Undecidable Problems About Languages
Reducibility
Clique and Independent Set

**CLIQUE** = \{<G,k> \mid G \text{ is a graph with a } k\text{-clique}\}

**INDEPENDENT** = \{<G,k> \mid G \text{ is a graph containing an independent set of size } k\}
CLIQUE reduces to INDEPENDENT
Certified Impossible

**Theorem.** \( A_{TM} = \{ <M, w> \mid M \text{ is a TM and } M \text{ accepts } w \} \) is undecidable.

**Definition.** \( HALT_{TM} = \{ <M, w> \mid M \text{ is a TM and } M \text{ halts on input } w \} \)
The Halting Problem (Again!)

**Theorem.** \( \text{HALT}_{TM} \) is undecidable.

**Proof Idea.** We know \( A_{TM} \) is undecidable. We need to reduce one of \( \text{HALT}_{TM} \) or \( A_{TM} \) to the other.

Which way to go?
Proof. Suppose $R$ decides $\text{HALT}_{\text{TM}}$. Define

$$S = \text{"On input } <M, w>\text{, where } M \text{ is a TM and } w \text{ a string:}$$

1. Run TM $R$ on input $<M, w>$.
2. If $R$ rejects, then reject.
3. If $R$ accepts, simulate $M$ on input $w$ until it halts.
4. If $M$ enters its accept state, accept. If $M$ enters its reject state, reject."
Definition. \[ E_{TM} = \{ \langle M \rangle \mid M \text{ is a TM and } L(M) = \emptyset \} \]

Theorem. \[ E_{TM} \text{ is undecidable.} \]
Proof. Given an input \( <M, w> \) we construct a machine \( M_w \) as follows:

\[ M_w = \text{"On input } x:\]

1. If \( x \neq w \), reject.
2. If \( x = w \), run \( M \) on input \( w \) and accept if \( M \) does.

... to be continued ...
The Proof Continues

Proof continued.

Suppose TM \( R \) decides \( E_{TM} \). Define

\[ S = \text{"On input } \langle M, w \rangle \text{:\n1. Use the description of } M \text{ and } w \text{ to construct } M_w. \n2. \text{ Run } R \text{ on input } \langle M_w \rangle. \n3. \text{ If } R \text{ accepts, reject. If } R \text{ rejects, accept."} \]
With Power Comes Uncertainty

$M$ accepts $w$  \hspace{1cm}  $L(M) = \emptyset$  \hspace{1cm}  $L(M_1) = L(M_2)$

Turing machines

Pushdown machines

Finite machines
Rice’s Theorem. Any *nontrivial property* of the languages recognized by Turing machines is undecidable.
For Example

Definition. \( \text{REGULAR}^\text{TM} = \{ \langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is regular} \} \).

Theorem. \( \text{REGULAR}^\text{TM} \) is undecidable.
**REGULAR\_TM is undecidable**

Proof. Let $R$ be a TM that decides $\text{REGULAR}\_\text{TM}$. Define

$$S = \text{"On input } <M, w>:\n$$

1. Construct TM

   \[M_2 = \text{"On input } x:\]
   1. If $x$ has the form $0^n1^n$, accept.
   2. Otherwise, run $M$ on input $w$ and accept if $M$ accepts $w$.

2. Run $R$ on input $<M_2>$.

3. If $R$ accepts, accept. Otherwise, if $R$ rejects, reject."
Problems

• Let $EQ_{TM} = \{ \langle M_1, M_2 \rangle \mid M_1 \text{ and } M_2 \text{ are TMs and } L(M_1) = L(M_2) \}$. Show that $EQ_{TM}$ is undecidable by reducing $E_{TM}$ to $EQ_{TM}$.

• Consider the problem of determining whether a two-tape TM ever writes a nonblank symbol on its second tape when run on input $w$. Formulate this problem as a language and show that it is undecidable. (Hint: create an intermediary TM $T$ that writes a nonblank symbol on its second tape iff $M$ accepts $w$.)