Gödel's First Incompleteness Theorem

UIT2206: The Importance of Being Formal

Martin Henz

March 26, 2014

Generated on Wednesday 26th March, 2014, 09:48

Predicate logic: Terms

$$t ::= \ x \mid c \mid f(t, \ldots, t)$$

where

- x ranges over a given set of variables $\mathcal{V}$,
- c ranges over nullary function symbols in $\mathcal{F}$, and
- f ranges over function symbols in $\mathcal{F}$ with arity $n > 0$.

Predicate logic: Formulas

$$\begin{aligned}\phi \quad ::= \quad & P(t, \ldots, t) \mid (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid \\ & (\phi \to \phi) \mid (\forall x \phi) \mid (\exists x \phi)\end{aligned}$$

where

- $P \in \mathcal{P}$ is a predicate symbol of arity $n \geq 0$,
- t are terms over $\mathcal{F}$ and $\mathcal{V}$, and
- x are variables in $\mathcal{V}$.

Foundational crisis of mathematics

Wish for consistent foundation

In early 20th century, mathematicians were aiming for a common consistent foundation for mathematics

Foundational crisis of mathematics

Wish for consistent foundation

In early 20th century, mathematicians were aiming for a common consistent foundation for mathematics

Paradoxes

Problems such as Russell's Paradox indicated the difficulty of the task

Foundational crisis of mathematics

Wish for consistent foundation

In early 20th century, mathematicians were aiming for a common consistent foundation for mathematics

Paradoxes

Problems such as Russell's Paradox indicated the difficulty of the task

Hilbert's program

In 1920s, David Hilbert called for a concerted effort towards a consistent foundation, using logic and deduction as the tools of choice: "Develop a finite set of axioms in predicate logic that allows the proof of all known mathematics"

Entscheidungsproblem

A very useful tool...

Entscheidungsproblem

A very useful tool...

...in Hilbert's program would be a method to decide whether a given sentence in predicate logic is valid or not, the "Entscheidungsproblem" (in English: decision problem)

Entscheidungsproblem

A very useful tool...

...in Hilbert's program would be a method to decide whether a given sentence in predicate logic is valid or not, the "Entscheidungsproblem" (in English: decision problem)

Challenge

Hilbert posed this problem in 1928. If it could be solved, all problems that can be stated in predicate logic would be automatically solvable.

Undecidability of Predicate Logic

Theorem (Church, Turing: 1936)

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

Undecidability of Predicate Logic

Theorem (Church, Turing: 1936)

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

Proof sketch

- Establish that the Post Correspondence Problem (PCP) is undecidable

Undecidability of Predicate Logic

Theorem (Church, Turing: 1936)

The decision problem of validity in predicate logic is undecidable: no program exists which, given any language in predicate logic and any formula ϕ 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 ϕ.

Undecidability of Predicate Logic

Theorem (Church, Turing: 1936)

The decision problem of validity in predicate logic is undecidable: no program exists which, given any language in predicate logic and any formula ϕ 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 ϕ.
- Establish that $\models \phi$ holds if and only if C has a solution.

Undecidability of Predicate Logic

Theorem (Church, Turing: 1936)

The decision problem of validity in predicate logic is undecidable: no program exists which, given any language in predicate logic and any formula ϕ 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 ϕ.
- Establish that $\models \phi$ holds if and only if C has a solution.
- Conclude that validity of predicate logic formulas is undecidable.

Central Result of Natural Deduction

Theorem

$$\phi_1, \ldots, \phi_n \models \psi$$

iff

$$\phi_1, \ldots, \phi_n \vdash \psi$$

Central Result of Natural Deduction

Theorem

$$\phi_1, \ldots, \phi_n \models \psi$$

iff

$$\phi_1, \ldots, \phi_n \vdash \psi$$

proven by Kurt Gödel, in 1929 in his doctoral dissertation

Central Result of Natural Deduction

Theorem

$$\phi_1, \ldots, \phi_n \models \psi$$

iff

$$\phi_1, \ldots, \phi_n \vdash \psi$$

proven by Kurt Gödel, in 1929 in his doctoral dissertation (just one year before his most famous result, the incompleteness results of predicate logic)

A more modest program

Hilbert's program

In 1920s, David Hilbert called for a concerted effort towards a consistent foundation, using logic and deduction as the tools of choice: "Develop a finite set of axioms in predicate logic that allows the proof of all known mathematics"

A more modest program

Hilbert's program

In 1920s, David Hilbert called for a concerted effort towards a consistent foundation, using logic and deduction as the tools of choice: "Develop a finite set of axioms in predicate logic that allows the proof of all known mathematics"

Limitations due to undecidability

The undecidability of predicate logic shows that these proofs cannot be automatically obtained

A more modest program

Hilbert's program

In 1920s, David Hilbert called for a concerted effort towards a consistent foundation, using logic and deduction as the tools of choice: "Develop a finite set of axioms in predicate logic that allows the proof of all known mathematics"

Limitations due to undecidability

The undecidability of predicate logic shows that these proofs cannot be automatically obtained

Hilbert's more modest program

would provide a sound and complete proof theory for mathematics: All valid theorems are provable and every proof is valid

Can predicate logic “express” arithmetics?

Idea: introduce constant symbol 0 and “successor” function S.

Can predicate logic "express" arithmetics?

Idea: introduce constant symbol 0 and "successor" function S.

Example

1 + 2 = 3 is expressed as

$$plus(S(0), S(S(0))) = S(S(S(0)))$$

First 8 Peano Axioms

1. 0 is a natural number.
2. For every natural number x, x = x. (reflexive)
3. For all natural numbers x and y, if x = y, then y = x. (symmetric)
4. For all natural numbers x, y and z, if x = y and y = z, then x = z. (transitive)
5. For all a and b, if a is a natural number and a = b, then b is also a natural number.
6. For every natural number n, S(n) is a natural number.
7. For every natural number n, S(n) = 0 is false.
8. For all natural numbers m and n, if S(m) = S(n), then m = n.

Elusive number 9

Ninth Peano Axiom in second-order predicate logic

If P is a unary predicate such that:

- $P(0)$ is true, and
- for every natural number n, if $P(n)$ is true, then $P(S(n))$ is true,

then $P(n)$ is true for every natural number n.

Back to Hilbert's program

Recall: Hilbert's more modest program

would provide a sound and complete proof theory for mathematics: All valid theorems are provable and every proof is valid

Back to Hilbert's program

Recall: Hilbert's more modest program

would provide a sound and complete proof theory for mathematics: All valid theorems are provable and every proof is valid

Arithmetics is a must

Surely arithmetics should be covered by the proof theory for mathematics

Back to Hilbert's program

Recall: Hilbert's more modest program

would provide a sound and complete proof theory for mathematics: All valid theorems are provable and every proof is valid

Arithmetics is a must

Surely arithmetics should be covered by the proof theory for mathematics

More concrete program

Find a sound and complete proof theory for second-order predicate logic

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).

Consequence for second-order predicate logic

Theorem

For second-order predicate logic, there is no deduction system $\vdash$ such that

$$\phi_1, \ldots, \phi_n \models \psi$$

iff

$$\phi_1, \ldots, \phi_n \vdash \psi$$

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."

Reading guide

- Read material before page 259 “From Mumon to the MU-puzzle” for your own entertainment (and edification)
- Ignore references to tortoises (or read GEB over the holidays)
- Central Dogma of Mathematical Logic: TNT $\Rightarrow$ N $\Rightarrow$ meta-TNT
- What is TNT?
- What is MIU?

What is TNT?

Hint

It's not Trinitrotoluene (explosive)

What is TNT?

Hint

It's not Trinitrotoluene (explosive)

Hint

It's not Trinitrotoluene (explosive)

What is TNT?

Hint

It's not Trinitrotoluene (explosive)

Hint

It's not Trinitrotoluene (explosive)

Motivation

Devise logic that is just expressive enough for arithmetics

What is TNT?

Hint

It's not Trinitrotoluene (explosive)

Hint

It's not Trinitrotoluene (explosive)

Motivation

Devise logic that is just expressive enough for arithmetics
Hofstadter calls this calculus "TNT": Typographical Number Theory

What is MIU?

- Symbols: M, I, U
- Axiom: MI
- Rules:
  1. If xI is a theorem, so is xIU.
  2. IF Mx is a theorem, so is Mxx.
  3. In any theorem, III can be replaced by U.
  4. UU can be dropped from any theorem.