Computability and the Foundations of Mathematics
A formal system describes a formal language for
- writing (finite) mathematical statements,
- has a definition of what statements are “true”
- has a definition of a proof of a statement

Example: Every TM $M$ defines some formal system $F$
{Mathematical statements in $F)} = \Sigma^*$
String $w$ represents the statement “$M$ accepts $w$”
✓ {True statements in $F)} = L(M)$
✓ A proof that “$M$ accepts $w$” can be defined to be an accepting computation history for $M$ on $w$
Interesting Formal Systems

Define a formal system $F$ to be interesting if:

1. Mathematical statements that can be precisely described in English should be expressible in $F$

2. Proofs are “convincing” – a TM can check that a proof of a theorem is correct (decidable)

3. Simple proofs that can be precisely described in English should be expressible in $F$
Define a formal system $F$ to be interesting if:

1. Any mathematical statement about computation can be (computably) described as a statement of $F$.

   Given $(M, w)$, there is a (computable) $S_{M,w}$ in $F$ such that $S_{M,w}$ is true in $F$ if and only if $M$ accepts $w$.

2. Proofs are “convincing” – a TM can check that a proof of a theorem is correct

   This set is decidable: $\{(S, P) \mid P \text{ is a proof of } S \text{ in } F\}$

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

   If $M$ accepts $w$, then there is a proof $P$ in $F$ of $S_{M,w}$
Consistency and Completeness

A formal system $F$ is consistent or sound if no false statement has a valid proof in $F$ (Proof in $F$ implies Truth in $F$)

A formal system $F$ is complete if every true statement has a valid proof in $F$ (Truth in $F$ implies Proof in $F$)
Limitations on Mathematics

For every consistent and interesting $F$,

Theorem 1. (Gödel 1931) $F$ is incomplete:
There are mathematical statements in $F$ that are true but cannot be proved in $F$.

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 $F$ is incomplete: there are true statements that cannot be proved.

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

Proof: Define Turing machine $G(x)$:
1. Obtain own description $G$ [Recursion Theorem]
2. Construct statement $S' = \neg S_{G,\varepsilon}$
3. Search for a proof of $S'$ in $F$ over all finite length strings. Accept if a proof is found.

Claim: $S'$ is true in $F$, but has no proof in $F$ ($S'$ basically says “There is no proof of $S'$ in $F$”)
(Gödel 1931) The consistency of \( F \) cannot be proved within any interesting consistent \( F \)

Proof: Suppose we can prove “\( F \) is consistent” in \( F \)

We constructed \( \neg S_{G, \varepsilon} = \text{“} G \text{ does not accept } \varepsilon \text{”} \) which we showed is true, but has no proof in \( F \)

\( G \) does not accept \( \varepsilon \) \( \iff \) There is no proof of \( \neg S_{G, \varepsilon} \) in \( F \)

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

“If \( S_{G, \varepsilon} \) is true, then there is a proof in \( F \) of \( \neg S_{G, \varepsilon} \)

\( F \) is consistent, therefore \( \neg S_{G, \varepsilon} \) is true.

But \( S_{G, \varepsilon} \) and \( \neg S_{G, \varepsilon} \) cannot both be true.

Therefore, \( \neg S_{G, \varepsilon} \) is true”
Proof:
Suppose \( \text{PROVABLE}_F \) is decidable with TM \( P \).
Then we can decide \( A_{\text{TM}} \) using the following procedure:
On input \((M, w)\), run the TM \( P \) on input \( S_{M,w} \).
If \( P \) accepts, examine all possible proofs in \( F \).
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?
Parting thoughts:
If a formal mathematical system is consistent and interesting:
• True statements that cannot be proved (including the consistency of the system),
• Provable statements we cannot tell are provable.
What about statements with short proofs?