Three Messages. 2. -- but not as much a barrier as we thought 40 and 80 years ago (Turing 1936)

3. Reductions include positive aspects of "negative" results.

Example: \( \text{HALT}_{TM} \) is undecidable.

The language \( \text{HALT}_{TM} \) is undecidable.

The language \( \text{HALT}_{TM} = \{ \langle M, x \rangle \mid M \text{ accepts } x \} \).

Direct: Suppose we had a \( \text{total} \) TM \( Q \) s.t. \( L(Q) = \text{HALT}_{TM} \). Then we could use \( Q \) to get a \( \text{total} \) TM \( R \) s.t. \( L(R) = \text{A}_{TM} \) as follows:

1. Convert \( M \) to \( M' \) s.t. if \( M \) goes to \( Q \) on \( x \), then \( M' \) loops instead.
2. The effect of this is that for all \( x \), \( M'(x) \downarrow \iff M \) accepts \( x \).

Feed \( \langle M', x \rangle \) to \( Q \) (by assumption, \( \text{solid box} \)).

If \( Q \) accepts \( L(M') \), accept; else reject.

Let \( r(n) \) this will be due by a reduction.

Diagram: (Two TMs denoted by boxes with arrows and labels.)

1. \( M \) has states and inputs.
2. \( M' \) has states and is modified.
3. \( Q \) accepts \( L(M') \) or rejects.
4. \( S' = S \cup \{ \text{new state} \} \) for all \( C \).
5. \( C \) and \( S' \) are included in \( L(R) \).
Then \( R \) accepts \( \langle M, x \rangle \) \( \iff \) \( \alpha \) accepts \( \langle M', x \rangle \)
\( \iff M'(x) \) accepts \( \iff M(x) \) accepts \( \iff \langle M, x \rangle \) \( \in \text{ATM} \).
\( \circ R \) is total and \( L(R) = \text{ATM} \), but this is impossible.
\( \text{Last lecture showed ATM is undecidable.} \)

Hence there is no such \( \alpha \). \( \Box \)

\text{Note: Halt}_{TM} \text{ is recognizable. } \text{We can code } R' \text{ to run } M(x) \text{ and accept if it halts. But if } M(x) \text{ then our } R' \text{ won't halt either. } \text{Contradiction is } \alpha \text{ or } R \text{ being total.}

\text{The complement } \overline{\text{Halt}}_{TM} \text{ is not even recognizable.}

\text{Example 2: Emptiness and Nonemptiness}

\( \text{NE}_{TM} = \text{INSTANCE: A Turing Machine } M = \langle \text{just an } M \rangle \)
\( \text{question: } \exists \ M \in \text{L(M)} \neq \emptyset ? \)
\( \text{ETM: \langle M \rangle \sim \exists \ M \in \text{L(M)} \neq \emptyset ? \)
\( \text{ETM = \langle M \rangle \sim \emptyset \implies \text{NE}_{TM} \}

\text{If } \langle M \rangle \text{ includes all strings, then } \text{ETM literally } = \overline{\text{NE}_{TM}}.

\text{If we consider any invalid code to yield the empty language, then } \text{ETM = } \exists \ x \in \text{L(M) \ or \ L(M) = \emptyset} \implies \overline{\text{NE}_{TM}}.

\text{Generally, it will be OK to ignore the issue of invalid codes.}

\text{Theorem: NE}_{TM} \text{ is recognizable but undecidable and so ETM is not even recognizable.
Proof: Suppose we had a total TM $Q$ s.t. $L(Q) = \text{NTRM}$.

Then we could build a total TM $R$ deciding ATTM as follows:

1. input to $R$ is $\langle M, x \rangle$

   - $M'$ is a machine that takes $M$ and $x$ as a fixed subroutine.
   - If $Q$ accepts, accept;
   - Else, simulate $M$ on $x$.

Analysis: $M$ accepts $x \iff L(M') \neq \emptyset$

$\langle M, x \rangle \in \text{ATTM} \Rightarrow M(x) \text{ accepts} \Rightarrow \forall w \in \Sigma^*, M(w) \in L(M') \neq \emptyset$

$\langle M, x \rangle \in \text{ATTM} \Rightarrow M(x) \text{ does not accept} \Rightarrow \forall w \in \Sigma^*, M(w) \not\in L(M') = \emptyset$

$\Rightarrow L(R) = \text{ATTM} \land R$ is total, contradiction.

Consequences:

- Not only is $R$ Turing, but also it is not even Turing reducible to $\text{NTRM}$.
- $\text{ATTM}$ is undecidable.
- Let $Q'$ be another TM.
- Suppose we had a total TM $Q'$ s.t. $L(Q') = \text{ATTM}$.
- Using $Q'$ in place of $Q$ makes $R$ behave the same.
- i.e., $L(R) = \text{ATTM}$, contradiction, so $Q'$ does not exist.

Note: This does not say that $\text{ATTM}$ is a non-reducible TM.

Example:

- $E(\emptyset) = \emptyset^*$ is not decidable.
- $E(\{0,1\}) = L(\emptyset) = \emptyset$ is decidable.

But we will see $E(\{0\}) = L(\emptyset) = \Sigma^*$ is undecidable.

\[ \text{Remember: Is } L(M) = \Sigma^? \text{ is undecidable.} \]