1. Universal Turing Machine

<u>Th</u> The Language $A_{TM} = \{\langle M, w\rangle \mid M$ is TM that accepts $w\}$ is undecidable.

<u>Th</u> The Language $A_{TM} = \{\langle M, w\rangle \mid M$ is TM that accepts $w\}$ is recognizable

<u>Proof by contradiction</u>

Assume A_{TM} is decidable.

Simulate/run M on w on a universal TM $\mathcal{U}$ (recognizability)

M accepts/rejects $w \rightarrow \mathcal{U}$ will halt and accept/reject w.

M loops on $w \rightarrow \mathcal{U}$ will not halt. However, a decoder can never loop.

$\therefore$ Contradiction: A_{TM} is undecidable.

2. Non-Turing-Recognizability

$f: A \rightarrow B$

"one-to-one": $\forall a, a' \in A,\ a \neq a' \rightarrow f(a) \neq f(a')$

"onto": $\forall b \in B,\ \exists a \in A$ s.t. $f(a) = b$

"correspondence": $\forall b \in B,\ \exists! a \in A$ s.t. $f(a) = b$

Two sets have the same size iff $\exists$ correspondence between them.

A set is countable iff it has finite size or has correspondence with $\mathbb{N} = \{1, 2, \ldots\}$.

Odd numbers $\rightarrow$ Countably infinite $(f(x) = 2x - 1)$, also subset of $\mathbb{N}$

Rational numbers, $\mathbb{Q} = \{\frac{M}{N} \mid M, N \in \mathbb{N}\} \rightarrow$ Countably infinite

Irrational/Real numbers $\rightarrow$ Uncountable infinite

<u>Th</u> The set of all Turing Machines is Countably Infinite.

<u>Corollary</u> The set of all Turing Recognizable Languages is Countably Infinite.

<u>Th</u> The set of all infinite length strings over $\{0, 1\}$ is Uncountably Infinite.

<u>Th</u> The set of all Languages is Uncountably Infinite

<u>Corollary</u> Some Languages are not Turing Recognizable.

<u>Th</u> L - decidable $\leftrightarrow L, \overline{L}$ - Turing Recognizable

M_1, M_2 - recognizers for $L, \overline{L}$; run in parallel, one will halt and accept

<u>Def.</u> Language is Co-Turing Recognizable if its complement is Turing Recognizable.

A_{TM} - Turing Recognizable, not decidable

$\Rightarrow \overline{A_{TM}}$ - Non-Turing Recognizable

3. Reducibility (Undecidability)

Known fact: A_{TM} is undecidable.

Prove: Problem P is undecidable.

<u>Proof by contradiction</u>

Assume P is decidable.

Reduce A_{TM} to P.

Use solution of P to solve A_{TM}.

→ Use decidability of P to find algorithm to decide A_{TM}.

→ Build TM to decide A_{TM} using TM to decide P as subroutine.

However, decider for A_{TM} cannot exist.

∴ Contradiction: P is undecidable.

<u>Th</u> The Language $HALT_{TM} = \{\langle M, w\rangle \mid M \text{ is TM that halts on } w\}$ is undecidable.

<u>Proof by contradiction</u>

Assume $HALT_{TM}$ is decidable.

⇒ $\exists$ TM R that decides $HALT_{TM}$.

Use R to build another TM S that decides A_{TM} (S runs R as subroutine).

Reduce A_{TM} to $HALT_{TM}$ (~~S~~ If it halts, then either accepts or rejects).

However, A_{TM} is undecidable.

∴ Contradiction: $HALT_{TM}$ is undecidable.

<u>Th</u> The Language $E_{TM} = \{\langle M\rangle \mid M \text{ is TM and } L(M) = \emptyset\}$ is undecidable.

<u>Proof by contradiction</u>

Assume E_{TM} is decidable by TM R.

Construct M' s.t.:

→ Take M.

→ Add condition $x = w$.

→ Pass control to M.

$L(M') = \{w\}$ if M accepts w. $\emptyset$ otherwise

$\left\{\begin{array}{l} R \text{ accepts} \Rightarrow L(M') = \emptyset \Rightarrow M \text{ rejects} \\ R \text{ rejects} \Rightarrow L(M') = \{w\} \Rightarrow M \text{ accepts} \end{array}\right\}$

Use R to decide whether $L(M')$ is empty or not.

⇒ R decides A_{TM}.

∴ Contradiction: E_{TM} is undecidable.