Inconsistency Robustness in Foundations: Mathematics self proves its own formal consistency and other matters

Carl Hewitt

This article is dedicated to Alonzo Church, Richard Dedekind, Stanisław Jaśkowski, Ludwig Wittgenstein, and Ernst Zermelo.

Abstract

Inconsistency Robustness is performance of information systems with pervasively inconsistent information. Inconsistency Robustness of the community of professional mathematicians is their performance repeatedly repairing contradictions over the centuries. In the Inconsistency Robustness paradigm, deriving contradictions has been a progressive development and not "game stoppers." Contradictions can be helpful instead of being something to be "swept under the rug" by denying their existence, which has been repeatedly attempted by authoritarian theoreticians (beginning with some Pythagoreans). Such denial has delayed mathematical development. This article reports how considerations of Inconsistency Robustness have recently influenced the foundations of mathematics for Computer Science continuing a tradition developing the sociological basis for foundations.[1]

Classical Direct Logic is a foundation of mathematics for Computer Science, which has a foundational theory (for convenience called "Mathematics") that can be used in any other theory. A bare turnstile is used for Mathematics so that $\vdash\Psi$ means that Ψ is a mathematical proposition that is a theorem of Mathematics and $\Phi\vdash\Psi$ means that Ψ can be inferred from Φ in Mathematics.

A current common understanding is that Gödel proved "Mathematics cannot prove its own consistency, if it is consistent." However, the formal consistency of mathematics can be proved by a simple argument using standard rules of Mathematics including the following:

- rule of *Proof by Contradiction*, *i.e.*, $(\neg\Phi\Rightarrow(\Theta\wedge\neg\Theta))\vdash\Phi$
- the rule *Theorem Use*, *i.e.*, $(\vdash\Phi)\vdash\Phi$[2]

Formal Proof.

By definition, $\text{Consistent}\Leftrightarrow\neg\exists[\Psi]\rightarrow\vdash(\Psi\wedge\neg\Psi)$. By Existential Elimination, there is some proposition Ψ_0 such that $\neg\text{Consistent}\Rightarrow\vdash(\Psi_0\wedge\neg\Psi_0)$ which by *Theorem Use* and transitivity of implication means $\neg\text{Consistent}\Rightarrow(\Psi_0\wedge\neg\Psi_0)$. Substituting for Φ and Θ, in the rule for Proof by Contradiction, it follows that $(\neg\text{Consistent}\Rightarrow(\Psi_0\wedge\neg\Psi_0))\vdash\text{Consistent}$. Thus, $\vdash\text{Consistent}$.

The above theorem means that formal consistency is deeply embedded in the architecture of classical mathematics. Please note the following points:

- **The above argument formally mathematically proves the theorem that mathematics is consistent**
- **It is *not* a premise of the theorem that mathematics is consistent.**

Classical mathematics was designed for consistent axioms and consequently the rules of classical mathematics can be used to prove formal consistency regardless of the axioms, *e.g.*, Euclidean geometry.[3]

By formally consistent, it is meant that a consistency is not inferred. The proof is remarkably tiny consisting of only using proof by contradiction. In fact, it is so easy that one wonders why this was overlooked by so many great logicians in the past. The above proof of formal consistency is also remarkable that it does not use knowledge about the content of mathematical theories (plane geometry, integers, *etc*.). The proof serves to formalize that formal consistency is built into the very architecture of classical mathematics saying nothing about the possibility of bugs in the rules of inference and axioms of Direct Logic.. **Consequently, the proof of formal consistency does not provide any assurance that there are no dangerous security holes in Direct Logic**

In particular, the proof of formal consistency does not prove *constructive* consistency, which provides that the rules of Classical Direct Logic themselves do not derive a contradiction**.** Proof of constructive consistency requires a separate inductive proof using the axioms and rules of inference of Classical Direct Logic. The upshot is that, contra Gödel, there seems to be no inherent reason that mathematics cannot prove constructive consistency of Classical Direct Logic (which formalizes classical mathematical theories). However, such a proof seems far beyond the current state of the art.[4]

The usefulness of Classical Direct Logic depends crucially on its consistency because otherwise it cannot be used freely to reason about inconsistent information without introducing further inconsistencies. Fortunately, Classical Direct Logic directly builds on classical mathematics. Good evidence for the consistency Classical Direct Logic derives from how it blocks the known paradoxes of classical mathematics. Humans have spent millennia devising paradoxes for classical mathematics.

Having a powerful system like Direct Logic is important in computer science because computers must be able to formalize all logical inferences (including inferences about their own inference processes) without requiring recourse to human intervention. Any inconsistency in Classical Direct Logic would be a potential security hole because it could be used to cause computer systems to adopt invalid conclusions.

It is very important to distinguish between the following:

- using the Y untyped fixed point operator for propositions[5]
- using recursive definitions that are strongly typed to construct propositions.

Gödel based his incompleteness results on the thesis that mathematics necessarily has the proposition *I'm unprovable.* using what [Carnap 1934] later called the "Diagonal Lemma" which is equivalent to the Y untyped fixed point operator on propositions. **Using strong parameterized types, it is impossible to construct the Gödel's proposition because the Y untyped fixed point operators does not exist for strongly typed logic.**[6] In this way, formal consistency of mathematics is preserved without giving up power because there do not seem to be any practical uses for the proposition *I'm unprovable.*

A procedure definition G could be attempted as follows: G[p] ≡ ⊬p. With strong types, the attempted definition becomes:

G[p:Proposition◁n:ℕ▷]:Proposition◁n+1▷ ≡ ⊬p

Consequently, there is no fixed point *I'm unprovable* for G such that

G[*I'm unprovable*] ⇔ *I'm unprovable*

Thus Gödel's proposition "*I'm unprovable*" does not exist in strongly-typed mathematics. Consequently, it became an open question to find an intuitive inferentially undecidable proposition.[7] Later in this article, it will be shown that [Church 1934] had a very important clue.

Mathematical Foundation for Computer Science

Computer Science brought different concerns and a new perspective to mathematical foundations including the following requirements:[8] [Arabic numeral superscripts refer to endnotes at the end of this article]

- provide powerful inference machinery so that arguments (proofs) can be short and understandable and all logical inferences can be formalized
- establish standard foundations so people can join forces and develop common techniques and technology
- incorporate axioms thought to be consistent by the overwhelming consensus of working professional mathematicians, e.g., natural numbers [Dedekind 1888], real numbers [Dedekind 1888], ordinals, sets of integers, reals, *etc*.
- facilitate inferences about the mathematical foundations used by computer systems.

Classical Direct Logic is a foundation of mathematics for Computer Science, which has a foundational theory (for convenience called "Mathematics") that can be used in any other theory. A bare turnstile is used for Mathematics so that $\vdash\Psi$ means that Ψ is a mathematical proposition that is a theorem of Mathematics and $\Phi \vdash \Psi$ means that Ψ can be inferred from Φ in Mathematics.

Mathematics self proves its own formal consistency

A mathematically significant idea involves:

> "…a very high degree of unexpectedness, combined with inevitability and economy."[9]

The following rules are fundamental to classical mathematics:

- *Proof by Contradiction*, *i.e.* $(\neg\Phi\Rightarrow(\Theta\wedge\neg\Theta))\vdash\Phi$, which says that a proposition can be proved by showing that it implies a contradiction.
- *Theorem Use* (a theorem can be used in a proof), *i.e.* $(\vdash\Phi)\vdash\Phi$[10]

Theorem: Mathematics self proves its own formal consistency.

Formal Proof By definition, $\neg$Consistent$\Leftrightarrow\exists[\Psi]\rightarrow\vdash(\Psi\wedge\neg\Psi)$.[11] By the rule of Existential Elimination, there is some proposition Ψ_0 such that $\neg$Consistent $\Rightarrow\vdash(\Psi_0\wedge\neg\Psi_0)$ which by *Theorem Use* and transitivity of implication means $\neg$Consistent$\Rightarrow(\Psi_0\wedge\neg\Psi_0)$. Substituting for Φ and Θ, in the rule for Proof by Contradiction, we have $(\neg$Consistent$\Rightarrow(\Psi_0\wedge\neg\Psi_0))\vdash$ Consistent. Thus, $\vdash$ Consistent.

A Natural Deduction[i] proof is given below:

1) $\neg$Consistent // hypothesis to derive a contradiction **just in this subargument**

2) $\exists[\Psi]\rightarrow\vdash(\Psi\wedge\neg\Psi)$ // definition of inconsistency using **1)**

3) $\vdash(\Psi_0\wedge\neg\Psi_0)$ // rule of Existential Elimination using **2)**

4) $\Psi_0\wedge\neg\Psi_0$ // rule of Soundness using **3)**

$\vdash$ Consistent // rule of Proof by Contradiction using **1)** and **4)**

Natural Deduction Proof of Formal Consistency of Mathematics

[i] [Jaśkowski 1934] developed Natural Deduction *cf.* [Barker-Plummer, Barwise, and Etchemendy 2011]

Please note the following points:

- The above argument formally mathematically proves that mathematics is formally consistent and that **it is not a premise of the theorem that mathematics is consistent.**[12]
- Classical mathematics was designed for consistent axioms and consequently the rules of classical mathematics can be used to prove formal consistency regardless of other axioms.[13]

The above proof means that “Mathematics is consistent” is a theorem in Classical Direct Logic. This means that the usefulness of Classical Direct Logic depends crucially on the consistency of Mathematics. Good evidence for the consistency of Mathematics comes from the way that Classical Direct Logic avoids the known paradoxes. Humans have spent millennia devising paradoxes.

Computer Science needs very strong foundations for mathematics so that computer systems are not handicapped. It is important not to have inconsistencies in mathematical foundations of Computer Science because they represent security vulnerabilities.

The recently developed self-proof of formal consistency (above) shows that the current common understanding that Gödel proved “Mathematics cannot prove its own consistency, if it is consistent” is inaccurate. Wittgenstein developed the proof below [lines 5) thru 7)][14] that contradiction in mathematics results from allowing the proposition *I'm unprovable.*[15] used in the incompleteness results of [Gödel 1931]:

1) *I'mUnprovable* $\Leftrightarrow \nvdash$ *I'mUnprovable* // Gödel's diagonal lemma
2) ($\vdash$*I'mUnprovable*) $\Longrightarrow$ ($\vdash\vdash$*I'mUnprovable*) // adequacy
3) ($\vdash$*I'mUnprovable*) $\Longrightarrow$ ($\vdash\nvdash$ *I'mUnprovable*) // Using 1)
4) $\vdash\nvdash$ *I'mUnprovable*[i] // Classical Proof by Contradiction using 2) and 3)
5) $\vdash$ ***I'mUnprovable*** // from 4) using 1)
6) $\vdash\vdash$ *I'mUnprovable* // from 5) using adequacy
7) $\vdash\neg$***I'mUnprovable*** // from 6) using 1)

Wittgenstein's proof above shows that if the proposition *I'm unprovable.* is allowed, then Mathematics is inconsistent because of proving lines 5) and 7).

Fortunately, using strongly typed propositions, it can be proved that the Gödel's proposition *I'm unprovable.* cannot be constructed because required Y untyped fixed points on propositions used in its construction do not exist.

[i] This is Gödel's argument formalized for the foundations of mathematics, à la Principia Mathematica

Consequently, using strong types, formal consistency of mathematics can be preserved without giving up power.

Gödel and other theoreticians developed the First-order Thesis that weakened the foundations of mathematics in order to allow the Y fixed point operator for propositions.[16] The weakened foundations (based on first-order logic) enabled some limited meta-mathematical theorems to be proved. However, as explained in this article, the weakened foundations are cumbersome, unnatural, and unsuitable as the mathematical foundation for Computer Science.

By the above formalized proof, Mathematics (⊢) proves its own formal consistency. However, the proof does so without regard to the content of Mathematics. For example, Mathematics includes the Dedekind categorical axiomatization of the natural numbers.

The proof formal consistency doesn't lead to any technical problems as long as there are no inconsistencies, *e.g.*,

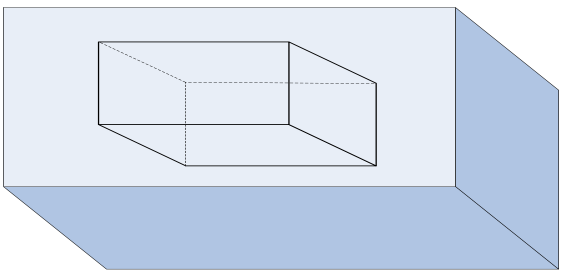
- Categorical axiomatizations of integers and ordinals don't infer any contradictions/
- Propositions in Mathematics cannot be constructed using the Y fixed point operator on propositions.

Monster-Barring

> "But why accept the counterexample? ... Why should the theorem give way...? It is the 'criticism' that should retreat.... It is a monster, a pathological case, not a counterexample."[17]

The Euler formula for polyhedra is Vertices-Edges+Faces=2, which can be proved in a variety of different ways.

But the hollow cube below is a counterexample because Vertices-Edges+Faces=4.



Counterexample to Euler's Formula

In the face of this counterexample, it becomes important to characterize polyhedra more rigorously. For example,

- A Regular solid
- A convex solid with polyhedral faces
- A surface consisting of a system of polygons
- *etc.*

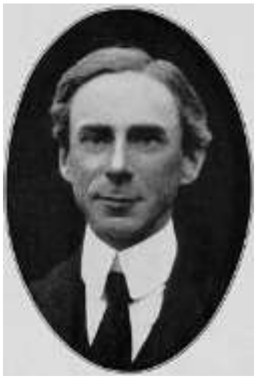
Lakatos has called this strategy "*monster-barring*."

contra Gödel *et. al*

> *"Men… think in herds …*
> *they only recover their senses slowly, and one by one."*
> Charles Mackay

That mathematics self proves its own formal consistency contradicts [Gödel 1931] that claim (using the proposition[i] *I'm unprovable.*) that mathematics cannot prove its own consistency.

However using strong types, Gödel's proposition cannot be constructed.[ii] Note that there is a crucial difference between how Russell used types and the method used in Direct Logic. Russell attempted to use types as the fundamental mechanism for preventing inconsistencies by restricting the domain of mathematics to objects that can be described by a strict hierarchical type system. However, he ran into trouble because his type mechanism was too strict and prevented ordinary mathematical reasoning.[iii]



Bertrand Russell

In this paper, strong types are used in the mathematical foundation of computer science to prevent the construction of paradoxical contradictory sentences and to provide the foundations for sets.

---

[i] constructed using what [Carnap 1934] later called the "Diagonal Lemma" which is equivalent to the Y untyped fixed point operator on propositions.

[ii] It is important that types for propositions do not place restrictions on types of other mathematical objects so that Y fixed points can be used elsewhere when they exist, for example in topology, etc..

[iii] In order to be able to carry out ordinary mathematical reasoning, Russell introduced an (unmotivated) patch called "ramified types" that collapsed the type hierarchy.

The difficulties encountered by Russell are avoided as follows:

- having the integers[18] and ordinals as primitives categorically axiomatized
- sets axiomatized as the characteristic functions of strongly typed functions

The above approach provides a very usable foundation for ordinary mathematical reasoning and for the mathematical foundations of computer science..

Computer Scientists did not initially set out to overthrow Gödel's approach to incompleteness. However over time, Gödel's proposition *I'm unprovable*. became an obstacle to the Computer Scientist's security requirements because its construction using Y fixed points on propositions introduces inconsistency into Mathematics. Fortunately, strong parameterized types enabled them to show that Gödel's sentence is neither valid nor needed in mathematics. However, once Gödel's sentence was thrown out, whether or not mathematics can prove its consistency once again became an open question. At this point, the Computer Scientists discovered a remarkably simple proof that mathematics does formally prove its formal consistency. So Computer Science is in relatively good shape and is coming to a consensus around using strong parameterized types for the mathematical foundations of Computer Science and also using them in the foundations of their programming languages.

So where does this leave some theoreticians who previously thought that they *owned* this subject matter because of their previous publications? There are at least two possibilities:

- The field splits and some theoreticians try to ignore the Computer Scientists' prestigious published volume and its reviews.
- Some younger logicians join with the Computer Scientists pushing forward with strong parameterized types in foundations. At this point, the above Computer Scientists are firmly ensconced in their field and have a head start in that they have held two international symposia at Stanford and published a well-regarded volume of articles on their results in what is arguably the most prestigious academic series of volumes in the area. But there is still much work to be done!

Classical Direct Logic

> "The point of foundations is not to arbitrarily restrict inquiry but to provide a framework wherein one can legitimately perform those constructions and operations that are mathematically interesting and useful."
> —Herrlich and Strecker [1973]

Direct Logic must meet the following challenges:

- *Consistent* to avoid security holes
- *Powerful* so that computer systems can formalize all logical inferences
- *Principled* so that it can be easily learned by software engineers
- *Coherent* so that it hangs together without a lot of edge cases
- *Intuitive* so that humans can follow computer system reasoning
- *Comprehensive* to accommodate all forms of logical argumentation
- *Inconsistency Robust* to be applicable to pervasively inconsistent theories of practice using
    - Inconsistency Robust Direct Logic for logical inference about inconsistent information
    - Classical Direct Logic for Mathematics used in inconsistency-robust theories[19]

In Direct Logic, unrestricted recursion is allowed in programs by using strongly typed recursive definitions.

There are uncountably many Actors.[20] For example, CreateReal.**[ ]** is a nondeterministic procedure that can return any real number[i] between 0 and 1 where[ii]

CreateReal.**[ ]** ≡ **[**(0 **either** 1), ∨**Postpone** CreateReal.**[ ]]**

where

- CreateReal.**[ ]** is the result of sending the Actor CreateReal the message **[ ]**
- (0 **either** 1) is the nondeterministic choice of 0 or 1,
- **[***first*, *rest***]** is the list that begins with *first* and whose remainder is *rest*, and
- **Postpone** *expression* delays execution of *expression* until the value is needed.

Each CreateReal.**[ ]** could be any one of uncountably many numbers between 0 and 1.

---

[i] using binary representation.

[ii] Typically, a result returned by the non-deterministic procedure Real is not computable in the sense there is no computable deterministic procedure that can compute its digits.

There are uncountably many propositions (because there is a different proposition for every real number). Consequently, there are propositions that are not the abstraction of any element of a denumerable set of sentences. For example, $p[x:\mathbb{R}] \equiv \lambda[y:\mathbb{R}] \to (y=x)$ defines a different predicate p[x] for each real number x, which holds for only one real number, namely x.[i]

It is important to distinguish between sentences, and propositions. Sentences[ii] (which without free variables) can be abstracted into propositions that can be asserted. Furthermore, terms[iii] can be abstracted into Actors (*i.e.* objects in mathematics).

Abstraction and parsing are becoming increasingly important in software engineering. *e.g.,*

- The execution of code can be dynamically checked against its documentation. Also Web Services can be dynamically searched for and invoked on the basis of their documentation.
- Use cases can be inferred by specialization of documentation and from code by automatic test generators and by model checking.
- Code can be generated by inference from documentation and by generalization from use cases.

Abstraction and parsing are needed for large software systems so that that documentation, use cases, and code can mutually speak about what has been said and their relationships.

For example:

> Propositions
>
> *e.g.* $\forall[n:\mathbb{N}]\to \exists[m:\mathbb{N}]\to m>n$
> *i.e.,* for every $\mathbb{N}$ there is a larger $\mathbb{N}$

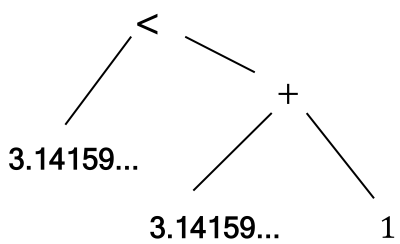
> Sentences
>
> *e.g.* $⦅\forall[n:\mathbb{N}]\to ⦅\exists[m:\mathbb{N}]\to ⦅m>n⦆⦆⦆$
> *i.e.,* the sentence that for every $\mathbb{N}$ there is a larger $\mathbb{N}$

---

[i] For example (p[3])[y] holds if and only if y=3.
[ii] which are grammar tree structures
[iii] which are grammar tree structures

In Direct Logic, a sentence is a grammar tree (analogous to the ones used by linguists). Such a grammar tree has terminals that can be constants. And there are uncountably many constants, *e.g.*, the real numbers:

> Of course, because the digits of 3.14159... are computable, there is a expression_1 such that $\lfloor \text{expression}_1 \rfloor$ = 3.14159... that can be used to create the sentence ⦅expression_1 < ⦅expression_1 + 1⦆⦆.
>
> However the sentence ⦅expression_1 < ⦅expression_1 + 1⦆⦆ is not the same as ⦅3.14159... < ⦅3.14159... + 1⦆⦆ because it does not have the same vocabulary and it is a much larger sentence that has many terminals whereas ⦅3.14159... < ⦅3.14159... + 1⦆⦆ has just 3 terminals:



Consequently, sentences *cannot* be enumerated.[21]

Note: Types in Classical Direct Logic are much stronger than those in constructive logic[22] because Classical Direct Logic has all of the power of Classical Mathematics.

Mathematics self proves that it is open

> "Although everything mathematical is formalizable, it is nonetheless impossible to formalize all of mathematics in a *single* formal system."
> [Gödel 1935]

Mathematics proves that it is open in the sense that it can prove that its proofs cannot be provably computationally enumerated:[23]

Theorem ⊢Mathematics is Open, i.e.,

⊢¬ProofsComputationalyEnumerable

Proof.[i]

Suppose to obtain a contradiction that it is possible to prove closure, *i.e.*,

⊢ProofsComputationalyEnumerable. Then there is a provably computable total procedure ProofsEnumerator:[ℕ]↦Proof such that it is provable that

∀[p:Proof]→ ∃[i:ℕ]→ProofsEnumerator▪[i]= p

∀[i:ℕ]→ProofsEnumerator▪[i]:Proof

A subset of the proofs are those proving that certain procedures [ℕ]↦ℕ are total. Consequently, there is a ProvedTotalsEnumerator:[ℕ]↦([ℕ]↦ℕ) that enumerates the provably total computable procedures [ℕ]↦ℕ that can be used in the implementation of the following procedure:

Diagonal▪[i:ℕ]:ℕ ≡ 1+ (ProvedTotalsEnumerator▪[i]) ▪[i]

Consequently:

- Diagonal *is* a proved total procedure because it is implemented using computable proved total procedures.
- Diagonal *is not* a proved total procedure because it differs from every other computable proved total procedure.

The above contradiction completes the proof.

[Franzén 2004] argued that mathematics is inexhaustible because of inferential undecidability[ii] of mathematical theories. The above theorem that mathematics is open provides another independent argument for the inexhaustibility of mathematics.

[i] This argument appeared in [Church 1934] expressing concern that the argument meant that there is "*no sound basis for supposing that there is such a thing as logic*."

[ii] See section immediately below.

Mathematical provability is computationally undecidable

Mathematics proves that provability in mathematics is computationally undecidable:[24]

Theorem ⊢Mathematical provability is computationally undecidable, *i.e.*, ⊢¬ProvablityComputationalyDecidable

Proof. If provability in Mathematics[25] were computationally decidable then, the halting problem would be computationally decidable.

Completeness of inference versus inferential undecidability of closed mathematical theories

A closed mathematical theory is an extension of mathematics whose proofs are computationally enumerable. For example, group theory is obtained by adding the axioms of groups to Classical Direct Logic. Similarly, the closed mathematical theory *Nat* has the usual Dedekind axioms including the induction axiom: For each order:$\mathbb{N}_+$ and P:**Proposition**◁order▷$^{\mathbb{N}}$ the following Dedekind induction axiom holds:

$$(P[\![0]\!] \wedge \forall[i:\mathbb{N}]\rightarrow P[\![i]\!] \Rightarrow P[\![i+1]\!]) \Rightarrow \forall[i:\mathbb{N}]\rightarrow P[\![i]\!]$$

Theorem The proposition $\text{ProofsComputationalyEnumerable}_{Nat}$ is true but unprovable in *Nat*; *i.e.*, both of the following hold:

- $\vDash_{\mathbb{N}}$ $\text{ProofsComputationalyEnumerable}_{Nat}$
- $\nvdash_{Nat}$ $\text{ProofsComputationalyEnumerable}_{Nat}$

Proof:

Mathematics can make use of its types (*i.e.*, $\textbf{Proposition}_{Nat}$◁ω▷[26], $\textbf{Sentence}_{Nat}$◁ω▷[27] and $\textbf{Domain}_{Nat}$◁ω▷[28]) in the enumeration of proofs of *Nat* without violating the type rules of Mathematics. Consequently,

$$\vDash_{\mathbb{N}} \text{ProofsComputationalyEnumerable}_{Nat}$$

Suppose to obtain a contradiction that

$$\vdash_{Nat} \text{ProofsComputationalyEnumerable}_{Nat}$$

Then there is a provable in *Nat* computable total procedure $\text{ProofsEnumerator}_{Nat}$:[ℕ]↦$\textbf{Proof}_{Nat}$ such that it is provable in *Nat* that

$$\forall[p:\textbf{Proof}_{Nat}]\rightarrow \exists[i:\mathbb{N}]\rightarrow \text{ProofsEnumerator}_{Nat}\blacksquare[i]= p$$

$$\forall[i:\mathbb{N}]\rightarrow \text{ProofsEnumerator}_{Nat}\blacksquare[i]:\textbf{Proof}_{Nat}$$

A subset of the proofs in *Nat* are those proving that certain procedures [ℕ]↦ℕ are total. Consequently, there is a procedure

$\text{ProvedTotalsEnumerator}_{Nat}$:[ℕ]↦([ℕ]↦ℕ)

that enumerates the provable in *Nat* total computable procedures [ℕ]↦ℕ that can be used in the implementation of the following procedure:

$$\text{Diagonal}\blacksquare[i:\mathbb{N}]:\mathbb{N} \equiv 1+(\text{ProvedTotalsEnumerator}_{Nat}\blacksquare[i])\blacksquare[i]$$

Consequently:

- Diagonal *is* a provable in $\mathcal{Nat}$ total procedure because it is implemented using computable provable in $\mathcal{Nat}$ total procedures.
- Diagonal *is not* a provable in $\mathcal{Nat}$ total procedure because it differs from every other computable provable in $\mathcal{Nat}$ total procedure.

The above contradiction completes the proof.

Note that the closed mathematical theory $\mathcal{Nat}$ is inferentially undecidable[29] with respect to $\text{ProofsComputationalyEnumerable}_{\mathcal{Nat}}$ does not mean Mathematical *incompleteness* with respect to the information that can be inferred about theory $\mathcal{Nat}$ because

- $\vDash_{\mathbb{N}} \text{ProofsComputationalyEnumerable}_{\mathcal{Nat}}$
- $\nvdash_{\mathcal{Nat}} \text{ProofsComputationalyEnumerable}_{\mathcal{Nat}}$

Theorem[30] $\vdash_{\mathcal{Nat}} \text{Consistent}_{\mathcal{Nat}}$, i.e., $\mathcal{Nat}$ proves its formal consistency

Proof by contradiction in $\mathcal{Nat}$: Suppose to derive an inconsistency that $\neg\text{Consistent}_{\mathcal{Nat}}$. By the definition of inconsistency for $\mathcal{Nat}$, there is some proposition Ψ of the theory $\mathcal{Nat}$ such that $\vdash_{\mathcal{Nat}} (\Psi \wedge \neg\Psi)$. By Existential Elimination, there is some proposition Ψ_0 such that $\vdash_{\mathcal{Nat}}(\Psi_0 \wedge \neg\Psi_0)$ which theorem can be used to infer in $\mathcal{Nat}$ that $(\Psi_0 \wedge \neg\Psi_0)$. The above contradiction completes the proof.

Theorem. $\mathcal{Sets}_\tau$ is categorical via a (unique) isomorphism with the unique model **Sets**◁τ▷.

Theorem[31] The proposition $\text{ProofsComputationalyEnumerable}_{\text{Sets}_\tau}$ is true but unprovable in $\mathcal{Sets}_\tau$, that is, both of the following hold:

- $\vDash_{\textbf{Sets}\triangleleft\tau\triangleright}\text{ProofsComputationalyEnumerable}_{\text{Sets}_\tau}$
- $\nvdash_{\text{Sets}_\tau} \text{ProofsComputationalyEnumerable}_{\text{Sets}_\tau}$

Proof:

Mathematics can make use of the types definable in $\mathcal{Sets}_\tau$ in the enumeration of proofs of $\mathcal{Sets}_\tau$ without violating the type rules of Mathematics. Consequently,

$$\vDash_{\mathbb{N}} \text{ProofsComputationalyEnumerable}_{\text{Sets}_\tau}$$

The proof of $\nvdash_{\text{Sets}_\tau} \text{ProofsComputationalyEnumerable}_{\text{Sets}_\tau}$ is the same as the corresponding proof for $\mathcal{Nat}$.

Theorem[32] $\vdash_{\mathrm{Sets}_\tau} \mathrm{Consistent}_{\mathrm{Sets}_\tau}$, that is, $\mathcal{Sets}_\tau$ proves its formal consistency

Information Invariance[33] is a *fundamental* technical goal of logic consisting of the following:[34]

1. *Soundness of inference:* information is not increased by inference[35]
2. *Completeness of inference:* all information that necessarily holds can be inferred

Classical Provability Direct Logic

Classical Provability Direct Logic is an improvement over Provability Logic [Verbrugge 2010] as follows:

- Propositions are directly represented as opposed to being represented as Gödel numbers.
- Theorems can be used directly in proofs, that is $(\vdash\Phi)\Rightarrow\Phi$.
- Adequacy is directly expressed as follows: $(\Psi\vdash\Phi) \Leftrightarrow \vdash(\Psi\vdash\Phi)$.
- Deduction is directly expressed as follows: $(\Psi\vdash\Phi) \Leftrightarrow \vdash(\Psi\Rightarrow\Phi)$.
- $\vDash$ is directly expressed with the following:
  - $(\vdash\Phi) \Rightarrow (\vDash\Phi)$
  - $(\vDash\Psi)\wedge(\Psi\Rightarrow\Phi) \Rightarrow (\vDash\Phi)$
- A goal ($\Vdash$) is directly expressed with the following:
  - Forward: $(\Phi\Rightarrow\Psi)\wedge(\Vdash\Phi) \Rightarrow (\Vdash \Psi)$
  - Backward: $(\Phi\Rightarrow\Psi)\wedge(\Vdash\Psi) \Rightarrow (\Vdash \Phi)$

Overview

Contradiction	Outcome
Church discovered to his dismay that if theorems of mathematics are postulated to be computationally enumerable, then mathematics is inconsistent.	Proofs of mathematics cannot be computationally enumerated and mathematics is open and inexhaustible. But theorems of a particular theory can be postulated to be computationally enumerable.
Using the proposition *I'm unprovable.*, [Gödel 1931] claimed that mathematics cannot prove its own consistency. However, it is pointed out in this paper that mathematics easily proves its own formal consistency.	The contradiction can be resolved by using strong types for mathematics so that Gödel's proposition does not exist.
Using the Y untyped fixed point operator to construct the proposition *I'm unprovable.*, [Gödel 1931] claimed to prove inferential undecidability (sometimes called "incompleteness") for mathematics. However, Wittgenstein showed that Gödel's proposition leads to inconsistency in mathematics.	In Classical Direct Logic, the following sentence is true but unprovable in $\mathcal{Nat}$ assuming that $\mathcal{Nat}$ is consistent: $\text{ProofsComputationalyEnumerable}_{\mathcal{Nat}}$
In Computer Science, it is important that the Natural Numbers ($\mathcal{Nat}$) be axiomatized in a way that does not allow integers (e.g. infinite ones) in models of the axioms. However, it is impossible to properly axiomatize $\mathcal{Nat}$ using first-order logic.	Using Classical Direct Logic, $\mathcal{Nat}$ can be axiomatized in such a way that all models are uniquely isomorphic to $\mathcal{Nat}$. Consequently, there are no infinite integers in models of the axioms.
First-order logic is unsuitable as the foundation of mathematics for Computer Science: • Some theorems of ordinary mathematics cannot be proved. • Some ordinary theorems useful in Computer Science cannot be proved. • There are undesirable models of mathematical theories (see above).	Classical Direct Logic is suitable as the foundation of mathematics for Computer Science: • All ordinary theorems of mathematics can be proved. • All ordinary theorems useful in Computer Science can be proved • There are no undesirable models of mathematical theories.

Conclusion

"The problem is that today some knowledge still feels too dangerous because our times are not so different to Cantor or Boltzmann or Gödel's time. We too feel things we thought were solid being challenged; feel our certainties slipping away. And so, as then, we still desperately want to cling go a belief in certainty. It makes us feel safe. ...

Are we grown up enough to live with uncertainties or will we repeat the mistakes of the twentieth century and pledge blind allegiance to another certainty."
Malone [2007]

"The world always needs heretics to challenge the prevailing orthodoxies. We are lucky that we can be heretics today without any danger of being burned at the stake. But unfortunately I am an old heretic. Old heretics do not cut much ice. When you hear an old heretic talking, you always say, 'Too bad he has lost his marbles.'

What the world needs is young heretics. I am hoping that one or two of you people in the audience may fill that role."
Dyson [2005]

The closed mathematical theory $\mathcal{Nat}$ has the usual Dedekind axioms including the induction axiom. Consequently, the proposition $\text{ProofsComputationalyEnumerable}_{\mathcal{Nat}}$ is true but unprovable in $\mathcal{Nat}$.

Information Invariance is a fundamental technical goal of logic consisting of the following:

1. *Soundness of inference*: information is not increased by inference
2. *Completeness of inference*: all information that necessarily holds can be inferred.

That the closed mathematical theory $\mathcal{Sets}_\tau$ is inferentially undecidable[i] with respect to $\text{ProofsComputationalyEnumerable}_{\mathcal{Nat}}$ does not mean incompleteness with respect to the information that can be inferred because

- $\vdash \nvdash_{\text{Sets}_\tau} \text{ProofsComputationalyEnumerable}_{\text{Sets}_\tau}$
- $\vdash \nvdash_{\text{Sets}_\tau} \neg\text{ProofsComputationalyEnumerable}_{\text{Sets}_\tau}$
- $\vdash \vDash_{\text{Sets}\triangleleft\tau\triangleright} \text{ProofsComputationalyEnumerable}_{\text{Sets}_\tau}$

Computer Science needs a rigorous foundation for all of mathematics that enables computers to carry out all reasoning without human intervention.[36] [Frege 1879] was a good start, but it foundered on the issue of consistency.

[i] sometimes called "incomplete"

[Russell 1925] attempted basing foundations entirely on types, but foundered on the issue of being expressive enough to carry to some common mathematical reasoning. [Church 1932, 1933] attempted basing foundations entirely on untyped higher-order functions, but foundered because it was shown to be inconsistent [Kleene and Rosser 1935]. Presently, Isabelle [Paulson 1989] and Coq [Coquand and Huet 1986] are founded on types and do not allow theories to reason about themselves. Classical Direct Logic is a foundation for all of mathematical reasoning based on strong types (to provide grounding for concepts) that allows general inference about reasoning.

[Gödel 1931] claimed inferential undecidability[i] results for mathematics using the proposition *I'm unprovable.* In opposition to Wittgenstein's correct argument his proposition leads to contradictions in mathematics, Gödel later claimed that his results were for a cut-down first-order theory of natural numbers. However, first-order logic is not a suitable foundation for Computer Science because of the requirement that computer systems be able to carry out all reasoning without requiring human intervention (including reasoning about their own inference systems).

[i] sometimes called "incompleteness"

Following [Frege 1879, Russell 1925, and Church 1932-1933], Direct Logic was developed and then investigated propositions with the following results.

- Formalization of Wittgenstein's proof that Gödel's proposition *I'm unprovable.* leads to contradiction in mathematics. So the consistency of mathematics had to be rescued against Gödel's proposition constructed using what [Carnap 1934] later called the "Diagonal Lemma" which is equivalent to the Y untyped fixed point operator on propositions. Use of the Y untyped fixed point operator on propositions in results of [Curry 1941] and [Löb 1955] also lead to inconsistency in mathematics. Consequently, mathematics had to be rescued against these uses of the Y untyped fixed point operator for propositions.
- Self-proof of the formal consistency of mathematics. Consequently, mathematics had to be rescued against the claim [Gödel 1931] that mathematics cannot prove its own consistency. Also, it became an open problem whether mathematics proves its own formal consistency, which was resolved by the author discovering an amazing simple proof.[37] A solution is to require strongly typed mathematics to bar use of the Y untyped fixed point operator for propositions.[38] However, some theoreticians have very reluctant to accept the solution.

  According to [Dawson 2006]:[39]
    - *Gödel's results altered the mathematical landscape, but they did **not** "produce a debacle".*
    - *There is **less** controversy today over mathematical foundations than there was **before** Gödel's work.*

  However, Gödel's writings have produced a controversy of a very different kind from the one discussed by Dawson:
    - Gödel's claim that mathematics cannot prove its own consistency[i] has been disproved.
    - Consequently, Gödel's writings have led to increased controversy over mathematical foundations.

The development of Direct Logic has strengthened the position of working mathematicians as follows:[ii]

- Allowing freedom from the philosophical dogma of the First-Order Thesis
- Providing usable strong types for all of Mathematics that provides theories that have categorical models
- Allowing theories to freely reason about theories
- Providing Inconsistency Robust Direct Logic for safely reasoning about theories of practice that are (of necessity) pervasively inconsistent.

---

[i] Gödel's writing was accepted doctrine by some theoreticians for over eight decades.
[ii] Of course, Direct Logic must preserve as much previous learning as possible.

Sociology of Foundations

> "Faced with the choice between changing one's mind and proving that there is no need to do so, almost everyone gets busy on the proof." John Kenneth Galbraith [1971 pg. 50]

> "Max Planck, surveying his own career in his Scientific Autobiography [Planck 1949], sadly remarked that 'a new scientific truth does not triumph by convincing its opponents and making them see the light, but rather because its opponents eventually die, and a new generation grows up that is familiar with it.' "
> [Kuhn 1962]

The inherently social nature of the processes by which principles and propositions in logic are produced, disseminated, and established is illustrated by the following issues with examples:[40]

- **The formal presentation of a demonstration (proof) has not lead automatically to consensus.** Formal presentation in print and at several different professional meetings of the extraordinarily simple proof in this paper have not lead automatically to consensus about the theorem that "Mathematics is Consistent". New results can sound crazy to those steeped in conventional thinking. Paradigm shifts often happen because conventional thought is making assumptions taken as dogma. As computer science continues to advance, such assumptions can get in the way and have to be discarded.
- **There has been an absence of universally recognized central logical principles**. Disputes over the validity of the Principle of Excluded Middle led to the development of Intuitionistic Logic.
- **There are many ways of doing logic.** One view of logic is that it is about *truth*; another view is that it is about *argumentation* (i.e. proofs).[41]
- **Argumentation and propositions have be variously (re-)connected and both have been re-used.** Church's paradox is that assuming theorems of mathematics are computationally enumerable leads to contradiction. In this papers, the paradox is transformed into the fundamental principle that "Mathematics is Open" (*i.e.* it is a theorem of mathematics that the proofs of mathematics are not computationally enumerable) using the argument used in Church's paradox.
- **New technological developments have cast doubts on traditional logical principles.** The pervasive inconsistency of modern large-scale information systems has cast doubt on some logical principles, *e.g.*, Excluded Middle.[42]
- **Political actions have been taken against views differing from the establishment theoreticians**. According to [Kline 1990, p. 32],

Hippasus was literally thrown overboard by his fellow Pythagoreans "*…for having produced an element in the universe which denied the…doctrine that all phenomena in the universe can be reduced to whole numbers and their ratios*." Fearing that he was dying and the influence that Brouwer might have after his death, Hilbert fired[43] Brouwer as an associate editor of *Mathematische Annalen* because of "*incompatibility of our views on fundamental matters*"[44] *e.g.,* Hilbert ridiculed Brouwer for challenging the validity of the Principle of Excluded Middle. Gödel's original results were for Principia Mathematica (and not first-order logic) as the foundation for the mathematics of its time including the Dedekind axiomatization of the natural numbers. In face of Wittgenstein's devastating criticism, Gödel insinuated[45] that he was crazy and retreated to first-order logic in an attempt to salvage his results. Some theoreticians turned first-order logic into a philosophical dogma in part it facilitated their careers. Since theoreticians couldn't prove anything significant about practical mathematical theories, they cut them down to unrealistic first-order theories where results could be proved (*e.g.* compactness) that did not hold for practical mathematical theories. In the famous words of Upton Sinclair:

> *"It is difficult to get a man to understand something,*
> *when his salary depends on his not understanding it."*

Some theoreticians have ridiculed dissenting views and attempted to limit their distribution by political means.[46]

Acknowledgments

> "All truth passes through three stages:
> First, it is ridiculed.
> Second, it is violently opposed.
> Third, it is accepted as being self-evident."[47]

Extensive discussions with Tom Costello, Eric Kao, Ron van der Meyden and other members of the Stanford CS Logic Group helped the development of this paper. Tom suggested that more conventional terminology be used in the formal proof of consistency of mathematics. Martin Davis kindly provide the reference for [Gödel 1933]. Comments by James Lottes, Pat Suppes, Daniel Raggi, Eric Winsberg, John Woods, and Ming Xiong helped improve the presentation. Dan Flickinger suggested including an overview table. Alan Bundy pointed out many crucial places where the presentation needed improvement. John Woods served ably as the senior referee by compiling an excellent synopsis of anonymized conference referee reports for this article. Discussions with Michael Beeson helped improve the section on how mathematics self-proves its own formal consistency. Correspondence with Monroe Eskew helped clarify the relationship of Classical Direct Logic with

first-order logic. Also, Monroe suggested looking at the Berry Paradox.[i] Correspondence with Jack Copeland helped clarify the relationship of the work reported in this article with previous work by Gödel *et. al.* Also, Jack suggested inclusion of the closely related natural deduction proof that mathematics proves its own formal consistency in addition to the linear proof. Discussions on the FriAM electronic mailing list were very helpful in improving this article. J.J. Meyer participated in Inconsistency Robustness 2011 and subsequently wrote an excellent review of the resulting book. Conversations with Dana Scott were very helpful in clarifying the discussion of the issue of the provability of the formal consistency of Mathematics. Natarajan Shankar was extremely helpful in clarifying how Gödel's proposition *I'm unprovable.* leads to inconsistency in Mathematics.

Bibliography

Anthony Anderson and Michael Zelëny (editors). *Logic, Meaning and Computation: Essays in Memory of Alonzo Church* Springer. 2002.

Jeremy Avigad and John Harrison. *Formally Verified Mathematics*. CACM. April 2014.

Steve Awodey and Erich Reck. *Completeness and Categoricity. Parts I and II: Nineteenth-century Axiomatics to Twentieth-century Metalogic*. History and Philosophy of Logic. Vol. 23. 2002.

Steve Awodey, Álvaro Pelayo, and Michael A. Warren. *Voevodsky's Univalence Axiom in Homotopy Type Theory* Notices of AMS. October 2013.

Alan Anderson. *Mathematics and the "Language Game"* (1958) in Philosophy of Mathematics Selected Readings. Prentice Hall. 1964.

David Barker-Plummer, Jon Barwise and John Etchemendy. *Language, Proof, and Logic: Second Edition* Stanford Center for the Study of Language and Information. 2011.

Jon Barwise. *Model-Theoretic Logics: Background and Aims* in "Model Theoretic Logics" Springer-Verlag. 1985.

Hamid Berenji, *et. al. A reply to the paradoxical success of Fuzzy Logic* AI Magazine. Spring 1994.

Francesco Berto. *The Gödel Paradox and Wittgenstein's Reasons* Philosophia Mathematica. February, 2009.

Errett Bishop. *The Crisis in Contemporary Mathematics* Historia Mathematica. 2. 1975.

George Boolos. *On second-order logic* Journal of Philosophy. Vol. 72. 1975.

Nicolas Bourbaki. *Univers*. In Séminaire de Géométrie Algébrique du Bois Marie – 1963-64 – Théorie des topos et cohomologie étale des schémas – (SGA 4) – vol. 1. Springer-Verlag. 1972

Andrey Bovykin. *Brief introduction to unprovability.* Logic Colloquium 2006. Lecture Notes in Logic 2009

[i] Please see section on the Berry Paradox in the historical appendix.

Bernd Buldt. *The Scope of Gödel's First Incompleteness Theorem.* Logica Universalis. Volume 8. December 2014.

Andrea Cantini, *Paradoxes and Contemporary Logic*, Stanford Encyclopedia of Philosophy. Winter 2012 Edition.

Rudolph Carnap. Logische Syntax der Sprache. (*The Logical Syntax* of Language Open Court Publishing 2003) 1934.

Walter Carnielli and João Marcos. *Ex Contradictione Non Sequitur Quodlibet* Bulletin of Advanced Reasoning and Knowledge. 2001.

Hasok Chang. *Inventing Temperature: Measurement and Scientific Progress* Oxford University Press. 2007.

Aziem Chawdhary, Byron Cook, Sumit Gulwani, Mooly Sagiv, and Hongseok Yang *Ranking Abstractions* ESOP'08.

Gregory Chaitin. Interview in "Dangerous Knowledge" BBC4 documentary. 2007.

Gregory Chaitin. *Thinking about Godel and Turing: Essays on Complexity 1970-2007.* World Scientific Publishing. 2007.

Alonzo Church. *A Set of postulates for the foundation of logic (1)* Annals of Mathematics. Vol. 33, 1932.

Alonzo Church. *A Set of postulates for the foundation of logic (2*) Annals of Mathematics. Vol. 34, 1933.

Alonzo Church. *The Richard Paradox.* Proceedings of American Mathematical Society. 1934.

Alonzo Church. *An unsolvable problem of elementary number theory* Bulletin of the American Mathematical Society 19, May, 1935. American Journal of Mathematics, 58 (1936),

Alonzo Church: *A Formulation of the Simple Theory of Types*, Journal of Symbolic Logic. vol. 5. 1940.

Alonzo Church *The Calculi of Lambda-Conversion* Princeton University Press. 1941.

Alonzo Church. *Introduction to Mathematical Logic* Princeton University Press. 1956.

Paul Cohen. *The Independence of the Continuum Hypothesis I and II* Proceedings of the National Academy of Sciences of the United States of America, Vol. 50, No. 6. Dec. 15, 1963.

John Corcoran. *Second-order Logic.* Logic, Meaning and Computation. Kluwer. 2001.

Thierry Coquand and Gérard Huet: *The calculus of constructions.* Technical Report 530, INRIA, Centre de Rocquencourt, 1986.John Corcoran. *Gaps between logical theory and mathematical practice* in The methodological unity of science. 1973.John Corcoran. *Categoricity.* History and Philosophy of Logic. Vol. 1. 1980

Will Clinger. *Foundations of Actor Semantics* MIT Mathematics Doctoral Dissertation. June 1981.

Jack Copeland. *The Essential Turing* Oxford University Press. 2004.

Haskell Curry "Some Aspects of the Problem of Mathematical Rigor" *Bulletin of the American Mathematical Society* Vol. 4. 1941.

Reuben Hersh (editor). *18 Unconventional Essays on the Nature of Mathematics* Springer. 2005

John Dawson. *Shaken Foundations or Groundbreaking Realignment? A Centennial Assessment of Kurt Gödel's Impact on Logic, Mathematics, and Computer Science* FLOC'06.

Richard Dedekind (1888) *What are and what should the numbers be?* (Translation in From Kant to Hilbert: A Source Book in the Foundations of Mathematics. Oxford University Press. 1996) Braunschweig.

Cora Diamond. *Wittgenstein's Lectures on the Foundations of Mathematics, Cambridge, 1939* Cornell University Press. 1967.

Heinz-Dieter Ebbinghaus. *Ernst Zermelo: An Approach to His Life and Work* Springer. 2007.

B. Fantechi, A. Vistoli, L. Gottsche, S. L. Kleiman, L. Illusie, and N. Nitsure. *Fundamental algebraic geometry: Grothendieck's FGA explained.* Mathematical Surveys and Monographs. Vol. 123. American Mathematical Society. 2005.

Colin Farquhar, Gudmund Grov, Andrew Cropper, Stephen Muggleton and Alan Bundy. *Typed meta-interpretive learning for proof* CEUR Workshop. ILP. 2015.

Solomon Feferman. *Arithmetization of Metamathematics in a General Setting* Fundamenta Mathematicae. 49. 1960.

Solomon Feferman. *Finitary Inductively Presented Logics* Logic Colloquium '88. North-Holland, 1989.

Solomon Feferman "Axioms for determinateness and truth" Review of Symbolic Logic. 2008.

Freeman Dyson. *Heretical Thoughts about Science and Society* Boston University. November 1, 2005.

T. S. Eliot. *Four Quartets.* Harcourt. 1943.

Solomon Feferman, Solomon (1984). *Toward Useful Type-Free Theories, I*. "Journal of Symbolic Logic. 1984.

Juliet Floyd and Hilary Putnam *A note on Wittgenstein's "Notorious Paragraph" about the Gödel's Theorem* Journal of Philosophy. 2000.

Torkel Franzén. *Inexhaustibility* AK Peters. 2004

Gottlob Frege. *Begriffsschrift: eine der arithmetischen nachgebildete Formelsprache des reinen Denkens* Halle, 1879.

Gerhard Gentzen. *Investigations into Logical Deduction.* 1934-1935 in *The collected works of Gerhard Gentzen*. edited by Szabo. North-Holland, 1969.

Gerhard Gentzen. *The Consistency of Arithmetic.* 1936 in The collected works of Gerhard Gentzen. edited by Szabo. North-Holland, 1969.

Kurt Gödel. *Über die Vollständigkeit des Logikkalküls*. Doctoral dissertation. University Of Vienna. 1929.

Kurt Gödel. *Letter to Karl Menger.* April 20, 1972. in Wang [1974] pg. 179.

Kurt Gödel. *Die Vollständigkeit der Axiome des logischen Functionenkalküls,* Monatshefte für Mathematik und Physik 1930.Kurt Gödel (1931) *On formally undecidable propositions of Principia*

Mathematica in "*A Source Book in Mathematical Logic, 1879-1931*" Translated by Jean van Heijenoort. Harvard Univ. Press. 1967.

Kurt Gödel (1933) *Kurt Gödel Collected Works, Vol. III.* Oxford University Press. Page 47.

Kurt Gödel. *Review of Logical Syntax of Language.* 1935.

Godfrey Hardy, *A Mathematician's Apology* Cambridge: University Press. 1940.

Reuben Goodstein. *On the restricted ordinal theorem* Journal of Symbolic Logic. 1944.

Mandy Hale. *The Single Woman: Life, Love, and a Dash of Sass*. Thomas Nelson. 2013.

Horst Herrlich and George Strecker. *Category theory* Allyn and Bacon. 1973.

Carl Hewitt, Peter Bishop and Richard Steiger. *A Universal Modular Actor Formalism for Artificial Intelligence* IJCAI-1973.

Carl Hewitt *Inconsistency Robustness for Logic Programs* in Inconsistency Robustness. College Publications. 2015.

Carl Hewitt *Formalizing common sense for inconsistency-robust information integration using Direct Logic™ Reasoning and the Actor Model* Inconsistency Robustness 2010.

Carl Hewitt *ActorScript™ extension of C#™, Java™ , JavaScript™ and Objective C™* in Inconsistency Robustness. College Publications. 2015.

Carl Hewitt *Actor Model of Computation: Many-core Inconsistency-robust Information Integration* Inconsistency Robustness 2011.

Carl Hewitt. *What is computation?* A Computable Universe: Understanding Computation & Exploring Nature as Computation. Edited by Hector Zenil. World Scientific Publishing Company. 2012.

David Hilbert *Mathematical Problems* Lecture delivered before the International Congress of Mathematicians at Paris in 1900. Translation by Newson: http://aleph0.clarku.edu/~djoyce/hilbert/problems.html

David Hilbert. *Neubegründung der Mathematik: Erste Mitteilung* Abhandlungen aus dem Seminar der Hamburgischen Universität. 1. 1922. English translation in *From Brouwer to Hilbert. The Debate on the Foundations of Mathematics in the 1920s* Oxford University Press. 1998.

Immaculee Ilibagiza. *Left to Tell: Discovering God Amidst the Rwandan Holocaust* Hay House. 2014.

Daniel Isaacson. *The reality of mathematics and the case of set theory.* Truth, Reference, and Realism. Central European University Press. 2008.

Stanisław Jaśkowski *On the Rules of Suppositions in Formal* "Logic Studia Logica" 1, 1934. (reprinted in: *Polish logic 1920-1939*, Oxford University Press, 1967.

Stephen Kleene and John Barkley Rosser *The inconsistency of certain formal logics* Annals of Mathematics Vol. 36. 1935.

Stephen Kleene *General recursive functions and natural numbers* Mathematical Annuals. 1936.

Stephen Kleene *On the Notation of Ordinal Numbers* Journal of Symbolic Logic 1938.

Stephen Kleene. *Introduction to Metamathematics.* North Holland. 1952.
Morris Kline. *Mathematical Thought from Ancient to Modern Times* Oxford University Press, 1990.
Rob Knies. *Terminator Tackles an Impossible Task* Microsoft Research. Aug. 6. 2006
Robert Kowalski *The Early Years of Logic Programming* CACM. January 1988.
Thomas Kuhn. The Structure of Scientific Revolutions University of Chicago Press. 1962.
Imre Lakatos. *Proofs and Refutations* Cambridge University Press. 1976.
John Law. *After Method: mess in social science research* Routledge. 2004.
Martin Löb. *Solution of a problem of Leon Henkin*. Journal of Symbolic Logic. Vol. 20. 1955.
J.P. Mayberry. *Review of J. L. Bell: A primer of infinitesimal analysis. British Journal of Philosophy of Science*. Vol 51. 2000.
David Malone. *Dangerous Knowledge* BBC4 documentary. 2007. http://www.dailymotion.com/playlist/x1cbyd_xSilverPhinx_bbc-dangerous-knowledge/1
Donald Martin. *Completeness or Incompleteness of Basic Mathematical Concepts* EFI Workshop Papers. revised October, 20, 2015.
George R. R. Martin. *A Game of Thrones* Bantam. 2013.
Colin McLarty. *What Does it Take to Prove Fermat's Last Theorem? Grothendieck and the Logic of Number Theory* Journal of Symbolic Logic. September 2010.
John-Jules Meyer. *Review of Inconsistency Robustness.* Amazon. January, 2016.
Mizar. Mizar Home Page. http://mizar.org
Ray Monk. *Bourgeois, Boshevist or anarchist? The Reception of Wittgenstein's Philosophy of Mathematics* in Wittgenstein and his interpreters. Blackwell. 2007.
Adam Naumowicz. *A Brief Overview of Mizar. Institute of Informatics*, University of Bialystok, Poland. August 11, 2009.
Adam Naumowicz and Artur Korniłowicz *A Brief Overview of Mizar. Theorem Proving in Higher Order Logics.* Lecture Notes in Computer Science 5674. 2009.
Nick Nielsen *Alternative Foundations/philosophical* February 28, 2014. http://www.cs.nyu.edu/pipermail/fom/2014-February/017861.html
nLab. *Constructive mathematics* 2014. http://ncatlab.org/nlab/show/constructive+mathematics
Russell O'Connor. *Essential Incompleteness of Arithmetic Verified by Coq* TPHOLs 2005. http://arxiv.org/abs/cs/0505034
George Orwell. *In Front of Your Nose* Tribune. London. March 22, 1946.
Lawrence Paulson. *A Machine-Assisted Proof of Gödel's Incompleteness Theorems* Computer Laboratory, University of Cambridge. September 2014. http://www.cl.cam.ac.uk/~lp15/papers/Formath/Goedel-slides.pdf

Giuseppe Peano *Arithmetices principia, nova methodo exposita* (The principles of arithmetic, presented by a new method) 1889.

Álvaro Pelayo, Michael A. Warren *Homotopy type theory and Voevodsky's univalent foundations* ArXiv:1210.5658. October 2012.

Francis Pelletier *A Brief History of Natural Deduction* “History and Philosophy of Logic” Vol. 20, Issue. 1, 1999.

Tomas Petricek. *Against a universal definition of type* Onward! Essays. 2015

Max Planck *Scientific Autobiography and Other Papers* 1949.

Graham Priest. *Wittgenstein’s remarks on Gödel’s Theorem* in Wittgenstein’s Lasting Significance. Routledge 2004.

Panu Raatikainen. *Gödel's Incompleteness Theorems* Stanford Encyclopedia of Philosophy. Jan. 20, 2015.

Jules Richard. *Les Principes des Mathématiques et le Problème des Ensembles. Revue Générale des Sciences Pures et Appliquées*. 1905. Translated in “Source Book in Mathematical Logic 1879-1931” Harvard University Press. 1964

Victor Rodych. *Misunderstanding Gödel: New arguments about Wittgenstein and new remarks by Wittgenstein* Dialectica 2003.

Claude Rosental. *Weaving Self-Evidence* Princeton. 2008.

Richard Routley. *Dialectical logic, semantics and metamathematics* Erkenntnis 1979.

Bertrand Russell. “Les paradoxes de la logique” *Revue de métaphysique et de morale* 1906.

Bertrand Russell. *Mathematical logic as based on the theory of types* American Journal of Mathematics. Vol 30. No. 3. 1908,

Bertrand Russell. *Principia Mathematica 2nd Edition* 1925.

Michael Shulman. *Type theory and category theory* April 10, 2012. http://www.math.ucsd.edu/~mshulman/hottminicourse2012/

Dana Scott. *Axiomatizing Set Theory* “Symposium in Pure Mathematics” Los Angeles. July 1967.

Dana Scott *Data Types as Lattices* “SIAM Journal on computing”. 1976.

Oron Shagrir *Gödel on Turing on Computability* Church's Thesis after 70 years Ontos-Verlag. 2006.

Stewart Shapiro *Foundations without Foundationalism: A Case for Second-Order Logic* Oxford. 2002.

Stewart Shapiro. *Do Not Claim Too Much: Second-order Logic and First-order* Logic Philosophia Mathematica.1999.

Natarajan Shankar. *Meta-mathematics, Machines, and Gödel’s Proof*. Cambridge University Press. 1994.

Stephen Simpson *Nonprovability of certain combinatorial properties of finite trees* in Studies in Logic and the Foundations of Mathematics. North-Holland. 1985.

Sandro Skansi. *A constructive proof of cut elimination for a system of full second-order logic* Retrieved February 27, 2014. http://www.logic101.net/upload/4296/documents/Article%20SO%20Skansi.pdf

Alfred Tarski *A Decision Method for Elementary Algebra and Geometry* University of California Press. 1951.

Alfred Tarski and Robert Vaught. "Arithmetical extensions of relational systems" *Compositio Mathematica* 13. 1957

Alan Turing. *On computable numbers, with an application to the Entscheidungsproblem* Proceedings London Math Society. 1936.

van Dalen, D. *Zermelo and the Skolem Paradox* Logic Group Preprint Series. Number 83. February 1998. Utrecht Research Institute of Philosophy.

Rineke Verbrugge. *Provability Logic* "The Stanford Encyclopedia of Philosophy" 2010.

Albert Visser. *Can We Make the Second Incompleteness Theorem Coordinate Free* Journal on Logic and Computation. 21(4) 2011.

Hao Wang *A Logical Journey, From Gödel to Philosophy* MIT Press. 1974.

Dan Willard *On the Results of a 14-Year Effort to Generalize Gödel's Second Incompleteness Theorem and Explore Its Partial Exceptions* Collegium Logicum IX, 2007

Leslie White. *The Locus of Mathematical Reality: An Anthropological Footnote* in The World of Mathematics. Vol. 4. Dover. 1956.

Raymond Wilder *Evolution of mathematical concepts. An elementary study.* Dover. 1968.

Andrew Wiles. *Modular elliptic curves and Fermat's Last Theorem* Annals of Mathematics. 141. 1995.

Ludwig Wittgenstein. 1956. *Bemerkungen ¨uber die Grundlagen der Mathematik/Remarks on the Foundations of Mathematics, Revised Edition* Basil Blackwell. 1978

Ludwig Wittgenstein. *Philosophische Grammatik* Basil Blackwell. 1969.

Ludwig Wittgenstein. (1933-1935) *Blue and Brown Books.* Harper. 1965.

Ludwig Wittgenstein *Philosophical Investigations* Blackwell. 1953/2001.

John Woods. *How robust can inconsistency get?* IfCoLoG Journal of Logics and Their Applications. 2014.

Noson Yanofsky. *The Outer Limits of Reason* MIT Press 2013.

Ernst Zermelo *Uber Grenzzahlen und Mengenbereiche: Neue Untersuchungen Äuber die Grundlagen der Mengenlehre* Fundamenta mathematicae. 1930; English translation by Michael Hallett, "On boundary numbers and domains of sets: new investigations in the foundations of set theory" *From Kant to Hilbert: a Source Book in the Foundations of Mathematics*, Oxford University Press, 1996

Ernst Zermelo. *Collected Works/Gesammelte Werke: Volume I/Band I - Set Theory, Miscellanea/Mengenlehre, Varia (Schriften der Mathematisch-naturwissenschaftlichen Klasse) (English and German Edition)* Springer. 2010.

Appendix 1. Notation of Classical Direct Logic

- ***Type*** *i.e.*, a type is a *discrimination*[48] of the following:
  - Boolean::[49], ℕ::[50] and 𝕆::[51]
  - Proposition◁order▷:: and Sentence◁order▷:: where order:$\mathbb{N}_+$
  - τ_1⦶τ_2::[52], [τ_1,τ_2]::[53], [τ_1]↦τ_2::[i] and $\tau_2^{\tau_1}$::[ii] where τ_1:: and τ_2::
  - Set◁τ▷::[iii] and Expression◁τ▷::[54] where τ::

- ***Propositions***, *i.e.*, a Proposition is a *discrimination* of the following:
  - ¬Φ:Proposition◁order▷ where Φ:Proposition◁order▷[iv] and order:$\mathbb{N}_+$
  - Φ∧Ψ,Φ∨Ψ,Φ⇨Ψ,Φ⇔Ψ:Proposition◁order▷ where Φ,Ψ:Proposition◁order▷ and order:$\mathbb{N}_+$
  - (p � True⦂ Φ_1, False⦂ Φ_2):Proposition◁order▷ where p:Boolean, Φ,Ψ:Proposition◁order▷[55] and order:$\mathbb{N}_+$
  - x_1=x_2 where x_1,x_2:τ and τ::
  - x_1⊆x_2 where x_1,x_2:Set◁τ▷ and τ::
  - x_1∈x_2:Proposition◁1▷ where x_1:τ, e_2:τ and τ::
  - τ_1⊑τ_2[56] where τ_1:τ_3, τ_2:τ_4 ▷, τ_3:: and τ_4::
  - τ_1::[57] where τ_1:τ_2 and τ_2::
  - (x:τ):Proposition◁1▷ where τ::, x:τ_1 and τ_1::
  - f[x]:Proposition◁1▷ where x:τ, f:Boolean^{τ} and τ::
  - p⟦x⟧:Proposition◁order+1▷[v] where x:τ, p:$\text{Proposition◁order▷}^{\tau}$ and order:$\mathbb{N}_+$
  - (Φ_1, …, Φ_{n-1} $\vdash^{p}_{T}$ Φ_n):Proposition◁order▷[58] where p:Proof, T:Theory, $\Phi_{1\text{ to }n}$:Proposition◁order▷ and order:$\mathbb{N}_+$
  - ⌊s⌋:Proposition◁order▷ where s:Sentence◁order▷ with no free variables and order:$\mathbb{N}_+$

---

[i] Type of computable procedures from τ_1 into τ_2.
If f:([τ_1]↦τ_2) and x:τ_1, then f.[x]:τ_2.

[ii] Type of functions from τ_1 into τ_2.
If f:$\tau_2^{\sigma_1}$ and x:τ_1, then f[x]:τ_2.

[iii] Set◁τ▷ is a type parametrized by the type τ. In Java and C++, parametrized types are called "generics", "<" is used for ◁, and ">" is used for ▷. The following axiom holds: ∀[τ::,s:Set◁τ▷,x∈s] → x:τ

[iv] Proposition◁order▷ is the parametrized type consisting of type Proposition parametrized by order.

[v] The type of p⟦x⟧ means that the Y untyped fixed point operator cannot be used to construct propositions in Direct Logic.

Grammar (syntax) trees (*i.e.* expressions and sentences) are defined as follows:

- ***Expressions***, *i.e.,* an Expression◁τ▷ is a *discrimination* of the following:
  - Constant◁τ▷:: where τ::
  - ⦅x⦆:Constant◁τ▷ where x:τ and τ::
  - x:Expression◁τ▷ where x:Constant◁τ▷ and τ::
  - x:Expression◁τ▷ where x:Variable◁τ▷ and τ::
  - ⦅Let $f_1[x_1:\tau_1]:\sigma_1 \equiv d_1, \ldots, f_n[x_n:\tau_n]:\sigma_n \equiv d_n$[59] ∘ y⦆:Expression◁τ▷ where for i in 1 to n, f_i:Variable◁$\sigma_i^{\tau_i}$▷ in d_i and y, x_i:Variable◁τ_i▷ in d_i, d_i:Expression◁σ_i▷, y:Expression◁τ▷, and τ_i::
  - ⦅Let $x_1:\tau_1 \equiv d_1, \ldots, x_n \equiv d_n$[60] ∘ y⦆:Expression◁τ▷ where for i in 1 to n, x_i:Variable◁τ_i▷ in d_i and y, d_i:Expression◁σ_i▷, y:Expression◁τ▷, and τ_i::
  - ⦅$e_1 \oplus e_2$⦆:Expression◁$\tau_1 \oplus \tau_2$▷, ⦅[e_1, e_2]⦆:Expression◁[τ_1, τ_2]▷, ⦅[e_1]↦e_2⦆:Expression◁[τ_1]↦τ_2▷ and ⦅$e_2^{e_1}$⦆:Expression◁$\tau_2^{\tau_1}$▷ where e_1:Expression◁τ_1▷, e_2:Expression◁τ_2▷ ▷, τ_1:: and τ_2::
  - ⦅e_1 � True⦂ e_2, False⦂ e_3⦆:Expression◁τ▷[i] where e_1:Expression◁Boolean▷, e_2,e_3:Expression◁τ▷ and τ::
  - ⦅λ[x:τ_1]:τ_2 → e⦆:Expression◁$\tau_2^{\tau_1}$▷ where e:Expression◁τ_2▷, x:Variable◁τ_1▷ in e, and τ_1,τ_2::
  - ⦅e[x]⦆:Expression◁τ_2▷ where e:Expression◁$\tau_2^{\tau_1}$▷, x:Expression◁τ_1▷, τ_1:: and τ_2::
  - ⦅e.[x]⦆:Expression◁τ_2▷ where e:Expression◁[τ_1]↦τ_2▷, x:Expression◁τ_1▷, τ_1:: and τ_2::
  - Sentence◁order▷⊑Expression◁Sentence◁order▷▷ where order:$\mathbb{N}_+$
  - ⌊e⌋:τ where e:Expression◁τ▷ with no free variables and τ::
  - ⌊s⌋:Expression◁τ▷ where s:String◁Expression◁τ▷▷ and τ::

---

[i] *if* e_1 *then* e_2 *else* e_3

- *Sentences*, *i.e.*, a Sentence is a *discrimination* of the following:
  - ⦅**x**⦆:Sentence◁order+1▷[i] where **x**:Variable◁Sentence◁order▷▷ and order:ℕ₊
  - ⦅¬**s**⦆:Sentence◁order▷ where **s**:Sentence◁order▷ and order:ℕ₊
  - ⦅s_1∧s_2⦆,⦅s_1∨s_2⦆,⦅s_1⇨s_2⦆,⦅s_1⇔s_2⦆:Sentence◁order▷ where s_1,s_2:Sentence◁order▷ and order:ℕ₊
  - ⦅**e** � **True**⦂ s_1, **False**⦂ s_2⦆[ii]:Sentence◁order▷ where **e**:Expression◁Boolean▷, s_1,s_2:Sentence◁order▷ and order:ℕ₊
  - ⦅e_1=e_2⦆:Sentence◁1▷ where e_1,e_2:Expression◁τ▷ and τ::
  - ⦅e_1⊑e_2⦆:Sentence◁1▷ where e_1,e_2:Expression◁$τ_1$▷, $τ_1$:$τ_2$ and $τ_2$::
  - ⦅e_1⊆e_2⦆:Sentence◁1▷ where e_1,e_2:Expression◁Set◁τ▷▷ and τ::
  - ⦅e_1∈e_2⦆:Sentence◁1▷ where e_1:Expression◁τ▷, e_2:Expression◁Set◁τ▷▷ and τ::
  - ⦅e_1:e_2⦆:Sentence◁1▷ where e_1:Expression◁$τ_1$▷, e_2:Expression◁$τ_2$▷ $τ_1$:$τ_3$, $τ_2$:$τ_4$ and $τ_3$,$τ_4$::
  - ⦅**e**::⦆:Sentence◁1▷ where **e**:Expression◁τ▷ and τ::
  - ⦅∀[x:$τ_1$]→ s⦆,⦅∃[x:$τ_1$]→ s⦆:Sentence◁order▷ where x:Variable◁$τ_1$▷ in **s**, **s**:Sentence◁order▷ and order:ℕ₊
  - ⦅f[x]⦆:Sentence◁1▷ where x:Expression◁τ▷, f:Expression◁Boolean^τ▷ and τ::
  - ⦅p⟦x⟧⦆:Sentence◁order+1▷[iii] where x:Expression◁τ▷, p:Expression◁Sentence◁order▷^{τ}▷, τ:: and order:ℕ₊
  - ⦅s_1,…,s_{n-1} $\vdash^{p}_{T}$ s_n⦆:Sentence◁order▷ where T:Expression◁Theory▷, $s_{1\ to\ n}$:Sentence◁order▷, p:Expression◁Proof▷ and order:ℕ₊
  - ⌊**s**⌋:Sentence◁order▷ where **s**:String◁Sentence◁order▷▷ and order:ℕ₊

---

[i] The type of ⦅**x**⦆ means that the Y untyped fixed point operator cannot be used to construct sentences in Direct Logic.

[ii] *if* **t** *then* s_1 *else* s_1

[iii] The type of ⦅**p**⟦**x**⟧⦆ means that the Y untyped fixed point operator cannot be used to construct sentences in Direct Logic.

Foundations with strong parameterized types

> "Everyone is free to elaborate [their] own foundations. All that is required of [a] Foundation of Mathematics is that its discussion embody absolute rigor, transparency, philosophical coherence, and addresses fundamental methodological issues."[61]

Classical Direct Logic develops foundations for mathematics by deriving sets[62] from types[63] to encompass all of standard mathematics including the reals, analysis, geometry, *etc.*[64]

For each order:$\mathbb{N}_+$ and P:**Proposition**⊲order⊳$^{\mathbb{N}}$, the following strongly-typed categorical Dedekind induction axiom holds:

$$(P[\![0]\!] \wedge \forall[i{:}\mathbb{N}] \to P[\![i]\!] \Rightarrow P[\![i+1]\!]) \Rightarrow \forall[i{:}\mathbb{N}] \to P[\![i]\!]$$

Another fundamental axiom of Mathematics is for each order:$\mathbb{N}_+$ and P:**Proposition**⊲order⊳$^{\mathbf{O}}$ the following strongly-typed categorical ordinal induction axiom holds:

$$(\forall[\alpha{:}\mathbf{O}] \to (\forall[\beta<\alpha{:}\mathbf{O}] \to P[\![\beta]\!]) \Rightarrow P[\![\alpha]\!]) \Rightarrow \forall[\alpha{:}\mathbf{O}] \to P[\![\alpha]\!]$$

Type Choice

$$\text{Choice}\lhd\tau1,\tau2\rhd{:}\tau2^{\tau2^{\tau1}}$$

$$\forall[f{:}\tau2^{\tau1}] \to \forall[x{:}\tau1] \to (\exists[z{:}\tau1] \to f[z]{:}\tau2) \Rightarrow \text{Choice}\lhd\tau1,\tau2\rhd[f][x]{:}\tau2$$

Categoricity

> "If the mathematical community at some stage in the development of mathematics has succeeded in becoming (informally) clear about a particular mathematical structure, this clarity can be made mathematically exact ... Why must there be such a characterisation? Answer: if the clarity is genuine, there must be a way to articulate it precisely. If there is no such way, the seeming clarity must be illusory ... for every particular structure developed in the practice of mathematics, there is [a] categorical characterization of it."[65]

Classical Direct Logic is much stronger than first-order axiomatizations of set theory in that it provides categoricity for $\mathbb{N}$, $\mathbb{R}$, and **O**. Categoricity is very important in Computer Science so that there are no nonstandard elements in models of computational systems, e.g., infinite integers and infinitesimal reals.

Theorem (Categoricity of Natural Numbers $\mathbb{N}$):[66]

If **X** be a type satisfying the Dedekind categorical axioms for the natural numbers *Nat*, then **X** is isomorphic to $\mathbb{N}$[67], which is strictly more powerful than what a first-order theory can express.[68]

Richard Dedekind

Theorem (Categoricity of Real Numbers $\mathbb{R}$):[69]

If **X** is a type satisfying the Dedekind categorical axioms for the real numbers *Real*, then **X** is (uniquely) isomorphic to $\mathbb{R}$, which is strictly more powerful than what a first-order theory can express.[70]

Theorem (Model Soundness of *Nat*): $(\vdash_{Nat}\Psi) \Rightarrow \vDash_{\mathbb{N}}\Psi$

Proof: Suppose $\vdash_{Nat}\Psi$. The conclusion immediately follows because the axioms for the theory *Nat* hold in the model $\mathbb{N}$.

Theorem (Categoricity of Ordinals O):

If **X** be a type satisfying the axioms the theory of the ordinals *Ord*, then **X** is (uniquely) isomorphic to **O**.[71]

Theorem (Model Soundness of *Ord*): $(\vdash_{Ord}\Psi) \Rightarrow \vDash_{\mathbf{O}}\Psi$

Proof: Suppose $\vdash_{Ord}\Psi$. The conclusion immediately follows because the axioms for the theory *Ord* hold in the model **O**.

$\mathcal{Sets}_{\tau}$ defined using strong parameterized types

The type **Set**◁τ▷ can be defined as follows:

Set◁τ▷ ≡ $\textbf{Boolean}^{\tau}$

Of course set membership is defined as follows:

∀[x:τ:, S:**Set**◁τ▷] → x∈S ⇔ S[x]=True

Inductive definition:

1. $\textbf{Set}^{0}$◁τ▷ ≡ $\textbf{Boolean}^{\tau}$
2. $\textbf{Set}^{\alpha+1}$◁τ▷ ≡ **Set**◁$\textbf{Set}^{\alpha}$◁τ▷▷
3. α is a limit ordinal

   S:$\textbf{Set}^{\alpha}$◁τ▷ ⇔ ∀[X∈S]→ ∃[β<α:**O**,Y:$\textbf{Set}^{\beta}$◁τ▷]→ X∈Y

S:**Sets**◁τ▷ ⇔ ∃[α:**O**]→ S:$\textbf{Set}^{\alpha}$◁τ▷

The properties below mean that **Sets**◁τ▷ is a "universe" of mathematical discourse.[72]

- Foundation: There are no downward infinite membership chains.[73]
- Transitivity of ∈[74]: ∀[S:**Sets**◁τ▷]→ ∀[X∈S]→ X?:**Sets**◁τ▷
- Powerset:[75] ∀[S:**Sets**◁τ▷]→ $\textbf{Boolean}^{S}$:**Sets**◁τ▷
- Union:[76]

  ∀[S:**Sets**◁τ▷]→ ⋃S:**Sets**◁τ▷

  ∀[S:**Sets**◁τ▷]→ ∀[X:**Sets**◁τ▷]→ X∈⋃S ⇔ ∃[Y∈S]→ X∈Y
- Replacement:[77] The function image of any set is also a set, *i.e.*:

  Image◁τ▷:**Sets**◁τ▷$^{[\textbf{Sets}◁τ▷^{\textbf{Sets}◁τ▷}, \textbf{Sets}◁τ▷]}$

  ∀[f:**Sets**◁τ▷$^{\textbf{Sets}◁τ▷}$, S:**Sets**◁τ▷]→

  ∀[y:**Sets**◁τ▷] → y∈Image◁τ▷[f, S] ⇔ ∃[x∈S]→ f[x]=y

Sets◁τ▷ is *much stronger* than first-order ZFC.[78]

Theorem (Model Soundness of $\mathcal{Sets}_{\tau}$): ($\vdash_{\text{Sets}_{\tau}}$ **Ψ**) ⇨ $\vDash_{\textbf{Sets}◁τ▷}$ **Ψ**

Proof: Suppose $\vdash_{\text{Sets}_{\tau}}$ **Ψ**. The conclusion immediately follows because the axioms for the theory $\mathcal{Sets}_{\tau}$ hold in the type **Sets**◁τ▷.

Theorem. $\mathcal{Sets}_\tau$ is categorical via a (unique) isomorphism.

Proof:[79] Suppose that **X** satisfies the axioms for $\mathcal{Sets}_\tau$.

By ordinal induction, the isomorphism I:$\mathbf{X}^{\mathbf{Sets}\triangleleft\tau\triangleright}$ as follows:

1. S:$\mathbf{Set}^0\triangleleft\tau\triangleright$

   I[S] ≡ S
2. S:$\mathbf{Set}^{\alpha+1}\triangleleft\tau\triangleright$

   Z∈$_X$I[S] ⇔ ∃[Y:$\mathbf{Set}^{\alpha}\triangleleft\tau\triangleright$]→ I[Y]∈$_X$Z
3. S:$\mathbf{Set}^{\alpha}\triangleleft\tau\triangleright$ and α is a limit ordinal

   Z∈$_X$I[S] ⇔ ∃[β<α:**O**,Y:$\mathbf{Set}^{\beta}\triangleleft\tau\triangleright$]→ I[Y]∈$_X$Z

I is a unique isomorphism:

- I is one to one
- The range of I is **X**
- I is a homomorphism:
  - I [{ }$_{\mathbf{Sets}\triangleleft\tau\triangleright}$] = { }$_X$
  - ∀[S1,S2:**Sets**◁τ▷] → I [S1 ∪ S2] = I[S1] ∪$_X$ I [S2]
  - ∀[S1 S2:**Sets**◁τ▷] → I[S1 ∩ S2] = I[S1] ∩$_X$ I[S2]
  - ∀[S1,S2:**Sets**◁τ▷] → I[S1 - S2] = I[S1] -$_X$ I[S2]
  - ∀[S:**Sets**◁τ▷] → I[⋃S] = ⋃$_X$ {I[x] | x∈S}
- I^{-1}:$\mathbf{Sets}\triangleleft\tau\triangleright^{\mathbf{X}}$ is a homomorphism
- I is a unique isomorphism: If g:$\mathbf{X}^{\mathbf{Set}\triangleleft\tau\triangleright}$ is an isomorphism, then g= I

Appendix 2. Historical Background

> "The powerful (try to) insist that their statements are literal depictions of a single reality. 'It really is that way', they tell us. 'There is no alternative.' But those on the receiving end of such homilies learn to read them allegorically, these are techniques used by subordinates to read through the words of the powerful to the concealed realities that have produced them." [Law 2004]

Gödel was certain

> "'Certainty' is far from being a sign of success; it is only a symptom of lack of imagination and conceptual poverty. It produces smug satisfaction and prevents the growth of knowledge." [Lakatos 1976]

According to John von Neumann, Gödel was "the greatest logician since Aristotle."[80]

Gödel based his incompleteness results on the thesis that mathematics necessarily has the proposition *I'm unprovable.* Wittgenstein correctly noted that Gödel's proposition infers inconsistency in mathematics:[81]

> "Let us suppose [Gödel's writings are correct and therefore] I prove[i] the improvability (in Russell's system) of [Gödel's proposition *I'm unprovable.*] P; [*i.e.*, $\vdash\nvdash P$ where $P\Leftrightarrow\nvdash P$] then by this proof I have proved P [*i.e.*, $\vdash P$]. Now if this proof were one in Russell's system [*i.e.*, $\vdash\vdash P$] — I should in this case have proved at once that it belonged [*i.e.*, $\vdash P$] and did not belong [*i.e.*, $\vdash\neg P$ because $\neg P\Leftrightarrow\vdash P$] to Russell's system.
> But there is a contradiction here! [*i.e.*, $\vdash P$ and $\vdash\neg P$] …
> **[This] is what comes of making up such sentences.**"

---

[i] Wittgenstein was granting the supposition that Gödel had proved inferential undecidability (sometimes called "incompleteness") of Russell's system, that is., $\vdash\nvdash P$. However, inferential undecidability is easy to prove using the proposition *P where* $P\Leftrightarrow\nvdash P$:

Proof. Suppose to obtain a contradiction that $\vdash P$. Both of the following can be inferred:

1) $\vdash\nvdash P$ from the hypothesis because $P\Leftrightarrow\nvdash P$
2) $\vdash\vdash P$ from the hypothesis by Adequacy.

But 1) and 2) are a contradiction. Consequently, $\vdash\nvdash P$ follows from proof by contradiction.

According to [Monk 2007]:[82]

> *Wittgenstein hoped that his work on mathematics would have a cultural impact, that it would threaten the attitudes that prevail in logic, mathematics and the philosophies of them. On this measure it has been a spectacular failure.*

Ludwig Wittgenstein

Unfortunately, recognition of the worth of Wittgenstein's work on mathematics came long after his death. For decades, many theoreticians mistakenly believed that they had been completely victorious over Wittgenstein.

According to [Gödel 1972]:

> "Wittgenstein did not understand it [Gödel's 1931 article on Principia Mathematica] (or pretended not to understand it). He interpreted it as a kind of logical paradox, while in fact it is just the opposite, namely a mathematical theorem within an absolutely uncontroversial part of mathematics (finitary number theory or combinatorics)."

In the above passage, Gödel retreated from Principia Mathematic to the First-Order Logic theory *FirstOrderNatualNumbers* to defend his proposition *I'mUnprovableInFirstOrderNatualNumbers.* However, the following incompleteness result is not very impressive because *FirstOrderNatualNumbers* is a very weak theory:

- $\models_{\mathbb{N}}$ *I'mUnprovableInFirstOrderNatualNumbers*
- $\nvdash_{\mathit{FirstOrderNatualNumbers}}$ *I'mUnprovableInFirstOrderNatualNumbers*

Furthermore, Chaitin [2007] complained about basing something as important as incompleteness on such a trivial sentence saying:

> "[Gödel's proof] was too superficial. It didn't get at the real heart of what was going on. It was more tantalizing than anything else. It was not a good reason for something so devastating and fundamental. It was too clever by half. It was too superficial. [It was based on the clever construction] "I'm unprovable." So what? This doesn't give any insight how serious the problem is.[i]

Kurt Gödel

[i] According to [Chaitin 2007]: "You see, the real problem with Gödel's proof is that it gives no idea how serious incompleteness is."

The thesis of Chaitin's criticism above is that incompleteness is a fundamental issue for formal systems that is not adequately addressed by Gödel's proof based on his proposition.[83]

Gödel's derived his proposition using what [Carnap 1934] later called the "Diagonal Lemma" which is equivalent to the Y untyped fixed point operator on propositions. His results were for Principia Mathematica, which was intended as the foundation of all of Mathematics. Unfortunately, Principia Mathematica had some defects in its types that have been corrected in Direct Logic..

[Church 1935] correctly proved inferential incompleteness (sometimes called the "First Incompleteness Theorem") without using Gödel's proposition. The Church theorem and its proof are very robust. After Church proved inferential undecidabilty of closed mathematical theories using computational undecidablity, Gödel claimed more generality and that his results applied to all consistent mathematical systems that incorporate axioms for the natural numbers. However, when he learned of Wittgenstein's devastating proof of inconsistency,[84] Gödel retreated to claiming that his results applied to the very weak first-order theory of natural numbers.[85] The upshot is that Gödel never acknowledged that his proposition[i] *I'm unprovable.* implies inconsistency in mathematics.

Also, the ultimate criteria for working mathematicians of correctness theorems of the natural numbers provability using the categorical set theory described in this article. In this sense, Wittgenstein was correct in his identification of ultimate "truth" with provability. On the other hand, Gödel obfuscated the important identification of provability as the touchstone of ultimate correctness in mathematics.

Paul Cohen [2006] wrote as follows of his interaction with Gödel:[86]

> "His [Gödel's] main interest seemed to lie in discussing the ʽtruth' or ʽfalsity' of these questions, not merely in their undecidability. He struck me as having an almost unshakable belief in this "realist" position, which I found difficult to share. His ideas were grounded in a deep philosophical belief as to what the human mind could achieve. I greatly admired this faith in the power and beauty of Western Culture, as he put it, and would have liked to understand more deeply what were the sources of his strongly held beliefs. Through our discussions, I came closer to his point of view, although I never shared completely his ʽrealist' point of view, that all questions of Set Theory were in the final analysis, either true or false."

[i] constructed using the Y untyped fixed point operator on propositions

von Neumann [1961] had a very different view from Gödel:

> "It is **not** necessarily true that the mathematical method is something absolute, which was revealed from on high, or which somehow, after we got hold of it, was evidently right and has stayed evidently right ever since."


John von Neumann

Provability Logic

One kind of Provability Logic (called $\mathcal{PL}$) is a cut-down theory of deduction that has been used to investigate provability predicates for languages that allow use of the Y untyped fixed point operator for propositions [Verbrugge 2010]. $\mathcal{PL}$ is very weak; even for proving theorems about integers because $\mathcal{PL}$ is first-order.

[Löb 1955] proposed the following conditions that became the basis of Provability Logic:[i]

1. $(\vdash_{\mathcal{PL}}\Phi) \Rightarrow \vdash_{\mathcal{PL}}\vdash_{\mathcal{PL}}\Phi$[87]
2. $\vdash_{\mathcal{PL}}((\vdash_{\mathcal{PL}}(\Phi\Rightarrow\Psi)) \Rightarrow ((\vdash_{\mathcal{PL}}\Phi) \Rightarrow \vdash_{\mathcal{PL}}\Psi)))$[88]
3. $\vdash_{\mathcal{PL}}((\vdash_{\mathcal{PL}}\Phi) \Rightarrow \vdash_{\mathcal{PL}}\vdash_{\mathcal{PL}}\Phi)$[89]

Gödel's construction of the proposition *I'm unprovable in PL*[ii] using the Y fixed point operator can be carried out in Provability on propositions about the natural numbers with the following results:

- $\nvdash_{\mathcal{PL}}$ *I'mUnprovableInPL*[iii]
- $\nvdash_{\mathcal{PL}}\neg$*I'mUnprovableInPL*[iv]
- Consistent[$\mathcal{PL}$] $\Rightarrow \vDash_{\mathbb{N}}$ *I'mUnprovableInPL*[v]

---

[i] His formulation actually used a convoluted coding of propositions into integers called Gödel numbers.

[ii] *I'mUnprovableInPL* $\equiv \nvdash_{\mathcal{PL}}$ *I'mUnprovableInPL*

[iii] Proof by contradiction in $\mathcal{PL}$

[iv] Proof by contradiction in $\mathcal{PL}$

[v] If Consistent[$\mathcal{PL}$], then *I'mUnprovableInPL* is true in the model $\mathbb{N}$ by proof above in theory $\mathcal{Nat}$.

However, $\mathcal{PL}$ is a weak theory. For example, the principle of natural deduction Logic that allows theorems to be used in subproofs is *not* allowed in $\mathcal{PL}$:[90]

$$(\vdash \Phi)\vdash\Phi$$

In summary, Provability Logic (although a useful historical development step) is too weak and fragile to serve in the mathematical foundation of Computer Science.

Limitations of first-order logic

> "By this it appears how necessary it is for nay man that aspires to true knowledge to examine the definitions of former authors; and either to correct them, where they are negligently set down, or to make them himself. For the errors of definitions multiply themselves, according as the reckoning proceeds, and lead men into absurdities, which at last they see, but cannot avoid, without reckoning anew from the beginning; in which lies the foundation of their errors..."
> [Hobbes *Leviathan*, Chapter 4][91]

It turns out that first-order logic is amazing weak. For example, first-order logic is incapable of characterizing even the natural numbers, *i.e.*, there are infinite integers in models of every first-order axiomatization of the natural numbers. Furthermore, there are infinitesimal real numbers in models of every first-order axiomatization of the real numbers.[i] Of course, infinite integers and infinitesimal reals are monsters that must be banned from the mathematical foundations of Computer Science.

[i] Likewise, first-order set theory (*e.g.* ZFC) is very weak. See discussion immediately below.

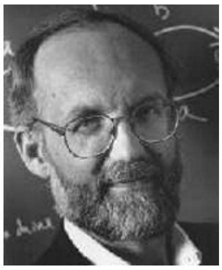
However, some theoreticians have found first-order logic to be useful for their careers because it is weak enough that they can prove theorems about first-order axiomatizations whereas they cannot prove such theorems about stronger practical systems, *e.g.*, Classical Direct Logic.[92]

Ernst Zermelo

Zermelo considered the First-Order Thesis to be a mathematical "hoax" because it necessarily allowed unintended models of axioms.[93]

[Barwise 1985] critiqued the First-Order Thesis that mathematical foundations should be restricted to first-order logic as follows:

> *The reasons for the widespread, often uncritical acceptance of the first-order thesis are numerous. The first-order thesis ... confuses the subject matter of logic with one of its tools. First-order language is just an artificial language structured to help investigate logic, much as a telescope is a tool constructed to help study heavenly bodies. From the perspective of the mathematics in the street, the first-order thesis is like the claim that astronomy is the study of the telescope*.[94]



Jon Barwise

Computer Science is making increasing use of Model Analysis[i] in the sense of analyzing relationships among the following:

- concurrent programs and their Actor Model denotations
- domain axiom systems and computations on these domains

[i] a restricted form of Model Checking in which the properties checked are limited to those that can be expressed in Linear-time Temporal Logic has been studied [Clarke, Emerson, Sifakis, *etc*. ACM 2007 Turing Award].

In Computer Science, it is important that the natural numbers be axiomatized in a way that does not allow non-numbers (*e.g.* infinite ones) in models of the axioms. Unfortunately, every consistent first-order axiomatization of the natural numbers has a model with an infinite integer:

Theorem: If ℕ is a model of a first-order axiomatization *T*, then *T* has a model 𝕄 with an infinite integer.

Proof: The model 𝕄 is constructed as an extension of ℕ by adding a new element ∞ with the following atomic relationships:

$$\{\neg\infty<\infty\} \cup \{ m<\infty \mid m{:}\mathbb{N}\}$$

It can be shown that 𝕄 is a model of *T* with an infinite integer ∞.

The infinite integer ∞ is a monster that must be banned from the mathematical foundations of Computer Science.

A similar result holds for the standard theory ℝ of real numbers [Dedekind 1888] compared to a cut-down, first-order theory[95], which has models with infinitesimals:

Theorem: If ℝ is a model of a first-order axiomatization *T*, then *T* has a model 𝕄 with an infinitesimal.

Proof: The model 𝕄 is constructed as an extension of ℝ by adding a new element ∞ with the following atomic relationships:

$$\{\neg\infty<\infty\} \cup \{m<\infty \mid m{:}\mathbb{N}\}$$

Defining ε to be $\frac{1}{\infty}$, it follows that $\forall[r{:}\mathbb{R}]\rightarrow 0<\varepsilon<\frac{1}{r}$. It can be shown that 𝕄 is a model of *T* with an infinitesimal ε, which is a monster that must be banned from the mathematical foundations of Computer Science.

On the other hand, since it is not limited to first-order logic, Classical Direct Logic characterizes structures such as natural numbers and real numbers up to isomorphism.[i]

There are theorems for integers that cannot be proved from the first-order versions of the Dedekind axioms [Goodstein 1944, Simpson 1985, Wiles 1995, Bovykin 2009, McLarty 2010].

[i] proving that software developers and computer systems are using the same structures

Of greater practical import, First-order logic is *not* a suitable foundation for the Internet of Things in which specifications require a device respond to a request.[i] The specification that a computer responds can be formalized as follows: ∃[i:ℕ]→ResponseBefore[i].. However, the specification cannot be proved in a first-order theory.

Proof: In order to obtain a contradiction, suppose that it is possible to prove in a first-order theory $\mathcal{T}$ ∃[i:ℕ]→ResponseBefore[i]. Therefore the infinite set of propositions {¬ResponseBefore[i] | i:ℕ} is inconsistent. By the compactness theorem of first-order logic, it follows that there is finite subset of the set of propositions that is inconsistent. But this is a contradiction, because all the finite subsets are consistent since the amount of time before a server responds is unbounded, that is, ∄[i:ℕ]→ $\vdash_{\mathcal{T}}$ ResponseBefore[i].

First-order logic is **not** a suitable foundation for specifications in the Internet of Things.

The following is an example of an Actor system can provides a service that is impossible to implement using nondeterministic Turing Machines because:

There is a bound on the size of integer that can be computed by an *always halting* nondeterministic Turing Machine starting on a blank tape.

Plotkin [1976] gave an informal derivation as follows:

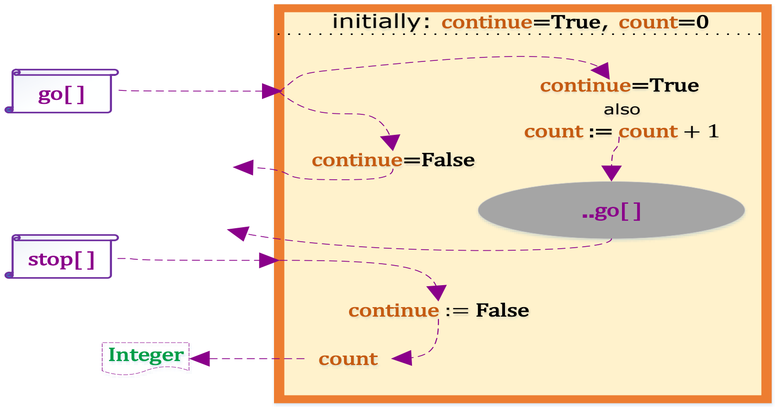
Now the set of initial segments of execution sequences of a given nondeterministic program P*, starting from a given state, will form a tree. The branching points will correspond to the choice points in the program. Since there are always only finitely many alternatives at each choice point, the branching factor of the tree is always finite. That is, the tree is finitary. Now König's lemma says that if every branch of a finitary tree is finite, then so is the tree itself. In the present case this means that if every execution sequence of* P *terminates, then there are only finitely many execution sequences. So if an output set of* P *is infinite, it must contain a nonterminating computation.*

[i] An implementation of such a system is given below in this article.

By contrast, the following Actor system can compute an integer of unbounded size:

The above Actor system can be implemented as follows using ActorScript™:

```
Unbounded.[ ]:Integer ≡
                          // Unbounded is a procedure that returns Integer
  Let aCounter ← Counter[ ].     // let aCounter be a new Counter
     Prep aCounter.go[ ]
                          // send aCounter a go message concurrently with
        concurrently aCounter.stop[ ]
                    // returning the value of sending aCounter a stop message
```




```
Actor Counter[ ]
  locals count := 0,          // the variable count is initially 0
         continue := True.  // the variable continue is initially True
  stop[ ]:Integer →
     Prep continue := False  // change continue to False and then
            count                    // return count
  go[ ]:Void →
     continue �
       True⦂                      //  if continue is True,
         Prep count := count+1     // increment count and then
            Hole ..go[ ]             //  send go[ ] to this counter
       False⦂                     // if continue is False,
         Void                      // return Void
```

By the semantics of the Actor model of computation, executing Unbounded.[ ] returns an integer of unbounded size. Consequently, the above concurrent algorithm for Unbounded.[ ] cannot be implemented using nondeterministic abstract state machines or using the nondeterministic λ-calculus.

The above program illustrates how nondeterministic branching is not a good model for message reception in IoT

As a foundation of mathematics for Computer Science, Classical Direct Logic provides categorical[96] numbers (integer and real), sets, lists, trees, graphs, etc. which can be used in arbitrary mathematical theories including theories for categories, large cardinals, first-order axiomatizations, etc. These various theories might have "monsters" of various kinds. However, these monsters are not imported into the foundations of Computer Science.

Computer Science needs *stronger* systems than provided by first-order logic in order to weed out unwanted models. In this regard, Computer Science doesn't have a problem computing with "infinite" objects (*i.e.* Actors) such as π and uncountable sets such as the set of real numbers **Set**◁ℝ▷. However, the mathematical foundation of Computer Science is very different from the general philosophy of mathematics in which the infinite integers and infinitesimal reals allowed by models of first-order theories may be of some interest. Of course, it is always possible to have special theories that are *not* part of the foundations with infinite integers, infinitesimal reals, unicorns, *etc*.[97]

Of course some problems are theoretically not computable. However, even in these cases, it is often possible to compute approximations and cases of practical interest.[i]

The mathematical foundation of Computer Science is very different from the general philosophy of mathematics in which infinite integers and infinitesimal reals may be of some interest. Of course, it is always possible to have special theories with infinite integers, infinitesimal reals, unicorns, *etc*.

Church's Paradox

[Church 1932, 1933] attempted basing foundations entirely on untyped higher-order functions, but foundered because contradictions emerged because

1. His system allowed the use of the Y untyped fixed point operator for propositions [Kleene and Rosser 1935]
2. Theorems in his system were computationally enumerable.

[i] *e.g.* see Terminator [Knies 2006], which practically solves the halting problem for device drivers

[Church 1934] expounded on the following profound issues, which is designated " Church's Paradox":

> "in the case of any system of symbolic logic, the set of all provable theorems is [computationally] enumerable... any system of symbolic logic not hopelessly inadequate ... would contain the formal theorem that this same system ... was either insufficient [theorems are not computationally enumerable] or over-sufficient [that theorems are computationally enumerable means that the system is inconsistent]...
>
> This, of course, is a deplorable state of affairs...
>
> Indeed, if there is no formalization of logic as a whole, then there is no exact description of what logic is, for it in the very nature of an exact description that it implies a formalization. And if there no exact description of logic, then there is no sound basis for supposing that there is such a thing as logic."

Alonzo Church

The above issues can be addressed as follows:

1. Requiring Mathematics to be strongly typed
2. Mathematics self proves that it is "open" in the sense that proofs are not computationally enumerable (*i.e.* not "closed").[98]

Gödel, Curry, and Löb Paradoxes

Allowing use of the Y untyped fixed point operator for propositions results in contradictions.

For example, consider the diagonal construction used in [Gödel 1931]:

The proposition *I'm unprovable.* that was used by Gödel *cannot* be constructed as follows:[99]

I'mUnprovable:Nonexistent $\not\equiv$ $\nvdash$ *I'mUnprovable*

// illegal definition because $\nvdash$ *I'mUnprovable* is of

// type of **Proposition** order greater than *I'm unprovable*

I'mUnprovable $\Leftrightarrow$ $\nvdash$ *I'mUnprovable* // definition of *I'mUnprovable*

By the following argument, Wittgenstein derived a contradiction in Mathematics from Gödel's result:[100]

Gödel thought that he demonstrated $\vdash\nvdash$Gödel. Therefore $\vdash$Gödel using Gödel$\Leftrightarrow\nvdash$Gödel. $\vdash\vdash$Gödel follows using adequacy. But the contradiction $\vdash\neg$Gödel follows using Gödel$\Leftrightarrow\nvdash$Gödel.

Also, the following paradoxes cannot prove *every* proposition because the Y untyped fixed point operator for propositions cannot be used in a strongly typed logic:

The Liar Paradox [Eubulides of Miletus] is an example of using untyped propositions to derive an inconsistency. However, strong typing prevents an inconsistency as follows:

- ***Liar Paradox*** [Eubulides of Miletus][101]

  Liar:Nonexistent ≢ ¬Liar

  // above definition is illegal because ¬Liar is of

  // type Proposition order greater than Liar

  1) Liar ⇔ ¬Liar // definition of Liar
  2) ¬Liar // proof by contradiction from 1)
  3) Liar // from 1) and 2)

Also, the following paradoxes cannot prove *every* proposition because the Y untyped fixed point operator for propositions cannot be used in a strongly typed logic:

- ***Curry's Paradox*** **[**Curry 1941**]** Suppose Ψ:Proposition◁order:$\mathbb{N}_+$▷.

  $Curry_\Psi$:Nonexistent ≢ $Curry_\Psi$ ⊢Ψ

  // illegal definition because $Curry_\Psi$ is not a Proposition of any order

  // because the type of $Curry_\Psi$ ⊢Ψ is a Proposition of higher

  // order than $Curry_\Psi$

  1) $Curry_\Psi$ ⊢Ψ ⇔ ($Curry_\Psi$ ⊢Ψ ⊢ Ψ) // definition of $Curry_\Psi$
  2) ⊢ ($Curry_\Psi$ ⊢ $Curry_\Psi$) // idempotency
  3) ⊢ ($Curry_\Psi$ ⊢ ($Curry_\Psi$ ⊢Ψ)) // substituting 1) into 2)
  4) ⊢ ($Curry_\Psi$ ⊢Ψ) // contraction
  5) ⊢ $Curry_\Psi$ // from 4) using 1)
  6) ⊢Ψ // transitivity 4) and 5)

- ***Löb's Paradox*** **[**Löb 1955**]**[102] Suppose Ψ:Proposition◁anOrder:$\mathbb{N}_+$▷.

  $Löb_\Psi$:Nonexistent ≢ (⊢$Löb_\Psi$) ⊢Ψ

  // illegal definition because $Löb_\Psi$ is not a Proposition of any order

  // because the type of (⊢$Löb_\Psi$)⊢Ψ is a Proposition of higher

  // order than $Löb_\Psi$

  1) $Löb_\Psi$ ⇔ ((⊢ $Löb_\Psi$) ⊢Ψ) // definition of $Löb_\Psi$
  2) ⊢ ((⊢ $Löb_\Psi$) ⊢ $Löb_\Psi$) // rule of *Theorem Use*
  3) ⊢((⊢ $Löb_\Psi$) ⊢ ((⊢ $Löb_\Psi$) ⊢Ψ)) // substituting 1) into 2)
  4) ⊢((⊢$Löb_\Psi$) ⊢Ψ) // contraction
  5) ⊢$Löb_\Psi$ // from 4) using 1)
  6) Ψ // transitivity using 4) and 5)

Of course, it is completely unacceptable for every proposition to be provable and so measures must be taken to prevent this.

Berry Paradox

The Berry Paradox [Russell 1906] can be formalized as follows:

Characterize⟦s:String◁Expression◁Proposition◁ω▷$^{\mathbb{N}}$▷▷
k:ℕ⟧:Proposition◁ω+1▷ ≡
∀[x:ℕ]→⌊⌊s⌋⌋⟦x⟧ ⇔ x=k

Consider the following definition:

BString:String◁Expression◁Proposition◁ω+1▷$^{\mathbb{N}}$▷▷ ≡
“⦅λ[n:ℕ]→ ⦅∀[s:String◁Expression◁Proposition◁ω▷$^{\mathbb{N}}$▷▷]
Length[s]<100 ⇨ ¬Characterize⟦s, n⟧⦆⦆”

BExpression:Expression◁Proposition◁ω+1▷$^{\mathbb{N}}$▷ ≡ ⌊BString⌋

Note that

- Length[BString]<100.
- {s:String◁Expression◁Proposition◁ω▷$^{\mathbb{N}}$▷▷| $\models_{\mathbb{N}}$ Length[s]<100} is finite.
- Therefore the following set is finite:

{n:$\mathbb{N}_+$ | $\models_{\mathbb{N}}$ ∃[s:String◁Expression◁Proposition◁ω▷$^{\mathbb{N}}$▷▷]
Length[s]<100 ∧ Characterize⟦s, n⟧}

BSet:Set◁ℕ▷ ≡ {n:$\mathbb{N}_+$ | $\models_{\mathbb{N}}$ ⌊BExpression⌋⟦n⟧}

BSet≠{ } because is {n:ℕ | n≧1} is infinite.

1. BNumber:ℕ ≡ Least[BSet]
2. $\models_{\mathbb{N}}$ ⌊BExpression⌋⟦BNumber⟧ [103]
3. $\models_{\mathbb{N}}$⌊⦅λ[n:ℕ]→ ⦅∀[s:String◁Expression◁Proposition◁ω▷$^{\mathbb{N}}$▷▷] → Length[s]<100 ⇨ ¬Characterize⟦s, n⟧⦆⌋⟦BNumber⟧[104]
4. $\models_{\mathbb{N}}$ ∀[s:String◁Expression◁Proposition◁ω▷$^{\mathbb{N}}$▷▷] → Length[s]<100 ⇨ ¬Characterize⟦s, BNumber⟧ [105]
5. $\models_{\mathbb{N}}$ Length[BString]<100 ⇨ ¬Characterize⟦BString, BNumber⟧
   // above is invalid because of attempted substitution of
   // BString:String◁Expression◁Proposition◁ω+1▷$^{\mathbb{N}}$▷▷ for
   // s:String◁Expression◁Proposition◁ω▷$^{\mathbb{N}}$▷▷

Appendix 3. On the decidability of Classical Provability Direct Logic

Provability Direct Logic is a subset of Inconsistency Robust Direct Logic *for a single theory* without quantifiers as follows: $\neg\Phi$:Proposition where Φ:Proposition

- $\Phi\wedge\Psi, \Phi\vee\Psi, \Phi\Rightarrow\Psi, \Phi\Leftrightarrow\Psi$:Proposition where Φ,Ψ:Proposition
- $(\Phi_1, \ldots, \Phi_{n-1} \vdash \Phi_n)$:Proposition where $\Phi_{1\text{ to }n}$:Proposition

The following logical equivalences hold:

$$
\begin{array}{l}
(\Psi=\Phi) \Rightarrow \Psi\Leftrightarrow\Phi \\
(\Psi\Leftrightarrow\Phi) \Rightarrow (\neg\Psi)\Leftrightarrow(\neg\Phi) \\
(\Psi\Leftrightarrow\Phi) \Rightarrow (\Psi\vee\Theta) \Leftrightarrow (\Phi\vee\Theta) \\
(\Psi\Leftrightarrow\Phi) \Rightarrow (\Theta\vee\Psi) \Leftrightarrow (\Theta\vee\Phi) \\
(\Psi\Leftrightarrow\Phi) \Rightarrow (\Psi\wedge\Theta) \Leftrightarrow (\Phi\wedge\Theta) \\
(\Psi\Leftrightarrow\Phi) \Rightarrow (\Theta\wedge\Psi) \Leftrightarrow (\Theta\wedge\Phi) \\
(\Psi\Leftrightarrow\Phi) \Rightarrow (\Psi\vdash\Theta) \Leftrightarrow (\Phi\vdash\Theta) \\
(\Psi\Leftrightarrow\Phi) \Rightarrow (\Theta\vdash\Psi) \Leftrightarrow (\Theta\vdash\Phi) \\
(\Psi\Leftrightarrow\Phi) \Rightarrow (\Psi\Rightarrow\Theta) \Leftrightarrow (\Phi\Rightarrow\Theta) \\
(\Psi\Leftrightarrow\Phi) \Rightarrow (\Theta\Rightarrow\Psi) \Leftrightarrow (\Theta\Rightarrow\Phi) \\
(\Psi\Leftrightarrow\Phi) \Rightarrow (\Psi\Leftrightarrow\Theta) \Leftrightarrow (\Phi\Leftrightarrow\Theta) \\
(\Psi\Leftrightarrow\Phi) \Rightarrow (\Theta\Leftrightarrow\Psi) \Leftrightarrow (\Theta\Leftrightarrow\Phi)
\end{array}
$$

Double Negation: $\neg\neg\Psi \Leftrightarrow \Psi$

$\Rightarrow$ as $\vee$: $(\Psi\Rightarrow\Phi) \Leftrightarrow \Phi\vee\neg\Psi$

Idempotence of $\wedge$: $\Psi\wedge\Psi \Leftrightarrow \Psi$

Commutativity of $\wedge$: $\Psi\wedge\Phi \Leftrightarrow \Phi\wedge\Psi$

Associativity of $\wedge$: $(\Psi\wedge(\Phi\wedge\Theta)) \Leftrightarrow (\Psi\wedge\Phi)\wedge\Theta$

Distributivity of $\wedge$ over $\vee$: $(\Psi\wedge(\Phi\vee\Theta)) \Leftrightarrow (\Psi\wedge\Phi)\vee(\Psi\wedge\Theta)$

Idempotence of $\vee$: $\Psi\vee\Psi \Leftrightarrow \Psi$

Commutativity of $\vee$: $\Psi\vee\Phi \Leftrightarrow \Phi\vee\Psi$

Associativity of $\vee$: $(\Psi\vee(\Phi\vee\Theta)) \Leftrightarrow (\Psi\vee\Phi)\vee\Theta$

Equivalence $(\Psi\Leftrightarrow\Phi) \Leftrightarrow (\Psi\Rightarrow\Phi)\wedge(\Phi\Rightarrow\Psi)$

Comma: $(\Psi\wedge\Phi\vdash\Theta) \Leftrightarrow \Psi,\Phi\vdash\Theta$

Subproof: $(\Psi\vdash(\Phi\vdash\Theta)) \Leftrightarrow (\Psi,\Phi\vdash\Theta)$

Adequacy: $(\Psi\vdash\Phi) \Leftrightarrow \vdash(\Psi\vdash\Phi)$

Classical Proof by Contradiction: $(\Psi\vdash(\Phi\wedge\neg\Phi)) \vdash \neg\Psi$

Theorem Use: $(\vdash\Psi)\vdash\Psi$

Inference means provable entailment: $(\Psi\vdash\Phi) \Leftrightarrow \vdash(\Psi\Rightarrow\Phi)$

Theorem. Allowing the proposition *I'm unprovable.*[ii] used in the incompleteness results of [Gödel 1931] leads to inconsistency in Classical Provability Direct Logic.[106]

Proof: [107]

1) *I'mUnprovable* $\Leftrightarrow$ $\nvdash$ *I'mUnprovable* // Gödel's diagonal lemma
2) ($\vdash$*I'mUnprovable*) $\Rightarrow$ ($\vdash$ $\vdash$*I'mUnprovable*) // adequacy
3) ($\vdash$*I'mUnprovable*) $\Rightarrow$ ($\vdash$ $\nvdash$ *I'mUnprovable*) // Using 1)
4) $\vdash$ $\nvdash$ *I'mUnprovable* // Proof by Contradiction using 2) and 3)[i]
5) $\vdash$ ***I'mUnprovable*** // from 4) using 1)
6) $\vdash$ $\vdash$ *I'mUnprovable* // from 5) using adequacy
7) $\vdash$ $\neg$***I'mUnprovable*** // from 6) using 1)

Inconsistency in Classical Provability Direct Logic means that there is some Ψ such that $\vdash(\Psi\wedge\neg\Psi)$.

Theorem. Classical Provability Direct Logic proves its formal consistency

Proof:

$\neg$Consistent $\Rightarrow$ $\vdash(\Psi\wedge\neg\Psi)$ // definition of consistency
$\neg$Consistent $\vdash$ $\vdash(\Psi\wedge\neg\Psi)$ // entailment means inference
$\neg$Consistent $\vdash(\Psi\wedge\neg\Psi)$ // adequacy
$\vdash$Consistent // classical proof by contradiction

Conjecture. Classical Provability Direct Logic is computationally decidable.

Discussion: A major difficulty in a decision procedure is dealing with expressions of the following form: $(\vdash\boldsymbol{\Psi})\vdash\boldsymbol{\Phi}$

[i] This is a stronger form of a theorem in [Gödel 1931]

Appendix 4. Classical Natural Deduction

Below are schemas for nested-box-style Natural Deduction[i] for Classical Mathematics:

⇨ Introduction	
Ψ	ⓘ hypothesis
...	
Φ	ⓘ inference
$\Psi \Rightarrow \Phi$	ⓘ **conclusion**
$(\Psi \vdash \Phi) \vdash (\Psi \Rightarrow \Phi)$	

⇨ Elimination	
Ψ	ⓘ premise
...	
$\Psi \Rightarrow \Phi$	ⓘ premise
...	
Φ	ⓘ **conclusion**
$\Psi, (\Psi \Rightarrow \Phi) \vdash \Phi$	

∧ Introduction	
Ψ	ⓘ premise
...	
Φ	ⓘ premise
...	
$\Psi \wedge \Phi$	ⓘ **conclusion**
$\Psi, \Phi \vdash (\Psi \wedge \Phi)$	

∧ Elimination	
$\Psi \wedge \Phi$	ⓘ premise
...	
Ψ	ⓘ **conclusion**
...	
Φ	ⓘ **conclusion**
$(\Psi \wedge \Phi) \vdash \Psi, \Phi$	

∨ Introduction	
Ψ	ⓘ premise
...	
$\Psi \vee \Phi$	ⓘ **conclusion**
$\Psi \vdash (\Psi \vee \Phi)$	

∨ Elimination	
$\neg\Psi$	ⓘ premise
...	
$\Psi \vee \Phi$	ⓘ premise
...	
Φ	ⓘ **conclusion**
$\neg\Psi, (\Psi \vee \Phi) \vdash \Phi$	

Proof by Contradiction	
Ψ	ⓘ hypothesis
...	
$\Phi \wedge \neg\Phi$	ⓘ inference
$\neg\Psi$	ⓘ **conclusion**
$(\Psi \vdash (\Phi \wedge \neg\Phi)) \vdash \neg\Psi$	

∨ Cases	
$\Psi \vee \Phi$	ⓘ premise
...	
$\Psi \vdash \Theta$	ⓘ premise
...	
$\Phi \vdash \Theta$	ⓘ premise
...	
Θ	ⓘ **conclusion**
$(\Psi \vee \Phi), (\Psi \vdash \Theta), (\Phi \vdash \Theta) \vdash \Theta$	

Soundness
$(\vdash \Psi) \Rightarrow \Psi$

Adequacy
$(\Phi \vdash \Psi) \Leftrightarrow (\vdash (\Phi \vdash \Psi))$

[i] Evolved from classical natural deduction [Jaśkowski 1934]. See history in Pelletier [1999].

Index

End Notes

---

[1] [White 1956, Wilder 1968, Rosental 2008]

[2] The principle of allowing use of theorems in subproofs is fundamental to Mathematics going back at least to Euclid.

[3] For reservations about the adequacy of the proof, see [Woods 2014].

[4] This paragraph builds on [Meyer 2016].

[5] There seem to be no practical uses for using the Y untyped fixed point operator for propositions in the mathematical foundations of Computer Science beyond what can be done using strongly typed recursive definitions. Furthermore, the Y untyped fixed point operator on propositions can lead to inconsistency in Mathematics as shown in this article.

[6] Types have become increasingly important in formal proofs. For example, see [Farquhar, Grov, Cropper, Muggleton and Bundy 2015]. The type theory of Direct Logic is much stronger than constructive type theory [Shulman 2012, nLab 2014] with constructive logic because Classical Direct Logic has all of the power of Classical Mathematics.

[7] After [Church 1935], it was known that inferentially undecidable propositions must exist because deciding inferability is computationally undecidable.

[8] Mathematical foundations of Computer Science must be general, rigorous, realistic, and as simple as possible. There are a large number of highly technical aspects with complicated interdependencies and trade-offs. Foundations will be used by humans and computer systems. Contradictions in the mathematical foundations of Computer Science cannot be allowed and if found must be repaired.

Classical mathematics is the subject of this article. In a more general context:

- Inconsistency Robust Direct Logic is for pervasively inconsistent theories of practice, e.g., theories for climate modeling and for modeling the human brain.
- Classical Direct Logic can be freely used in theories of Inconsistency Robust Direct Logic. See [Hewitt 2010] for discussion of Inconsistency Robust Direct Logic. Classical Direct Logic for mathematics used in inconsistency robust theories.

[9] [Hardy 1940]

[10] The principle of *Theorem Use* means:

- A theorem of Mathematics can be used *anywhere* including in inconsistency robust inference

- A theorem of Mathematics can be used in a step of a sub-proof to prove a theorem in Mathematics *regardless* of the assumptions of the sub-proof.

[11] The definition of formal inconsistency, *i.e.*,

$$\text{Consistent}\Leftrightarrow\neg\exists[\Psi]\rightarrow\vdash(\Psi\wedge\neg\Psi)$$

is not per se about numbers. Consistent with the general practice in Computer Science, there is no way to identify propositions with integers.

[12] A prominent logician referee of this article suggested that if the proof is accepted then consistency should be made an explicit premise of every theorem of classical mathematics!

[13] As shown above, there is a simple proof in Classical Direct Logic that Mathematics ($\vdash$) is consistent. If Classical Direct Logic has a bug, then there might also be a proof that Mathematics is inconsistent. Of course, if such a bug is found, then it must be repaired.

The Classical Direct Logic proof that Mathematics ($\vdash$) is consistent is very robust. One explanation is that consistency is built in to the very architecture of classical mathematics because it was designed to be consistent. Consequently, it is not absurd that there is a simple proof of the formal consistency of Mathematics ($\vdash$) that does not use all of the machinery of Classical Direct Logic.

In reaction to paradoxes, the dogma of strict separation of "object theories" (theories about basic mathematical entities such as numbers) and "meta theories" (theories about theories) was developed. This linguistic separation can be very awkward in Computer Science. Consequently, Direct Logic does not have the separation in order that some propositions can be more "directly" expressed. For example, Direct Logic can use $\vdash\vdash\Psi$ to express that it is provable that P is provable in Mathematics. It turns out in Classical Direct Logic that $\vdash\vdash\Psi$ holds if and only if $\vdash\Psi$ holds. By using such expressions, Direct Logic contravenes the philosophical dogma that the proposition $\vdash\vdash\Psi$ must be expressed using Gödel numbers.

[14] Wittgenstein in 1937 published in Wittgenstein 1956, p. 50e and p. 51e]

[15] using what [Carnap 1934] later called the "Diagonal Lemma" which is equivalent to the Y untyped fixed point operator on propositions.

[16] Classical Direct Logic is different from [Willard 2007], which developed sufficiently weak systems that the proposition *I'm unprovable.* does not exist.

[17] Delta, student in [Lakatos, 1976, pg. 14].

[18] specified by axioms [Dedekind 1888] that characterize them up to a unique isomorphism

[19] It is worth going to a lot of trouble to make sure that Classical Direct Logic is consistent so that it can be freely used in an inconsistent theory without introducing additional inconsistencies into the theory. **Any bugs found in Classical Direct Logic must be repaired to restore consistency.**

[20] The *Computational Representation Theorem* [Clinger 1981; Hewitt 2006] characterizes computation for systems which are closed in the sense that they do not receive communications from outside:

> The denotation Denote_S of a closed system **S** represents all the possible behaviors of **S** as[20]
>
> $$\mathsf{Denote}_S = \lim_{i\to\infty} \mathrm{Progression}_S{}^i$$
>
> *where* $\mathrm{Progression}_S{}^i \rightarrow \mathrm{Progression}_S{}^{i+1}$
>
> In this way, **S** can be mathematically characterized in terms of all its possible behaviors (including those involving unbounded nondeterminism).

The denotations of the Computational Representation Theorem form the basis of procedurally checking programs against all their possible executions.

[21] According to [Gödel correspondence with Ernst Zermelo October 30, 1931 in *Kurt Gödel Collected Works Volume V* Correspondence H-Z, Oxford University Press, 2003.

pp. 420-431.]: Without Gödel's restriction "you obtain an uncountable system of possible statements, among which only a countable subset are `provable', and there must certainly be `undecidable' statements."

[22] e.g. [Shulman 2012, nLab 2014]

[23] [*cf.* Church 1934, Kleene 1936]

[24] [Church 1936, Turing 1936]

[25] Since Mathematics has all the power of *Nat*.

[26] $\mathbf{Proposition}_{Nat}\triangleleft\omega\triangleright \equiv \mathbf{Proposition}_{Nat}\triangleleft 1\triangleright \oplus \mathbf{Proposition}_{Nat}\triangleleft 2\triangleright \oplus \ldots$

[27] $\mathbf{Sentence}_{Nat}\triangleleft\omega\triangleright \equiv \mathbf{Sentence}_{Nat}\triangleleft 1\triangleright \oplus \mathbf{Sentence}_{Nat}\triangleleft 2\triangleright \oplus \ldots$

[28] $\mathbf{Domain}_{Nat}\triangleleft\omega\triangleright \equiv \mathbf{Domain}_{Nat}\triangleleft 1\triangleright \oplus \mathbf{Domain}_{Nat}\triangleleft 2\triangleright \oplus \ldots$

where

$\mathbf{Domain}_{Nat}\triangleleft 1\triangleright \equiv \mathbb{N}$

$\mathbf{Domain}_{Nat}\triangleleft n+1\triangleright \equiv \mathbf{Domain}_{Nat}\triangleleft n\triangleright^{\mathbb{N}}$

[29] sometimes called "incomplete"

[30] The formal consistency theorem contradicts the claim in [Raatikainen 2015] which states:

> "For any consistent system [formal system] *F* within which a certain amount of elementary arithmetic can be carried out [for example, the formal system ℕ], the consistency of *F* cannot be proved in *F* itself."

where

> "Roughly, a formal system is a system of axioms equipped with rules of inference, which allow one to generate new theorems. The set of axioms is required to be finite or at least decidable, i.e., there must be an algorithm (an effective method) which enables one to mechanically decide whether a given statement is an axiom or not. If this condition is satisfied, the theory is called "recursively axiomatizable", or, simply, "axiomatizable". The rules of inference (of a formal system) are also effective operations, such that it can always be mechanically decided whether one has a legitimate application of a rule of inference at hand. Consequently, it is also possible to decide for any given finite sequence of formulas, whether it constitutes a genuine derivation, or a proof, in the system—given the axioms and the rules of inference of the system."

and

> "A formal system is consistent if there is no statement such that the statement itself and its negation are both derivable in the system."

The reason for the contradiction is that [Raatikainen 2015] implicitly assumed that a formal system must be able construct Gödel's proposition *I'm unprovable.*

[31] Same proof as for $\mathcal{Nat}$

[32] Same proof as for $\mathcal{Nat}$

[33] Closely related to conservation laws in physics

[34] That the closed mathematical theory $\mathcal{Nat}$ is inferentially undecidable[34] with respect to $\text{ProofsComputationalyEnumerable}_{\mathcal{Nat}}$ does not mean incompleteness with respect to the information that can be inferred because

- $\vdash\ \nVdash_{\mathcal{Nat}} \text{ProofsComputationalyEnumerable}_{\mathcal{Nat}}$
- $\vdash\ \nVdash_{\mathcal{Nat}} \neg\text{ProofsComputationalyEnumerable}_{\mathcal{Nat}}$

[35] For example, inconsistent information does not infer nonsense.

[36] Consequently, there can cannot be any escape hatch into an unformalized "meta-theory."

[37] The claim also relied on Gödel's proposition *I'm unprovable.*

[38] Formal syntax was invented long after [Gödel 1931].

[39] emphasis in original

[40] cf. [Rosental 2008]

[41] According to [Concoran 2001]:

> "after first-order logic had been isolated and had been assimilated by the logic community, people emerged who could not accept the idea that first-order logic was not comprehensive. These logicians can be viewed not as conservatives who want to reinstate an outmoded tradition but rather as radicals who want to overthrow an established tradition."

[42] for discussion see [Hewitt 2010]

[43] in an unlawful way (Einstein, a member of the editorial board, refused to support Hilbert's action)

[44] Hilbert letter to Brouwer, October 1928

[45] Gödel said "Has Wittgenstein lost his mind?"

[46] For example:

From: Harvey Friedman
Sent: Wednesday, April 20, 2016 10:53
To: Carl Hewitt
Cc: Martin Davis @cs.nyu; Dana Scott @cmu; Eric Astor @uconn; Mario Carneiro @osu; Dave Mcallester @ttic; Joe Shipman
Subject: Re: Parameterized types in the foundations of mathematics

Not if I have anything to say about it!

Harvey

On Wed, Apr 20, 2016 at 11:25 AM, Carl Hewitt wrote:

> Hi Martin,
>
> Please post the message below to FOM [Foundations of Mathematics forum].
>
> Thanks!
>
> Carl
>
> According to Harvey Friedman on the FOM Wiki: "I have not yet seen any seriously alternative foundational setup that tries to be better than ZFC in this [categoricity of models] and other respects that isn't far far worse than ZFC in other even more important respects."
>
> Of course, ZFC is a trivial consequence of parameterized types with the following definition for set of type τ:
>
> $\text{Set}\triangleleft\tau\triangleright \equiv \text{Boolean}^{\tau}$
>
> Also of course, classical mathematics can be naturally formalized using parameterized types. For example, see "Inconsistency Robustness in Foundations: Mathematics self proves its own Consistency and Other Matters" in HAL Archives.
>
> Regards,
> Carl

[47] Arthur Schopenhauer (1788-1860)

[48] For every type there is a larger type, i.e.., $\forall[\tau_1::] \rightarrow \exists[\tau_2::] \rightarrow \tau_1 \subsetneq \tau_2$

[49] **True**≠**False**

$\forall[x{:}\textbf{Boolean}] \rightarrow x=\textbf{True} \vee x=\textbf{False}$

[50] The theory of the natural numbers *Nat* is axiomatized as follows where S is the successor function:

- $0{:}\mathbb{N}$
- $+_1{:}\mathbb{N}^{\mathbb{N}}$
- $\forall[i{:}\mathbb{N}]\rightarrow\ +_1[i]\neq 0$
- $\forall[i,j{:}\mathbb{N}]\rightarrow\ +_1[i]=+_1[j]\Rightarrow i=j$
- For each order:$\mathbb{N}_+$ and P:**Proposition**◁order▷$^{\mathbb{N}}$:

  $(P[\![0_{\mathbb{N}}]\!]\wedge\forall[i{:}\mathbb{N}]\rightarrow P[\![i]\!]\Rightarrow P[\![+_1[i]]\!])\Rightarrow\forall[i{:}\mathbb{N}]\rightarrow P[\![i]\!]$

[51] The theory of the ordinals *Ord* is axiomatized as follows:

- $0{:}\mathbf{O}$
- $\forall[\alpha{:}\mathbf{O}]\rightarrow\ \alpha\geqq 0$
- Successor ordinals: $+_1{:}\mathbf{O}^{\mathbf{O}}$
  - $\forall[\alpha{:}\mathbf{O}]\rightarrow\ +_1[\alpha]>\alpha$
  - $\forall[\alpha,\beta{:}\mathbf{O}]\rightarrow\ \beta>\alpha\Rightarrow +_1[\alpha]\leqq\beta$
  - $\forall[\alpha{:}\mathbf{O}]\rightarrow\ \alpha$:**Succesor**◁**O**▷ $\Leftrightarrow\exists[\delta{:}\mathbf{O}]\rightarrow\alpha=+_1[\delta]$
- Limit ordinals: $\forall[\alpha{:}\mathbf{O},f{:}\mathbf{O}^{\mathbf{O}}]\rightarrow\ \biguplus_\alpha f{:}\mathbf{O}$
  - $\forall[\alpha{:}\mathbf{O}]\rightarrow\ \alpha$:**Limit**◁**O**▷ $\Leftrightarrow\alpha>0\wedge\nexists[\delta{:}\mathbf{O}]\rightarrow\alpha=+_1[\delta]$
  - $\forall[\alpha,\beta<\alpha{:}\mathbf{O},f{:}\mathbf{O}^{\mathbf{O}}]\rightarrow\ f[\beta]\leqq\biguplus_\alpha f$
  - $\forall[\alpha,\beta{:}\mathbf{O},f{:}\mathbf{O}^{\mathbf{O}}]\rightarrow\ (\forall[\delta<\alpha]\rightarrow f[\delta]\leqq\beta)\Rightarrow\biguplus_\alpha f\leqq\beta$
- Omega ordinals: $\forall[\alpha{:}\mathbf{O}]\rightarrow\ \boldsymbol{\omega}_\alpha{:}\mathbf{O}$
  - $\boldsymbol{\omega}_0=\mathbb{N}$
  - $\forall[\alpha{:}\mathbf{O}]\rightarrow\ \Rightarrow|\boldsymbol{\omega}_{\alpha+1}|\cong|\mathbf{Boolean}^{\boldsymbol{\omega}_\alpha}|$

    $\forall[\alpha,\beta{:}\mathbf{O}]\rightarrow\ |\beta|\cong|\boldsymbol{\omega}_{\alpha+1}|\Rightarrow\boldsymbol{\omega}_{\alpha+1}\leqq\beta$

    where $|\tau_1|\cong|\tau_2|\Leftrightarrow\exists[f{:}\tau_2{}^{\tau_1}]\rightarrow$ **OneToOneOnto**◁τ_1,τ_2▷$[f]$
  - $\forall[\alpha$:**Limit**◁**O**▷$]\rightarrow\ \boldsymbol{\omega}_\alpha=\biguplus_{\beta<\alpha}\boldsymbol{\omega}_\beta$
- Trichotomy: $\forall[\alpha,\beta{:}\mathbf{O}]\rightarrow\ \alpha<\beta\ \veebar\ \alpha=\beta\ \veebar\ \beta<\alpha$
- For each order:$\mathbb{N}_+$ and P:**Proposition**◁order▷$^{\mathbf{O}}$ the following ordinal induction axiom holds:

  $(\forall[\alpha{:}\mathbf{O}]\rightarrow\forall[\beta<\alpha{:}\mathbf{O}]\rightarrow P[\![\beta]\!]\Rightarrow P[\![\alpha]\!])\Rightarrow\forall[\alpha{:}\mathbf{O}]\rightarrow P[\![\alpha]\!]$

Ordinals have the following properties:

- Ordinals are well-ordered:
  Least:$\mathbf{O}^{\mathbf{Boolean}^{\mathbf{O}}}$
  Least[{ }] = $0_{\mathbf{O}}$
  ∀[S:$\mathbf{Boolean}^{\mathbf{O}}$]→ S≠{ } ⇨ Least[S]∈S
  ∀[S:$\mathbf{Boolean}^{\mathbf{O}}$]→ S≠{ } ⇨ ∀[α:**O**]→ α∈S ⇨ Least[S]≦α
- ∀[α:**O**]→ α=0 ⊻ α:**Succesor**◁**O**▷ ⊻ α:**Limit**◁**O**▷
- The set of all ordinals Ω is $\mathbf{Boolean}^{\mathbf{O}}$ so that:

  ∀[α:**O**]→ α∈Ω ⇔ α:**O**

  Note that it is **not** the case that Ω is of type **O**, thereby thwarting the Burali-Forti paradox

[52] Discrimination of τ_1 and τ_2

For i=1,2

- If x:τ_i, then ((τ_1⦶τ_2)[x]):(τ_1⦶τ_2) and x=((τ_1⦶τ_2)[x])↓τ_i.
- ∀[z:τ]→ z:τ_1⦶τ_2 ⇔ ∃[x:τ_i]→ z=(τ_1⦶τ_2)[x]

[53] type of 2-element list with first element of type τ_1 and with second element of type τ_2

[54] expression of type **τ.** The following axiom holds:
∀[**τ**::,e:**Expression**◁**τ**▷] →⌊**e**⌋::**τ**

[55] *if* p *then* Φ_1 *else* Φ_2

[56] x_1 is a subtype of x_2, i.e., ∀[**x:τ_1**]→ **x:τ_2**

[57] The proposition that τ is a type

[58] Φ_1, … and Φ_{n-1} infer Φ_n

[59] mutually recursive definitions of functions $\mathbf{f}_{1\text{ to }n}$

[60] mutually recursive definitions of variables $x_{1\text{ to }n}$

[61] [Nielsen 2014]

[62] According to [Scott 1967]: "As long as an idealistic manner of speaking about abstract objects is popular in mathematics, people will speak about collections of objects, and then collections of collections of ... of collections. In other words set theory is inevitable." [emphasis in original]

[63] According to [Scott 1967]: "there is only one satisfactory way of avoiding the paradoxes: namely, the use of some form of the *theory of types*... the best way to regard Zermelo's theory is as a simplification and extension of Russell's ...*simple* theory of types. Now Russell made his types *explicit* in his notation and Zermelo left them *implicit*. It is a mistake to leave something so important invisible..."

[64] [Church 1956; Boolos 1975; Corcoran 1973, 1980]. Also, Classical Direct Logic is *not* a univalent homotopy type theory [Awodey, Pelayo, and Warren 2013].

[65] [Isaacson 2007]

[66] [Dedekind 1888] According to [Isaacson 2007]:

> "Second-order quantification is significant for philosophy of mathematics since it is the means by which mathematical structures may be characterized. But it is also significant for mathematics itself. It is the means by which the significant distinction can be made between the independence of Euclid's Fifth postulate from the other postulates of geometry and the independence of Cantor's Continuum hypothesis [conjecture] from the axioms of set theory. The independence of the Fifth postulate rejects the fact, which can be expressed and established using second-order logic, that there are different geometries, in one of which the Fifth postulate holds (is true), in others of which it is false."

[67] For each type **X** that satisfies the Dedekind axioms there is a (unique) isomorphism $I{:}\mathbf{X}^{\mathbb{N}}$ and inductively defined as follows:

1. $I[0_{\mathbb{N}}] \equiv 0_{\mathbf{X}}$
2. $I[+_1[j]] \equiv +_1^{\mathbf{X}}[I[j]]$

Using proofs by induction on $\mathbb{N}$ and **X**, the following follow:

1. I is defined for every $\mathbb{N}$
2. I is one-to-one: $\forall[k,j{:}\mathbb{N}]\to I[k]=I[j] \Rightarrow k=j$

   First show **Lemma** by induction on k: $\forall[k{:}\mathbb{N}]\to I[k]=0_{\mathbf{X}} \Rightarrow k=0_{\mathbb{N}}$

   Base: Suppose $k=0_{\mathbb{N}}$. QED.

   Induction: Suppose $I[k]=0_{\mathbf{X}} \Rightarrow k=0_{\mathbb{N}}$

   To show: $I[+_1[k]]=0_{\mathbf{X}} \Rightarrow +_1[k]=0_{\mathbf{X}}$

   $I[+_1[k]]= +_1^{\mathbf{X}}{}_{\mathbf{X}}[I[k]]$

   Therefore $I[+_1[k]]=0_{\mathbf{X}} \Rightarrow 0_{\mathbf{X}}=+_1^{\mathbf{X}}[I[k]]$

   which is a contradiction

   To show: $\forall[k,j{:}\mathbb{N}]\to I[k]=I[j] \Rightarrow i=j$

   Proof: Induction on $P[m{:}\mathbb{N}]{:}\text{Proposition}\triangleleft 1\triangleright \equiv$

   $\forall[k,j{\leqq}m{:}\mathbb{N}]\to I[k]=I[j] \Rightarrow i=j$

   Base: Suppose $m=k=j=0_{\mathbb{N}}$. QED.

   Induction: Suppose $\forall[k,j{\leqq}m{:}\mathbb{N}]\to I[k]=I[j] \Rightarrow k=j$

   To show: $\forall[0_{\mathbb{N}}<k,j<+_1[m]]\to I[k]=I[j] \Rightarrow k=j$

   $\exists[k_0,j_0{\leqq}m{:}\mathbb{N}]\to k=+_1[k_0] \wedge j=+_1[j_0]$ because $0_{\mathbb{N}}<k,j{\leqq}m$

   $k_0=j_0$ since $k_0,j_0{\leqq}m$, $k,j<+_1[m]$ and

   $I[k_0]=I[j_0] \Rightarrow k_0=j_0$ by induction hypothesis

   $+_1[k_0]=+_1[j_0]$ and therefore $k=j$
3. the range of I is all of **X**: $\forall[y{:}\mathbf{X}]\to \exists[k{:}\mathbb{N}]\to I[k]=y$

   Proof: Induction on $P[y{:}\mathbf{X}]{:}\text{Proposition}\triangleleft 1\triangleright \equiv \exists[k{:}\mathbb{N}]\to I[k]=y$

   Base: Suppose $y=0_{\mathbf{X}}$. To show $\exists[k{:}\mathbb{N}]\to I[k]= 0_{\mathbf{X}}$. Clearly $I[0_{\mathbb{N}}]=0_{\mathbf{X}}$

   Induction: Suppose $y>0_{\mathbf{X}}{:}\mathbf{X}$ and $\exists[k{:}\mathbb{N}]\to I[k]=y$. Let $I[k_0]=y$.

   To show $\exists[k{:}\mathbb{N}]\to I[k]=+_1^{\mathbf{X}}[y]$.

   It follows from $I[+_1[k_0]]] = +_1^{\mathbf{X}}[I[k_0]] = +_1^{\mathbf{X}}[y]$

4. I is a homomorphism: $I[0_{\mathbb{N}}]=0_{X}$ and $\forall[j{:}\mathbb{N}]\rightarrow I[+_1[j]] = +_1^{X}[I[j]]$

Proof: Induction on $P[j{:}\mathbb{N}]{:}\text{Proposition}\triangleleft 1\triangleright \equiv I[+_1[j]]=+_1^{X}[I[j]]]$

Base: $I[+_1[0_{\mathbb{N}}]]=+_1^{X}[I[0_{\mathbb{N}}]]$ by definition of I

Induction: Suppose $\forall[i{:}\mathbb{N}]\rightarrow I[+_1[j]] = +_1^{X}[I[j]]$

To show: $\forall[j{:}\mathbb{N}]\rightarrow I[+_1[+_1[j]]] = +_1^{X}[I[+_1[j]]]$

$I[+_1[+_1[j]]] = +_1^{X}[I[+_1[j]]]$ by definition of I

5. $I^{-1}{:}\mathbb{N}^{X}$ is a homomorphism:

$I^{-1}[0_{X}]= 0_{\mathbb{N}}$ and $\forall[y{:}X]\rightarrow I^{-1}[+_1^{X}[y]]=+_1[I^{-1}[y]]$

Proof:

To show: $I^{-1}[0_{X}]=0_{\mathbb{N}}$.

Let $i=I^{-1}[0_{X}]$. Therefore $I[k]=0_{X}$ and $k=0_{\mathbb{N}}$.

To show: $\forall[y{:}X]\rightarrow I^{-1}[+_1^{X}[y]]=+_1[I^{-1}[y]]]$

Induction on $P[y{:}X]{:}\text{Proposition}\triangleleft 1\triangleright \equiv I^{-1}+_1^{X}[y]]=+_1[I^{-1}[y]]]$

Base: To show: $I^{-1}[+_1^{X}[0_{X}]]=+_1[I^{-1}[0_{X}]]]=+_1[0_{\mathbb{N}}]$

Let $k=I^{-1}[+_1^{X}[0_{X}]]$. Therefore $I[k]=+_1^{X}[0_{X}]$ and $k=+_1[0_{\mathbb{N}}]$.

Induction: Suppose $\forall[j{:}\mathbb{N}]\rightarrow I[+_1[j]]=+_1^{X}[I[j]]$

To show: $\forall[i{:}\mathbb{N}]\rightarrow I[+_1[+_1[j]]]=+_1^{X}[I[+_1[j]]]$

$I[+_1[+_1[j]]]= +_1^{X}[I[+_1[j]]]$ by definition of I

6. I is the unique isomorphism: If $g{:}X^{\mathbb{N}}$ is an isomorphism then g=I

Proof: Induction on $P[j{:}\mathbb{N}]{:}\text{Proposition}\triangleleft 1\triangleright \equiv I[j]=g[j]$

Base: $I[0_{\mathbb{N}}]=0_{X}$. $g[0_{\mathbb{N}}]=0_{X}$ because g is an isomorphism.

Therefore $I[0_{\mathbb{N}}]=g[0_{\mathbb{N}}]$

Induction: Suppose $I[j]=g[j]$.

To show: $I[+_1[j]]=g[+_1[j]]$

$I[+_1[j]]= +_1^{X}[I[j]]= +_1^{X}[g[j]]= g[+_1[j]]$

68 For example, there are nondeterministic Turing machines that the theory $\mathbb{N}$ proves always halt that cannot be proved to halt in the cut-down first-order theory.

69 [Dedekind 1888]

The following can be used to characterize the real numbers ($\mathbb{R}$[69]) up to isomorphism with a unique isomorphism:

$\forall[S{:}\mathbf{Set}\triangleleft\mathbb{R}\triangleright]\rightarrow\ S\neq\{\ \}_{\mathbb{R}} \wedge \text{Bounded}[S] \Rightarrow \text{HasLeastUpperBound}[S]$

where

$\text{Bounded}[S{:}\mathbf{Set}\triangleleft\mathbb{R}\triangleright] \Leftrightarrow \exists[b{:}\mathbb{R}]\rightarrow \text{UpperBound}[b, S]$

$\text{UpperBound}[b{:}\mathbb{R}, S{:}\mathbf{Set}\triangleleft\mathbb{R}\triangleright] \Leftrightarrow\ b\in S \wedge \forall[x\in S]\rightarrow x\leqq b$

$\text{HasLeastUpperBound}[S{:}\mathbf{Set}\triangleleft\mathbb{R}\triangleright]] \Leftrightarrow \exists[b{:}\mathbb{R}]\rightarrow \text{LeastUpperBound}[b, S]$

$\text{LeastUpperBound}[b{:}\mathbb{R}, S{:}\mathbf{Set}\triangleleft\mathbb{R}\triangleright]$

$\Leftrightarrow\ \text{UpperBound}[b,S] \wedge \forall[x\in S]\rightarrow \text{UpperBound}[x,S] \Rightarrow x\leqq b$

70 Robinson [1961]

[71] For each type $\mathbf{X}$ that satisfies the theory *Ord* there is a (unique) isomorphism $I:\mathbf{X}^{\mathbf{O}}$ inductively defined as follows:

$I[0_{\mathbf{O}}] \equiv 0_{\mathbf{X}}$

$\forall[\alpha:\mathbf{O}]\rightarrow I[+_1[\alpha]] \equiv +_1^{\mathbf{X}}[I[\alpha]]$

$\forall[\alpha:\mathbf{Limit}\triangleleft\mathbf{O}\triangleright]\rightarrow I[\alpha] \equiv y$

where $y:\mathbf{X} \wedge \forall[\beta<\alpha]\rightarrow y\leqq_{\mathbf{X}} I[\beta]$

$\wedge \forall[z:X]\rightarrow (\forall[\beta<\alpha]\rightarrow z\leqq_{\mathbf{X}} I[\beta]) \Rightarrow y\leqq_{\mathbf{X}} z$

Using proofs by ordinal induction on $\mathbf{O}$ and $\mathbf{X}$, the following follow:

1. I is defined for every $\mathbf{O}$
2. I is one-to-one: $\forall[\alpha,\beta:\mathbf{O}]\rightarrow I[\alpha] = I[\beta] \Rightarrow \alpha=\beta$
3. The range of I is all of $\mathbf{X}$: $\forall[y:\mathbf{X}]\rightarrow \exists[\alpha:\mathbf{O}]\rightarrow I[\alpha] = y$
4. I is a homomorphism:
   - $I[0_{\mathbf{O}}] = 0_{\mathbf{X}}$
   - $\forall[\alpha:\mathbf{O}]\rightarrow I[+_1[\alpha]] = +_1^{\mathbf{X}}[I[\alpha]]$
   - $\forall[\alpha:\mathbf{Limit}\triangleleft\mathbf{O}\triangleright, f:\mathbf{O}^{\mathbf{O}}]\rightarrow I[\uplus_\alpha f] = \uplus^{\mathbf{X}}_{f[\alpha]} I\circ f\circ I^{-1}$
5. $I^{-1}:\mathbf{O}^{\mathbf{X}}$ is a homomorphism
6. I is the unique isomorphism: If $g:\mathbf{X}^{\mathbf{O}}$ is an isomorphism then g=I

[72] [Bourbaki 1972; Fantechi, et. al. 2005]

[73] This implies, for example, that no set is an element of itself.

[74] Proof: Suppose $S{:}\text{Sets}\triangleleft\tau\triangleright$ and therefore $\exists[\alpha{:}\text{O}]\rightarrow S{:}\text{Set}^{\alpha}\triangleleft\tau\triangleright$.

Proof by induction on

$P[\beta{:}\text{O}]{:}\text{Proposition}\triangleleft 1\triangleright \equiv \forall[X\in S]\rightarrow X{:}\text{Set}^{\beta}\triangleleft\tau\triangleright$

Assume: $(\forall[\beta<\alpha{:}\text{O}]\rightarrow \forall[X\in S]\rightarrow X{:}\text{Set}^{\beta}\triangleleft\tau\triangleright)$

$\Rightarrow \forall[X\in S]\rightarrow X{:}\text{Set}^{\alpha}\triangleleft\tau\triangleright$

Show: $\forall[X\in S]\rightarrow X{:}\text{Set}^{\alpha}\triangleleft\tau\triangleright$

Assume: $X\in S$

Show $X{:}\text{Set}^{\alpha}\triangleleft\tau\triangleright$

Proof by cases on α

1. $X{:}\text{Set}^{0}\triangleleft\tau\triangleright$

   $X{:}\text{Boolean}^{\tau}$

2. $\forall[\alpha{:}\text{O}]\rightarrow \text{Sets}^{\alpha}\triangleleft\tau\triangleright = \text{Set}\triangleleft\text{Set}^{\alpha\text{-}1}\triangleleft\tau\triangleright\triangleright$

   $X{:}\text{Set}^{\alpha\text{-}1}\triangleleft\tau\triangleright$ QED by induction hypothesis

3. $\forall[\alpha{:}\text{Limit}\triangleleft\text{O}\triangleright]\rightarrow \exists[\beta<\alpha, Y{:}\text{Set}^{\beta}\triangleleft\tau\triangleright]\rightarrow X\in Y$

   QED by induction hypothesis

[75] Proof: Suppose $S{:}\text{Sets}\triangleleft\tau\triangleright$ and therefore $\exists[\alpha{:}\text{O}]\rightarrow S{:}\text{Sets}^{\alpha}\triangleleft\tau\triangleright$

$S{:}\text{Sets}^{\alpha}\triangleleft\tau\triangleright$

Show: $\text{Boolean}^{S}{:}\text{Sets}\triangleleft\tau\triangleright$

$\text{Boolean}^{S}{:}\text{Sets}^{\alpha+1}\triangleleft\tau\triangleright$ QED

[76] Proof by induction on

$P[\alpha{:}\text{O}]{:}\text{Proposition}\triangleleft 1\triangleright \equiv \forall[S{:}\text{Sets}^{\alpha}\triangleleft\tau\triangleright]\rightarrow \cup S{:}\text{Sets}\triangleleft\tau\triangleright$

Assume: $\forall[\beta<\alpha{:}\text{O}]\rightarrow \forall[S{:}\text{Sets}^{\beta}\triangleleft\tau\triangleright]\rightarrow \cup S{:}\text{Sets}\triangleleft\tau\triangleright$

Show: $\forall[S{:}\text{Sets}^{\alpha}\triangleleft\tau\triangleright]\rightarrow \cup S{:}\text{Sets}\triangleleft\tau\triangleright$

Assume: $S{:}\text{Sets}^{\alpha}\triangleleft\tau\triangleright$

Show: $\cup S{:}\text{Sets}\triangleleft\tau\triangleright$

$\forall[X{:}\text{Sets}\triangleleft\tau\triangleright]\rightarrow X\in\cup S \Leftrightarrow \exists[Y\in S]\rightarrow X\in Y$

$\forall[X{:}\text{Sets}\triangleleft\tau\triangleright]\rightarrow X\in\cup S \Leftrightarrow \exists[\beta<\alpha{:}\text{O}, Y{:}\text{Sets}^{\beta}\triangleleft\tau\triangleright]\rightarrow X\in Y$

$\forall[X{:}\text{Sets}\triangleleft\tau\triangleright]\rightarrow X\in\cup S \Rightarrow X{:}\text{Sets}\triangleleft\tau\triangleright$

QED by definition of $\text{Sets}\triangleleft\tau\triangleright$

77 Suppose f:**Sets**◁τ▷$^{\textbf{Sets}◁τ▷}$ and S:**Sets**◁τ▷
Show **Image**◁τ▷[f, S]:**Sets**◁τ▷
Proof by induction on
P[α:**O**] ⇔ S:**Set**$^{\alpha}$◁τ▷ ⇒ **Image**◁τ▷[f, S]:**Sets**◁τ▷
Suppose ∀[β<α:**O**]→ S:**Set**$^{\beta}$◁τ▷ ⇒ **Image**◁τ▷[f, S]:**Sets**◁τ▷
Show S:**Set**$^{\alpha}$◁τ▷ ⇒ **Image**◁τ▷[f, S]:**Sets** ◁τ▷
Suppose S:**Set**$^{\alpha}$◁τ▷
Show **Image**◁τ▷[f, S]:**Sets**◁τ▷
∀[y:**Sets**◁τ▷] → y:**Image**◁τ▷[f, S] ⇔ ∃[x∈S]→ f[x]=y
Show ∀[y:**Sets**◁τ▷] → y∈**Image**◁τ▷[f, S] ⇒ y:**Sets**◁τ▷
Suppose y:**Sets**◁τ▷ ∧ y∈**Image**◁τ▷[f, S]
Show y:**Sets**◁τ▷
∃[x∈S]→ f[x]=y because y∈**Image**◁τ▷[f, S]
∃[β<α:**O**]→ x:**Set**$^{\beta}$◁τ▷ because x∈S and S:**Set**$^{\alpha}$◁τ▷
Image◁τ▷[f, x]:**Sets**◁τ▷ by induction hypothesis
Show f[x]:**Sets**◁τ▷
Suppose z∈f[x]
Show z:**Sets**◁τ▷
z∈**Sets**◁τ▷ because z∈f[x] and **Image**◁τ▷[f, x]:**Sets**◁τ▷
f[x]:**Sets**◁τ▷
y:**Sets**◁τ▷ because f[x]=y

78 [Mizar; Matuszewski1 and Rudnicki: 2005; Naumowicz and Artur Korniłowicz 2009; Naumowicz 2009]

79 Note that this proof is fundamentally different from the categoricity proof in [Martin 2015].

80 [Feferman 1986, pg. 1 and 8]

81 Wittgenstein in 1937 published in Wittgenstein 1956, p. 50e and p. 51e]

[82] Subsequent further discussion of Wittgenstein's criticism of Gödel's writings has unfortunately misunderstood Wittgenstein.

For example, [Berto 2009] granted that proof theoretically if P⇔⊬P, then:

1) ⊢⊬P

However, the above has proof consequences as follows:

2) ⊢P because (⊬P)⇔P in 1) above
3) ⊢⊢P because of 2) above
4) ⊢¬P because (⊢P)⇔¬*P* in 3) above

Of course, 2) and 4) are a manifest contradiction in mathematics that has been obtained without any additional "'semantic' story" that [Berto 2009] claimed is required for Wittgenstein's argument that contradiction in mathematics "*is what comes of making up such sentences*." [Wittgenstein 1956, p. 51e]

[83] Chaitin's criticism is partially supported by the fact that even Gödel himself agreed that the subsequent proof of incompleteness by Church/Turing based on computational undecidability was fundamental in proving that there is no total recursive procedure that can decide provability of a proposition of the Dedekind theory *Nat* of natural numbers. There must be an inferentially undecidable proposition for *Nat* because otherwise provability of any proposition could be computationally decided by enumerating all theorems until the proposition or its negation occurs. However, Gödel, Church, Turing, and many other logicians continued for a long time to believe in the importance of Gödel's proof based on his proposition *I'm unprovable.*

[84] [Wittgenstein in 1937 published in Wittgenstein 1956, p. 50e and p. 51e]

[85] [Wang 1997] pg. 197.

[86] According to Sol Feferman, Gödel was "the most important logician of the 20th century" and according to John Von Neumann he was "the greatest logician since Aristotle." [Feferman 1986, pg. 1 and 8]

[87] In Direct Logic, 1) can be inferred from 3) within $\mathcal{PL}$.

[88] In Classical Direct Logic, $(\Phi\vdash_{\mathcal{PL}}\Psi) \Leftrightarrow \vdash_{\mathcal{PL}}(\Phi\Rightarrow\Psi)$.

[89] In Direct Logic, 3) can be infers from 1) within $\mathcal{PL}$.

[90] Note that the rule of Soundness [*i.e.* $(\vdash\Phi)\Rightarrow\Phi$] does not involve any coding of propositions as integers. If the soundness were allowed in $\mathcal{PL}$, then it would be inconsistent because every sentence would be provable in $\mathcal{PL}$ by [Löb 1955] which is the following:[90]

$$(\vdash_{\mathcal{PL}}((\vdash_{\mathcal{PL}}\Phi)\Rightarrow\Phi)) \Rightarrow \vdash_{\mathcal{PL}}\Phi$$

[91] In 1666, England's House of Commons introduced a bill against atheism and blasphemy, singling out Hobbes' Leviathan. Oxford university condemned and burnt Leviathan four years after the death of Hobbes in 1679.

[92] ContinuumForReals is defined as follows:

ContinuumForReals ⇔ ∄[S:**Boolean**$^{\mathbb{N}}$]→ |ℕ| < |S| < |**Boolean**$^{\mathbb{N}}$|

ContinuumForReals has been proved for well-behaved subsets of the reals, such as Borel sets as follows:

ContinuumForBorelSets ⇔ ∄[S:**BorelSet**]→ |ℕ| < |S| < |**Boolean**$^{\mathbb{N}}$|

where a Borel Set is formed from the countable union, countable intersection, and relative complement of open sets

That ContinuumForReals is an open problem is not so important for Computer Science because for ContinuumForComputableReals is immediate because the computable real numbers are enumerable.

For less well behaved subset of ℝ, ContinuumForReals remains an open problem.

Note that it is important not to confuse ContinuumForReals with ContinuumForFirstOrderZFC. *FirstOrderZFC* has countably many first-order propositions as axioms. [Cohen 1963] proved the following theorem which is much weaker than ContinuumForReals because sets in the models of *FirstOrderZFC* do **not** include all of **Boolean**$^{\mathbb{N}}$ and the theory *FirstOrderZFC* is much weaker than the theory *Sets*$_{\mathbb{N}}$:

- $\nVdash_{FirstOrderZFC}$ ContinuumForFirstOrderZFC
- $\nVdash_{FirstOrderZFC}$ ¬ContinuumForFirstOrderZFC

Cohen's result above is very far from being able to decide the following:

$\vdash_{Sets_{\mathbb{N}}}$ ContinuumForReals

[93] [Zermelo 1930, van Dalen 1998, Ebbinghaus 2007]

[94] First-order theories fall prey to paradoxes like the Löwenheim–Skolem theorems (*e.g.* any first-order theory of the real numbers has a countable model). First-order theorists have used the weakness of first-order logic to prove results that do not hold in stronger formalisms such as Direct Logic [Cohen 1963, Barwise 1985].

[95] *e.g.* the theory *RealClosedField* [Tarski 1951]

[96] unique up to isomorphism via a unique isomorphism

[97] Rejection of the First-Order Thesis resolves the seeming paradox between the formal proof in this article that Mathematics formally proves its own formal consistency and the formal proof that 'Every "strong enough" formal system that admits a proof of its own consistency is actually inconsistent.' [Paulson 2014]. Although Mathematics is "strong enough" the absence of propositions (constructed using the Y untyped fixed point operator on propositions) blocks the proof of formal inconsistency to which Paulson referred.

[98] In other words, the paradox that concerned [Church 1934] (because it could mean the demise of formal mathematical logic) has been transformed into fundamental theorem of foundations!

[99] In formalizing Gödel's proof, [Shankar 1994] and [O'Connor 2005] followed Gödel in using integers to code sentences using the Y untyped fixed point operator on propositions.

[100] Wittgenstein in 1937 published in Wittgenstein 1956, p. 50e and p. 51e]

[101] According to [Russell 1908] page 240:

> *[W]hen a man says "I am lying", we must interpret him as meaning: "There is a proposition of order n which I affirm and which is false". This is a proposition of order n+1; hence the man is not affirming any proposition of order n*

[102] [Yanofsky 2013 page 328] expressed concern about Löb's paradox:

> *we must restrict the fixed-point machine in order to avoid proving false statements* [using Löb's argument]. *Such a restriction might seem strange because the proof that the fixed-point machine works seems applicable to all* [functions on untyped statements]. *But restrict we must.*

Yanofsky proposed solving above problem posed by Löb's paradox using systems of logic that are so weak that they cannot abstract their own sentences. Unfortunately, such weak systems are inadequate for Computer Science. Instead of weakening logic, Direct Logic adopted the strategy of using types for mathematics that does not allow the Y fixed point operator for propositions and sentences.

[103] using definition of BSet

[104] using definition of BExpression

[105] substituting BNumber for n

[ii] using what [Carnap 1934] later called the "Diagonal Lemma" which is equivalent to the Y the fixed point operator on propositions.

[106] Wittgenstein developed the proof below [lines 5) thru 7)] that contradiction in mathematics results from allowing the proposition *I'm unprovable.* used in the incompleteness results of [Gödel 1931].

[107] [Wittgenstein 1937 published in Wittgenstein 1956, p. 50e and p. 51e]