Lecture 13:
Review Recursion Theorem,
Foundations of Mathematics and
Kolmogorov Complexity
The Recursion Theorem

Theorem: For every TM $T$ computing a function
$$t : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$$
there is a Turing machine $R$ computing a function $r : \Sigma^* \rightarrow \Sigma^*$, such that for every string $w$,
$$r(w) = t(R, w)$$
For every computable t, there is a computable r such that \( r(w) = t(R,w) \) where

\( R \) is a description of a TM computing \( r \)

Moral: Suppose we can design a TM \( T \) of the form

“On input \((x,w)\), do bla bla with \( x \),
do bla bla bla bla with \( w \), etc. etc.”

We can always find a TM \( R \) with the behavior:

“On input \( w \), do bla bla bla with code of \( R \),
do bla bla bla bla with \( w \), etc. etc.”

We can use the operation:

“Obtain your own description”
in Turing machine pseudocode!
Theorem: \( A_{\text{TM}} \) is undecidable

Proof (using the recursion theorem)

Assume \( H \) decides \( A_{\text{TM}} \)

Define a TM \( T \) as follows:

\[
T(M,w) := \text{Run } H \text{ on } (M,w) \text{ and output the opposite}
\]

By the Recursion Theorem,

There is a TM \( B \) such that for all \( w \), \( B(w) = T(B,w) \).

Then, \( B \) on input \( w \) always does the opposite of what \( H(B,w) \) said \( B \) would do! Contradiction!
Theorem: \( A_{\text{TM}} \) is undecidable

Proof (using the recursion theorem)

Assume \( H \) decides \( A_{\text{TM}} \)

Construct machine \( B \) such that on input \( w \):

1. Obtains own description \( B \)
2. Runs \( H \) on \((B, w)\) and flips the output

Running \( B \) on any input \( w \) always does the \textit{opposite} of what \( H(B,w) \) says \( B \) would do! Contradiction!

A formalization of “free will” paradoxes!
No single machine can predict behavior of all others
Turing Machine Minimization

MIN = \{ M \mid M \text{ is a minimal-state TM over } \Gamma = \{ 0, 1, \square \} \}

Theorem: MIN is undecidable

Proof: Suppose we could recognize MIN with TM \( M' \)

\[ M(x) := \text{Obtain the description of } M. \]

For \( k = 1, 2, 3, \ldots \)

Run \( M' \) on the first \( k \) TMs \( M_1, \ldots, M_k \) for \( k \) steps,

Until \( M' \) accepts some \( M_i \) with more states than \( M \)

Output \( M_i(x) \).

Why does \( M_i \) exist?

We have: 1. \( L(M) = L(M_i) \) [by construction]

2. \( M \) has fewer states than \( M_i \)

3. \( M_i \) is minimal [by definition of MIN]

CONTRADICTION!
Computability and the Foundations of Mathematics
Formal Systems of Mathematics

A formal system describes a formal language for
- writing (finite) mathematical statements as strings,
- has a definition of a proof of a statement (as strings)
- has a notion of “true” statements

Example: Every TM M can be used to define a formal system $\mathcal{F}$ with the properties:

- $\{\text{Mathematical statements in } \mathcal{F}\} = \Sigma^*$
  String $w$ represents the statement “M halts on $w$”
- A proof of “M halts on $w$” can be defined as the computation history of M on $w$: the sequence of configurations $C_0 \ C_1 \ \cdots \ C_t$ that M goes through while computing on $w$
Interesting Systems of Mathematics

Define a formal system $\mathcal{F}$ to be *interesting* if:

1. Mathematical statements about computation can be (computably) described as a statement of $\mathcal{F}$. Given $(M, w)$, there is a (computable) $S_{M,w}$ of $\mathcal{F}$ such that $S_{M,w}$ is true in $\mathcal{F}$ if and only if $M$ accepts $w$.

2. Proofs are “convincing” – a TM can check that a candidate proof of a theorem is correct. This set is decidable: $\{ (S, P) | P$ is a proof of $S$ in $\mathcal{F} \}$. 

3. If $S$ is in $\mathcal{F}$ and there is a proof of $S$ describable as a computation history, then there’s a proof of $S$ in $\mathcal{F}$.

If TM $M$ halts on $w$, then there’s either a proof $P$ of $S_{M,w}$ or a proof $P$ of $\neg S_{M,w}$.
Consistency and Completeness

A formal system $F$ is inconsistent if there is a statement $S$ in $F$ such that both $S$ and $\neg S$ are provable in $F$. $F$ is consistent if it is NOT inconsistent.

A formal system $F$ is incomplete if there is a statement $S$ in $F$ such that neither $S$ nor $\neg S$ are provable in $F$. $F$ is complete if it is NOT incomplete.

We want consistent and complete systems!
Limitations on Mathematics!

For every consistent and interesting $F$,

Theorem 1. (Gödel 1931) $F$ must be *incomplete*!
“The there are mathematical statements that are true but cannot be proved.”

Theorem 2. (Gödel 1931)
The consistency of $F$ cannot be proved in $F$.

Theorem 3. (Church-Turing 1936) The problem of checking whether a given statement in $F$ has a proof is undecidable.
Unprovable Truths in Mathematics

(Gödel) Every consistent interesting $\mathcal{F}$ is incomplete: there are statements that cannot be proved or disproved.

Let $S_{M,w}$ in $\mathcal{F}$ be true if and only if TM M accepts string w

Proof: Define TM $G(w)$:

1. Obtain own description $G$ [Recursion Theorem!]
2. For all strings P in lexicographical order,
   If (P is a proof of $S_{G,w}$ in $\mathcal{F}$) then reject
   If (P is a proof of $\neg S_{G,w}$ in $\mathcal{F}$) then accept

Note: If $\mathcal{F}$ is complete then $G$ cannot run forever!

1. If ($G$ accepts w) then have proof P of “$G$ doesn’t accept w”
2. If ($G$ rejects w) then have proof P of “$G$ accepts w”

In either case, $\mathcal{F}$ is inconsistent! Proof of $S_{G,w}$ and $\neg S_{G,w}$
Unprovable Truths in Mathematics

(Gödel) Every consistent interesting $\mathcal{F}$ is incomplete: there are statements that cannot be proved or disproved.

Let $S_{M,w}$ in $\mathcal{F}$ be true if and only if TM M accepts string w

Proof: Define TM $G(w)$:

1. Obtain own description $G$ [Recursion Theorem!]
2. For all strings $P$ in lexicographical order,
   If ($P$ is a proof of $S_{G,w}$ in $\mathcal{F}$) then reject
   If ($P$ is a proof of $\neg S_{G,w}$ in $\mathcal{F}$) then accept

   Note: If $\mathcal{F}$ is complete then $G$ cannot run forever!

Conclusion: $G$ must run forever.
So in fact $\neg S_{G,w}$ is a true statement, but it has no proof in $\mathcal{F}$!

    In either case, $\mathcal{F}$ is inconsistent! Proof of $S_{G,w}$ and $\neg S_{G,w}$
(Gödel 1931) The consistency of $F$ cannot be proved within any interesting consistent $F$

Proof Sketch: Assume we can prove “$F$ is consistent” in $F$

We constructed $\neg S_{G, w} = “G$ does not accept $w”$

which has no proof in $F$

$G$ accepts $w$ $\Rightarrow$ There are proofs of $S_{G, w}$ and $\neg S_{G, w}$ in $F$

But if there’s a proof of “$F$ is consistent” in $F$, then there is a proof of $\neg S_{G, w}$ in $F$ (here’s the proof):

“$F$ is consistent, because <insert proof here>.
If $S_{G, w}$ is true, then both $S_{G}$ and $\neg S_{G, w}$ have proofs in $F$.

But $F$ is consistent, so this is a contradiction. Therefore, $\neg S_{G, w}$ is true.”

This contradicts the previous theorem!
Undecidability in Mathematics

\[ \text{PROVABLE}_F = \{ S \mid \text{there's a proof in } F \text{ of } S, \text{ or there's a proof in } F \text{ of } \neg S \} \]

(Church-Turing 1936) For every interesting consistent \( F \), \( \text{PROVABLE}_F \) is undecidable

Proof: Suppose \( \text{PROVABLE}_F \) is decidable with TM P. Then we could decide \( A_{\text{TM}} \) with the following procedure:

On input \((M, w)\), run the TM P on input \( S_{M,w} \)
If P accepts, examine all proofs in lex order
If a proof of \( S_{M,w} \) is found then accept
If a proof of \( \neg S_{M,w} \) is found then reject
If P rejects, then reject.

Why does this work?
Kolmogorov Complexity:
A Universal Theory of Data Compression