The Importance of Being Formal

Martin Henz

March 12, 2014
1. Infinity
2. Decidability
3. (In)completeness
4. Undefinability
1. **Infinity**
   - Finite Sets
   - Countable and Uncountable Sets
   - The Cantor-Schröder-Bernstein Theorem

2. **Decidability**

3. **(In)completeness**

4. **Undefinability**
Finite sets

There is a finite number that represents the cardinality of the set.

Example

$S = \{a, b, c, d, e\}$: The number 5 is the cardinality of $S$.

How about this set?

$\mathbb{N} = \{0, 1, 2, 3, 4, \ldots\}$ What is the cardinality of $\mathbb{N}$?
We count finite sets by establishing a function that is one-to-one and onto between the set and the numbers \( \{1, 2, \ldots, n\} \).

We say the two sets are *equinumerous*. 
Equinumerous Sets

Definition

Suppose $A$ and $B$ are sets. We say that $A$ is equinumerous with $B$ if there is a function $f : A \rightarrow B$ that is one-to-one and onto, denoted $A \sim B$. For each natural number $n$, let $l_n = \{i \in \mathbb{Z}^+ | i \leq n\}$.

Definition

A set $A$ is called finite if there is a natural number $n$ such that $A \sim \{i \in \mathbb{Z}^+ | i \leq n\}$.
Surprising Example

\[ \mathbb{Z}^+ \text{ and } \mathbb{Z} \text{ are equinumerous} \]

\[ \mathbb{Z}^+ \sim \mathbb{Z} \]

Proof

\[ f(n) = \begin{cases} 
\frac{n}{2} & \text{if } n \text{ is even} \\
\frac{1-n}{2} & \text{if } n \text{ is odd} 
\end{cases} \]
Even More Surprising

\[ \mathbb{Z}^+ \times \mathbb{Z}^+ \text{ and } \mathbb{Z}^+ \text{ are equinumerous} \]

\[ \mathbb{Z}^+ \times \mathbb{Z}^+ \sim \mathbb{Z}^+ \]
Equinumerosity is an Equivalence Relation

Theorem

For any sets $A$, $B$, $C$:

1. $A \sim A$
2. If $A \sim B$ then $B \sim A$.
3. If $A \sim B$ and $B \sim C$, then $A \sim C$. 

Definition

A set $A$ is called *denumerable* if $\mathbb{Z}^+ \sim A$.

Definition

A set $A$ is called *countable* if it is either finite or denumerable.
**Theorem**

Suppose $A$ and $B$ are countable sets. Then:

1. $A \times B$ is countable.
2. $A \cup B$ is countable.

**Theorem**

The union of countably many countable sets is countable.

**Theorem**

Let $A$ be a countable set. The set of all finite sequences of elements of $A$ is countable.
Cantor’s Theorem

\( \mathcal{P}(\mathbb{Z}^+) \) is uncountable.

**Corollary**

\( \mathbb{R} \) is uncountable.
Domination

Definition

We say $B$ dominates $A$, written $A \preceq B$, if there is a function $f : A \rightarrow B$ that is one-to-one.
Suppose $A$ and $B$ are sets. If $A \preccurlyeq B$ and $B \preccurlyeq A$, then $A \sim B$.

**Corollary**

$\mathbb{R} \sim \mathcal{P}(\mathbb{Z}^+)$
Continuum Hypothesis

Hypothesis
There is no set $X$ such that $\mathbb{Z}^+ \subset X \subset \mathbb{R}$.

Impossibility of Proof
Gödel and Cohen proved that it is impossible to prove the continuum hypothesis, and it is also impossible to disprove it.
<table>
<thead>
<tr>
<th>Q</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>If $\text{Term}$ is countable, is its Traditional Logic countable?</td>
<td>yes</td>
</tr>
<tr>
<td>If $A$ is countable, is its Propositional Logic countable?</td>
<td>yes</td>
</tr>
</tbody>
</table>

Other countable sets

predicate logic, modal logic, all proofs in natural deduction
1. Infinity
2. Decidability
3. (In)completeness
4. Undefinability
**Definition**

A *decision problem* is a question in some formal system with a yes-or-no answer.

**Examples**

The question whether a given propositional formula is satisifiable (unsatisfiable, valid, invalid) is a decision problem.

The question whether two given propositional formulas are equivalent is also a decision problem.
## How to Solve the Decision Problem?

### Question

How do you decide whether a given propositional formula is satisfiable/valid?

### The good news

We can construct a truth table for the formula and check if some/all rows have $\top$ in the last column.

### Algorithm

A precise step-by-step procedure for solving a problem is called an *algorithm* for the problem.
Decidability

Definition
Decision problems for which there is an algorithm computing “yes” whenever the answer is “yes”, and “no” whenever the answer is “no”, are called *decidable*.

An algorithm for satisfiability
Using a truth table, we can implement an *algorithm* that returns “yes” if the formula is satisfiable, and that returns “no” if the formula is unsatisfiable.

Decidability of satisfiability
The question, whether a given propositional formula is satisfiable, is decidable.
Is termination of algorithms decidable?

The Halting Problem
For a given algorithm (program) $P$ and a given input data $D$, decide whether $P$ terminates on $D$.

The bad news
The Halting Problem is not decidable

Language does not matter
It does not matter whether you decide to use JavaScript or C or a Turing Machine or the lambda calculus
Decidability of Propositional Logic

**Theorem**

The decision problem of validity in propositional logic is decidable: There are algorithms which, given any formula $\phi$ of propositional logic, decides whether $\models \phi$.

**Proof**

One such algorithm builds the full truth table for the given formula and then checks whether the last column has no $F$. 

---

The Importance of Being Formal

08—The Ugly Corners of Math, Logic and Computation
Theorem

The decision problem of validity in predicate logic is undecidable: no program exists which, given any language in predicate logic and any formula $\phi$ in that language, decides whether $\models \phi$.

Proof sketch

- Establish that the Post Correspondence Problem (PCP) is undecidable
- Translate an arbitrary PCP, say $C$, to a formula $\phi$.
- Establish that $\models \phi$ holds if and only if $C$ has a solution.
- Conclude that validity of predicate logic formulas is undecidable.
1. Infinity
2. Decidability
3. (In)completeness
4. Undefinability
Natural Deduction in Propositional Logic

\[ \phi_1, \ldots, \phi_n \models \psi \iff \phi_1, \ldots, \phi_n \vdash \psi \]

**Proof sketch**

"\( \leftarrow \)" : Show that each proof rule does the right thing, semantically. Structural induction.

"\( \Rightarrow \)" : Construct a proof based on the truth table (tedious).
Natural Deduction in Predicate Logic

\[ \phi_1, \ldots, \phi_n \models \psi \]

iff

\[ \phi_1, \ldots, \phi_n \vdash \psi \]

proven by Kurt Gödel, in 1929 in his doctoral dissertation
Second-order Predicate Logic

Definition
Second-order predicate logic has the same definition as first order predicate logic, but after $\forall$ and $\exists$ predicate symbols are allowed.

Example
$\forall P \forall x (P(x) \lor \neg P(x))$
Incompleteness of Second-order Logic

There is no deductive system (that is, no notion of provability) for second-order formulas that simultaneously satisfies the following:

**Soundness:** Every provable second-order sentence is universally valid, i.e., true in every model.

**Completeness:** Every universally valid second-order formula, under standard semantics, is provable.

**Effectiveness:** There is a proof-checking algorithm that can correctly decide whether a given sequence of symbols is a valid proof or not.
Gödel’s First Incompleteness Result

Theorem
No consistent system of axioms whose theorems can be listed by an algorithm is capable of proving all truths about the relations of the natural numbers (arithmetic).

Proof sketch
Represent formulas by natural numbers. Express provability as a property of these numbers. Construct a bomb: “This formula is valid, but not provable.”
Gödel’s Second Incompleteness Result

**Theorem**

For any formal effectively generated theory $T$ including basic arithmetical truths and also certain truths about formal provability, if $T$ includes a statement of its own consistency then $T$ is inconsistent.
Tarski’s Undefinability Result

Theorem
Given some formal arithmetic, the concept of truth in that arithmetic is not definable using the expressive means that arithmetic affords.