.

Proof: 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 $A_{TM}$.

---

[Diagram shown: Decision process involving $HALT_{TM}$ and $A_{TM}$, with arrows indicating flow and decision logic.]
$E_{TM} = \{<M> | M \text{ is a } TM \text{ and } L(M) = \emptyset \}$

\[\text{Decider } R \text{ for } E_{TM}\]
- Accept $<M>$ if $L(M) = \emptyset$
- Reject $<M>$ if $L(M) \neq \emptyset$

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 $L(M') = \begin{cases} \{w\} & \text{if } M \text{ accepts } w \\ \emptyset & \text{if } M \text{ rejects } w \text{ or keeps searching.} \end{cases}$
$S$: On input $<M, w>$

1) Construct $M'$ using description of $M$, and $w$

2) Run $R$, the decider for $E_{TM}$, on $<M'>$

3) If $R$ accepts then $S$ rejects
   else $R$ rejects, then $S$ accepts

---

$M'$ (built using input $<M, w>$): On any input $x$:

1) If $x \neq w$, then reject \(\text{if } x \notin L(M')\)

2) Else $x = w$, then run $M'$ on $w$ and accept if $M'$ accepts $w$.

\(\text{if } x = w; \text{ then } x \in L(M')\)

---

**Note:** $M'$ 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 $w$
Th: $EQ_{TM}$ is undecidable

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

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

$S = E_{TM}$

\[ \langle M \rangle \rightarrow \langle M, M_\phi \rangle \rightarrow R \text{ decides } EQ_{TM} \rightarrow \text{ Accept} \]

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

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

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

Def: Function \( f: \Sigma^* \rightarrow \Sigma^* \) 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 \( f: \Sigma^* \rightarrow \Sigma^* \), where for every \( w \),

\( \omega \in A \iff f(\omega) \in B \)

\[ \omega \in A \iff f(\omega) \in B \]

\[ A \leq_m B \iff \overline{A} \leq_m \overline{B} \]

Th: If \( A \leq_m B \) and \( B \) is decidable, then \( A \) is decidable

Th: \( T-R \)

Cor: and \( A \) is undecidable, then \( B \) is undecidable

Cor: not \( T-R \)

Exercise: show mapping reducibility \( A_{TM} \leq_m HALT_{TM} \)

\( E_{TM} \leq_m EQ_{TM} \)
Th: $\overline{EQ^{TM}}$ is not T-R nor co-T-R.

1. $\overline{A^{TM}} \leq_m EQ^{TM} = A^{TM} \leq_m \overline{EQ^{TM}}$

2. $\overline{A^{TM}} \leq_m EQ^{TM}$

$\overline{A^{TM}} \leq_m EQ^{TM}$

$A^{TM} \leq_m \overline{EQ^{TM}}$