Last time:

Recursion Theorem:

\[ \forall \text{ computable } t : \Sigma^* \times \Sigma^* \to \Sigma^* \]

\[ \exists \text{ TM } \ R \ \text{ s.t.} \]

\[ x \rightarrow R \rightarrow t(<R>,x) \]

Applications:

We now can design TM as follows:

\[ M = \text{ "On input } x, \]

1. Get own description \(<M>\).

2. Do some computation on \(<M> \& x\).

Self-Print = \{ \text{ } <M> | \text{ TM } M \text{ outputs } <M> \}

Claim: Self-Print is undecidable.
Proof: Reduce from \( A_{\text{TM}} \).

\[
\text{Aim:} \quad (M, w) \rightarrow M^	ext{'}
\]

Want: \( M \text{ acc } w \iff M^	ext{'} \text{ prints } <M^	ext{'}> \)

\[
M^	ext{'} = "\text{On input } x,\\n1. \text{ Get own description } <M^	ext{'}>\\n2. \text{ Simulate } M \text{ on } w\\n3. \text{ If } M \text{ acc } w, \text{ then print } <M^	ext{'}>"."
\]

---

**Fixed-Point Theorem**

\( A \) computable \( t : \Sigma^* \rightarrow \Sigma^* \)

\( \exists \text{ TM } F \text{ s.t.} \)

\[
L(F) = L(t(<F>,<G>))<"G">
\]

Proof:

\( F = "\text{On input } x,\quad x.\)
F = "On input x,
1. Get own desc. <F>
2. <G> = t(<F>)
3. Simulate G on x."

L(F) = L(G) = L(t(<F>)). ✓

---

Minimal TMs

Define \( \text{MIN}_{TM} = \{ <M> | M \text{ is a minimal TM} \} \)

M is called minimal TM if

\( \forall M', |<M'>| < |<M>| \)

\( \Rightarrow L(M') \neq L(M) \).

Claim: \( \text{MIN}_{TM} \) is not semi-decidable.

Proof:

Suppose \( \text{MIN}_{TM} \) is semi-decidable
Suppose $\text{MIN}_T$ is semi-decidable by some $TM$ $M$.

$c = " \text{ On input } w, \n1. \text{ Get own code } <c> \n2. \text{ Let } M_1, M_2, M_3, \ldots \text{ be an enumeration of all TMs s.t. } |<M_i>| > |<c>| \n3. \text{ for } i = 1 \text{ to } \infty \n\% \text{ want to find } \text{min}_T M_i \n\% \text{ for some } i \n\text{ Simulate } M \text{ on each } M_1, M_2, \ldots, M_i \n\text{ for } \leq i \text{ steps} \n\text{ If } M \text{ accepts some } M_k \n\text{ ( } 1 \leq k \leq i \text{ ) then } \n\text{ go to Step 4.} \n4. \text{ Simulate } M_k \text{ on } w. \n
L(c) = L(M_k) \n|<c>| < |<M_k>| \n
\text{so } M_k \text{ is not } \text{min}. \text{ Contrad.} \quad \square
Claim: There are infinitely many
with TMs.

\[ \lambda(M_1) = \lambda(M) \]
\[ \forall M \in \text{Eq. Class} \]

Summary of Basic Computability

1. TM = algorithm
2. \# undecidable problems
3. Diagonalization method
4. Reductions: (1) new algs
   (2) new lower bounds
5. Rice's Theorem
6. Recursion Theorem:
   (1) G computer viruses
   (2) Virus checking is
      undecidable