Recap

- **Fact**: Th(\(\mathbb{N},+\)) is decidable (this is review from last lecture).

  For Example: \(\forall p \exists q : (q = p + 1)\)

- **Theorem**: Th(\(\mathbb{N}, +, x\)) is undecidable.

  For Example: \(\forall q \exists p \forall x, y : (p > q \land (x, y > 1 \rightarrow p \neq xy))\)

- **Basic Idea**: For every TM M and input w, there is a formula \(\phi_{M,w}\) with one free variable x such that [M accepts w \(\iff \exists x \phi_{M,w}\) is true].
  
  - \(\phi_{M,w}\) is in the language of Th(\(\mathbb{N},+\),x)
  
  - Given M and w, there exists a TM that computes \(\phi_{M,w}\)

- **Exact Proof**: Assume Th(\(\mathbb{N},+\),x) is decidable by a TM R. We define a machine N as follows:
  1. "On input \(< M, w >:\"
  2. Compute \(\phi_{M,w}\)
  3. Simulate R on \(\exists x \phi_{M,w}\)
  4. If R accept, ACCEPT
  5. If R reject, REJECT"

N decides \(A_{TM}\) which is a contradiction and implies that Th(\(\mathbb{N},+\),x) is undecidable

In order to prove \([\exists x \phi_{M,w} \iff M \text{ accept } w \text{ true}]\) we define x. x is a sequence of TM configurations represented as

\[
x = "c_1 \rightarrow c_2 \rightarrow c_3 \rightarrow ... \rightarrow c_m"
\]

Where \(c_1\) is the start state configuration of M on w, \(c_i\) is a valid next step configuration of M on w, and \(c_m\) is the accept state config of M on w.
**Example:** How to encode a configuration as a number sequence:

- If the current state of a TM is 0110q60110, we can represent q6 as a base 2 number, but with 3 → 0 and 4 → 1
- The encoding of this state therefore looks like 201104430110
- Multiple states can be encoded as follows: 201104430110 | 201100444110 | ...

$\phi_{M,w}$ is true $\iff$ our long number of states (the encoding of $x$ shown above) is a valid set of configurations for $M$ accepting $w$. In order to determine whether this is the case, we must be able to randomly access a digit of $x$. The process for doing so is shown in the example below.

**Example:** Accessing the $k$'th digit of a configuration:

- $mod(x, y, z) = \exists k \ s.t. \ (yk + z = x \land (z < y))$
  - Tests if $x \ mod \ y == z$
- $div(x, y, z) = \exists r \ s.t. \ (yz + r = x \land (r < y))$
  - Tests if the quotient of $x/y == z$
- $digit(x, k, d) = \exists q \ s (div(x, 10^{k-1}, q) \ and \ mod(q, 10, d))$
  - Tests if the $k$'th digit of $x$ (from the right) == $d$
  - Exponentiation is allowed for this function

If we let $x, x'$ be two arbitrary configurations, we can check whether TM $M$ in configuration $x$ goes to $x'$ by creating a giant table of all changes that can be made to string $x$. We can then find the differences between $x$ and $x'$ and check our table to see if these are acceptable differences. This can be implemented using $digit(x, k, d)$. Care must be taken for the front and back of strings $x$ and $x'$ but no further detail was given. Lastly, if the above is true for all sequences involving $x$ and $x'$ then $[\phi_{M,w} \iff M \text{ accepts } w]$ has been proven.

**Definition:** Proof System (from 311). If we want to prove a sentence $\phi$, we use a sequence of statements $S_1, S_2 \ldots S_m = \phi$. Each statement $S_i$ is either an axiom or follows logically from previous statements.

**Definition:** Provability $\phi$ is provable if $\phi$ has a proof.

**Definition:** Soundness $\phi$ is provable $\Rightarrow$ $\phi$ is true.

**Fact:** The set of provable sentences is turing recognizable (there is a TM that if given a provable sentence will accept)

**Proof:**
1. "On input $\phi$
2. Enumerate all the proofs : In lexicographic order, $pi_1, pi_2, \ldots pi_n$
3. For all i, check whether $pi_i$ is a valid proof of phi. If so, ACCEPT"
**Theorem:** There is a true sentence in $\text{Th}(\mathbb{N},+)$ that is unprovable.

**Proof:** Suppose that every true sentence of $\text{Th}(\mathbb{N},+)$ is provable. We define the following TM, $TM_{FINAL}$:

1. Given $\phi$: Either $\phi$ is true or $\neg\phi$ is true
2. Run the provable recognizer on $\phi$ and $\neg\phi$ in parallel
3. The one that is provable will eventually be accepted
4. If $\phi$ is accepted, ACCEPT
5. If $\neg\phi$ is accepted REJECT

Since $\text{Th}(\mathbb{N},+)$ is undecidable, our supposition must be wrong, meaning there must be an unprovable true statement.

**Example:** $\psi = "This sentence is not provable"

**Example:** TM $S =$
1. "On any input:
2. Obtain my source code $<S>$ by Recursion Thm
3. Compute the formula $psi = \neg(\exists x\phi_{S,0})$
4. If $\psi$ is provable , ACCEPT"

**Claim:** $\psi$ is true but unprovable due to the following contradictions:

- If $\psi$ is false, $S$ accepts 0, meaning $\psi$ provable and therefore $\psi$ is true
- If $\psi$ is unprovable, $S$ doesnt accept 0, which means $\psi$ is provable