



HAL
open science

Strong Types for Direct Logic

Carl Hewitt

► **To cite this version:**

Carl Hewitt. Strong Types for Direct Logic. Symposium on Logic and Collaboration for Intelligent Applications, Mar 2017, Stanford, United States. hal-01566393v1

HAL Id: hal-01566393

<https://hal.science/hal-01566393v1>

Submitted on 21 Jul 2017 (v1), last revised 17 Oct 2018 (v14)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Strong Types for Direct Logic

Carl Hewitt

<http://plus.google.com/+CarlHewitt-StandardIoT>

This article is dedicated to Alonzo Church, Richard Dedekind, Bertrand Russell, and Ernst Zermelo.

Abstract

This article follows on the introductory article “Direct Logic for Intelligent Applications” [Hewitt 2017a]. Strong Types enable new mathematical theorems to be proved including the Formal Consistency of Mathematics. Also, Strong Types are extremely important in Direct Logic because they block all known paradoxes [Cantini and Bruni 2017]. Blocking known paradoxes makes Direct Logic safer for use in Intelligent Applications by preventing security holes.

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

Mathematics here means the common foundation of all classical mathematical theories from Euclid to the mathematics used to prove Fermat's Last [McLarty 2010]. 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.

Mathematical Foundation for Computer Science

Computer Science brought different concerns and a new perspective to mathematical foundations including the following requirements:² [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 .

Formalism of Direct Logic

The aims of logic should be the creation of “a unified conceptual apparatus which would supply a common basis for the whole of human knowledge.”
[Tarski 1940]

In Direct Logic, unrestricted recursion is allowed in programs. For example,

- There are uncountably many Actors.³ For example, $\text{Real}_{\bullet}[\]$ can output any real numberⁱ between 0 and 1 where

$\text{Real}_{\bullet}[\] \equiv [(0 \text{ either } 1), \forall \text{Postpone Real}_{\bullet}[\]]$

where

- $(0 \text{ either } 1)$ is the nondeterministic choice of 0 or 1,
 - $[\textit{first}, \forall \textit{rest}]$ is the list that begins with *first* and whose remainder is *rest*, and
 - **Postpone expression** delays execution of *expression* until the value is needed.
- 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,

$p \equiv \lambda[x:\mathbb{R}] (\lambda[y:\mathbb{R}] (y=x))$

defines a different predicate $p[x]$ for each real number x , which holds for only one real number, namely x .ⁱⁱ

Sentencesⁱⁱⁱ can be abstracted into propositions that can be asserted. Furthermore, expressions^{iv} can be abstracted into Actors (*e.g.*, 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 can be generated by inference from documentation and by generalization from use cases.

ⁱ using binary representation.

ⁱⁱ For example $(p[3])[y]$ holds if and only if $y=3$.

ⁱⁱⁱ which are grammar tree structures

^{iv} which are grammar tree structures

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. $\forall [n:\mathbb{N}] \exists [m:\mathbb{N}] m > n$

i.e., proposition that for every \mathbb{N} there is a larger \mathbb{N}

Sentences

e.g. $(\forall [n:\mathbb{N}] (\exists [m:\mathbb{N}] (m > n)))$

*i.e., sentence for proposition that f
for every \mathbb{N} there is a larger \mathbb{N}*

Strings

e.g. $"(\forall [n:\mathbb{N}] (\exists [m:\mathbb{N}] (m > n)))"$

*i.e., string for sentence for proposition that
for every \mathbb{N} there is a larger \mathbb{N}*

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:

Note: types in Direct Logic are much stronger than constructive types with constructive logic because Classical Direct Logic has all of the power of Classical Mathematics.

Mathematics Self Proves that it is Open

Mathematics here means the common foundation of all classical mathematical theories from Euclid to the mathematics used to prove Fermat's Last [McLarty 2010].ⁱ Mathematics proves that it is open in the sense that it can prove that its theorems cannot be provably computationally enumerated:

Theorem \vdash Mathematics is Open, *i.e.*,

$\neg \vdash$ TheoremsComputationallyEnumerable

Proof.ⁱⁱ

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

\vdash TheoremsComputationallyEnumerable. Then there is a provably computable total procedure TheoremsEnumerator: $[\mathbb{N}] \rightarrow \mathbf{Theorem}$ such that it is provable that the following holds:

$\forall [p: \mathbf{Theorem}] (\vdash p) \Rightarrow \exists [i: \mathbb{N}] \text{TheoremsEnumerator} \cdot [i] = p$

A subset of the theorems of order n are those proving that certain procedures $[\mathbb{N}] \rightarrow \mathbb{N}$ are total. Consequently, there is a ProvedTotalsEnumerator: $[\mathbb{N}] \rightarrow ([\mathbb{N}] \rightarrow \mathbb{N})$ that computationally enumerates the provably total computable procedures $[\mathbb{N}] \rightarrow \mathbb{N}$ that can be used in the implementation of the following procedure:

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

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 undecidabilityⁱⁱⁱ of mathematical theories. The above theorem that Mathematics is open provides another independent argument for the inexhaustibility of Mathematics.

ⁱ Consequently, Mathematics evolves and is not fixed.

ⁱⁱ 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.*”

ⁱⁱⁱ See section immediately below.

Mathematics self proves its own consistency (contra Gödel *et. al.*)

The following rules are fundamental to Mathematics⁴:

- Derivation by Contradiction, *i.e.* $(\neg\Phi \Rightarrow (\Theta \wedge \neg\Theta)) \vdash \Phi$, which says that a proposition can be proved showing that its negation implies a contradiction.
- A theorem can be used in a proof, *i.e.* $(\vdash\Phi) \Rightarrow \Phi$

Theorem: Mathematics self proves its own consistency.

Formal Derivation. Suppose to obtain a contradiction, that mathematics is formally inconsistent, *i.e.*, $\neg\text{Consistent}$. By definition of formal consistency, there is some mathematical proposition Ψ such that $\vdash(\Psi \wedge \neg\Psi)$. By the rule of Existential Elimination, there is some proposition Ψ_0 such that $\vdash(\Psi_0 \wedge \neg\Psi_0)$ which by the rule of Theorem Use means $\Psi_0 \wedge \neg\Psi_0$, which is a contradiction. Thus, $\vdash\text{Consistent}$ by the rule of Proof by Contradiction.

1) $\neg\text{Consistent}$ // hypothesis to derive a contradiction just in this subargument
2) $\exists[\Psi:\text{Proposition}] \rightarrow \vdash(\Psi \wedge \neg\Psi)$ // 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)
$\vdash\text{Consistent}$ // rule of Proof by Contradiction using 1) and 4)

Natural Deductionⁱ Derivation of Consistency of Mathematics

ⁱ [Jaśkowski 1934] developed Natural Deduction

Please note the following points:

- The above argument formally mathematically proves that Mathematics is consistent and that **it is not a premise of the theorem that Mathematics is consistent.**
- Mathematics was designed for consistent axioms and consequently the rules of Mathematics can be used to prove consistency regardless of other axioms.⁵

The above derivation 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.

The above recently developed self-proof of consistency shows that the current common understanding that Gödel proved “Mathematics cannot prove its own consistency, if it is consistent” is inaccurate.⁶

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.”⁷

Direct Logic develops foundations for Mathematics by deriving sets from types *and* categorical axioms for the natural numbers and ordinals.

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 characterization? 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.”⁸

Classical Direct Logic is much stronger than first-order axiomatizations of set theory in that it provides categoricity for natural numbers \mathbb{N} , reals \mathbb{R} , and ordinals \mathbb{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. For example, nonstandard models cause problems in model checking if a model has specified properties.

Proof by Natural Number Induction

The mathematical theory \mathcal{Nat} categorically axiomatizes using the following induction axiom:

$$\forall [P:\text{Proposition}\langle 1 \rangle^{\mathbb{N}}] (P[0] \wedge \forall [i:\mathbb{N}] P[i] \Rightarrow P[i+1]) \Rightarrow \forall [i:\mathbb{N}] P[i]$$

The above proposition is of type $\text{Proposition}\langle 2 \rangle$. However, $\forall [i:\mathbb{N}] P[i]$ in the above proposition is of type $\text{Proposition}\langle 1 \rangle$. Quine famously criticized 2nd-order logic as nothing more than “set theory in sheep’s clothing” [Quine 1970, pg. 66]. However, the induction axiom above does simply formalizes more a more natural form the induction schema than used in the infinitely large number of axioms used to axiomatized the Natural Numbers in 1st order logic.⁹

The other axioms of \mathcal{Nat} are as follows:

- $0:\mathbb{N}$
- $\forall [i:\mathbb{N}] +_1[i]:\mathbb{N}$
- $\exists [i:\mathbb{N}] +_1[i]=0$
- $\forall [i,j:\mathbb{N}] +_1[i]=+_1[j] \Rightarrow i=j$

Theorem $\text{ProofsComputationallyEnumerable}\langle \text{Nat} \rangle$ is unprovable in Nat .

Proof:¹⁰

Suppose to obtain a contradiction that

$$\vdash_{\text{Nat}} \text{ProofsComputationallyEnumerable}[\text{Nat}]$$

Then there is a provable in Nat computable total procedure $\text{TheoremsEnumerator}\langle \text{Nat} \rangle: [\mathbb{N}] \rightarrow \text{Theorem}\langle \text{Nat} \rangle$ such that it is provable in Nat that

$$\forall [p: \text{Theorem}\langle \text{Nat} \rangle] \exists [i: \mathbb{N}] \text{TheoremsEnumerator}\langle \text{Nat} \rangle.[i] = p$$

A subset of the proofs in Nat are those proving that certain procedures $[\mathbb{N}] \rightarrow \mathbb{N}$ are total. Consequently, there is a procedure

$$\text{ProvedTotalsEnumerator}\langle \text{Nat} \rangle: [\mathbb{N}] \rightarrow ([\mathbb{N}] \rightarrow \mathbb{N})$$

that enumerates the provable in Nat total computable procedures $[\mathbb{N}] \rightarrow \mathbb{N}$ that can be used in the implementation of the following procedure:

$$\text{Diagonal}.[i: \mathbb{N}]: \mathbb{N} \equiv 1 + (\text{ProvedTotalsEnumerator}\langle \text{Nat} \rangle.[i]).[i]$$

Consequently:

- Diagonal is a provable in Nat total procedure because it is implemented using computable provable in Nat total procedures.
- Diagonal is not a provable in Nat total procedure because it differs from every other computable provable in Nat total procedure.

The above contradiction completes the proof.¹¹

Theorem $\vdash_{\text{Nat}} \text{Consistent}[\text{Nat}]$

Proof: Suppose to derive an inconsistency that $\neg \text{Consistent}[\text{Nat}]$. By the definition of inconsistency for Nat , there is some proposition Ψ such that

$$\vdash_{\text{Nat}} (\Psi \wedge \neg \Psi).$$

By Existential Elimination, there is some proposition Ψ_0 such that $\vdash_{\text{Nat}} (\Psi_0 \wedge \neg \Psi_0)$ which can be used to infer in Nat that $\Psi_0 \wedge \neg \Psi_0$.

The above contradiction completes the proof.

Theorem (Categoricity of Natural Numbers \mathbb{N}):¹²

If \mathbf{X} be a type satisfying the categorical axioms for the natural numbers Nat , then \mathbf{X} is isomorphic to \mathbb{N}^{13} , which is strictly more powerful than a 1st order theory of Natural Numbers.¹⁴



Richard Dedekind

Theorem (Categoricity of Real Numbers \mathbb{R}):¹⁵

If \mathbf{X} is a type satisfying the categorical axioms for the real numbers $Real$, then \mathbf{X} is (uniquely) isomorphic to \mathbb{R} , which is strictly more powerful than the first-order theory of real closed fields.¹⁶

Theorem (Model Soundness of Nat): $(\vdash_{Nat} \Psi) \Leftrightarrow \models \Psi$

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

Theory of Ordinals

A theory of the ordinals can be axiomatized¹⁷ using a 2nd order ordinal induction axiom as follows: For each order: \mathbb{N}_+ and P : **Proposition** $\langle order \rangle^O$:

$$(\forall [\alpha:O] \forall [\beta < \alpha:O] P[\beta] \Rightarrow P[\alpha]) \Leftrightarrow \forall [\alpha:O] P[\alpha]$$

In order to fill out the ordinals, the following limit axioms are included:

- $\forall [\alpha:O, f:O^O] \cup_{\alpha} f:O$
- $\forall [\alpha, \beta:O, f:O^O] \beta < \cup_{\alpha} f \Leftrightarrow \exists [\delta < \alpha] \beta \leq f[\delta]$
- $\forall [\alpha, \beta:O, f:O^O] (\forall [\delta < \alpha] f[\delta] \leq \beta) \Leftrightarrow \cup_{\alpha} f \leq \beta$

In order to guarantee that there are uncountable ordinals, the following axioms are included:

- $\omega_0 = \mathbb{N}$
- $\forall [\alpha:O] \alpha > 0_O \Leftrightarrow |\omega_{\alpha}| = |\mathbf{Boolean}^{\cup_{\beta < \alpha} \omega_{\beta}}|$
- $\forall [\alpha, \beta:O] |\beta| = |\omega_{\alpha}| \Leftrightarrow \omega_{\alpha} \leq \beta$
 - where $|\tau_1| = |\tau_2| \Leftrightarrow \exists [f:\tau_2^{\tau_1}] \text{OneToOneOnto} \langle \tau_1, \tau_2 \rangle [f]$
 - $\text{OneToOne} \langle \tau_1, \tau_2 \rangle [f:\tau_2^{\tau_1}] \Leftrightarrow \forall [x_1, x_2:\tau_1] f[x_1] = f[x_2] \Rightarrow x_1 = x_2$
 - $\text{OneToOneOnto} \langle \tau_1, \tau_2 \rangle [f:\tau_2^{\tau_1}] \Leftrightarrow \text{OneToOne} \langle \tau_1, \tau_2 \rangle [f:\tau_2^{\tau_1}] \wedge \forall [y:\tau_2] \exists [x:\tau_1] f[x] = y$

Theorem (Categoricity of Ordinals \mathbf{O}):

If \mathbf{X} be a type satisfying the axioms the theory of the ordinals \mathcal{Ord} , then \mathbf{X} is (uniquely) isomorphic to \mathbf{O} .¹⁸

Theorem (Model Soundness of \mathcal{Ord}): $(\vdash_{\mathcal{Ord}} \Psi) \Rightarrow \models \Psi$

Proof: Suppose $\vdash_{\mathcal{Nat}} \Psi$. The conclusion immediately follows because the axioms for the theory \mathcal{Ord} hold in the type \mathbf{O} .

Type Choice

$$\forall [f: (\mathbf{Boolean}^\sigma)^\tau] \exists [\text{choice}: \sigma^\tau] \forall [x: \tau] f[x] \neq \{\} \Rightarrow \text{choice}[x] \in f[x]$$

Sets _{τ} defined using strong parameterized types

The type $\mathbf{Set}\langle\tau\rangle$ can be defined as follows:

$$\mathbf{Set}\langle\tau\rangle \equiv \mathbf{Boolean}^\tau$$

Of course set membership is defined as follows:

$$\forall [x: \tau, S: \mathbf{Set}\langle\tau\rangle] x \in S \Leftrightarrow S[x] = \text{True}$$

Inductive definition:

1. $\mathbf{Set}^0\langle\tau\rangle \equiv \mathbf{Boolean}^\tau$
2. $\mathbf{Set}^{\alpha+1}\langle\tau\rangle \equiv \mathbf{Set}\langle\mathbf{Set}^\alpha\langle\tau\rangle\rangle$
3. $\alpha: \mathbf{Limit}\langle\mathbf{O}\rangle \Rightarrow (S: \mathbf{Set}^\alpha\langle\tau\rangle \Leftrightarrow \forall [X \in S] \exists [\beta < \alpha: \mathbf{O}, Y: \mathbf{Set}^\beta\langle\tau\rangle] X \in Y)$

$$S: \mathbf{Sets}\langle\tau\rangle \Leftrightarrow \exists [\alpha: \mathbf{O}] S: \mathbf{Set}^\alpha\langle\tau\rangle$$

The properties below mean that $\mathbf{Sets}\langle\tau\rangle$ is a "universe" of mathematical discourse.¹⁹

- Foundation: There are no downward infinite membership chains.²⁰
- Transitivity of \in :²¹ $\forall[S:\mathbf{Sets}\langle\tau\rangle] \forall[X\in S] X?:\mathbf{Sets}\langle\tau\rangle$
- Powerset:²² $\forall[S:\mathbf{Sets}\langle\tau\rangle] \mathbf{Boolean}^S:\mathbf{Sets}\langle\tau\rangle$
- Union:²³

$$\forall[S:\mathbf{Sets}\langle\tau\rangle] \cup S:\mathbf{Sets}\langle\tau\rangle$$

$$\forall[S:\mathbf{Sets}\langle\tau\rangle] \forall[X:\mathbf{Sets}\langle\tau\rangle] X\in\cup S \Leftrightarrow \exists[Y\in S] X\in Y$$

- Replacement:²⁴ The function image of any set is also a set, *i.e.*:

$$\text{Image}\langle\tau\rangle:\mathbf{Sets}\langle\tau\rangle^{[\mathbf{Sets}\langle\tau\rangle^{\mathbf{Sets}\langle\tau\rangle}, \mathbf{Sets}\langle\tau\rangle]}$$

$$\forall[f:\mathbf{Sets}\langle\tau\rangle^{\mathbf{Sets}\langle\tau\rangle}, S:\mathbf{Sets}\langle\tau\rangle]$$

$$\forall[y:\mathbf{Sets}\langle\tau\rangle] y\in\text{Image}\langle\tau\rangle[f, S] \Leftrightarrow \exists[x\in S] f[x]=y$$

$\mathbf{Sets}\langle\tau\rangle$ is *much stronger* than first-order ZFC.²⁵

Theorem. \mathbf{Set}_{τ} is categorical via a (unique) isomorphism.

Proof:²⁶ Suppose that \mathbf{X} satisfies the axioms for \mathbf{Set}_{τ} .

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

$$1. S:\mathbf{Set}^0\langle\tau\rangle$$

$$I[S] \cong S$$

$$2. S:\mathbf{Set}^{\alpha+1}\langle\tau\rangle$$

$$Z\in\mathbf{x}I[S] \Leftrightarrow \exists[Y:\mathbf{Set}^{\alpha}\langle\tau\rangle] I[Y]\in\mathbf{x}Z$$

$$3. S:\mathbf{Set}^{\alpha}\langle\tau\rangle \text{ and } \alpha:\mathbf{Limit}\langle\mathbf{O}\rangle$$

$$Z\in\mathbf{x}I[S] \Leftrightarrow \exists[\beta<\alpha:\mathbf{O}, Y:\mathbf{Set}^{\beta}\langle\tau\rangle] I[Y]\in\mathbf{x}Z$$

I is a unique isomorphism:

- I is one to one
- The range of I is \mathbf{X}
- I is a homomorphism:
 - $I[\{ \ } \mathbf{Sets}_{\langle \tau \rangle}] = \{ \ } \mathbf{X}$
 - $\forall [S1, S2: \mathbf{Sets}_{\langle \tau \rangle}] I[S1 \cup S2] = I[S1] \cup_{\mathbf{X}} I[S2]$
 - $\forall [S1, S2: \mathbf{Sets}_{\langle \tau \rangle}] I[S1 \cap S2] = I[S1] \cap_{\mathbf{X}} I[S2]$
 - $\forall [S1, S2: \mathbf{Sets}_{\langle \tau \rangle}] I[S1 - S2] = I[S1] -_{\mathbf{X}} I[S2]$
 - $\forall [S: \mathbf{Sets}_{\langle \tau \rangle}] I[US] = \cup_{\mathbf{X}} \{I[x] \mid x \in S\}$
- $I^{-1}: \mathbf{Sets}_{\langle \tau \rangle} \mathbf{X}$ is a homomorphism
- I is a unique isomorphism: If $g: \mathbf{X} \mathbf{Set}_{\langle \tau \rangle}$ is an isomorphism, then $g = I$

Theorem (Model Soundness of \mathbf{Set}_{τ}): $(\vdash_{\mathbf{Set}_{\tau}} \Psi) \Rightarrow \models \Psi$

Proof: Suppose $\vdash_{\mathbf{Set}_{\tau}} \Psi$. The conclusion immediately follows because the axioms for the theory \mathbf{Set}_{τ} hold in the type $\mathbf{Sets}_{\langle \tau \rangle}$.

Lambda Induction

The axiom is of Lambda Induction is as follows:ⁱ

$$\begin{aligned}
 & \forall [P: \mathbf{Proposition}_{\langle 1 \rangle} \Lambda_{\langle \tau \rangle}] \\
 & (P[I_{\langle \tau_1 \rangle}] \wedge P[K_{\langle \tau_1, \tau_2 \rangle}] \wedge P[S_{\langle \tau_1, \tau_2, \tau_3 \rangle}] \wedge P[\mathbf{Fix}_{\langle \tau_1 \rangle}] \\
 & \quad \wedge \forall [f_1: \tau_1, f_2: \tau_2] P[f_1] \wedge P[f_2] \Rightarrow P[K_{\langle \tau_1, \tau_2 \rangle} \cdot [f_1, f_2]] \\
 & \quad \wedge \forall [f_1: \tau_1, f_2: \tau_2, f_3: \tau_3] P[f_1] \wedge P[f_2] \wedge P[f_3] \Rightarrow P[S_{\langle \tau_1, \tau_2, \tau_3 \rangle} \cdot [f_1, f_2, f_3]] \\
 & \quad \wedge \forall [f_1: \tau_1] P[f_1] \Rightarrow P[\mathbf{Fix}_{\langle \tau_1 \rangle} \cdot [f_1]]) \\
 & \Rightarrow \forall [f: \Lambda_{\langle \tau \rangle}] P[f]
 \end{aligned}$$

Type $\Lambda_{\langle \tau \rangle}$ defined using strong parameterized types

The type $\Lambda_{\langle \tau \rangle}$ can be defined by Induction as follows:

1. $([\tau] \rightarrow \tau): \mathbf{Type}_{\langle \Lambda_{\langle \tau \rangle} \rangle}$
2. $\tau_1, \tau_2: \mathbf{Type}_{\langle \Lambda_{\langle \tau \rangle} \rangle} \Rightarrow ([\tau_1] \rightarrow \tau_2), [\tau_1, \tau_2], \tau_1 \oplus \tau_2: \mathbf{Type}_{\langle \Lambda_{\langle \tau \rangle} \rangle}$

Functional Definition: $\mathbf{Functional}_{\langle \tau_1, \tau_2 \rangle} \equiv [([\tau_1] \rightarrow \tau_2)] \rightarrow ([\tau_1] \rightarrow \tau_2)$

ⁱ $\tau_1, \tau_2, \tau_3: \mathbf{Type}_{\langle \Lambda_{\langle \tau \rangle} \rangle}$

Theory Lam_τ

In addition to Lambda Induction (above), the theory Lam_τ has the following axioms:ⁱ

1. $I\langle\tau_1\rangle:([\tau_1]\rightarrow\tau_1)$
 $I\langle\tau_1\rangle.\mathbf{f}[\tau_1] = f$
2. $K\langle\tau_1,\tau_2\rangle:([\tau_1,\tau_2]\rightarrow\tau_1)$
 $K\langle\tau_1,\tau_2\rangle.\mathbf{f}_1[\tau_1,\tau_2] = f_1$
3. $S\langle\tau_1,\tau_2,\tau_3\rangle:([\tau_1]\rightarrow([\tau_2]\rightarrow\tau_3), [[\tau_1]\rightarrow\tau_2])\rightarrow\tau_3, \tau_3\rightarrow\tau_3)$
 $S\langle\tau_1,\tau_2,\tau_3\rangle.\mathbf{f}_1([\tau_1]\rightarrow([\tau_2]\rightarrow\tau_3)), \mathbf{f}_2([\tau_1]\rightarrow\tau_2), \mathbf{f}_3:\tau_1] = (f_1.\mathbf{f}_3).\mathbf{f}_2.\mathbf{f}_3]$
4. $\mathbf{Fix}\langle\tau_1,\tau_1\rangle:([\mathbf{Functional}\langle\tau_1,\tau_1\rangle]\rightarrow\mathbf{Functional}\langle\tau_1,\tau_1\rangle)^{27}$
 $\mathbf{Fix}\langle\tau_1,\tau_1\rangle.\mathbf{F}[\mathbf{Functional}\langle\tau_1,\tau_1\rangle] = F.\mathbf{Fix}\langle\tau_1,\tau_1\rangle.\mathbf{F}]$

Equivalence for Lam_τ :ⁱⁱ $\forall[f:([\tau_1]\rightarrow\tau_2)] f = \lambda[x:\tau_1] f.\mathbf{x}]$

Convergence: $\forall[f:([\tau_1]\rightarrow\tau_2),x:\tau_1] f.\mathbf{x}] \downarrow \Leftrightarrow \exists[y:\tau_2] f.\mathbf{x}] = y$

Approximation: $\forall[f_1,f_2:([\tau_1]\rightarrow\tau_2)] f_1 \leq f_2 \Leftrightarrow \forall[x:\tau_1] f_1.\mathbf{x}] \downarrow \Rightarrow f_1.\mathbf{x}] = f_2.\mathbf{x}]$

Bottom: $\perp\langle\tau_1,\tau_2\rangle.\mathbf{x}:\tau_1]:\tau_2 \equiv x$

Note that $\forall[x:\tau_1] \neg\perp\langle\tau_1,\tau_2\rangle.\mathbf{x}] \downarrow$ and $\forall[f:([\tau_1]\rightarrow\tau_2)] \perp\langle\tau_1,\tau_2\rangle \leq f$

Monotone Definition:

$$F:\mathbf{Monotone}\langle\tau_1,\tau_2\rangle \Leftrightarrow F:\mathbf{Functional}\langle\tau_1,\tau_2\rangle \wedge \forall[g:([\tau_1]\rightarrow\tau_2)] g \leq F.\mathbf{g}]$$

Limit Theorem: $\forall[F:\mathbf{Monotone}\langle\tau_1,\tau_2\rangle] F = \text{limit}_{i:\mathbb{N}_+} F^i.\mathbf{[\perp\langle\tau_1,\tau_2\rangle]}^{28}$

Adequacy Theorem: Every Direct Logic procedure in over τ can be implemented in Lam_τ .

ⁱ $\tau_1,\tau_2,\tau_3:\mathbf{Type}\langle\Lambda\langle\tau\rangle\rangle$

ⁱⁱ Because of Equivalence for Lam_τ , the domain of [Scott 2015] is *not* a valid model of Lam_τ .

Theorem. Lam_τ is categorical via a (unique) isomorphism.

Proof: Suppose that \mathbf{X} satisfies the axioms for Lam_τ .

By lambda induction, the isomorphism $I: \mathbf{X}^{\Lambda\langle\tau\rangle}$ is defined as follows:ⁱ

1. $I[I\langle\tau_1\rangle] \equiv \mathbf{I}_x\langle\tau_1\rangle$
2. $I[K\langle\tau_1, \tau_2\rangle] \equiv \mathbf{K}_x\langle\tau_1, \tau_2\rangle$
3. $I[S\langle\tau_1, \tau_2, \tau_3\rangle] \equiv \mathbf{S}_x\langle\tau_1, \tau_2, \tau_3\rangle$
4. $I[\mathbf{Fix}\langle\tau_1\rangle] \equiv \mathbf{Fix}_x\langle\tau_1\rangle$
5. $\forall [f_1:\tau_1, f_2:[\tau_1]\rightarrow\tau_2] I[f_2.\mathbf{[}f_1]] \equiv I[f_2]_x.\mathbf{[}I[f_1]]$

I is a unique isomorphism:

- I is one to one
- The range of I is \mathbf{X}
- I is a homomorphism:
- $I^{-1}:\Lambda\langle\tau\rangle^{\mathbf{X}}$ is a homomorphism
- I is a unique isomorphism: If $g:\mathbf{X}^{\Lambda\langle\tau\rangle}$ is an isomorphism, then $g=I$

Theorem (Model Soundness of Lam_τ): $(\vdash_{Lam_\tau} \Psi) \Leftrightarrow \vDash \Psi$

Proof: Suppose $\vdash_{Lam_\tau} \Psi$. The conclusion immediately follows because the axioms for the theory Lam_τ hold in the type $\Lambda\langle\tau\rangle$.

ⁱ $\tau_1, \tau_2, \tau_3: \mathbf{Type}\langle\Lambda\langle\tau\rangle\rangle$

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:²⁹

- **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).³⁰
- **Argumentation and propositions have been 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 1934].
- **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.³¹
- **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 fired³² Brouwer as an associate editor of *Mathematische Annalen* because of “incompatibility of our views on fundamental matters”³³ 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 categorical axiomatization of the natural numbers. In face of Wittgenstein's devastating criticism, Gödel insinuated³⁴ 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.³⁵

Conclusion

Strong Types enable new mathematical theorems to be proved including the Formal Consistency of Mathematics. Also, Strong Types enable proofs of the Categoricity of axiomatizations of the ordinals and the cumulative hierarchy of sets of a type.

Furthermore, Strong Types are extremely important in Direct Logic because they block all known paradoxes [Cantini and Bruni 2017]. Blocking known paradoxes makes Direct Logic safer for use in Intelligent Applications by preventing security holes. For example, Strong Types block the following paradoxes: Berry [Russell 1906], Burali-Forti [Burali-Forti 1897], Church [Church 1934], Curry [Curry 1941], Girard [Coquand 1986], and Liar [Eubulides of Miletus], and Löb [Löb 1955].

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.

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. [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 undecidabilityⁱ 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

ⁱ sometimes called “incompleteness”

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).

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'mUnprovable* 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 consistencyⁱ has been disproved.
- Consequently, Gödel's writings have led to increased controversy over mathematical foundations.

ⁱ Gödel's writing was accepted doctrine by some theoreticians for over eight decades.

The development of Direct Logic has strengthened the position of working mathematicians as follows:ⁱ

- 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.

Acknowledgements

Tom Costello, Eric Martin, Per Stenstrom, and Johan van Benthem made very helpful comments and suggestions. Interactions with John Woods were very helpful in developing a resolution to Church's Paradox. A suggestion by Gordon Plotkin led to the introduction of the fixed point operator in the categorical theory of the lambda calculus.

ⁱ Of course, Direct Logic must preserve as much previous learning as possible.

Bibliography

- A. I. Arruda. "Aspects of the historical development of paraconsistent logic" In *Paraconsistent Logic: Essays on the Inconsistent* Philosophia Verlag. 1989
- Andrea Asperti, Herman Geuvers, Raja Natarajan. *Social processes, program verification and all that* "Mathematical Structures in Computer Science" Cambridge University Press. 2009.
- Jean-Yves Béziau, Walter Carnielli, and Dov Gabbay. Ed. *Handbook of Paraconsistency* College Publications Kings College London. 2007
- Cristian Calude and Giuseppe Longo. "The Deluge of Spurious Correlations in Big Data" *Foundations of Science* Springer. March 2016.
- Carlo Cellucci "Gödel's Incompleteness Theorem and the Philosophy of Open Systems" *Kurt Gödel: Actes du Colloque, Neuchâtel 13-14 juin 1991*, Travaux de logique N. 7, Centre de Recherches Sémiologiques, University de Neuchâtel. <http://w3.uniroma1.it/cellucci/documents/Goedel.pdf>
- Gregory Chaitin Interview in *Dangerous Knowledge* BBC4 documentary. 2007.
- Alan Chalmers. "What is this thing called science?" Open University Press. 1999.
- Haskell Curry. "Some Aspects of the Problem of Mathematical Rigor" *Bulletin of the American Mathematical Society* Vol. 4. 1941.
- 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.
- Will Clinger. *Foundations of Actor Semantics* MIT Mathematics Doctoral Dissertation. June 1981.
- 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.
- Michael Dunn and Greg Restall. "Relevance Logic" in *The Handbook of Philosophical Logic, second edition*. Dov Gabbay and Franz Guenther (editors), Kluwer. 2002.
- Freeman Dyson. *Heretical Thoughts about Science and Society* Boston University. November 1, 2005.
- T. S. Eliot. *Four Quartets*. Harcourt. 1943.
- Solomon Feferman "Axioms for determinateness and truth" *Review of Symbolic Logic*. 2008.

- A. C. W. Finkelstein, D. Gabbay, A. Hunter, J. Kramer, and B. Nuseibeh, “Inconsistency Handling in Multi-Perspective Specifications” *Transactions on Software Engineering*, August 1994.
- Dov Gabbay and Anthony Hunter. “Making inconsistency respectable: A logical framework for inconsistency in reasoning (Part 1). *Fundamentals of Artificial Intelligence Research '91*, Springer-Verlag.. 1991.
- Dov Gabbay and Anthony Hunter. “Making inconsistency respectable: A logical framework of inconsistency in reasoning (Part 2). *Symbolic and Quantitative Approaches to Reasoning and Uncertainty LNCS*, Springer-Verlag, 1992.
- 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.
- Louis Hansen. *Another approach to autonomous vehicles – slow and steady* SiliconValley.com. Jan. 27, 2017.
- Carl Hewitt. *Planner: A Language for Proving Theorems in Robots* IJCAI. 1969.
- Carl Hewitt. “Procedural Embedding of Knowledge In Planner” *IJCAI* 1971.
- Carl Hewitt and John Woods assisted by Jane Spurr, editors. *Inconsistency Robustness*. College Publications. 2015.
- Carl Hewitt. 2015a. *Actor Model of Computation for Scalable Robust Information Systems* in “Inconsistency Robustness” College Publications. 2015.
- Carl Hewitt. 2015b. *ActorScript™ extension of C#®, Java®, Objective C®, C++, JavaScript®, and SystemVerilog using iAdaptive™ concurrency* in *Inconsistency Robustness*. College Publications. 2015.
- Carl Hewitt. 2015c. *Inconsistency Robustness in Logic Programs* in “Inconsistency Robustness” College Publications. 2015.
- Carl Hewitt. 2015d. *Formalizing common sense reasoning for scalable inconsistency-robust information coordination using Direct Logic™ Reasoning and the Actor Model* in “Inconsistency Robustness” College Publications. 2015.
- Carl Hewitt. 2015e. *Inconsistency Robustness in Logic Programs* in “Inconsistency Robustness” College Publications. 2015.
- Carl Hewitt. 2015f. *Actors for CyberThings*. Erlang Keynote. YouTube. March 23, 2015.
- Carl Hewitt. 2016b. *Future Cyberdefenses Will Defeat Cyberattacks* CACM. August 2016.
- Carl Hewitt 2016c. *Islets™ Protect Sensitive IoT Information: Verifiably ending use of sensitive IoT information for mass surveillance fosters (international) commerce* SSRN WP 2836282. 2016.
- Carl Hewitt 2017a. *Direct Logic for Intelligent Applications* Logic and Collaboration for Intelligent Applications. Stanford. March 30-31, 2017.

- Carl Hewitt 2017b. *Axiomatics for Inconsistency Robust Direct Logic* Logic and Collaboration for Intelligent Applications. Stanford. March 30-31, 2017.
- John Ioannidis. 2005a “Why Most Published Research Findings Are False” *PLoS Medicine*. 2(8): e124.
- John Ioannidis. 2005b “Contradicted and Initially Stronger Effects in Highly Cited Clinical Research” *JAMA*. 294.2.218.
- John Ioannidis. “Why Most Published Research Findings Are False: Author's Reply to Goodman and Greenland” *PLoS Medicine*. 4(6): e215. 2007
- Stanisław Jaśkowski “On the Rules of Suppositions in Formal Logic” *Studia Logica* 1, 1934. (reprinted in: *Polish logic 1920-1939*, Oxford University Press, 1967.
- Stanisław Jaśkowski. “Propositional calculus for contradictory deductive systems” *Studia Logica*. 24 (1969) *Rachunek zdań dla systemów dedukcyjnych sprzecznych* in: *Studia Societatis Scientiarum Torunensis, Sectio A, Vol. I, No. 5*, Toruń 1948.
- Eric Kao. “Proof by self-refutation and excluded middle lead to explosion” *Inconsistency Robustness 2011* Stanford. August 16-18, 2011.
- Marsi Kendall. *Uber pulls self-driving cars from San Francisco streets, bowing to regulators' demands* Mercury News. December 21, 2016.
- Morris Kline. *Mathematical thought from ancient to modern times* Oxford University Press. 1972.
- 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.
- John McCarthy. *Programs with common sense* Symposium on Mechanization of Thought Processes. National Physical Laboratory. Teddington, England. 1958.
- Lorenzo Magnani. *The eco-cognitive model of abduction II: Irrelevance and implausibility exculpated* *Journal of Applied Logic*. February 2, 2016.
- David Malone. *Dangerous Knowledge* BBC4 documentary. 2007. http://www.dailymotion.com/playlist/x1cbyd_xSilverPhinx_bbc-dangerous-knowledge/1
- 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.
- Annemarie Mol and Marc Berg. “Principles and Practices of Medicine: the Coexistence of various Anaemias” *Culture, Medicine, and Psychiatry* 1994.

- Annemarie Mol. *The Body Multiple: ontology in medical practice* Duke University Press. 2002.
- Nick Nielsen Alternative Foundations/philosophical February 28, 2014. <http://www.cs.nyu.edu/pipermail/fom/2014-February/017861.html>
- Bashar Graham Priest, and Richard Routley “The History of Paraconsistent Logic” in *Paraconsistent Logic: Essays on the Inconsistent* Philosophia Verlag. 1989.
- William Quine. *Philosophy of Logic*. Prentice Hall. 1970.
- J. Rimland, M. Ballora, and W. Shumaker. *Beyond visualization of big data: a multi-stage data exploration approach using visualization, sonification, and storification* in SPIE Defense, Security, and Sensing. 2013.
- Scott Rosenberg. *Dreaming in Code*. Crown Publishers. 2007.
- Bertrand Russell. *Principles of Mathematics* Norton. 1903.
- Bertrand Russell. “Les paradoxes de la logique” *Revue de métaphysique et de morale*. 1906.
- Bertrand Russell and Alfred Whitehead, *Principia Mathematica* (3 volumes). Cambridge University Press. 1910-1913.
- Dana Scott. *Setoids/Modest Sets/PERS: Adding and Using Types with a Type-free λ -Calculus* Domains XII. August 2015.
- Alfred Tarski *Introduction to Logic* Oxford University Press. 1940 (and many subsequent editions).
- Rineke Verbrugge “Provability Logic” *The Stanford Encyclopedia of Philosophy* 2010.
- Ludwig Wittgenstein. 1956. *Bemerkungen über die Grundlagen der Mathematik/Remarks on the Foundations of Mathematics, Revised Edition* Basil Blackwell. 1978
- John Woods. “Inconsistency: Its present impacts and future prospects” *Inconsistency Robustness* 2015.

Appendix 1: Notation of Direct Logic

- *Type i.e.*, a type is a *discrimination*⁴¹ of the following:
 - **Boolean**::⁴², **N**::⁴³, **O**::⁴⁴ and **Type** $\langle\tau\rangle$:: where τ ::⁴⁵
 - **Proposition** $\langle\text{order}\rangle$:: and **Sentence** $\langle\text{order}\rangle$:: where order:**N**₊
 - $\tau_1 \circ \tau_2$::⁴⁶, $[\tau_1, \tau_2]$::⁴⁷, $[\tau_1] \rightarrow \tau_2$::ⁱ and $\tau_2^{\tau_1}$::ⁱⁱ where τ_1 :: and τ_2 ::
 - **Set** $\langle\tau\rangle$::ⁱⁱⁱ and **Expression** $\langle\tau\rangle$::⁴⁸ where τ ::
- *Propositions, i.e.*, a **Proposition** is a *discrimination* of the following:
 - $\neg\Phi$:**Proposition** $\langle\text{order}\rangle$ where Φ :**Proposition** $\langle\text{order}\rangle$ ^{iv} and order:**N**₊
 - $\Phi \wedge \Psi, \Phi \vee \Psi, \Phi \Rightarrow \Psi, \Phi \Leftrightarrow \Psi$:**Proposition** $\langle\text{order}\rangle$ where Φ, Ψ :**Proposition** $\langle\text{order}\rangle$ and order:**N**₊
 - $(p \diamond \text{True}^s \Phi_1, \text{False}^s \Phi_2)$:**Proposition** $\langle\text{order}\rangle$ where p :**Boolean**, Φ, Ψ :**Proposition** $\langle\text{order}\rangle$ ⁴⁹ and order:**N**₊
 - $x_1 = x_2$:**Proposition** $\langle 1 \rangle$ where x_1, x_2 : τ and τ ::
 - $s_1 \subseteq s_2$:**Proposition** $\langle 1 \rangle$ where s_1, s_2 :**Set** $\langle\tau\rangle$ and τ ::
 - $x \in s$:**Proposition** $\langle 1 \rangle$ where x : τ , s :**Set** $\langle\tau\rangle$ and τ ::
 - $\tau_1 \sqsubseteq \tau_2$:**Proposition** $\langle 1 \rangle$ ⁵⁰ where τ_1 :: and τ_2 ::
 - $(x::)$:**Proposition** $\langle 1 \rangle$ ⁵¹
 - $(x:\tau)$:**Proposition** $\langle 1 \rangle$ where τ ::
 - $p[x]$:**Proposition** $\langle\text{order}+1\rangle$ ^v where x : τ , p :**Proposition** $\langle\text{order}\rangle$ ^{τ} and order:**N**₊ If τ is nonpropositional then $p[x]$ is also of type **Proposition** $\langle\text{order}\rangle$.
 - $(\Phi_1, \dots, \Phi_{n-1} \stackrel{D}{T} \Phi_n)$:**Proposition** $\langle\text{order}\rangle$ ⁵² where p :**Proof**, T :**Theory**, $\Phi_{1 \text{ to } n}$:**Proposition** $\langle\text{order}\rangle$ and order:**N**₊
 - $\lfloor s \rfloor$:**Proposition** $\langle\text{order}\rangle$ where s :**Sentence** $\langle\text{order}\rangle$ with no free variables and order:**N**₊

ⁱ Type of computable procedures from τ_1 into τ_2 .

If $f:([\tau_1] \rightarrow \tau_2)$ and $x:\tau_1$, then $f.[x]:\tau_2$.

ⁱⁱ Type of functions from τ_1 into τ_2 .

If $f:\tau_2^{\sigma_1}$ and $x:\tau_1$, then $f[x]:\tau_2$.

ⁱⁱⁱ **Set** $\langle\tau\rangle$ is a type parametrized by the type τ . In Java and C++, parametrized types are called "generics", "<" is used for \langle , and ">" is used for \rangle . The following axiom holds: $\forall[\tau::, s:\text{Set}\langle\tau\rangle, x \in s] x:\tau$

^{iv} **Proposition** $\langle\text{order}\rangle$ is the parametrized type consisting of type **Proposition** parametrized by order.

^v The type of $p[x]$ means that the **Y** fixed point construction 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** $\langle\tau\rangle$ is a *discrimination* of the following:
 - $(x):\mathbf{Constant}\langle\tau\rangle$ where $x:\tau$ and $\tau::$
 - $x:\mathbf{Expression}\langle\tau\rangle$ where $x:\mathbf{Constant}\langle\tau\rangle$ and $\tau::$
 - $x:\mathbf{Expression}\langle\tau\rangle$ where $x:\mathbf{Variable}\langle\tau\rangle$ and $\tau::$
 - $(f_1[x_1:\tau_1]:\sigma_1\equiv d_1, \dots, f_n[x_n:\tau_n]:\sigma_n\equiv d_n^{53}, y):\mathbf{Expression}\langle\tau\rangle$ where for i in 1 to n , $f_i:\mathbf{Variable}\langle\sigma_i^{\tau_i}\rangle$ in d_i and $y, x_i:\mathbf{Variable}\langle\tau_i\rangle$ in $d_i, d_i:\mathbf{Expression}\langle\sigma_i\rangle$, $y:\mathbf{Expression}\langle\tau\rangle$, and $\tau_i::$
 - $(x_1:\tau_1\equiv d_1, \dots, x_n\equiv d_n^{54}, y):\mathbf{Expression}\langle\tau\rangle$ where for i in 1 to n , $x_i:\mathbf{Variable}\langle\tau_i\rangle$ in d_i and $y, d_i:\mathbf{Expression}\langle\sigma_i\rangle, y:\mathbf{Expression}\langle\tau\rangle$, and $\tau_i::$
 - $(e_1\circ e_2):\mathbf{Expression}\langle\tau_1\circ\tau_2\rangle, ([e_1, e_2]):\mathbf{Expression}\langle[\tau_1, \tau_2]\rangle$, $([e_1]\rightarrow e_2):\mathbf{Expression}\langle[\tau_1]\rightarrow\tau_2\rangle$ and $(e_2^{e_1}):\mathbf{Expression}\langle\tau_2^{\tau_1}\rangle$ where $e_1:\mathbf{Expression}\langle\tau_1\rangle, e_2:\mathbf{Expression}\langle\tau_2\rangle, \tau_1::$ and $\tau_2::$
 - $(e_1 \diamond \mathbf{True}^e e_2, \mathbf{False}^e e_3):\mathbf{Expression}\langle\tau\rangle^i$ where $e_1:\mathbf{Expression}\langle\mathbf{Boolean}\rangle, e_2, e_3:\mathbf{Expression}\langle\tau\rangle$ and $\tau::$
 - $(\lambda[x:\tau_1] e):\mathbf{Expression}\langle\tau_2^{\tau_1}\rangle$ where $e:\mathbf{Expression}\langle\tau_2\rangle, x:\mathbf{Variable}\langle\tau_1\rangle$ in e , and $\tau_1, \tau_2::$
 - $(e[x]):\mathbf{Expression}\langle\tau_2\rangle$ where $e:\mathbf{Expression}\langle\tau_2^{\tau_1}\rangle, x:\mathbf{Expression}\langle\tau_1\rangle, \tau_1::$ and $\tau_2::$
 - $(e.[x]):\mathbf{Expression}\langle\tau_2\rangle$ where $e:\mathbf{Expression}\langle[\tau_1]\rightarrow\tau_2\rangle, x:\mathbf{Expression}\langle\tau_1\rangle, \tau_1::$ and $\tau_2::$
 - **Sentence** $\langle\text{order}\rangle \sqsubseteq \mathbf{Expression}\langle\mathbf{Sentence}\langle\text{order}\rangle\rangle$ where $\text{order}:\mathbf{N}_+$
 - $\lfloor e \rfloor:\tau$ where $e:\mathbf{Expression}\langle\tau\rangle$ with no free variables and $\tau::$

ⁱ (*if* e_1 *then* e_2 *else* e_3)

- **Sentences**, i.e., a **Sentence** is a *discrimination* of the following:
 - (\mathbf{x}) :**Sentence** \langle order+1 \rangle ⁱ where \mathbf{x} :**Variable** \langle **Sentence** \langle order \rangle \rangle and order:**N**₊
 - $(\neg\mathbf{s})$:**Sentence** \langle order \rangle where \mathbf{s} :**Sentence** \langle order \rangle and order:**N**₊
 - $(\mathbf{s}_1\wedge\mathbf{s}_2), (\mathbf{s}_1\vee\mathbf{s}_2), (\mathbf{s}_1\Rightarrow\mathbf{s}_2), (\mathbf{s}_1\Leftrightarrow\mathbf{s}_2)$:**Sentence** \langle order \rangle where $\mathbf{s}_1, \mathbf{s}_2$:**Sentence** \langle order \rangle and order:**N**₊
 - $(\mathbf{e} \diamond \text{True}^{\circ} \mathbf{s}_1, \text{False}^{\circ} \mathbf{s}_2)$ ⁱⁱ:**Sentence** \langle order \rangle where \mathbf{e} :**Expression** \langle **Boolean** \rangle , $\mathbf{s}_1, \mathbf{s}_2$:**Sentence** \langle order \rangle and order:**N**₊
 - $(\mathbf{e}_1=\mathbf{e}_2)$:**Sentence** \langle 1 \rangle where $\mathbf{e}_1, \mathbf{e}_2$:**Expression** \langle τ \rangle and τ ::
 - $(\mathbf{e}_1\sqsubseteq\mathbf{e}_2)$:**Sentence** \langle 1 \rangle where $\mathbf{e}_1, \mathbf{e}_2$:**Expression** \langle τ_1 \rangle , $\tau_1:\tau_2$ and τ_2 ::
 - $(\mathbf{e}_1\subseteq\mathbf{e}_2)$:**Sentence** \langle 1 \rangle where $\mathbf{e}_1, \mathbf{e}_2$:**Expression** \langle **Set** \langle τ \rangle \rangle and τ ::
 - $(\mathbf{e}_1\in\mathbf{e}_2)$:**Sentence** \langle 1 \rangle where \mathbf{e}_1 :**Expression** \langle τ \rangle , \mathbf{e}_2 :**Expression** \langle **Set** \langle τ \rangle \rangle and τ ::
 - $(\mathbf{e}_1:\mathbf{e}_2)$:**Sentence** \langle 1 \rangle where \mathbf{e}_1 :**Expression** \langle τ_1 \rangle , \mathbf{e}_2 :**Expression** \langle τ_2 \rangle $\tau_1:\tau_3$, $\tau_2:\tau_4$ and τ_3, τ_4 ::
 - $(\mathbf{e}::)$:**Sentence** \langle 1 \rangle where \mathbf{e} :**Expression** \langle τ \rangle and τ ::
 - $(\forall[\mathbf{x}:\tau_1] \mathbf{s}), (\exists[\mathbf{x}:\tau_1] \mathbf{s})$:**Sentence** \langle order \rangle where \mathbf{x} :**Variable** \langle τ_1 \rangle in \mathbf{s} , \mathbf{s} :**Sentence** \langle order \rangle and order:**N**₊
 - $(\mathbf{p}[\mathbf{x}])$:**Sentence** \langle order+1 \rangle ⁱⁱⁱ where \mathbf{x} :**Expression** \langle τ \rangle , \mathbf{p} :**Expression** \langle **Sentence** \langle order \rangle τ \rangle , τ :: and order:**N**₊ If τ is nonpropositional then $(\mathbf{p}[\mathbf{x}])$ is also of type **Sentence** \langle order \rangle .
 - $(\mathbf{s}_1, \dots, \mathbf{s}_{n-1} \vdash_{\tau}^{\mathbf{p}} \mathbf{s}_n)$:**Sentence** \langle order \rangle where τ :**Expression** \langle **Theory** \rangle , \mathbf{s}_1 to \mathbf{s}_n :**Sentence** \langle order \rangle , \mathbf{p} :**Expression** \langle **Proof** \rangle and order:**N**₊
 - $\lfloor \mathbf{s} \rfloor$:**Proposition** \langle order \rangle where \mathbf{s} :**Sentence** \langle order \rangle \rangle , order:**N**₊ and there are no free variables in \mathbf{s} .^{iv}

ⁱ The type of (\mathbf{x}) means that the **Y** fixed point construction cannot be used to construct sentences for “self-referential” propositions in Direct Logic.

ⁱⁱ *if \mathbf{t} then \mathbf{s}_1 else \mathbf{s}_1*

ⁱⁱⁱ The type of $(\mathbf{p}[\mathbf{x}])$ means that the **Y** fixed point construction cannot be used to construct sentences for “self-referential” propositions in Direct Logic.

^{iv} The type binding achieves much of what Russel sought to achieve in the ramified theory of types. [Russell and Whitehead 1910-1913]

- **Strings for sentences**, i.e., a string for a sentence is a *discrimination* of the following:
 - “**x**”:**String**⟨**Sentence**⟨order+1⟩⟩ⁱ where **x**:**Variable**⟨**String**⟨**Sentence**⟨order⟩⟩ and order:**N**₊
 - “**¬s**”:**String**⟨**Sentence**⟨order⟩⟩ where **s**:**String**⟨**Sentence**⟨order⟩⟩ and order:**N**₊
 - “**s₁ ∧ s₂**”, “**s₁ ∨ s₂**”, “**s₁ ⇒ s₂**”, “**s₁ ⇔ s₂**”:**String**⟨**Sentence**⟨order⟩⟩ where **s₁, s₂**:**String**⟨**Sentence**⟨order⟩⟩ and order:**N**₊
 - “**e** **⇔** **True**”ⁱⁱ, “**e** **⇔** **False**”ⁱⁱ:**String**⟨**Sentence**⟨order⟩⟩ where **e**:**String**⟨**Expression**⟨**Boolean**⟩⟩, **s₁, s₂**:**String**⟨**Sentence**⟨order⟩⟩ and order:**N**₊
 - “**e₁ = e₂**”:**String**⟨**Sentence**⟨1⟩⟩ where **e₁, e₂**:**String**⟨**Expression**⟨**τ**⟩⟩ and **τ**::
 - “**e₁ ⊆ e₂**”:**String**⟨**Sentence**⟨1⟩⟩ where **e₁, e₂**:**String**⟨**Expression**⟨**Set**⟨**τ**⟩⟩⟩ and **τ**::
 - “**e₁ ∈ e₂**”:**String**⟨**Sentence**⟨1⟩⟩ where **e₁**:**String**⟨**Expression**⟨**τ**⟩⟩, **e₂**:**String**⟨**Expression**⟨**Set**⟨**τ**⟩⟩⟩ and **τ**::
 - “**e₁:e₂**”:**String**⟨**Sentence**⟨1⟩⟩ where **e₁**:**String**⟨**Expression**⟨**τ₁**⟩⟩, **e₂**:**String**⟨**Expression**⟨**τ₂**⟩⟩, **τ₁:τ₃**, **τ₂:τ₄** and **τ₃, τ₄**::
 - “**e**”:**String**⟨**Sentence**⟨order⟩⟩ where **e**:**String**⟨**Expression**⟨**τ**⟩⟩ and **τ**::
 - “**∀[x:τ₁] s**”, “**∃[x:τ₁] s**”:**String**⟨**Sentence**⟨order⟩⟩ where **x**:**Variable**⟨**τ₁**⟩ in **s**, **s**:**String**⟨**Sentence**⟨order⟩⟩ and order:**N**₊
 - “**p[x]**”:**String**⟨**Sentence**⟨order+1⟩⟩ⁱⁱⁱ where **x**:**String**⟨**Expression**⟨**τ**⟩⟩, **p**:**String**⟨**Expression**⟨**Sentence**⟨order⟩**τ**⟩⟩, **τ**:: and order:**N**₊ If **τ** is nonpropositional then “**p[x]**” is also of type **String**⟨**Sentence**⟨order⟩⟩
 - “**s₁, ..., s_{n-1} $\frac{p}{T}$ s_n**”:**String**⟨order⟩ where **T**:**String**⟨**Expression**⟨**Theory**⟩⟩, **s₁ to s_n**:**String**⟨order⟩, **p**:**String**⟨**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** fixed point construction cannot be used to construct strings for “self-referential” propositions in Direct Logic.

ⁱⁱ *if t then s₁ else s₂*

ⁱⁱⁱ The type of “**p[x]**” means that the **Y** fixed point construction cannot be used to construct strings for “self-referential” propositions in Direct Logic.

- **String for expressions**, i.e., a string for an expression is a *discrimination* of the following:
 - “(x):String<Expression< τ >>” where x :String<Constant< τ >> and τ ::
 - “(x):String<Expression< τ >>” where x :String<Variable< τ >> and τ ::
 - “((f₁[x₁: τ_1]: $\sigma_1 \equiv d_1, \dots, f_n[x_n:\tau_n]:\sigma_n \equiv d_n^{55}, y))$):String<Expression< τ >>” where for i in 1 to n , f_i :String<Variable< $\sigma_i \tau_i$ >> in d_i and y , x_i :String<Variable< τ_i >> in d_i , d_i :String<Expression< σ_i >>, y :String<Expression< τ >>, and τ_i ::
 - “((x₁: $\tau_1 \equiv d_1, \dots, x_n \equiv d_n^{56} y))$):String<Expression< τ >>” where for i in 1 to n , x_i :String<Variable< τ_i >> in d_i and y , d_i :String<Expression< σ_i >>, y :String<Expression< τ >>, and τ_i ::
 - “(e₁⊙e₂):String<Expression< $\tau_1 \odot \tau_2$ >>”,
 “[e₁, e₂]:String<Expression<[τ_1, τ_2]>>”,
 “[e₁→e₂]:String<Expression<[τ_1 → τ_2]>>”, and
 (e₂^{e₁}):String<Expression< $\tau_2^{\tau_1}$ >>” where e_1 :String<Expression< τ_1 >>, e_2 :String<Expression< τ_2 >>, and τ_1 :: and τ_2 ::
 - “(e₁⋄ True^{e₂}, False^{e₃}):String<Expression< τ >>”ⁱ where e_1 :String<Expression<Boolean>>, e_2, e_3 :String<Expression< τ >> and τ ::
 - “(λ[x: τ_1] e):String<Expression< $\tau_2^{\tau_1}$ >>” where e :String<Expression< τ_2 >>, x :String<Variable< τ_1 >> in e , and τ_1, τ_2 ::
 - “(e[x]):String<Expression< τ_2 >>” where e :String<Expression< $\tau_2^{\tau_1}$ >>, x :String<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 ::
 - String<Sentence<order>>⊆String<Expression<Sentence<order>>> where order: \mathbb{N}_+
 - [e]:Expression< τ >, where e :String<Expression< τ >> and τ ::

ⁱ “if e₁ then e₂ else e₃”

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]

Paul Cohen [2006] wrote as follows of his interaction with Gödel:⁵⁷

“His [Gödel's] main interest seemed to lie in discussing the ‘truth’ or ‘falsity’ of these [mathematical] 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.



Kurt Gödel

Through our discussions, I came closer to his point of view, although I never shared completely his ‘realist’ point of view, that all questions of Set Theory were in the final analysis, either true or false.”

According to John von Neumann, Gödel was “the greatest logician since Aristotle.”⁵⁸ However, [von Neumann 1961] expressed a very different mathematical philosophy than Gödel:



John von Neumann

“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.”

Gödel based his incompleteness results on the thesis that mathematics necessarily has the proposition *I'mUnprovable* in Principia Mathematica [Russell 1902].

Wittgenstein correctly noted that Gödel's *I'mUnprovable* infers inconsistency in mathematics:⁵⁹

“Let us suppose [Gödel's writings are correct and therefore] I proveⁱ the improbability (in Russell's system) of [Gödel's *I'mUnprovable*] P ; [*i.e.*, $\vdash \nexists P$ where $P \Leftrightarrow \nexists P$] then by this proof I have proved P [*i.e.*, $\vdash P$]. Now if this proof were one in Russell's system [*i.e.*, $\vdash P$] — 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.” [*emphasis added*]

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 Mathematica to the First-Order Logic theory *FirstOrderNaturalNumbers* to defend his *I'mUnprovableInFirstOrderNaturalNumbers*. However, the following incompleteness result is not very impressive because *FirstOrderNaturalNumbers* is a very weak theory:

- $\vdash_{\mathbb{N}} I'mUnprovableInFirstOrderNaturalNumbers$
- $\nexists_{FirstOrderNaturalNumbers} I'mUnprovableInFirstOrderNaturalNumbers$

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

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

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

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

Trying to retain *I'mUnprovable* forced Gödel into a very narrow and constricted place of reducing propositions to strings for sentences and then to Gödel numbers axiomatized in a first-order theory in order to avoid Wittgenstein's devastating criticism. This narrow constricted place is intolerable for computer science, which needs to reason about propositions in a more natural and flexible way using Strong Types.

Let T be a theory capable of representing all computable functions on Strings and Natural Numbers with $\text{GödelNumber}[a\text{WellFormedString}]$ being the Gödel number of aWellFormedString, *where a well-formed string is here considered to be a proposition*. A Diagonal Lemma is:

If F is a well-formed string in the language with one free variable, then there is a well-formed string S such that the following is provable in T :

$$S \Leftrightarrow F[\text{GödelNumber}[S]]$$

Letting $\text{GödelNumberToWellFormedString}[n]$ be the well-formed string with Gödel number n , define Eubulides as follows (where " $\neg\text{GödelNumberToWellFormedString}[n]$ " is the string formed by prefixing the character \neg to the well-formed string with Gödel number n):

$\text{Eubulides}[n] \equiv \neg\text{GödelNumberToWellFormedString}[n]$

By the above Diagonal Lemma, there is a well-formed string *I'mFalse* such that the following is provable in T (where

" $\neg\text{GödelNumberToWellFormedString}[\text{GödelNumber}[\text{I'mFalse}]]$ " is the result of prefixing the well-formed string

$\text{GödelNumberToWellFormedString}[\text{GödelNumber}[\text{I'mFalse}]]$ with \neg):⁶⁰

$\text{I'mFalse} \Leftrightarrow \text{Eubulides}[\text{GödelNumber}[\text{I'mFalse}]]$
 $\Leftrightarrow \neg\text{GödelNumberToWellFormedString}[\text{GödelNumber}[\text{I'mFalse}]]$
 $\Leftrightarrow \neg\text{I'mFalse}$

[Chaitin 2007] complained about basing something as important as incompleteness something so trivial as *I'mUnprovable*:

"[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'mUnprovable* So what? This doesn't give any insight how serious the problem is."

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 categorical 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 is found. 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 *ImUnprovable*.⁶¹

According to [Monk 2007]:⁶²

“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.

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** fixed point operator for untyped propositions to construct “self-referential” propositions [Kleene and Rosser 1935]
2. Theorems in his system were computationally enumerable.



Alonzo Church

[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.”

The mathematical theory Nat_1 (1st order theory of Natural Numbers) non-categorically⁶³ formalizes the Natural Numbers using the following schema:¹

$$\forall [P:\text{String} \langle \text{Expression} \langle \text{Proposition} \langle 1 \rangle \rangle \rangle] \\ (\llbracket P \rrbracket [0] \wedge \forall [i:\mathbf{N}] \llbracket P \rrbracket [i] \Rightarrow \llbracket P \rrbracket [i+1]) \Rightarrow \forall [i:\mathbf{N}] \llbracket P \rrbracket [i]$$

Nat_1 has countably many instances of the above schema because there are only countably many strings.

¹ instead of using the categorical induction axiom of Nat

Church's Paradox:⁶⁴

1. $\vdash_{Nat_1} \text{TheoremsComputationallyEnumerable}[Nat_1]$
2. $\not\vdash_{Nat_1} \text{TheoremsComputationallyEnumerable}[Nat_1]$

Proof:

1. $\vdash_{Nat_1} \text{TheoremsComputationallyEnumerable}[Nat_1]$ because all of the instances of the induction axiom can be computationally enumerated and then used to computationally enumerate the theorems of Nat_1 .
2. Suppose to obtain a contradiction that

$$\vdash_{Nat_1} \text{TheoremsComputationallyEnumerable}[Nat_1]$$

Then there is a provable in Nat_1 computable total procedure

$$\text{TheoremsEnumerator}\langle Nat_1 \rangle : [\mathbb{N}] \rightarrow \text{Theorem}\langle Nat_1 \rangle$$

such that it is provable in Nat_1 that

$$\forall [p: \text{Theorem}\langle Nat_1 \rangle] \exists [i: \mathbb{N}] \text{TheoremsEnumerator}\langle Nat_1 \rangle \cdot [i] = p$$

A subset of the proofs in Nat_1 are those proving that certain procedures $[\mathbb{N}] \rightarrow \mathbb{N}$ are total. Consequently, there is a procedure

$$\text{ProvedTotalsEnumerator}\langle Nat_1 \rangle : [\mathbb{N}] \rightarrow ([\mathbb{N}] \rightarrow \mathbb{N})$$

that enumerates the provable in Nat_1 total computable procedures $[\mathbb{N}] \rightarrow \mathbb{N}$ that can be used in the implementation of the following procedure:

$$\text{Diagonal} \cdot [i: \mathbb{N}] : \mathbb{N} \equiv 1 + (\text{ProvedTotalsEnumerator}\langle Nat_1 \rangle \cdot [i]) \cdot [i]$$

Consequently:

- Diagonal *is* a provable in Nat_1 total procedure because it is implemented using computable provable in Nat_1 total procedures.
- Diagonal *is not* a provable in Nat_1 total procedure because it differs from every other computable provable in Nat_1 total procedure.

The above contradiction completes the proof

Church's Paradox is that Nat_1 (1st order theory of Natural Numbers) is *inconsistent*. [Church 1934] pointed out that there is no obvious way to remove the inconsistency concluding if Nat_1 is taken to be an exact description of logicⁱ then,

“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

ⁱ (in accord with the opinion of a large fraction of contemporary philosophers of logic)

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.”

The above issues can be addressed as follows:

1. Requiring Mathematics to be strongly typed using 2nd order logic so that
 - Mathematics self proves that it is “open” in the sense that proofs are not computationally enumerable.⁶⁵
 - Mathematics self proves that it is formally consistent.⁶⁶
2. It was initially thought that mathematics could be based just on character strings. Then diagonalization was discovered and things haven’t been the same since. The string for the 1st order Nat_1 non-categorical induction *schema* is as follows:ⁱ

" $\forall [P: \text{String} \langle \text{Expression} \langle \text{Proposition} \langle 1 \rangle^{\mathbb{N}} \rangle \rangle]$
 $(\llbracket P \rrbracket[0] \wedge \forall [i: \mathbb{N}] \llbracket P \rrbracket[i] \Rightarrow \llbracket P \rrbracket[i+1]) \Rightarrow \forall [i: \mathbb{N}] \llbracket P \rrbracket[i]$ "

which has countably many 1st order propositions as instances that are abstracted from the countably many character strings of type

$\text{String} \langle \text{Expression} \langle \text{Proposition} \langle 1 \rangle^{\mathbb{N}} \rangle \rangle$ and which differs fundamentally from the character string for the *more general* 2nd order categorical induction *axiom*, which is as follows:

" $\forall [P: \text{Proposition} \langle 1 \rangle^{\mathbb{N}}] (P[0]) \wedge \forall [i: \mathbb{N}] P[i] \Rightarrow P[i+1] \Rightarrow \forall [i: \mathbb{N}] P[i]$ "

Although the theory Nat has only *finitely* many axioms, the above string abstracted as a proposition has *uncountably* many 1st order propositions as instances.ⁱⁱ In this way, Nat differs fundamentally from the 1st order theory Nat_1 because, being uncountable, *not all* instances of the Nat induction axiom can be obtained by abstraction from character strings. Proofs abstracted from character strings for the axioms of Nat_1 can be computationally enumerated and are valid proofs in Nat , but this does not enumerate *all* of the proofs of Nat ! What is to be made of the *uncountable* number of theorems of Nat whose proofs cannot be written down in text?

Gödel, Curry, and Löb Paradoxes

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

ⁱ with the unfortunate consequence that the argument in Church’s Paradox shows that Nat_1 is inconsistent because it can provably computationally enumerate its theorems

ⁱⁱ with the consequence that the argument in Church’s Paradox is blocked in Nat because it has uncountably many theorems

The fixed point construction of the Diagonal Lemma *cannot* be used to construct Gödel's *I'mUnprovable* with the following definition of F because a fixed point does *not* exist:⁶⁷

$$F\langle n \rangle [p:\text{Proposition}\langle n \rangle]:\text{Proposition}\langle n+1 \rangle \equiv \not\vdash p$$

By the following argument, Wittgenstein derived a contradiction in Mathematics from Gödel's result:⁶⁸

Gödel thought that he demonstrated $\vdash \not\vdash I'mUnprovable$. Therefore $\vdash I'mUnprovable$ using $I'mUnprovable \Leftrightarrow \not\vdash I'mUnprovable$. $\vdash \vdash I'mUnprovable$ follows using adequacy. But the contradiction $\vdash \neg I'mUnprovable$ follows using $I'mUnprovable \Leftrightarrow \not\vdash I'mUnprovable$.

The following paradoxes cannot prove *every* proposition because the **Y** untyped fixed point operator for propositions cannot be used in a strongly typed logic:⁶⁹

- **Curry's Paradox** [Curry 1941] Suppose $\Psi:\text{Proposition}\langle \text{anOrder}:\mathbf{N}_+ \rangle$.

$$F\langle n:\mathbf{N}_+ \rangle [p:\text{Proposition}\langle n \rangle]:\text{Proposition}\langle \text{Max}[n+1, \text{anOrder}] \rangle \equiv p \Rightarrow \Psi$$

// above definition has no fixed point because $p \vdash \Psi$ has
// order greater than p

The following argument derives any proposition Ψ assuming the existence of a fixed point for F:

- 1) $\text{Curry}_\Psi \Leftrightarrow (\text{Curry}_\Psi \Rightarrow \Psi)$ // *nonexistent* fixed point of F
- 2) $\text{Curry}_\Psi \Rightarrow \text{Curry}_\Psi$ // idempotency
- 3) $\text{Curry}_\Psi \Rightarrow (\text{Curry}_\Psi \Rightarrow \Psi)$ // substituting 1) into 2)
- 4) $\text{Curry}_\Psi \Rightarrow \Psi$ // contraction
- 5) Curry_Ψ // from 4) using 1)
- 6) Ψ // transitivity 4) and 5)

- **Löb's Paradox** [Löb 1955]⁷⁰ Suppose $\Psi:\text{Proposition}\langle \text{anOrder}:\mathbf{N}_+ \rangle$.

$$F\langle n:\mathbf{N}_+ \rangle [p:\text{Proposition}\langle n \rangle]:\text{Proposition}\langle \text{Max}[n+1, \text{anOrder}] \rangle \equiv (\vdash p) \Rightarrow \Psi$$

// above definition has no fixed point because $\equiv (\vdash p)$ has
// order greater than p

The following argument derives any proposition Ψ assuming the existence of a fixed point for F:

- 1) $\text{Löb}_\Psi \Leftrightarrow ((\vdash \text{Löb}_\Psi) \Rightarrow \Psi)$ // *nonexistent* fixed point of F
- 2) $(\vdash \text{Löb}_\Psi) \Rightarrow \text{Löb}_\Psi$ // rule of *Theorem Use*
- 3) $(\vdash \text{Löb}_\Psi) \Rightarrow ((\vdash \text{Löb}_\Psi) \Rightarrow \Psi)$ // substituting 1) into 2)
- 4) $(\vdash \text{Löb}_\Psi) \Rightarrow \Psi$ // contraction
- 5) $\vdash \text{Löb}_\Psi$ // from 4) using 1)
- 6) Ψ // transitivity using 4) and 5)

Berry Paradox

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

Characterize[s:String<Expression<Proposition< ω >^N>>
 k:N]:Proposition< $\omega+1$ > \equiv
 $\forall[x:N] \lfloor \lfloor s \rfloor \rfloor [x] \Leftrightarrow x=k$

Consider the following definition:

BString:String<Expression<Proposition< $\omega+1$ >^N>> \equiv
 “($\lambda[n:N] (\forall[s:String<Expression<Proposition< ω >^N>>]
 Length[s]<100 \Leftrightarrow \neg Characterize[s, n]))$ ”

BExpression:Expression<Proposition< $\omega+1$ >^N> \equiv \lfloor BString \rfloor

Note that

- o Length[BString]<100.
- o {s:String<Expression<Proposition< ω >^N>> | Length[s]<100} is finite.
- o Therefore the following set is finite:

$\{n:N_+ \mid \exists[s:String<Expression<Proposition< ω >^N>>]
 Length[s]<100 \wedge Characterize[s, n]\}$

BSet:Set<N> \equiv {n:N₊ | \lfloor BExpression \rfloor [n]}

BSet \neq { } because is {n:N | n \geq 1} is infinite.

1. BNumber:N \equiv Least[BSet]
2. \lfloor BExpression \rfloor [BNumber]⁷¹
3. $\lfloor (\lambda[n:N] (\forall[s:String<Expression<Proposition< ω >^N>>]
 Length[s]<100 \Leftrightarrow \neg Characterize[s, n])) \rfloor$ [BNumber]⁷²
4. $\forall[s:String<Expression<Proposition< ω >^N>>]
 Length[s]<100 \Leftrightarrow \neg Characterize[s, BNumber]⁷³$
5. Length[BString]<100 \Leftrightarrow \neg Characterize[BString, BNumber]
 // above is invalid because of attempted substitution of
 // BString:String<Expression<Proposition< $\omega+1$ >^N>> for
 // s:String<Expression<Proposition< ω >^N>>

End Notes

¹ [White 1956, Wilder 1968, Rosental 2008]

² 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.

³ By the *Computational Representation Theorem* [Clinger 1981; Hewitt 2006], which can define all the possible executions of a procedure.

⁴ Again, *Mathematics* here means the common foundation of all classical mathematical theories from Euclid to the mathematics used to prove Fermat's Last [McLarty 2010].

⁵ As shown above, there is a simple proof in Classical Direct Logic that Mathematics (\vdash) is formally consistent. If Classical Direct Logic has a bug, then there might also be a proof that Mathematics is inconsistent. Of course, if a 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 Mathematics because it was designed to be consistent. Consequently, it is not absurd that there is a simple proof of the consistency of Mathematics (\vdash) that does not use all of the machinery of Classical Direct Logic.

In reaction to paradoxes, philosophers developed the dogma of the necessity of strict separation of “object theories” (theories about basic mathematical entities such as numbers) and “meta theories” (theories about theories). 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.

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

A procedure definition NotProvable could be attempted as follows:

$$\text{NotProvable}[p] \equiv \not\vdash p$$

With strong types, the attempted definition becomes:

$$\text{NotProvable}\langle n:\mathbb{N}_+\rangle[p:\text{Proposition}\langle n\rangle]:\text{Proposition}\langle n+1\rangle \equiv \not\vdash p$$

Consequently, there is no fixed point *I'mUnprovable* for the procedure NotProvable $\langle n:\mathbb{N}_+\rangle$ such that the following holds:

$$\text{NotProvable}\langle n:\mathbb{N}_+\rangle[\text{I'mUnprovable}] \Leftrightarrow \text{I'mUnprovable}$$

Thus Gödel's *I'mUnprovable* does not exist in Strongly-Typed Mathematics. See the discussion in this article on Provability Direct Logic for Wittgenstein's proof that *I'mUnprovable* leads to inconsistency in Mathematics.

In arguing against Wittgenstein's criticism, Gödel maintained that his results on *I'mUnprovable* followed from properties of \mathbb{N} using Gödel numbers for strings that are well-formed. The procedure NotProvable could be attempted for strings as follows: *NotProvable[s] ≡ “ $\not\vdash s$ ”* With strong types, the attempted definition becomes:

$$\text{NotProvable}\langle n:\mathbb{N}_+\rangle[s:\text{String}\langle \text{Proposition}\langle n\rangle \rangle]:\text{String}\langle \text{Proposition}\langle n+1\rangle \rangle \\ \equiv \text{“}\not\vdash s\text{”}$$

Consequently, there is no fixed point *I'mUnprovableString* for the procedure $\text{NotProvable}\langle n:\mathbb{N}_+\rangle$ such that the following holds where $\llbracket s \rrbracket$ is the proposition for well-formed string s :

$$\llbracket \llbracket \text{NotProvable}\langle n:\mathbb{N}_+\rangle [I'mUnprovableString] \rrbracket \rrbracket \Leftrightarrow \llbracket \llbracket I'mUnprovableString \rrbracket \rrbracket$$

Thus Gödel's *I'mUnprovableString* does not exist in Strongly-Typed Mathematics.

Furthermore, Strong Types thwart the known paradoxes while at the same time facilitating proof of new theorems, such as categoricity of the set theory.

Although Gödel's incompleteness results for *I'mUnprovable* have fundamental problems, the work was extremely significant. For example, the following paradoxes were developed following along Gödel's work:

- **Curry's Paradox** [Curry 1941] Suppose $\Psi:\text{Proposition}\langle \text{order}:\mathbb{N}_+\rangle$.
 $\text{Curry}\langle n:\mathbb{N}_+\rangle [p:\text{Proposition}\langle n \rangle]:\text{Proposition}\langle \text{Max}[n,\text{order}+1] \rangle \equiv p \Rightarrow \Psi$
 Curry's Paradox is blocked because the procedure Curry does *not* have a fixed point.
- **Löb's Paradox** [Löb 1955]⁶ Suppose $\Psi:\text{Proposition}\langle \text{order}:\mathbb{N}_+\rangle$.
 $\text{Löb}\langle n:\mathbb{N}_+\rangle [p:\text{Proposition}\langle n \rangle]:\text{Proposition}\langle \text{Max}[n,\text{order}+1] \rangle \equiv (\vdash p) \Rightarrow \Psi$
 Löb's Paradox is blocked because the procedure Löb does *not* have a fixed point.

A key difference is that Direct Logic works directly with propositions as opposed to the work of Gödel, Curry, and Löb, which was based on first-order theories of propositions for sentences strings coded as integers.

⁷ [Nielsen 2014]

⁸ [Isaacson 2007]

⁹ The induction axiom per se does not per se commit to sets such as those in **Boolean^N**. Also as illustrated in this article, strong types are in fact much stronger than first-order set theory.

¹⁰ 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.*”

¹¹ The argument above 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.*” However, there are **uncountably** many $P:\text{Proposition}\langle 1 \rangle^{\mathbb{N}}$ for which the following holds in \mathbb{N} : $(P[0] \wedge \forall [i:\mathbb{N}] P[i] \Rightarrow P[i+1]) \Leftrightarrow \forall [i:\mathbb{N}] P[i]$. Consequently, there is no way to computationally enumerate all such P by abstraction from strings of the following form: **String** $\langle \text{Proposition}\langle 1 \rangle^{\mathbb{N}} \rangle$. There are uncountably many propositions of *Nat* that are true in \mathbb{N} . However, only countably many can be abstracted from strings. And there are only

countably many propositions that can be proved by abstracting from strings that are the axioms of *Nat*. See the discussion of Church's Paradox in appendix of this article.

¹² [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.”

¹³ For each type \mathbf{X} that satisfies the categorical axioms there is a (unique) isomorphism $I:\mathbf{X}^{\mathbf{N}}$ and inductively defined as follows:

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

Using proofs by Natural Number induction on \mathbf{N} and \mathbf{X} , the following follow:

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

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

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

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

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

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

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

which is a contradiction

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

Proof: Induction on $P[m:\mathbf{N}]$: Proposition $\langle 1 \rangle \equiv$

$\forall[k,j \leq m:\mathbf{N}] I[k]=I[j] \Rightarrow k=j$

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

Induction: Suppose $\forall[k,j \leq m:\mathbf{N}] I[k]=I[j] \Rightarrow k=j$

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

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

$k_0 = j_0$ since $k_0, j_0 \leq m$, $k, j < +_1[m]$ 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 \mathbf{X} : $\forall[y:\mathbf{X}] \exists[k:\mathbf{N}] I[k]=y$
 Proof: Induction on $P[y:\mathbf{X}]$: Proposition <1> $\equiv \exists[k:\mathbf{N}] I[k]=y$
Base: Suppose $y=0_{\mathbf{X}}$. To show $\exists[k:\mathbf{N}] I[k]=0_{\mathbf{X}}$. Clearly $I[0_{\mathbf{N}}]=0_{\mathbf{X}}$
Induction: Suppose $y>0_{\mathbf{X}}$ and $\exists[k:\mathbf{N}] I[k]=y$. Let $I[k_0]=y$.
 To show $\exists[k:\mathbf{N}] I[k]=+_1^{\mathbf{X}}[y]$.
 It follows from $I[+_1[k_0]] = +_1^{\mathbf{X}}[I[k_0]] = +_1^{\mathbf{X}}[y]$
4. I is a homomorphism: $I[0_{\mathbf{N}}]=0_{\mathbf{X}}$ and $\forall[j:\mathbf{N}] I[+_1[j]] = +_1^{\mathbf{X}}[I[j]]$
 Proof: Induction on $P[j:\mathbf{N}]$: Proposition <1> $\equiv I[+_1[j]] = +_1^{\mathbf{X}}[I[j]]$
Base: $I[+_1[0_{\mathbf{N}}]] = +_1^{\mathbf{X}}[I[0_{\mathbf{N}}]]$ by definition of I
Induction: Suppose $\forall[i:\mathbf{N}] I[+_1[i]] = +_1^{\mathbf{X}}[I[i]]$
 To show: $\forall[j:\mathbf{N}] I[+_1[+_1[j]]] = +_1^{\mathbf{X}}[I[+_1[j]]]$
 $I[+_1[+_1[j]]] = +_1^{\mathbf{X}}[I[+_1[j]]]$ by definition of I
5. $I^{-1}:\mathbf{N}^{\mathbf{X}}$ is a homomorphism:
 $I^{-1}[0_{\mathbf{X}}] = 0_{\mathbf{N}}$ and $\forall[y:\mathbf{X}] I^{-1}[+_1^{\mathbf{X}}[y]] = +_1[I^{-1}[y]]$
 Proof:
 To show: $I^{-1}[0_{\mathbf{X}}] = 0_{\mathbf{N}}$.
 Let $i = I^{-1}[0_{\mathbf{X}}]$. Therefore $I[i] = 0_{\mathbf{X}}$ and $k = 0_{\mathbf{N}}$.
 To show: $\forall[y:\mathbf{X}] I^{-1}[+_1^{\mathbf{X}}[y]] = +_1[I^{-1}[y]]$
 Induction on $P[y:\mathbf{X}]$: Proposition <1> $\equiv I^{-1}[+_1^{\mathbf{X}}[y]] = +_1[I^{-1}[y]]$
Base: To show: $I^{-1}[+_1^{\mathbf{X}}[0_{\mathbf{X}}]] = +_1[I^{-1}[0_{\mathbf{X}}]] = +_1[0_{\mathbf{N}}]$
 Let $k = I^{-1}[+_1^{\mathbf{X}}[0_{\mathbf{X}}]]$. Therefore $I[k] = +_1^{\mathbf{X}}[0_{\mathbf{X}}]$ and $k = +_1[0_{\mathbf{N}}]$.
Induction: Suppose $\forall[j:\mathbf{N}] I[+_1[j]] = +_1^{\mathbf{X}}[I[j]]$
 To show: $\forall[i:\mathbf{N}] I[+_1[+_1[j]]] = +_1^{\mathbf{X}}[I[+_1[j]]]$
 $I[+_1[+_1[j]]] = +_1^{\mathbf{X}}[I[+_1[j]]]$ by definition of I
6. I is the unique isomorphism: If $g:\mathbf{X}^{\mathbf{N}}$ is an isomorphism then $g=I$
 Proof: Induction on $P[j:\mathbf{N}]$: Proposition <1> $\equiv I[j] = g[j]$
Base: $I[0_{\mathbf{N}}] = 0_{\mathbf{X}}$. $g[0_{\mathbf{N}}] = 0_{\mathbf{X}}$ because g is an isomorphism.
 Therefore $I[0_{\mathbf{N}}] = g[0_{\mathbf{N}}]$
Induction: Suppose $I[j] = g[j]$.
 To show: $I[+_1[j]] = g[+_1[j]]$
 $I[+_1[j]] = +_1^{\mathbf{X}}[I[j]] = +_1^{\mathbf{X}}[g[j]] = g[+_1[j]]$
- ¹⁴ For example, there are nondeterministic Turing machines that the theory *Nat* proves always halt that cannot be proved to halt in a first-order theory.

¹⁵ [Dedekind 1888]

The following can be used to characterize the real numbers (\mathbb{R} ¹⁵) up to isomorphism with a unique isomorphism:

$$\forall[S:\mathbf{Set}\langle\mathbb{R}\rangle] S \neq \{\} \mathbb{R} \wedge \text{Bounded}[S] \Rightarrow \text{HasLeastUpperBound}[S]$$

where

$$\text{Bounded}[S:\mathbf{Set}\langle\mathbb{R}\rangle] \Leftrightarrow \exists[b:\mathbb{R}] \text{UpperBound}[b, S]$$

$$\text{UpperBound}[b:\mathbb{R}, S:\mathbf{Set}\langle\mathbb{R}\rangle] \Leftrightarrow b \in S \wedge \forall[x \in S] x \leq b$$

$$\text{HasLeastUpperBound}[S:\mathbf{Set}\langle\mathbb{R}\rangle] \Leftrightarrow \exists[b:\mathbb{R}] \text{LeastUpperBound}[b, S]$$

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

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

¹⁶ Robinson [1961]

¹⁷ The theory of the ordinals \mathcal{Ord} is axiomatised as follows:

- $0_{\mathbf{O}}:\mathbf{O}$
- Successor ordinals
 - $\forall[\alpha:\mathbf{O}] \text{+}_1[\alpha]:\mathbf{O} \wedge \text{+}_1[\alpha] > \alpha$
 - $\forall[\alpha:\mathbf{O}] \nexists[\beta:\mathbf{O}] \alpha < \beta < \text{+}_1[\alpha]$
- Replacement for ordinals:
 - $\forall[\alpha:\mathbf{O}, f:\mathbf{O}^{\mathbf{O}}] \cup_{\alpha} f:\mathbf{O}$
 - $\forall[\alpha, \beta:\mathbf{O}, f:\mathbf{O}^{\mathbf{O}}] \beta \in \cup_{\alpha} f \Leftrightarrow \exists[\delta < \alpha] \beta \leq f[\delta]$
 - $\forall[\alpha, \beta:\mathbf{O}, f:\mathbf{O}^{\mathbf{O}}] (\forall[\delta < \alpha] f[\delta] \leq \beta) \Rightarrow \cup_{\alpha} f \leq \beta$

- Cardinal ordinals

$$\omega_0 = \mathbb{N}$$

$$\forall[\alpha:\mathbf{O}] \alpha > 0_{\mathbf{O}} \Rightarrow |\omega_{\alpha}| = |\mathbf{Boolean}^{\cup_{\beta < \alpha} \omega_{\beta}}|$$

$$\forall[\alpha, \beta:\mathbf{O}] |\beta| = |\omega_{\alpha}| \Rightarrow \omega_{\alpha} = \beta \vee \omega_{\alpha} \in \beta$$

where $|\tau_1| = |\tau_2| \Leftrightarrow \exists[f:\tau_2^{\tau_1}] \text{OneToOneOnto}\langle\tau_1, \tau_2\rangle[f]$

$$\text{OneToOne}\langle\tau_1, \tau_2\rangle[f:\tau_2^{\tau_1}]$$

$$\Leftrightarrow \forall[x_1, x_2:\tau_1] f[x_1] = f[x_2] \Rightarrow x_1 = x_2$$

$$\text{OneToOneOnto}\langle\tau_1, \tau_2\rangle[f:\tau_2^{\tau_1}]$$

$$\Leftrightarrow \text{OneToOne}\langle\tau_1, \tau_2\rangle[f:\tau_2^{\tau_1}] \wedge \forall[y:\tau_2] \exists[x:\tau_1] f[x] = y$$

- Transitivity of $<$

$$\forall[\alpha, \beta < \alpha, \delta < \beta:\mathbf{O}] \alpha < \delta$$
- $\forall[\alpha, \beta:\mathbf{O}] \alpha < \beta \vee \alpha = \beta \vee \beta < \alpha$
- $\forall[\alpha, \beta:\mathbf{O}] \alpha < \beta \Rightarrow \neg \beta < \alpha$

- For each order: \mathbb{N}_+ and $P: \text{Proposition} \langle \text{order} \rangle^{\mathbf{O}}$:
the following ordinal induction axiom holds:
$$(\forall [\alpha: \mathbf{O}] \forall [\beta < \alpha: \mathbf{O}] P[\beta] \Rightarrow P[\alpha]) \Rightarrow \forall [\alpha: \mathbf{O}] P[\alpha]$$

Ordinals have the following properties:

- Ordinals are well-ordered:
Least: $\mathbf{O}^{\text{Boolean}^{\mathbf{O}}}$
 $\text{Least}[\{\}] = 0_{\mathbf{O}}$
 $\forall [S: \text{Boolean}^{\mathbf{O}}] S \neq \{\} \Rightarrow \text{Least}[S] \in S$
 $\forall [S: \text{Boolean}^{\mathbf{O}}] S \neq \{\} \Rightarrow \forall [\alpha: \mathbf{O}] \alpha \in S \Rightarrow \text{Least}[S] \leq \alpha$
- Reals can be well-ordered
 $|\Omega_1| = |\mathbb{R}|$
- $\forall [\alpha: \mathbf{O}] \exists [\beta: \mathbf{O}] \alpha < \Omega_\beta$
- The set of all ordinals Ω is $\text{Boolean}^{\mathbf{O}}$ so that:

$$\forall [\alpha: \mathbf{O}] \alpha \in \Omega \Leftrightarrow \alpha: \mathbf{O}$$

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

¹⁸ For each type \mathbf{X} that satisfies the theory Ord there is a (unique) isomorphism $I: \mathbf{X}^{\mathbf{O}}$ inductively defined as follows:

$$\begin{aligned} I[0_{\mathbf{O}}] &\equiv 0_{\mathbf{X}} \\ \forall [\alpha: \mathbf{O}] I[+_1[\alpha]] &\equiv +_1^{\mathbf{X}}[I[\alpha]] \\ \forall [\alpha: \text{Limit} \langle \mathbf{O} \rangle] I[\alpha] &\equiv y \\ &\text{where } y: \mathbf{X} \wedge \forall [\beta < \alpha] y \leq_{\mathbf{X}} I[\beta] \\ &\quad \wedge \forall [z: \mathbf{X}] (\forall [\beta < \alpha] z \leq_{\mathbf{X}} I[\beta]) \Rightarrow y \leq_{\mathbf{X}} z \end{aligned}$$

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: $\forall [\alpha, \beta: \mathbf{O}] I[\alpha] = I[\beta] \Rightarrow \alpha = \beta$
3. The range of I is all of \mathbf{X} : $\forall [y: \mathbf{X}] \exists [\alpha: \mathbf{O}] I[\alpha] = y$
4. I is a homomorphism:
 - $I[0_{\mathbf{O}}] = 0_{\mathbf{X}}$
 - $\forall [\alpha: \mathbf{O}] I[+_1[\alpha]] = +_1^{\mathbf{X}}[I[\alpha]]$
 - $\forall [\alpha: \text{Limit} \langle \mathbf{O} \rangle, f: \mathbf{O}^{\mathbf{O}}] I[\cup_{\alpha} f] = \cup_{f[\alpha]}^{\mathbf{X}} I \circ f \circ I^{-1}$
5. $I^{-1}: \mathbf{O}^{\mathbf{X}}$ is a homomorphism

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

¹⁹ [Bourbaki 1972; Fantechi, et. al. 2005]

²⁰ This implies, for example, that no set is an element of itself.

²¹ Proof: Suppose $S: \mathbf{Sets} \langle \tau \rangle$ and therefore $\exists [\alpha: \mathbf{O}] S: \mathbf{Set}^{\alpha} \langle \tau \rangle$.

Proof by ordinal induction on

$$P[\beta: \mathbf{O}]: \text{Proposition} \langle 1 \rangle \equiv \forall [X \in S] X: \mathbf{Set}^{\beta} \langle \tau \rangle$$

$$\text{Assume: } (\forall [\beta < \alpha: \mathbf{O}] \forall [X \in S] X: \mathbf{Set}^{\beta} \langle \tau \rangle) \Rightarrow \forall [X \in S] X: \mathbf{Set}^{\alpha} \langle \tau \rangle$$

Show: $\forall [X \in S] X: \mathbf{Set}^{\alpha} \langle \tau \rangle$

Assume: $X \in S$

Show $X: \mathbf{Set}^{\alpha} \langle \tau \rangle$

Proof by cases on α

1. $X: \mathbf{Set}^0 \langle \tau \rangle$

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

2. $\forall [\alpha: \mathbf{O}] \mathbf{Sets}^{\alpha} \langle \tau \rangle = \mathbf{Set} \langle \mathbf{Set}^{\alpha-1} \langle \tau \rangle \rangle$

$X: \mathbf{Set}^{\alpha-1} \langle \tau \rangle$ QED by induction hypothesis

3. $\forall [\alpha: \mathbf{Limit} \langle \mathbf{O} \rangle] \exists [\beta < \alpha, Y: \mathbf{Set}^{\beta} \langle \tau \rangle] X \in Y$

QED by induction hypothesis

²² Proof: Suppose $S: \mathbf{Sets} \langle \tau \rangle$ and therefore $\exists [\alpha: \mathbf{O}] S: \mathbf{Sets}^{\alpha} \langle \tau \rangle$

$S: \mathbf{Sets}^{\alpha} \langle \tau \rangle$

Show: $\mathbf{Boolean}^S: \mathbf{Sets} \langle \tau \rangle$

$\mathbf{Boolean}^S: \mathbf{Sets}^{\alpha+1} \langle \tau \rangle$ QED

²³ Proof by ordinal induction on

$$P[\alpha: \mathbf{O}]: \text{Proposition} \langle 1 \rangle \equiv \forall [S: \mathbf{Sets}^{\alpha} \langle \tau \rangle] US: \mathbf{Sets} \langle \tau \rangle$$

$$\text{Assume: } \forall [\beta < \alpha: \mathbf{O}] \forall [S: \mathbf{Sets}^{\beta} \langle \tau \rangle] US: \mathbf{Sets} \langle \tau \rangle$$

Show: $\forall [S: \mathbf{Sets}^{\alpha} \langle \tau \rangle] US: \mathbf{Sets} \langle \tau \rangle$

Assume: $S: \mathbf{Sets}^{\alpha} \langle \tau \rangle$

Show: $US: \mathbf{Sets} \langle \tau \rangle$

$$\forall [X: \mathbf{Sets} \langle \tau \rangle] X \in US \Leftrightarrow \exists [Y \in S] X \in Y$$

$$\forall [X: \mathbf{Sets} \langle \tau \rangle] X \in US \Leftrightarrow \exists [\beta < \alpha: \mathbf{O}, Y: \mathbf{Sets}^{\beta} \langle \tau \rangle] X \in Y$$

$$\forall [X: \mathbf{Sets} \langle \tau \rangle] X \in US \Rightarrow X: \mathbf{Sets} \langle \tau \rangle$$

QED by definition of $\mathbf{Sets} \langle \tau \rangle$

-
- ²⁴ Suppose $f:\mathbf{Sets}\langle\tau\rangle^{\mathbf{Sets}\langle\tau\rangle}$ and $S:\mathbf{Sets}\langle\tau\rangle$
 Show $\mathbf{Image}\langle\tau\rangle[f, S]:\mathbf{Sets}\langle\tau\rangle$
 Proof by ordinal induction on
 $P[\alpha:\mathbf{O}] \Leftrightarrow S:\mathbf{Set}^\alpha\langle\tau\rangle \Rightarrow \mathbf{Image}\langle\tau\rangle[f, S]:\mathbf{Sets}\langle\tau\rangle$
 Suppose $\forall[\beta<\alpha:\mathbf{O}] S:\mathbf{Set}^\beta\langle\tau\rangle \Rightarrow \mathbf{Image}\langle\tau\rangle[f, S]:\mathbf{Sets}\langle\tau\rangle$
 Show $S:\mathbf{Set}^\alpha\langle\tau\rangle \Rightarrow \mathbf{Image}\langle\tau\rangle[f, S]:\mathbf{Sets}\langle\tau\rangle$
 Suppose $S:\mathbf{Set}^\alpha\langle\tau\rangle$
 Show $\mathbf{Image}\langle\tau\rangle[f, S]:\mathbf{Sets}\langle\tau\rangle$
 $\forall[y:\mathbf{Sets}\langle\tau\rangle] y:\mathbf{Image}\langle\tau\rangle[f, S] \Leftrightarrow \exists[x\in S] f[x]=y$
 Show $\forall[y:\mathbf{Sets}\langle\tau\rangle] y\in\mathbf{Image}\langle\tau\rangle[f, S] \Rightarrow y:\mathbf{Sets}\langle\tau\rangle$
 Suppose $y:\mathbf{Sets}\langle\tau\rangle \wedge y\in\mathbf{Image}\langle\tau\rangle[f, S]$
 Show $y:\mathbf{Sets}\langle\tau\rangle$
 $\exists[x\in S] f[x]=y$ because $y\in\mathbf{Image}\langle\tau\rangle[f, S]$
 $\exists[\beta<\alpha:\mathbf{O}] x:\mathbf{Set}^\beta\langle\tau\rangle$ because $x\in S$ and $S:\mathbf{Set}^\alpha\langle\tau\rangle$
 $\mathbf{Image}\langle\tau\rangle[f, x]:\mathbf{Sets}\langle\tau\rangle$ by induction hypothesis
 Show $f[x]:\mathbf{Sets}\langle\tau\rangle$
 Suppose $z\in f[x]$
 Show $z:\mathbf{Sets}\langle\tau\rangle$
 $z\in\mathbf{Sets}\langle\tau\rangle$ because $z\in f[x]$ and $\mathbf{Image}\langle\tau\rangle[f, x]:\mathbf{Sets}\langle\tau\rangle$
 $f[x]:\mathbf{Sets}\langle\tau\rangle$
 $y:\mathbf{Sets}\langle\tau\rangle$ because $f[x]=y$
- ²⁵ [Mizar; Matuszewski 1 and Rudnicki: 2005; Naumowicz and Artur Kornilowicz 2009; Naumowicz 2009]
- ²⁶ Note that this proof is fundamentally different from the categoricity proof in [Martin 2015].
- ²⁷ **Fix** implements recursion. For example, suppose
 $F[g:\mathbf{N}\rightarrow\mathbf{N}]:(\mathbf{N}\rightarrow\mathbf{N}) \equiv \lambda[i:\mathbf{N}] i=1 \blacklozenge \mathbf{True}: 1, \mathbf{False}: i*g.[i-1]$
 Therefore by the **Fix** axiom, $\mathbf{Fix}\langle\mathbf{N}, \mathbf{N}\rangle.[F] = F.[\mathbf{Fix}\langle\mathbf{N}, \mathbf{N}\rangle.[F]]$ and
 $F.[\mathbf{Factorial}] = \mathbf{Factorial}$ where
 $\mathbf{Factorial} \equiv \lambda[i:\mathbf{N}] i=1 \blacklozenge \mathbf{True}: 1, \mathbf{False}: i*\mathbf{Factorial}.[i-1]$
- ²⁸ where $F^1.[x] \equiv F.[x]$
 $F^{n+1}.[x] \equiv F^n.[F.[x]]$
- ²⁹ cf. [Rosental 2008]

³⁰ 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 [of Dedekind, etc.]”

³¹ for discussion see [Hewitt 2010]

³² in an unlawful way (Einstein, a member of the editorial board, refused to support Hilbert's action)

³³ Hilbert letter to Brouwer, October 1928

³⁴ Gödel said “Has Wittgenstein lost his mind?”

³⁵ 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 :
>
> $\text{Set}\langle\tau\rangle \equiv \text{Boolean}^{\tau}$
>

-
- > 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
- ³⁶ Consequently, there can not be any escape hatch into an unformalized "meta-theory."
- ³⁷ The claim also relied on Gödel's proposition *I'mUnprovable*.
- ³⁸ Formal syntax was invented long after [Gödel 1931].
- ³⁹ emphasis in original
- ⁴⁰ For every type there is a larger type, i.e., $\forall[\tau_1::] \exists[\tau_2::] \tau_1 \sqsubset \tau_2$
- ⁴¹ There is no universal type. Instead, **Type** is parameterized, e.g.,
Boolean:Type<Boolean> and **N:Type<N>**
- ⁴² **True≠False**, **True:Boolean**, and **False:Boolean**
 $\forall[x:\mathbf{Boolean}] x=\mathbf{True} \vee x=\mathbf{False}$
- ⁴³ The theory of the natural numbers *Nat* is axiomatized as follows where S is the successor function:
- $0:\mathbf{N}$
 - $+_1:\mathbf{N}^{\mathbf{N}}$
 - $\forall[i:\mathbf{N}] +_1[i] \neq 0$
 - $\forall[i, j:\mathbf{N}] +_1[i]=+_1[j] \Rightarrow i=j$
 - For each order: \mathbf{N}_+ and $P:\mathbf{Proposition}\langle\text{order}\rangle^{\mathbf{N}}$:
 $(P[0] \wedge \forall[i:\mathbf{N}] P[i] \Rightarrow P[+_1[i]]) \Rightarrow \forall[i:\mathbf{N}] P[i]$
- ⁴⁴ **O** is the type of the Ordinals
- ⁴⁵ $\forall[\tau::] \tau:\mathbf{Type}\langle\tau\rangle$
- ⁴⁶ Discrimination of τ_1 and τ_2
For $i=1,2$
- If $x:\tau_i$, then $((\tau_1 \circ \tau_2)[x]):(\tau_1 \circ \tau_2)$ and $x=((\tau_1 \circ \tau_2)[x]) \downarrow \tau_i$.
 - $\forall[z:\tau] z:\tau_1 \circ \tau_2 \Leftrightarrow \exists[x:\tau_i] z=(\tau_1 \circ \tau_2)[x]$
- ⁴⁷ type of 2-element list with first element of type τ_1 and with second element of type τ_2
- ⁴⁸ expression of type τ . The following axiom holds:
 $\forall[\tau::, e:\mathbf{Expression}\langle\tau\rangle] [e]::\tau$
- ⁴⁹ if p then Φ_1 else Φ_2
- ⁵⁰ \mathbf{x}_1 is a subtype of \mathbf{x}_2 , i.e., $\forall[x:\tau_1] x:\tau_2$
- ⁵¹ The proposition that τ is a type
- ⁵² Φ_1, \dots and Φ_{n-1} infer Φ_n

-
- ⁵³ mutually recursive definitions of functions $f_{1 \text{ to } n}$
- ⁵⁴ mutually recursive definitions of variables $x_{1 \text{ to } n}$
- ⁵⁵ mutually recursive definitions of functions $f_{1 \text{ to } n}$
- ⁵⁶ mutually recursive definitions of variables $x_{1 \text{ to } n}$
- ⁵⁷ 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]
- ⁵⁸ [Feferman 1986, pg. 1 and 8]
- ⁵⁹ Wittgenstein in 1937 published in Wittgenstein 1956, p. 50e and p. 51e]
- ⁶⁰ The Liar Paradox [Eubulides of Miletus] is an example of using untyped propositions to derive an inconsistency:
 $F \langle n \rangle [p:\text{Proposition} \langle n \rangle]:\text{Proposition} \langle n+1 \rangle \equiv \neg p$
 // above definition has no fixed point because $\neg p$ has
 // order greater than p
 The following argument derives a contradiction assuming the existence of a fixed point for F:
 1) $I'mFalse \Leftrightarrow \neg I'mFalse$ // *nonexistent* fixed point of F
 2) $\neg I'mFalse$ // proof by contradiction from 1)
 3) $I'mFalse$ // from 1) and 2)
- ⁶¹ [Church 1935] correctly proved inferential incompleteness (sometimes called the “First Incompleteness Theorem”) without using Gödel's *I'mUnprovable*. The Church theorem and its proof are very robust.
- ⁶² Subsequent further discussion of Wittgenstein's criticism of Gödel's writings has unfortunately misunderstood Wittgenstein.
 For example, [Berto 2009] granted that proof theoretically if $P \Leftrightarrow \neg P$, then:
 1) $\vdash \neg P$
 However, the above has proof consequences as follows:
 2) $\vdash P$ because $(\neg P) \Leftrightarrow P$ in 1) above
 3) $\vdash \vdash P$ because of 2) above
 4) $\vdash \neg P$ because $(\vdash P) \Leftrightarrow \neg P$ 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]

⁶³ Nat_1 is not a categorical theory because there are nonstandard (different from \mathbb{N}) countable types which satisfy the axioms of Nat_1 that have a (nonstandard) element that is larger than any number which can be reached by countably iterating the successor function starting with 0. Such a model can be constructed creating a new theory Nat_1^* by adding a new symbol ∞ and countably many axioms of the form $i < \infty$ for each natural number i . Since Nat_1^* is consistent, there is a type \mathbb{N}^* which satisfies the theory Nat_1^* . The type \mathbb{N}^* also satisfies the theory Nat_1 because the axioms of Nat_1 are a subset of the axioms of Nat_1^* .

⁶⁴ cf. [Church 1934]

⁶⁵ 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!

⁶⁶ Which is not the same as proving the much *stronger* proposition that Mathematics is structurally consistent, i.e., that there is no proof of contradiction from the inference rules of Direct Logic.

⁶⁷ In formalizing Gödel's proof, [Shankar 1994] and [O'Connor 2005] followed Gödel in using integers to code sentences using the \mathbf{Y} untyped fixed point operator on propositions.

⁶⁸ Wittgenstein in 1937 published in Wittgenstein 1956, p. 50e and p. 51e]

⁶⁹ Of course, it is completely unacceptable for every proposition to be provable and so measures must be taken to prevent this.

⁷⁰ [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 \mathbf{Y} fixed point operator for propositions and sentences.

⁷¹ using definition of BSet

⁷² using definition of BExpression

⁷³ substituting BNumber for n