



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-01566393v12

HAL Id: hal-01566393

<https://hal.science/hal-01566393v12>

Submitted on 17 Jun 2018 (v12), 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.

Copyright

Strong Types for Direct Logic

Carl Hewitt

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

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

Abstract

Mathematics in this article means the precise formulation of standard mathematical theories that axiomatize the following up to a unique isomorphism: booleans, natural numbers, reals, ordinals, sets, lambda calculus procedures, and Actors.

In a strongly typed mathematical theory, every proposition, term, and expression has a type where there is **no** universal type *Any*. Types are constructed bottom up from types that are categorically axiomatized.¹

Strong types are extremely important in Direct Logic because they block *all* known paradoxes including Berry [Russell 1906], Burali-Forti [Burali-Forti 1897], Church [Church 1934], Curry [Curry 1941], Girard [Girard 1972], and Liar [Eubulides of Miletus], Löb[Löb 1955], and Russell[Russell 1902].² Blocking known paradoxes makes Direct Logic safer for use in Intelligent Applications by preventing security holes. Consistent strong mathematical theories can be freely used without introducing additional inconsistent information into inconsistency robust empirical theories that will be the core of future Intelligent Applications.

Direct Logic³ is called “direct” because it directly deals with propositions instead of attempting to deal with them indirectly using Gödel numbers as is done in Provability Logic⁴. Because propositions in Direct Logic are uncountable, it is impossible to give a Gödel number to every proposition.⁵

Inconsistency Robustness is performance of information systems (including scientific communities) with pervasively inconsistent information.⁶ 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.

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.

After [Church 1934], logicians faced the following dilemma:

- 1st order theories cannot be powerful lest they fall into inconsistency because of Church’s Paradox.
- 2nd order theories contravene the philosophical doctrine that theorems must be computationally enumerable.

The above issues can be addressed by requiring Mathematics to be strongly typed using so that:

- Mathematics self proves that it is “open” in the sense that theorems are not computationally enumerable.⁷
- Mathematics self proves that it is *formally* consistent.⁸
- Strong mathematical theories for Natural Numbers, Ordinals, Set Theory, the Lambda Calculus, Actors, etc. are inferentially decidable, meaning that every true proposition is provable and every proposition is either provable or disprovable.⁹

Mathematical Foundation for Computer Science

All mathematical entities are instances of types.

Computer Science brought different concerns and a new perspective to mathematical foundations including the following requirements (building on [Maddy 2018]):

- **Practicality** is providing powerful machinery so that arguments (proofs) can be short and understandable and
- **Generality** is formalizing inference so that all of mathematics can take place side-by-side. Direct Logic provides a generality by formalizing theories of the natural numbers, reals, ordinals, set theory, groups, lambda calculus, and Actors side-by-side.
- **Shared Standard** of what counts as legitimate mathematics so people can join forces and develop common techniques and technology. According to [Burgess 2015]:

To guarantee that rigor is not compromised in the process of transferring material from one branch of mathematics to another, it is essential that the starting points of the branches being connected ... be compatible. ... The only obvious way ensure compatibility of the starting points ... is ultimate to derive all branches from a common unified starting point.

Direct Logic provides such a common unified starting point including natural numbers, reals, ordinals, set theory, group theory, geometry, algebra, lambda calculus, and Actors that are axiomatized up to a unique isomorphism.

- **Abstraction** so that fundamental mathematical structures can be characterized up to a unique isomorphism including natural numbers, reals, ordinals, set theory, groups, lambda calculus, and Actors.
- **Guidance** is for practioners in their day-to-day work by providing relevant structures and methods free of extraneous factors. Direct Logic provides guidance by providing strong parameterized types and intuitive categorical inductive axiomatizations of natural numbers, ordinals, sets, lambda calculus, and Actors.
- **Meta-Mathematics** is the formalization of logic and rules of inference. Direct Logic facilitates meta-mathematics because it is its own meta-theory.
- **Automation** is facilitated in Direct Logic by making type checking very easy and intuitive along as well as incorporating Jaśkowski natural deduction for building an inferential system that can be used in everyday work.

- **Risk Assessment** is the danger of contradictions emerging in classical mathematical theories. In this regard, Direct Logic formalizes long-established and well-tested mathematical practice while blocking all known paradoxes including Berry [Russell 1906], Burali-Forti [Burali-Forti 1897], Church [Church 1934], Curry [Curry 1941], Girard [Girard 1972], and Liar [Eubulides of Miletus], Löb[Löb 1955], and Russell[Russell 1902].

Intuitive categorical *inductive* axiomatizations of natural numbers, propositions, types, ordinals, sets, lambda calculus, and Actors promote confidence in consistency.

Consistent mathematical theories can be freely used in (inconsistent) empirical theories without introducing additional inconsistency.¹⁰

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.”¹¹

“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]

Note: parametrized types¹² in Direct Logic are more intuitive for classical mathematical theories than intuitionistic types [Martin-Löf 1998].¹³

Booleans are Propositions although Propositions are not reducible to Booleans:

- **True:** *Boolean*
- **False:** *Boolean*
- *Boolean* \sqsubseteq *Proposition* //each Boolean is a Proposition
- *Boolean* \neq *Proposition* //some Propositions are **not** Booleans
- $(3=3) \neq \mathbf{True}$ //the proposition $3=3$ is **not** equal to **True**
- $(3=3) \neq (4=4)$
//the proposition $3=3$ is **not** equal to the proposition $4=4$
- $(3=4) \neq \mathbf{False}$ //the proposition $3=4$ is **not** equal to **False**

In Direct Logic, unrestricted recursion is allowed in programs. For example, There are uncountably many Actors. For example, the procedure call $\text{Real}_\bullet[\]$ can output any real number using binary representation between 0 and 1 where¹⁴

$\text{Real}:[\] \rightarrow \mathcal{R}_{[0,1]}$ // Real is a procedure of
 // no arguments that returns a $\mathcal{R}_{[0,1]}$

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

where

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

Also, there are uncountably many propositions (because there is a different proposition for every real number). For example, the function p defined as follows:

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

defines a different predicate p[x] for each real number x, which holds for only one real number, namely x.¹⁵

Propositions are used to state theorems, conjectures, etc. of Mathematics.

Propositions

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

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

Strings are important as a means for constructing propositions. If s is a well formed string **with no free variables**, then [s] is a proposition. Since propositions are uncountable, there are propositions that cannot be abstracted from strings. Theorems are invariably abstracted from strings in mathematical practice.

Strings

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

i.e., **string** for *proposition* that

for every \mathcal{N} there is a larger \mathcal{N}

$[\forall [n:\mathcal{N}] \exists [m:\mathcal{N}] m > n] = \forall [n:\mathcal{N}] \exists [m:\mathcal{N}] m > n$

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 .

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

Higher Order Logic

“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.”¹⁶

Classical Direct Logic is much stronger than 1st order axiomatizations of set theory in that it provides categoricity for natural numbers \mathbb{N} , reals \mathbb{R} , ordinals \mathcal{O} . set theory, the lambda calculus and Actors. 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.

Natural Number Induction

The mathematical theory¹⁷ **Nat** that axiomatises the Natural Numbers has the following axioms:

- $\mathbb{N} ::$ // \mathbb{N} is a type
- $0:\mathbb{N}$ // 0 is of type \mathbb{N}
- $+_1:\mathbb{N}^{\mathbb{N}}$ // $+_1$ is of type $\mathbb{N}^{\mathbb{N}}$
- $\nexists[i:\mathbb{N}] +_1[i]=0$
- $\forall[i,j:\mathbb{N}] +_1[i]=+_1[j] \Leftrightarrow i=j$

In addition **Nat** has the following induction axiom, *which has uncountable instances*:¹⁸

$$\forall [P: \text{Proposition} \langle \mathbf{Nat} \rangle^{\mathbf{N}}] (P[0] \wedge \forall [i: \mathbf{N}] P[i] \Rightarrow P[+_1[i]]) \Rightarrow \forall [i: \mathbf{N}] P[i]$$

The above induction axiom makes use of the type $\text{Proposition} \langle \mathbf{Nat} \rangle$ which has the following axioms:¹⁹

- $\text{Proposition} \langle \mathbf{Nat} \rangle :: \quad \quad \quad // \text{Proposition} \langle \mathbf{Nat} \rangle$ is a type
- $\forall [t_1, t_2: \text{Type} \langle \mathbf{Nat} \rangle]; x: t_1] (x: t_2): \text{Proposition} \langle \mathbf{Nat} \rangle$
- $\forall [t: \text{Type} \langle \mathbf{Nat} \rangle; x_1, x_2: t] (x_1 = x_2): \text{Proposition} \langle \mathbf{Nat} \rangle$

In addition to the above, propositions of **Nat** have the following induction axiom:

$$\begin{aligned} \forall [Q: \text{Proposition} \langle \mathbf{Nat} \rangle^{\text{Proposition} \langle \mathbf{Nat} \rangle}] \\ & ((\forall [t: \text{Type} \langle \mathbf{Nat} \rangle; x_1, x_2: t] Q[x_1 = x_2]) \wedge \\ & (\forall [\Psi: \text{Proposition} \langle \mathbf{Nat} \rangle] Q[\Psi] \Rightarrow Q[\neg \Psi]) \wedge \\ & (\forall [\Psi_1, \Psi_2: \text{Proposition} \langle \mathbf{Nat} \rangle] Q[\Psi_1] \wedge Q[\Psi_2] \Rightarrow Q[\Psi_1 \wedge \Psi_2]) \wedge \\ & (\forall [t: \text{Type} \langle \mathbf{Nat} \rangle; P: \text{Proposition} \langle \mathbf{Nat} \rangle^t] (\forall t Q[P]) \Rightarrow Q[\forall t P])) \\ & \Rightarrow \forall [\Psi: \text{Proposition} \langle \mathbf{Nat} \rangle] Q[\Psi] \end{aligned}$$

The above axioms characterize $\text{Proposition} \langle \mathbf{Nat} \rangle$ up to a unique isomorphism.²⁰

Computational Undecidability of Provability in **Nat**

The halting problem is to computationally decide whether a given procedure (say $p: (\mathbf{N} \rightarrow \mathbf{N})$) halts on a given input (say $i: \mathbf{N}$), which is formally expressed as $\text{Halt}[p, i]$. [Church 1935; Turing 1936] proved that the halting problem is computationally undecidable. Inferability in **Nat** is computationally undecidable because the predicate Halt can be formalized in **Nat** such that:

$$\forall [i: \mathbf{N}, p: (\mathbf{N} \rightarrow \mathbf{N})] \text{Halt}[p, i] \Leftrightarrow \vdash_{\mathbf{Nat}} \text{Halt}[p, i]$$

Proof Checkers in Nat

$$\forall[\Psi_1, \Psi_2: \text{Proposition} \langle \text{Nat} \rangle]$$

$$(\Psi_1 \vdash_{\text{Nat}} \Psi_2) \Leftrightarrow \exists[p: \text{ProofChecker} \langle \text{Nat} \rangle] \Psi_1 \vdash_{\text{Nat}} \frac{p}{\text{Nat}} \Psi_2$$

where

$$\text{ProofChecker} \langle \text{Nat} \rangle \equiv \text{Total} \langle [\text{Proposition} \langle \text{Nat} \rangle, \text{Proposition} \langle \text{Nat} \rangle] \rightarrow_1 \text{Boolean} \rangle$$

Proof checking in **Nat** is **computationally decidable**.

$$\forall[\Psi_1, \Psi_2: \text{Proposition} \langle \text{Nat} \rangle, p: \text{ProofChecker} \langle \text{Nat} \rangle]$$

$$(\Psi_1 \vdash_{\text{Nat}} \frac{f}{\text{Nat}} \Psi_2) \Leftrightarrow p.[\Psi_1, \Psi_2] = \text{True}$$

Nat proof checkers can have uncountable domains

The $+_1$ proof checker below has a nondenumerable domain even though it is an effectively computable procedure.²¹

$+_1\text{Checker}: \text{ProofChecker} \langle \text{Nat} \rangle$ where

$+_1\text{Checker}.[\Psi_1, \Psi_2] \equiv \Psi_1 [?] (i: \mathbf{N}) \text{ then } \Psi_2 = (i+1: \mathbf{N}) \text{ else False}$

// If Ψ_1 is $i: \mathbf{N}$, then $\Psi_2 = (i+1: \mathbf{N})$, otherwise False

$$\forall[i: \mathbf{N}] (i: \mathbf{N}) \vdash_{\text{Nat}} \frac{+_1\text{Checker}}{\text{Nat}} (i+1: \mathbf{N})$$

Nat has uncountable proof checkers

There are uncountable proof checkers of the form $\text{ForAllElimChecker}[c]$ where $t: \text{Type} \langle \text{Nat} \rangle$ and $c: t$ such that

$\text{ForAllElimChecker}[c].[\Psi_1, \Psi_2] \equiv$

$\Psi_1 [?] (\forall[x: t] P[x]) \text{ then } \Psi_2 = P[c] \text{ else False}$

// If Ψ_1 is $\forall[x: t] P[x]$, then $\Psi_2 = P[c]$, otherwise False

Consequently,

$$(\forall[x: t] P[x]) \vdash_{\text{Nat}} \frac{\text{ForAllElimChecker}[c]}{\text{Nat}} P[c]$$

Theorem (Indiscernibility for Nat):

$$\forall [i,j:N] i=j \Leftrightarrow \forall [P:\text{Proposition} \langle \text{Nat} \rangle^N] P[i] \Leftrightarrow P[j]$$

Proof. Define Same: $(\text{Proposition} \langle \text{Nat} \rangle^N)^N$

$$\text{Same}[i] \equiv \lambda [j:N] i=j$$

$$\forall [i,j:N] (\text{Same}[i] \Leftrightarrow \text{Same}[j]) \Leftrightarrow i=j$$

Axiomatization of \models

Axioms of \models for **Nat** are as follows:

- $\models \mathbf{True}$
- $\neg \models \mathbf{False}$
- $(\models \Phi \wedge \Psi) \Leftrightarrow (\models \Phi) \wedge (\models \Psi)$
- $(\models \Phi \vee \Psi) \Leftrightarrow (\models \Phi) \vee (\models \Psi)$
- $(\models \neg \Phi) \Leftrightarrow \neg \models \Phi$
- $\forall [P:\text{Proposition} \langle \text{Nat} \rangle^N] (\models \forall [i:N] P[i]) \Leftrightarrow (\forall [i:N] \models P[i])$
- $\forall [P:\text{Proposition} \langle \text{Nat} \rangle^N] (\models \exists [i:N] P[i]) \Leftrightarrow (\exists [i:N] \models P[i])$

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

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

Theorem (Categoricity of Nat):²²

If \mathcal{X} be a type satisfying the axioms for the natural numbers \mathbf{Nat} , then there is a unique isomorphism I with \mathbf{N} defined as follows:

- $I: \mathcal{X}^{\mathbf{N}}$
- $I[0_{\mathbf{N}}] \equiv 0_{\mathcal{X}}$
- $I[+_1[j]] \equiv +_1^{\mathcal{X}}[I[j]]$

because

- I is defined on \mathbf{N}
- I is 1-1
- I is onto \mathcal{X}
- I is a homomorphism
 - $I[0_{\mathbf{N}}] \equiv 0_{\mathcal{X}}$
 - $\forall [i: \mathbf{N}] I[+_1[j]] \equiv +_1^{\mathcal{X}}[I[j]]$
- I^{-1} is a homomorphism
 - $I^{-1}[0_{\mathcal{X}}] \equiv 0_{\mathbf{N}}$
 - $\forall [z: \mathcal{X}] I^{-1}[+_1^{\mathcal{X}}[z]] \equiv [+_1[I^{-1}[z]]]$
- If g is an isomorphism with \mathcal{X} , then $g=I$



Richard Dedekind

Corollary There are no infinite numbers in models of the theory \mathbf{Nat} , i.e.,

$$\forall [\mathcal{X}:: \mathbf{Nat} \langle \mathcal{X} \rangle \Rightarrow \nexists [j: \mathcal{X}] \forall [i: \mathcal{X}] j < i$$

Proof by Contradiction in Nat

“[Proof by contradiction is] one of a mathematician’s first weapons It is a far finer gambit than any chess gambit: a chess player may offer the sacrifice of a pawn or even a piece, but a mathematician offers the game.”

G. H. Hardy ²³

Proof by Contradiction is one of the most fundamental principles of Classical Mathematics (going back to before Euclid), which can be formalized

- axiomatically to say that if Ψ implies Φ and $\neg\Phi$ then $\neg\Psi$:

$$(\Psi \Rightarrow \Phi \wedge \neg\Phi) \Rightarrow \neg\Psi$$

- proof theoretically to say that proving the theorem $\neg\Psi \Rightarrow \Phi \wedge \neg\Phi$ means that Ψ is a theorem:

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

- in [Jaśkowski 1934] natural deduction to say that (Ψ infers Φ and $\neg\Phi$) holds in a subproof ²⁴ of a proof infers that $\neg\Psi$ holds in the proof:

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

Proof by contradiction is much more powerful than has been supposed by many philosophers of mathematics. As shown in this article, it is sufficient to prove the consistency and the inferential completeness of categorical theories of the natural numbers, reals, ordinals, set theory, the lambda calculus, and Actors.

Theorem:²⁵ **Inferential completeness of Nat**

$$\forall[\Psi:\textit{Proposition} \langle \text{Nat} \rangle] (\models \Psi) \Rightarrow \vdash_{\text{Nat}} \Psi$$

Proof.

Suppose in **Nat**, $\Psi:\textit{Proposition} \langle \text{Nat} \rangle$ and $\models \Psi$. Further suppose to obtain a contradiction that $\neg\Psi$. Hence Ψ and $\neg\Psi$, which is a contradiction. Therefore $\vdash_{\text{Nat}} \Psi$ using proof by contradiction in **Nat**.²⁶

Although by the above theorem **Nat** is inferentially complete, some proofs are not expressible as character strings. It is an open problem to characterize theorems of **Nat** whose proofs cannot be expressed as character strings.

Corollary. Equivalence of satisfiability and provability in **Nat**, i.e.,

$$\forall[\Psi:\textit{Proposition}\langle\textit{Nat}\rangle] (\models \Psi) \Leftrightarrow (\vdash_{\textit{Nat}}\Psi)$$

Theorem. Inferential Decidability of **Nat**, i.e.,

$$\forall[\Psi:\textit{Proposition}\langle\textit{Nat}\rangle] (\vdash_{\textit{Nat}}\Psi) \vee (\vdash_{\textit{Nat}}\neg\Psi)$$

Proof. Follows immediately from $(\models \Psi) \Leftrightarrow (\vdash_{\textit{Nat}} \Psi)$

Theorem (Instance Adequacy of Nat):²⁷

$$\forall[\textit{P}:\textit{Proposition}\langle\textit{Nat}\rangle] (\forall[i:\mathcal{N}] \vdash_{\textit{Nat}} \textit{P}[i]) \Leftrightarrow \vdash_{\textit{Nat}} \forall[i:\mathcal{N}] \textit{P}[i]$$

Proof: Suppose $\forall[i:\mathcal{N}] \vdash_{\textit{Nat}} \textit{P}[i]$ which means by completeness $\forall[i:\mathcal{N}] \models \textit{P}[i]$.

Therefore $\forall[i:\mathcal{N}] \models \textit{P}[i]$ which means by completeness $\vdash_{\textit{Nat}} \forall[i:\mathcal{N}] \textit{P}[i]$

Definition Total: $\textit{Proposition}\langle\textit{Nat}\rangle^{[\mathcal{N}]\rightarrow\mathcal{N}}$ where

$$\text{Total}[f] \equiv \forall[i:\mathcal{N}] \exists[j:\mathcal{N}] f.[i]=j$$

Theorem. Theorems of the theory **Nat** are **not** computationally enumerable

$\vdash \text{Nat} \neg \text{NatTheoremsEnumerableByProvableTotalProcedure}$

Proof.²⁸

Suppose to obtain a contradiction that

$\text{NatTheoremsEnumerableByProvableTotalProcedure}$

Then by the definition of

$\text{NatTheoremsEnumerableByProvableTotalProcedure}$ there is a deterministic total procedure

$\text{NatTheoremsEnumerator} : \text{Total} \langle [N] \rightarrow_1 \text{Proposition} \langle \text{Nat} \rangle \rangle$ such that the following hold where:

- $\forall [i: N] \vdash \text{NatTheoremsEnumerator} \bullet [i]$
- $\forall [p: \text{Proposition} \langle \text{Nat} \rangle] (\vdash p) \Rightarrow \exists [i: N] \text{NatTheoremsEnumerator} \bullet [i] = p$

A subset of the theorems enumerated by $\text{NatTheoremsEnumerator}$ are those stating that certain real numbers are provably computable. Consequently, there is a deterministic total procedure

$\text{NatProvablyComputableR}_{[0,1]} \text{Enumerator} : \text{Total} \langle [N] \rightarrow_1 \text{NatProvablyComputableR}_{[0,1]} \rangle$

which enumerates provably computable real numbers:

- $\forall [i: N] (\text{NatProvablyComputableR}_{[0,1]} \text{Enumerator} \bullet [i]) : \text{NatProvablyComputableR}_{[0,1]}$
- $\forall [r: \text{NatProvablyComputableR}_{[0,1]}] \Rightarrow \exists [i: N] \text{NatProvablyComputableR}_{[0,1]} \text{Enumerator} \bullet [i] = r$

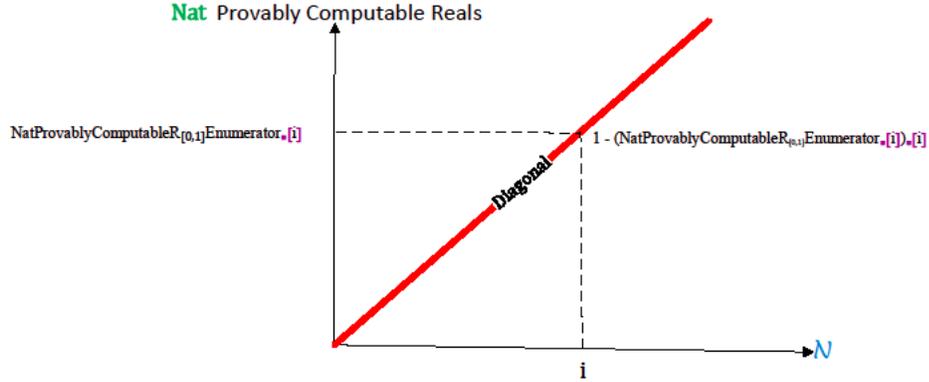
$\text{NatProvablyTotalEnumerator}$ can be used to implement the **Nat** provably deterministic total procedure $\text{Diagonal} : ([N] \rightarrow_1 \text{Boolean})$ as follows:

$\text{Diagonal} \bullet [i] \equiv 1 - (\text{NatProvablyComputableR}_{[0,1]} \text{Enumerator} \bullet [i]) \bullet [i]$

Consequently:

- $\text{Diagonal} : \text{NatProvablyComputableR}_{[0,1]}$ because it is the deterministic composition of provably total deterministic procedures.
- $\neg \text{Diagonal} : \text{NatProvablyComputableR}_{[0,1]}$ because Diagonal differs from every provably total deterministic procedure enumerated by $\text{NatProvablyComputableR}_{[0,1]} \text{Enumerator}$.

The above contradiction completes the proof.



$$\text{Diagonal}_{\bullet}[i:N]:\text{Boolean} \equiv 1 - (\text{NatProvablyComputableR}_{[0,1]}\text{Enumerator}_{\bullet}[i])_{\bullet}[i]$$

Theorem The theory **Nat** proves that its proofs cannot be expressed as character strings that are validity computationally decidable.

Proof: Suppose to obtain a contradiction that proofs can be expressed as character string that are validity computationally decidable. Since $\text{NatProvablyComputableR}_{[0,1]}$ is not computationally enumerable, proofs in **Nat** for $\text{NatProvablyComputableR}_{[0,1]}$ cannot be represented as character strings that are validity computationally decidable.

Corollary. There are theorems²⁹ in **Nat** that certain procedures are total whose proofs **cannot** be expressed as a character string.³⁰

Proof. Suppose to obtain a contradiction that all of the proofs of could be expressed using character strings. Then there is a contradiction because there is a provably total deterministic procedure

$$\text{NatTheoremsEnumerator}: [N] \rightarrow_1 \text{Proposition} \langle \text{Nat} \rangle$$

which enumerates the theorems of **Nat**, i.e.

- $\vdash_{\text{Nat}} \text{Total}[\text{NatTheoremsEnumerator}]$
because the procedure for enumerating character string proofs is total
- $\forall [p: \text{Theorem} \langle \text{Nat} \rangle] \exists [i:N] \text{NatTheoremsEnumerator}_{\bullet}[i]=p$
because every character string of a proof is enumerated
- $\forall [i:N] \vdash_{\text{Nat}} \text{NatTheoremsEnumerator}_{\bullet}[i]$
because only character strings of proofs are enumerated

Theorem The theory **Nat** proves that it is formally consistent:³¹

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

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

Ψ_0 : *Proposition* 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.

Uneven Progress

“All truth passes through three stages. First, it is ridiculed. Second, it is violently opposed. Third, it is accepted as being self-evident.”
Arthur Schopenhauer [1818]

“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:

- **The formal presentation of a demonstration (proof) has not led 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 proves that it is formally 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).³²
- **Argumentation and propositions have been variously (re-)connected and both have been re-used.** Church's paradox [Church 1934] is that assuming theorems of mathematics are computationally enumerable leads to contradiction. In this article, Church's 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). See discussion in this article.
- **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.³³ That there are proofs that cannot be expressed through text alone, overturns a long-held philosophical dogma about mathematical theories, i.e., that all theorems of a theory can be computationally generated by starting with axioms and mechanically applying rules of inference.
- **It has been easier to prove meta theorems for 1st order logic.** Since theoreticians found it difficult to prove anything significant about practical mathematical theories, they cut them down to unrealistic relational 1st 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.³⁴
- **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 fired³⁵ Brouwer as an associate editor of *Mathematische Annalen* because of "*incompatibility of our views on fundamental matters*"³⁶ e.g., Hilbert ridiculed Brouwer for challenging the validity of the Principle of Excluded Middle. [Gödel 1931] results were for Principia Mathematica 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 insinuated³⁷ that he was crazy and retreated to relational 1st order theory in an attempt to salvage his results.

Summary of Nat

The theory **Nat** can be summarized as follows:

- **Indiscernibility for Nat:**

$$\forall [i, j: \mathbb{N}] i=j \Leftrightarrow \forall [P: \text{Proposition} \langle \mathbb{N} \rangle^{\mathbb{N}}] P[i] \Leftrightarrow P[j]$$

- **Instance Adequacy of Nat:**

$$\forall [P: \text{Proposition} \langle \mathbb{N} \rangle^{\mathbb{N}}] (\forall [i: \mathbb{N}] \vdash_{\text{Nat}} P[i]) \Leftrightarrow \vdash_{\text{Nat}} \forall [i: \mathbb{N}] P[i]$$

- **Nat is categorical for \mathbb{N}**

$$\vdash_{\text{Nat}} \forall [X::] \text{Nat} \langle X \rangle \Leftrightarrow \text{UniquelyIsomorphic}[X, \mathbb{N}]$$

- **Nat proves its own consistency**

$$\vdash_{\text{Nat}} (\neg \exists [\Psi: \text{Proposition} \langle \mathbb{N} \rangle] \vdash_{\text{Nat}} \Psi \wedge \neg \Psi)$$

- **A proposition is true \Leftrightarrow it is provable in Nat**

$$\forall [\Psi: \text{Proposition} \langle \mathbb{N} \rangle] (\models \Psi) \Leftrightarrow (\vdash_{\text{Nat}} \Psi)$$

- **Nat is inferentially decidable**

$$\forall [\Psi: \text{Proposition} \langle \mathbb{N} \rangle] (\vdash_{\text{Nat}} \Psi) \vee (\vdash_{\text{Nat}} \neg \Psi)$$

- **$\vdash_{\text{Nat}} \Psi$ is computationally undecidable for $\Psi: \text{Proposition} \langle \mathbb{N} \rangle$**

Computational Limitations of Lambda Calculus, Turing Machines, and Pure Logic Programs

Actors can perform computations that cannot be implemented in the nondeterministic lambda calculus, Turing Machines, or using pure logic programs.³⁸

By contrast with the nondeterministic lambda calculus and pure Logic Programs, there is an always-halting Actor Unbounded that when sent a **start[]** message can compute an integer of unbounded size. This is accomplished by creating a counter with the following variables:

- **count** initially **0**
- **continue** initially **True**

and concurrently sending it both a **stop[]** message and a **go[]** message such that:

- When a **go[]** message is received:
 1. if **continue** is **True**, increment **count** by 1 and return the result of sending this counter a **go[]** message.
 2. if **continue** is **False**, return **Void**
- When a **stop[]** message is received, return **count** and set **continue** to **False** for the next message received.

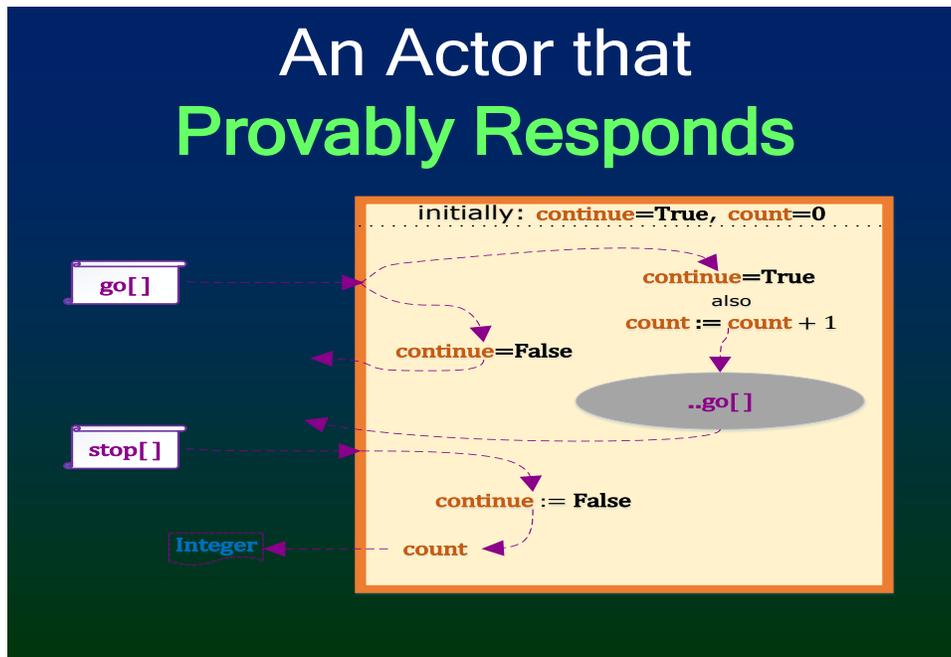
By the Actor Model of Computation, the above Actor will eventually receive the **stop[]** message and return an unbounded number.

Categorical Axiomatization of Actors

For each Actor x , $x[t]$ is the behavior³⁹ of x at time t of type $Time \langle x \rangle$ that specifies an outcome for a communication received in terms of a finite set of created Actors, a finite set of sent communications, and a behavior for the next communication received. The mathematical theory **Act** categorically axiomatises Actors using the following axioms where \rightsquigarrow is transitive and irreflexive relationship between events:

- Primitive Actors
 - $\forall [i: \mathbb{N}] \ i: Actor$ // natural numbers are Actors
 - $\forall [x_1, x_2: Actor] \ [x_1, x_2]: Actor$ // a tuple of Actors is an Actor
- An Actor's event ordering
 - $\forall [x: Actor, c_1, c_2: Com] \ c_1 \neq c_2 \Rightarrow Received_x[c_1] \rightsquigarrow Received_x[c_2]$
 $\vee Received_x[c_2] \rightsquigarrow Received_x[c_1]$
 - $\forall [x: Actor, c_1: Com]$
 $\nexists [c_2: Com] \ Received_x[c_1] \rightsquigarrow Received_x[c_2] \rightsquigarrow After_x[c_1]$
 - $\forall [x: Actor, c: Com] \ Initial_x \rightsquigarrow Received_x[c] \rightsquigarrow After_x[c]$
 - $\forall [x: Actor, c_1, c_2: Com]$
 $Finite[\{c: Com \mid Received_x[c_1] \rightsquigarrow Received_x[c] \rightsquigarrow Received_x[c_2]\}]$
- An Actor's behavior change
 - $\forall [x: Actor, c_1: Com] \ (\nexists [c_2: Com] \ Received_x[c_2] \rightsquigarrow Received_x[c_1])$
 $\Rightarrow x[Received_x[c_1]] = x[Initial_x]$
 - $\forall [x: Actor, c_1, c_2: Com]$
 $(\nexists [c_3: Com] \ After_x[c_1] \rightsquigarrow Received_x[c_3] \rightsquigarrow Received_x[c_2])$
 $\Rightarrow x[Received_x[c_2]] = x[After_x[c_1]]$
- Between Actors event ordering
 - $\forall [c: Com] \ Sent[c] \rightsquigarrow Received[c]$
 - $\forall [c_1, c_2: Com]$
 $Finite[\{c: Com \mid$
 $\exists [x_1, x_2: Actor] \ Sent[c_1] \rightsquigarrow Received_{x_1}[c] \rightsquigarrow Received_{x_2}[c_2]\}]$
- Actor Induction
 - $\forall [x: Actor, P: Proposition \textit{Behavior}]$
 $(P[x[Initial_x]] \wedge \forall [c: Com] \ P[x[Received_x[c]]) \Leftrightarrow P[x[After_x[c]])$
 $\Leftrightarrow \forall [c: Com] \ P[x[Received_x[c]]] \wedge P[x[After_x[c]]]$

An Actor that Provably Responds



The following hold:

- $\forall [t:\mathbb{N}] \not\vdash_{\text{Act}} \text{ResponseBefore}[t]$ // unbounded response time
- $\vdash_{\text{Act}} \exists [t:\mathbb{N}] \text{ResponseBefore}[t]$ // provably responds

Theorem. Unbounded Nondeterminacy of Actors

The Actor Unbounded described above cannot be implemented as a nondeterministic lambda calculus expression and cannot be implemented as a pure Logic Program.

Theorem. Computational Adequacy of Actors.

If for each $i:\mathbb{N}$, F_i is a nondeterministic λ expression such that

$$\forall [i:\mathbb{N}] F_i \sqsubseteq F_{i+1} \Leftrightarrow (\text{limit}_{i:\mathbb{N}} F_i) : \text{Actor}$$

Theorem. Categoricity of Act

If X be a type satisfying the axioms for **Act**, then there is a unique isomorphism between X and *Actor*.

Theorem: Inferential completeness of the theory Act

$$\forall[\Psi:\textit{Proposition}\langle\textit{Act}\rangle] (\models \Psi) \Rightarrow (\vdash_{\textit{Act}}\Psi)$$

Corollary. Equivalence of satisfiability and provability in Act, i.e.,

$$\forall[\Psi:\textit{Proposition}\langle\textit{Act}\rangle] (\models \Psi) \Leftrightarrow (\vdash_{\textit{Act}}\Psi)$$

Theorem. Inferential Decidability of Act, i.e.,

$$\forall[\Psi:\textit{Proposition}\langle\textit{Act}\rangle] (\vdash_{\textit{Act}}\Psi) \vee (\vdash_{\textit{Act}}\neg\Psi)$$

Proof. Follows immediately from $(\models \Psi) \Leftrightarrow (\vdash_{\textit{Act}}\Psi)$

Mathematics self proves its own formal consistency (contra [Gödel 1931])

The following are fundamental to Mathematics⁴⁰:

- Derivation by Contradiction, i.e. $\vdash(\neg\Phi \Rightarrow (\Theta \wedge \neg\Theta)) \Rightarrow \Phi$, which says that a proposition can be proved showing that its negation implies a contradiction.
- A theorem can be used in a proof⁴¹, i.e. $\vdash((\vdash\Phi) \Rightarrow \Phi)$

Theorem: Mathematics self proves its own formal consistency⁴², i.e., $\vdash\text{Consistent}$

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 proposition Ψ_0 such that $\vdash(\Psi_0 \wedge \neg\Psi_0)$ which by the Theorem Use means $\Psi_0 \wedge \neg\Psi_0$, which is a contradiction. Thus, $\vdash\text{Consistent}$ by Derivation by Contradiction.

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

2) $\vdash(\Psi_0 \wedge \neg\Psi_0)$ // definition of inconsistency using 1)

3) $\Psi_0 \wedge \neg\Psi_0$ // axiom of Soundness using 2)

$\vdash\text{Consistent}$ // axiom of Proof by Contradiction using 1) and 3)

Natural Deduction Proof of Formal Consistency of Mathematics

Please note the following points:

- The above argument mathematically proves that Mathematics is formally consistent and that **it is not a premise of the theorem that Mathematics is formally consistent.**
- Mathematics was designed for consistent theories and consequently Mathematics can be used to prove its own formal consistency regardless of other axioms.⁴³
- [Gödel 1931] used formal consistency as the basis of his results. The above theorem does not prove that Mathematics is operationally consistent, i.e., that no contradiction can be derived using the stated axioms and rules.

The above derivation means that “Mathematics is formally consistent” is a theorem in Classical Direct Logic.

The above self-proof of formal consistency shows that the current common understanding that [Gödel 1931] proved “Mathematics cannot prove its own formal consistency, if it is formally consistent” is inaccurate.⁴⁴

Mathematics Self Proves that it is Open.

Mathematics proves that it is open in the sense that it can prove that its theorems cannot be computationally enumerated by a provably total procedure:

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

$\vdash \neg$ TheoremsEnumerableByProvablyTotalProcedure

Proof.⁴⁵

Suppose to obtain a contradiction that

TheoremsEnumerableByProvablyTotalProcedure

Then by the definition of

TheoremsEnumerableByProvablyTotalProcedure there is a provably deterministic total procedure TheoremsEnumerator: $[N] \rightarrow_1$ Proposition

such that the following hold where:

- Total[TheoremsEnumerator]
- $\forall [i: N] \vdash$ TheoremsEnumerator. \blacksquare [i]
- $\forall [p: Proposition] (\vdash p) \Rightarrow \exists [i: N] \text{TheoremsEnumerator.}\blacksquare$ [i]=p

A subset of the theorems enumerated by TheoremsEnumerator are those stating that certain real numbers are provably computable. Consequently, there is a provably deterministic total procedure

ProvablyComputableR_[0,1] Enumerator: ($[N] \rightarrow_1$ ProvablyComputableR_[0,1])

which enumerates provably computable real numbers:

- Total[ProvablyComputableR_[0,1] Enumerator]
- $\forall [i: N] (\text{ProvablyComputableR}_{[0,1]} \text{ Enumerator.}\blacksquare$ [i]): ProvablyComputableR_[0,1]
- $\forall [r: ProvablyComputableR_{[0,1]}]$

$\exists [i: N] \text{ProvablyComputableR}_{[0,1]} \text{ Enumerator.}\blacksquare$ [i]=r

ProvablyTotalEnumerator can be used to implement the provably

deterministic total procedure Diagonal: ($[N] \rightarrow_1$ Boolean) as follows:

Diagonal. \blacksquare [i] \equiv 1 - (ProvablyComputableR_[0,1] Enumerator. \blacksquare [i]). \blacksquare [i]

Consequently:

- Diagonal: ProvablyComputableR_[0,1] because it is the deterministic composition of provably total deterministic procedures.
- \neg Diagonal: ProvablyComputableR_[0,1] because Diagonal differs from every provably computable real number enumerated by ProvablyComputableR_[0,1] Enumerator.

The above contradiction completes the proof.

Some but not all of the theorems of Mathematics can be computationally enumerated.

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

Conclusion

Strong Types are extremely important in Direct Logic because they block all know paradoxes.

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.⁴⁶ [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 undecidability⁴⁷ results for mathematics using the proposition *I'mUnprovable* In opposition to Wittgenstein's correct argument his proposition leads to contradictions in mathematics, Gödel claimed that the results of [Gödel 1931] were for a cut-down relational 1st order theory of natural numbers. However, relational 1st order theories are 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 [Russell 1925, and Church 1932-1933], Direct Logic was developed and then investigated propositions with results below.

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 formal 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.⁴⁸ A solution is to require strongly typed mathematics to bar use of the **Y** untyped fixed point operator for propositions.⁴⁹ However, some theoreticians have very reluctant to accept the solution.

According to [Dawson 2006]:⁵⁰

- *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 1931] has produced a controversy of a very different kind from the one discussed by Dawson:

- The common understanding that mathematics cannot prove its own formal consistency⁵¹ has been disproved.
- Consequently, [Gödel 1931] has now led to increased controversy over mathematical foundations.

Requirement to use higher order logic because moderately strong theories of 1st order logic are inconsistent. Categorical higher order theories of Natural Numbers, Reals, and Actors are inferentially complete and inferentially decidable. In general, theorems of theories in higher order logic are not computationally enumerable, inferability is computationally undecidable, proof checking is

computationally decidable, and some proofs are inexpressible as character strings. Consequently, it will be forever necessary to invent new proof notations that were previously not expressed in a process called “Progressive Knowing”.

Direct Logic with strong types categorically axiomatize up to a mathematical theory \mathbf{T} for the model \mathbf{M} for each of the following models: Natural Numbers, Real Numbers, Computable Procedures, and Actors as follows:

- **Indiscernibility for theory \mathbf{T} :**

$$\forall[x_1, x_2: \mathbf{M}] i=j \Leftrightarrow \forall[\mathbf{P}: \text{Proposition} \langle \mathbf{T} \rangle^{\mathbf{M}}] P[x_1] \Leftrightarrow P[x_2]$$
- **Instance Adequacy of \mathbf{T} :**

$$\forall[\mathbf{P}: \text{Proposition} \langle \mathbf{T} \rangle^{\mathbf{M}}] (\forall[x: \mathbf{M}] \vdash_{\mathbf{T}} P[x]) \Rightarrow \vdash_{\mathbf{T}} \forall[x: \mathbf{M}] P[x]$$
- **\mathbf{T} is categorical for \mathbf{M} :**

$$\vdash_{\mathbf{T}} \forall[\mathbf{X}::] \mathbf{T} \langle \mathbf{X} \rangle \Leftrightarrow \text{UniquelyIsomorphic}[\mathbf{X}, \mathbf{M}]$$
- **\mathbf{T} proves its own consistency**

$$\vdash_{\mathbf{T}} (\neg \exists[\Psi: \text{Proposition} \langle \mathbf{T} \rangle] \vdash_{\text{Theory}} \Psi \wedge \neg \Psi)$$
- **A proposition is true \Leftrightarrow it is provable in \mathbf{T}**

$$\forall[\Psi: \text{Proposition} \langle \mathbf{T} \rangle] (\models \Psi) \Leftrightarrow (\vdash_{\text{Theory}} \Psi)$$
- **\mathbf{T} is inferentially decidable**

$$\forall[\Psi: \text{Proposition} \langle \mathbf{T} \rangle] (\vdash_{\mathbf{T}} \Psi) \vee (\vdash_{\mathbf{T}} \neg \Psi)$$
- **$\vdash_{\mathbf{T}} \Psi$ is computationally undecidable for $\Psi: \text{Proposition} \langle \mathbf{T} \rangle$**
- **$\vdash_{\mathbf{T}}^c \Psi$ is computationally decidable for $c: \text{Checker} \langle \mathbf{T} \rangle$ and $\Psi: \text{Proposition} \langle \mathbf{T} \rangle$**

The development of Direct Logic has strengthened the position of working mathematicians as follows:⁵²

- Allowing freedom from the philosophical dogma of the 1st 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. Conversations with Gordon Plotkin were helpful and his suggestions led to the introduction of the fixed point operator in the categorical theory of the lambda calculus. Michael Beeson suggested a clarification in the argument of Church's Paradox. Hendrik Boom provided an excellent critique, which resulted in a fundamental reorganization. Wilfried Sieg provided information about provability in 1st order logic. John Woods and two anonymous referees provided excellent comments and suggestions that materially improved this article. Michael Beeson pointed out that I needed to use the terminology "proof checker."

Bibliography

- Anthony Anderson and Michael Zelëny (editors). *Logic, Meaning and Computation: Essays in Memory of Alonzo Church* Springer. 2002.
- Andrea Asperti, Herman Geuvers, Raja Natarajan. *Social processes, program verification and all that* “Mathematical Structures in Computer Science” Cambridge University Press. 2009.
- Jeremy Avigad and John Harrison. *Formally Verified Mathematics*. CACM. April 2014.
- Steve Awodey and Erich Reck. *Completeness and Categoricity. Parts I and II: Nineteenth-century Axiomatics to Twentieth-century Metalogic*. History and Philosophy of Logic. Vol. 23. 2002.
- Steve Awodey, Álvaro Pelayo, and Michael A. Warren. *Voevodsky’s Univalence Axiom in Homotopy Type Theory* Notices of AMS. October 2013.
- Jon Barwise. *Model-Theoretic Logics: Background and Aims* in “Model Theoretic Logics” Springer-Verlag. 1985.
- Francesco Berto. *The Gödel Paradox and Wittgenstein’s Reasons* Philosophia Mathematica. February, 2009.
- Andrey Bovykin. *Brief introduction to unprovability*. Logic Colloquium 2006. Lecture Notes in Logic 2009.
- Cesare Burali-Forti. *Una questione sui numeri transfiniti* Rendiconti del Circolo Matematico di Palermo. 1897.
- John Burgess. *Rigor and Structure* Oxford University Press. 2015.
- Andrea Cantini and Riccardo Bruni. *Paradoxes and Contemporary Logic* Stanford Encyclopedia of Philosophy. February 22, 2017.
- Georg Cantor. *Ueber eine elementare Frage der Mannigfaltigkeitslehre* Jahresbericht der Deutschen Mathematiker-Vereinigung. 1891
- 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. Vol. 41. No. 6. 1934.
- Alonzo Church. *An unsolvable problem of elementary number theory* Bulletin of the American Mathematical Society 19, May, 1935. American Journal of Mathematics, 58. 1936,
- Alonzo Church: *A Formulation of the Simple Theory of Types*, Journal of Symbolic Logic. vol. 5. 1940.

- Alonzo Church. *Introduction to Mathematical Logic* Princeton University Press. 1956.
- Will Clinger. *Foundations of Actor Semantics* MIT Mathematics Doctoral Dissertation. June 1981.
- Thierry Coquand and Gérard Huet: *The calculus of constructions*. Technical Report 530, INRIA, Centre de Rocquencourt, 1986.
- John Corcoran. *Gaps between logical theory and mathematical practice* in *The methodological unity of science*. 1973.
- John Corcoran. *Categoricity*. *History and Philosophy of Logic*. Vol. 1. 1980
- John Corcoran. *Second-order Logic*. *Logic, Meaning and Computation*. Kluwer. 2001.
- John Dawson. *Shaken Foundations or Groundbreaking Realignment? A Centennial Assessment of Kurt Gödel's Impact on Logic, Mathematics, and Computer Science* FLOC'06.
- Richard Dedekind (1888) "What are and what should the numbers be?" (Translation in *From Kant to Hilbert: A Source Book in the Foundations of Mathematics*. Oxford University Press. 1996) Braunschweig.
- Freeman Dyson. *Heretical Thoughts about Science and Society* Boston University. November 1, 2005.
- Heinz-Dieter Ebbinghaus. *Ernst Zermelo: An Approach to His Life and Work* Springer. 2007.
- Patrik Eklund, M. Angeles Galan, Robert Helgesson, and Jari Kortelainen. *Fuzzy Terms* *Fuzzy Sets and Systems*. 256. 2014.
- Erwin Engeler. *Algebras and Combinators* Algebra Universalis. 1981.
- Solomon Feferman. "Axioms for determinateness and truth" *Review of Symbolic Logic*. 2008.
- Mike Genesereth and Eric Kao. *The Herbrand Manifesto Thinking Inside the Box*. Rule ML. August 2-5, 2015.
- Jean-Yves Girard. *Interprétation fonctionnelle et Élimination des coupure de l'arithmétique d'ordre supérieur* 1972
- 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.
- G.H. Hardy. *A Mathematician's Apology* Cambridge University Press, 1992.
- 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.
- J. Roger Hindley and Jonathan Seldin. *λ -calculus and Combinators: An Introduction* Cambridge University Press. 2008.
- Wilfrid Hodges. *Mathematical Logic* Oxford University Press. 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.
- 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.
- Per Martin-Löf. *An intuitionistic theory of types* in *Twenty-Five Years of Constructive Type Theory*. Oxford University Press. 1998.
- Martin Löb. “Solution of a problem of Leon Henkin.” *Journal of Symbolic Logic*. Vol. 20. 1955.
- Penelope Maddy. *What do we want a foundation to do? Comparing set-theoretic, category-theoretic, and univalent approaches* *Reflections on Foundations: Univalent Foundations, Set Theory and General Thoughts*. 2018.
- 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.
- Ray Monk. *Bourgeois, Boshevist or anarchist? The Reception of Wittgenstein's Philosophy of Mathematics* in Wittgenstein and his interpreters. Blackwell. 2007.
- Nick Nielsen Alternative Foundations/philosophical February 28, 2014.
<http://www.cs.nyu.edu/pipermail/fom/2014-February/017861.html>Bashar
- Francis Pelletier *A Brief History of Natural Deduction* "History and Philosophy of Logic" Vol. 20, Issue. 1, 1999.
- William Quine. *Philosophy of Logic* Prentice Hall. 1970.
- Bertrand Russell. *Letter to Gottlob Frege* 1902.
- 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.
- Arthur Schopenhauer. *Die Welt als Wille und Vorstellung* 1818. English translation in *The World as Will and Representation* Volume I, Falcon's Wing Press. 1958.
- Natarajan Shankar. *Meta-mathematics, Machines, and Gödel's Proof*. Cambridge University Press. 1994.
- Dana Scott. *Setoids/Modest Sets/PERs: Adding and Using Types with a Type-free λ -Calculus* Domains XII. August 2015.
- Stewart Shapiro. *Foundations without Foundationalism: A Case for Second-Order Logic* Oxford University Press. 1991.
- Stephen Simpson *Nonprovability of certain combinatorial properties of finite trees* in *Studies in Logic and the Foundations of Mathematics*. North-Holland. 1985.
- Alfred Tarski *Introduction to Logic* Oxford University Press. 1940 (and many subsequent editions).
- R. Gregory Taylor. *Zermelo's Cantorian Theory of Systems of Infinitely Long Propositions* *Bulletin of Symbolic Logic* December, 2002.
- 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
- Hao Wang *A Logical Journey, From Gödel to Philosophy* MIT Press. 1974.

- Eugene Wigner. *The unreasonable effectiveness of mathematics in the natural sciences* Communications on Pure and Applied Mathematics. Vol. 13. 1960.
- Andrew Wiles. *Modular elliptic curves and Fermat's Last Theorem* Annals of Mathematics. 141. 1995.
- John Woods. *How robust can inconsistency get?* IfCoLoG Journal of Logics and Their Applications. 2014.
- John Woods. *Inconsistency: Its present impacts and future prospects* Inconsistency Robustness 2015.
- Noson Yanofsky. *The Outer Limits of Reason* MIT Press 2013.
- Ernst Zermelo *Über Grenzzahlen und Mengenbereiche: Neue Untersuchungen über die Grundlagen der Mengenlehre* Fundamenta mathematicae. 1930; English translation by Michael Hallett, "On boundary numbers and domains of sets: new investigations in the foundations of set theory" *From Kant to Hilbert: a Source Book in the Foundations of Mathematics*, Oxford University Press, 1996.
- Ernst Zermelo. *Über mathematische System und die Logik des Unendlichen* Forschungen and Fortschritte. Vol. 8. 1932.
- Ernst Zermelo. *Collected Works/Gesammelte Werke: Volume I/Band I - Set Theory, Miscellanea/Mengenlehre, Varia (Schriften der Mathematisch-naturwissenschaftlichen Klasse) (English and German Edition)* Springer. 2010.
- Yoni Zohar. *Reasoning Inside The Box: Gentzen Calculi for Herbrand Logics* Stanford Logic Group. June 7, 2017.

Appendix 1. 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:⁵³

“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. 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.”



Kurt Gödel

According to John von Neumann, Gödel was “the greatest logician since Aristotle.”⁵⁴ However, [von Neumann 1961] expressed a very different mathematical philosophy than Gödel:

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



John von Neumann

[Gödel 1931] based incompleteness results on the thesis that mathematics necessarily has the proposition *I'mUnprovable* in Principia Mathematica [Russell 1902].

Wittgenstein's Paradox

Wittgenstein correctly noted that Gödel's *I'mUnprovable* infers inconsistency in mathematics:⁵⁵

“Let us suppose [Gödel's writings are correct and therefore] I prove⁵⁶ the improbability (in Russell's system) of [Gödel's *I'mUnprovable*] P; [*i.e.*, $\vdash \neg P$ where $P \Leftrightarrow \neg P$] 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 [Monk 2007]:

“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.”

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.

Gödel's maintained:

“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).”⁵⁷



Ludwig Wittgenstein

In the above, Gödel retreated from the [Gödel 1931] results on Principia Mathematica to claiming that the results were for the relational 1st order theory

Relational1stOrderNaturalNumbers in order to defend his *I'mUnprovableInRelational1stOrderNaturalNumbers*. However, the [Gödel 1931] incompleteness result is not very impressive because **Relational1stOrderNaturalNumbers** is a very weak theory.

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 1st order theory 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 GödelNumber[aWellFormedString] 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 GödelNumberToWellFormedString[n] be the well-formed string with Gödel number n, define Eubulides as follows (where “¬GödelNumberToWellFormedString[n]” is the string formed by prefixing the character ¬ to the well-formed string with Gödel number n):

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

By the above Diagonal Lemma, there is a well-formed string *I'mFalse* such that the following is provable in **T** (where “¬GödelNumberToWellFormedString[GödelNumber[I'mFalse]]” is the result of prefixing the well-formed string GödelNumberToWellFormedString[GödelNumber[I'mFalse]] with ¬):⁵⁸

$$\begin{aligned} I'mFalse &\Leftrightarrow \text{Eubulides}[\text{GödelNumber}[I'mFalse]] \\ &\Leftrightarrow \text{“}\neg\text{GödelNumberToWellFormedString}[\text{GödelNumber}[I'mFalse]]\text{”} \\ &\Leftrightarrow \neg I'mFalse \end{aligned}$$

[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.”

[Gödel 1931] results can be formalized as follows using

NotProvable: *Proposition* $\langle n+1 \rangle$ *Proposition* $\langle n \rangle$ where

$$\text{NotProvable}\langle n \rangle[\Psi] \equiv \neg \vdash \Psi$$

The construction of *I'mUnprovable* is blocked because the procedure NotProvable does *not* have a fixed point (by Gödel’s Diagonal Lemma) *I'mUnprovable* such that *I'mUnprovable* $\Leftrightarrow \neg \vdash$ *I'mUnprovable* because the procedure NotProvable maps a proposition Ψ of degree n into a proposition $\neg \vdash \Psi$ of degree $n+1$.

However, Church⁵⁹, Turing, and many other logicians up to the present day continued to endorse Gödel’s argument based on the proposition *I'mUnprovable*.⁶⁰

Although Gödel’s incompleteness results for *I'mUnprovable* have fundamental problems, the work was extremely significant in further the development of the history of metamathematics. For example, the paradoxes of Curry and Löb were developed following along Gödel’s work.

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 relational 1st order theories with propositions from sentence strings coded as integers.

Attempted Theory Nat_1 to Axiomatize Natural Numbers

Nat_1 illustrates failure in attempting to axiomatize the Natural Numbers using a theory whose theorems are computationally enumerable:

- $\vdash_{\text{Nat}_1} 0:N$
- $\vdash_{\text{Nat}_1} \forall[i:N] +_1[i]:N$
- $\vdash_{\text{Nat}_1} \nexists[i:N] +_1[i]=0$
- $\vdash_{\text{Nat}_1} \forall[i, j:N] +_1[i]=+_1[j] \Leftrightarrow i=j$
- $\forall[P:\text{String} \langle \text{Proposition} \langle 1 \rangle \rangle^N]$
 $\vdash_{\text{Nat}_1} (\lfloor P[0] \rfloor \wedge \forall[i:N] \lfloor P[i] \rfloor \Leftrightarrow \lfloor P[i+1] \rfloor) \Leftrightarrow \forall[i:N] \lfloor P[i] \rfloor$
- $\forall[P:\text{String} \langle \text{Proposition} \langle 1 \rangle \rangle^N]$
 $\vdash_{\text{Nat}_1} (\forall[i:N] \vdash_{\text{Nat}_1} \lfloor P[i] \rfloor) \Leftrightarrow \forall[i:N] \lfloor P[i] \rfloor$ ⁶¹

Theorem: Axioms and theorems of Nat_1 are computationally enumerable

Theorem: The 1st order theory Nat_1 is inconsistent

- $\vdash_{\text{Nat}_1} \forall[i:N] \text{Nat}_1\text{ProvablyComputableR}_{[0,1]} \text{Enumerator}_{\bullet}[i]:$
 $\text{Nat}_1\text{ProvablyComputableR}_{[0,1]}$
- Proof:** Theorems of Nat_1 are computationally enumerable.
- $\nvdash_{\text{Nat}_1} \forall[i:N] \text{Nat}_1\text{ProvablyComputableR}_{[0,1]} \text{Enumerator}_{\bullet}[i]:$
 $\text{Nat}_1\text{ProvablyComputableR}_{[0,1]}$

Proof: Suppose to obtain a contradiction that

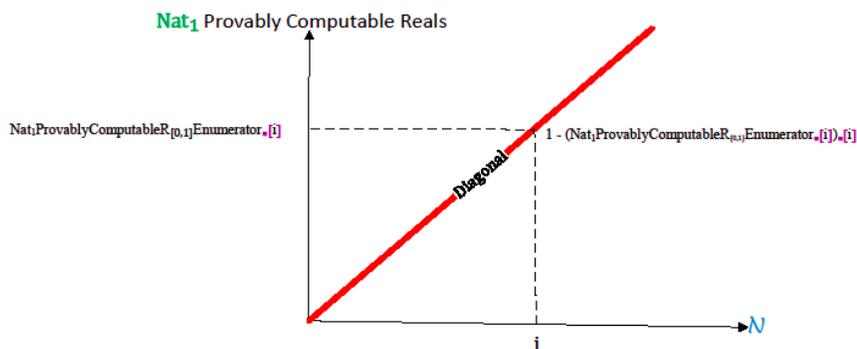
$$\vdash_{\text{Nat}_1} \forall[i:N] \text{Total}[\text{Nat}_1\text{ProvablyComputableR}_{[0,1]} \text{Enumerator}_{\bullet}[i]]$$

Define the Nat_1 provably deterministic total

procedure $\text{Diagonal}_{\bullet}[i] \rightarrow_1 \text{Boolean}$ as follows:

$$\text{Diagonal}_{\bullet}[i] \equiv 1 - (\text{Nat}_1\text{ProvablyComputableR}_{[0,1]} \text{Enumerator}_{\bullet}[i])_{\bullet}[i]$$

\therefore $\text{Diagonal}_{\bullet}:\text{Nat}_1\text{ProvablyComputableR}_{[0,1]}$ which is a contradiction



$$\text{Diagonal}_{[i:N]}:\text{Boolean} \equiv 1 - (\text{Nat1ProvablyComputableR}_{[0,1]}\text{Enumerator}_{[i]})_{[i]}$$

[Church 1934] pointed out that there is no obvious way to remove the inconsistency, which means that that if **Nat₁** is taken to be valid mathematical theory of truths of the natural numbers 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 it implies a formalization. And if there no exact description of logic, then there is no sound basis for supposing that there is such a thing as logic.”



Alonzo Church

After [Church 1934], logicians faced the following dilemma:

- 1st order theories cannot be powerful enough to be a foundation for Computer Science lest they fall into inconsistency because of Church’s Paradox.
- 2nd order theories contravene the philosophical doctrine that theorems must be computationally enumerable.

Attempted theory **Act₁** to Axiomatize Actors

Act₁ illustrates failure in attempting to axiomatize Actors using a theory whose theorems are computationally enumerable.

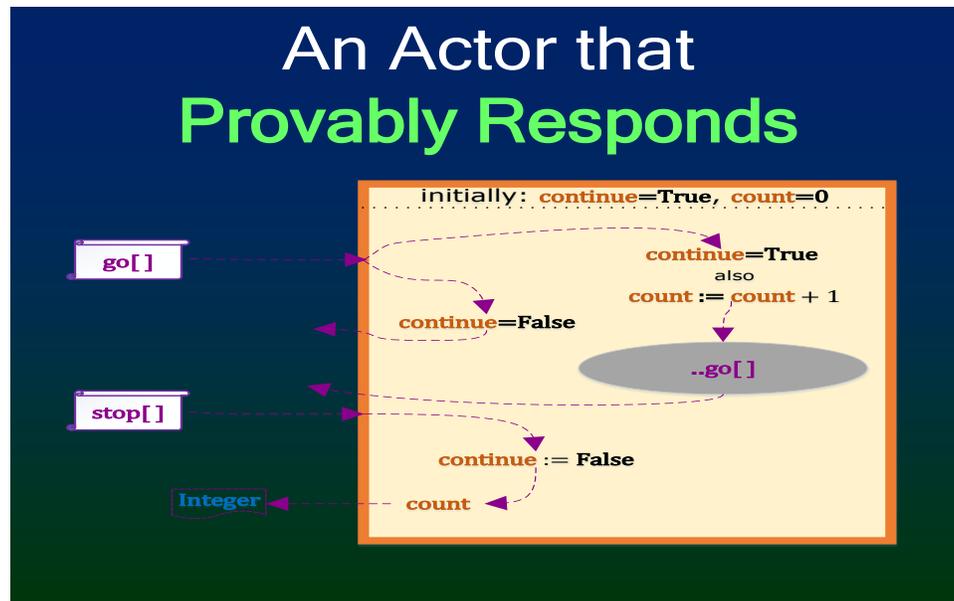
By contrast with the nondeterministic lambda calculus and pure Logic Programs, there is an always-halting Actor Unbounded that when sent a **start[]** message can compute an integer of unbounded size. This is accomplished by creating a counter with the following variables:

- **count** initially **0**
- **continue** initially **True**

and concurrently sending it both a **stop[]** message and a **go[]** message such that:

- When a **go[]** message is received:
 1. if **continue** is **True**, increment **count** by 1 and return the result of sending this counter a **go[]** message.
 2. if **continue** is **False**, return **Void**
- When a **stop[]** message is received, return **count** and set **continue** to **False** for the next message received.

By the Actor Model of Computation, the above Actor will eventually receive the **stop[]** message and return an unbounded number.



The following hold:

- $\forall [t:N] \not\vdash_{Act_1} \text{ResponseBefore}[t]$ // unbounded response time
- $\vdash_{Act_1} \exists [t:N] \text{ResponseBefore}[t]$ // provably responds
- $\models \exists [t:N] \text{ResponseBefore}[t]$

Theorem: Axioms and theorems of Act_1 are computationally enumerable.

Theorem $\not\vdash_{Act_1} \exists [t:N] \text{ResponseBefore}[t]$

Proof: Suppose to obtain a contradiction $\vdash_{Act_1} \exists [t:N] \text{ResponseBefore}[t]$,

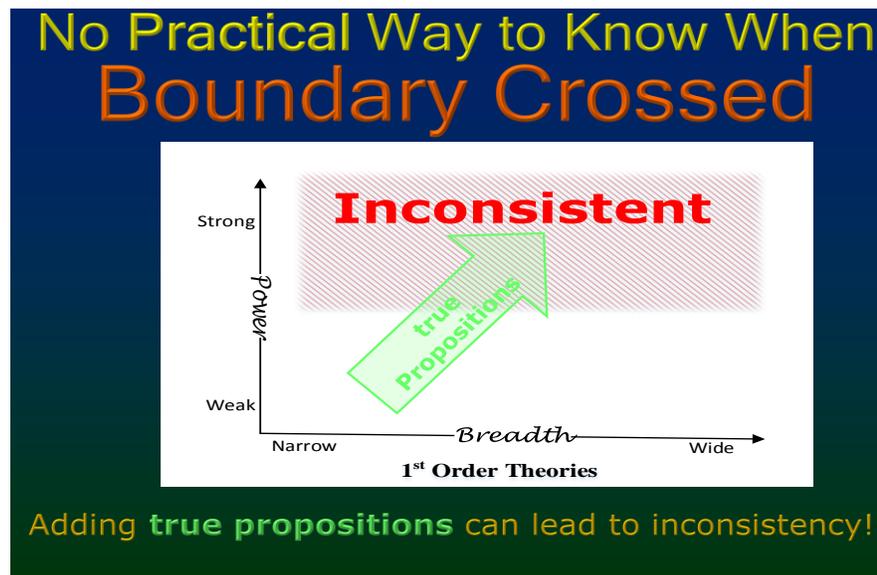
i.e., $\vdash_{Act_1} \neg \forall [t:N] \neg \text{ResponseBefore}[t]$ \therefore

$\neg \text{Consistent}[\{\neg \text{ResponseBefore}[t] \mid t:N\}]$ which by compactness for Nat_t

$\exists [S \subseteq \{\neg \text{ResponseBefore}[t] \mid t:N\} : \text{Finite} \langle \text{Boolean}^N \rangle] \neg \text{Consistent}[S]$

$\therefore \exists [t:N] \neg \text{Consistent}[\{\neg \text{ResponseBefore}[j] \mid j < t\}]$ meaning

$\neg \forall [t:N] \text{Consistent}[\{\neg \text{ResponseBefore}[j] \mid j < t\}]$ which is a contradiction



Since **Nat** is more powerful than **Nat₁**, it must be able to formalize the argument in Church's Paradox. The following section shows why the argument in Church's Paradox fails against **Nat**.

Church's Paradox fails for Higher Order Logic

Mathematics can formalize axioms for **Instance_i** which are strong enough to prove Church's Paradox using **Instance_i** provably computable reals, which can be defined as follows:

$Instance_i Provably Computable \mathcal{R}_{[0,1]} \equiv \mathcal{R}_{[0,1]} \ni \lambda[r] \text{ Computable}[r]: Instance_i$
 where **Instance_i** has axioms given just below:

- $(0:N): Instance_i$
- $(\forall[i:N] +_1[i]:N): Instance_i$
- $(\exists[i:N] +_1[i]=0): Instance_i$
- $(\forall[i,j:N] +_1[i]=+_1[j] \Rightarrow i=j): Instance_i$
- $\forall[P: String \langle Proposition \langle 1 \rangle \rangle^N]$
 $((\lfloor P[0] \rfloor \wedge \forall[i:N] \lfloor P[i] \rfloor \Rightarrow \lfloor P[i+1] \rfloor) \Rightarrow \forall[i:N] \lfloor P[i] \rfloor): Instance_i$
- $(Instance_i Provably Computable \mathcal{R}_{[0,1]} \text{ Enumerator}$
 $: Total \langle [N] \rightarrow_1 Instance_i Provably Computable \mathcal{R}_{[0,1]} \rangle): Instance_i$
- $(\forall[r: Instance_i Provably Computable \mathcal{R}_{[0,1]}]$
 $\exists[i:N] r = Instance_i Provably Computable \mathcal{R}_{[0,1]} \text{ Enumerator}_{\bullet}[i]): Instance_i$
- $(\forall[i:N] Instance_i Provably Computable \mathcal{R}_{[0,1]} \text{ Enumerator}_{\bullet}[i]$
 $: Instance_i Provably Computable \mathcal{R}_{[0,1]}): Instance_i$

Theorem. Axioms for *Instance₁* are inconsistent

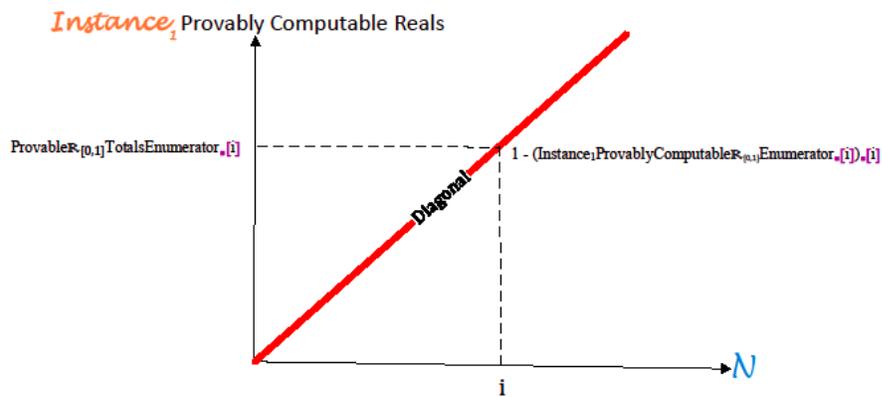
Define the **Nat** provably total deterministic procedure

Diagonal: $[i: \mathbb{N}] \rightarrow_1 \text{Boolean}$ as follows:

$$\text{Diagonal}_\bullet[i] \equiv 1 - (\text{Instance}_1 \text{Provably Computable } \mathbb{R}_{[0,1]} \text{ Enumerator}_\bullet[i])_\bullet[i]$$

Diagonal is not in the range of $\text{Instance}_1 \text{Provably Computable } \mathbb{R}_{[0,1]} \text{ Enumerator}_\bullet$, which is a contradiction because

$$\text{Instance}_1 \vdash_{\text{Nat}} \text{Diagonal: Instance}_1 \text{Provably Computable } \mathbb{R}_{[0,1]}$$



$$\text{Diagonal}_\bullet[i: \mathbb{N}]: \text{Boolean} \equiv 1 - (\text{Instance}_1 \text{Provably Computable } \mathbb{R}_{[0,1]} \text{ Enumerator}_\bullet[i])_\bullet[i]$$

Axioms for *Instance₁* do **not** stand as legitimate Mathematics because the axioms are “self-referential.” Therefore, it makes sense to use Inconsistency Robust logic for *Instance₁* instead of classical logic.

Discussion

Church's Paradox and other paradoxes raise a number of issues that can be addressed by requiring mathematics to be strongly typed and using higher order logic as follows:

1. Requiring Mathematics to be strongly typed using so that
 - Mathematics self proves that it is "open" in the sense that theorems are not computationally enumerable.⁶²
 - Mathematics self proves that it is *formally* consistent.⁶³
 - Strong mathematical theories for Natural Numbers, Ordinals, Set Theory, the Lambda Calculus, Actors, etc. are inferentially decidable, meaning that every true proposition is provable and every proposition is either provable or disprovable. Furthermore, theorems of these theories are not enumerable by a provably total procedure.
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 general 1st order non-categorical induction *schema* is as follows for each $P: \text{String} \langle \text{Proposition} \langle 1 \rangle \rangle^N$:

$$(\lfloor P[0] \rfloor \wedge \forall [i:N] \lfloor P[i] \rfloor \Rightarrow \lfloor P[i+1] \rfloor) \Rightarrow \forall [i:N] \lfloor P[i] \rfloor$$

which has countably many 1st order propositions as instances that are abstracted from the countably many character strings of type $\text{String} \langle \text{Proposition} \langle 1 \rangle \rangle$ and which differs fundamentally from the character string for the *more general* 2nd order categorical induction *axiom*, which is as follows:⁶⁴

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

Although the theory **Nat** has only *finitely* many axioms, the above string abstracted as a proposition has *uncountably* many 1st order propositions as instances.⁶⁵ In this way, **Nat** differs fundamentally from a 1st order theory 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 the natural numbers 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?



Ernst Zermelo

Zermelo suggested that the [Godel 1931] incompleteness results relied on an overly restrictive conception of quantification and proof. He took it as crucial that true proposition of set theory must be “provable” and that the validity of a proof can reasonably be decided. To this end, he developed an infinitary logic but did not carry it very far. In 1935 he was dismissed for failing to salute Hitler and his proposal was not pursued by anyone else until, perhaps, the revival of infinitary languages in the 1950s. [Shapiro 1991]

Additional limitations of Relational 1st order theories

“[F]rom the 1950s onward, classical mathematics had just one deductive system, namely, first-order Zermelo-Fraenkel Set Theory with [the Axiom of] Choice ...”

[Hodges 2007]

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

[Hobbes *Leviathan*, Chapter 4]⁶⁶

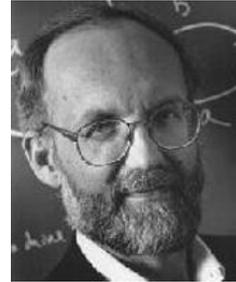
A relational 1st order theory is very weak. For example, a relational 1st order theory is incapable of characterizing even the natural numbers, *i.e.*, there are infinite integers in models of every relational 1st order axiomatization of the natural numbers. Furthermore, there are infinitesimal real numbers in models of every relational 1st order axiomatization of the real numbers.⁶⁷ Of course, infinite integers and infinitesimal reals are monsters that must be banned from the mathematical foundations of Computer Science.

However, some theoreticians have found relational 1st order theory to be useful for their careers because it is weak enough that they can prove theorems about relational 1st order axiomatizations whereas they cannot prove such theorems about stronger practical systems, *e.g.*, Classical Direct Logic.⁶⁸

Zermelo considered the 1st Order Thesis to be a mathematical “hoax” because it necessarily allowed unintended models of axioms.⁶⁹

[Barwise 1985] critiqued the 1st Order Thesis that mathematical foundations should be restricted to 1st order theories as follows:

*The reasons for the widespread, often uncritical acceptance of the first-order thesis are numerous. The first-order thesis ... confuses the subject matter of logic with one of its tools. First-order language is just an artificial language structured to help investigate logic, much as a telescope is a tool constructed to help study heavenly bodies. From the perspective of the mathematics in the street, the first-order thesis is like the claim that astronomy is the study of the telescope.*⁷⁰



Jon Barwise

Computer Science is making increasing use of Model Analysis⁷¹ in the sense of analyzing relationships among the following:

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

In Computer Science, it is important that the natural numbers be axiomatized in a way that does not allow non-numbers (e.g. infinite ones) in models of the axioms.

Theorem: If \mathcal{N} is a model of a 1st order axiomatization \mathbf{T} , then \mathbf{T} has a model \mathcal{M} with an infinite integer.

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

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

It can be shown that \mathcal{M} is a model of \mathbf{T} with an infinite integer ∞ .

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

Theorem: If \mathcal{R} is a model of a 1st order axiomatization \mathbf{T} , then \mathbf{T} has a model \mathcal{M} with an infinitesimal.

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

$$\{\neg \infty < \infty\} \cup \{m < \infty \mid m: \mathcal{R}\}$$

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

On the other hand, since it is not limited to 1st order propositions, Classical Direct Logic characterizes structures such as natural numbers and real numbers up to isomorphism.⁷²

There are many theorems of **Nat** that cannot be proved from 1st order axioms [Goodstein 1944, Simpson 1985, Wiles 1995, Bovykin 2009, McLarty 2010].

Unbounded Nondeterminism

Of greater practical import, 1st order theory is *not* a suitable foundation for the Internet of Things in which specifications require a device respond to a request.⁷³

The specification that a computer responds can be formalized as follows:

$\exists[i:\mathcal{N}] \text{ResponseBefore}[i]$. However, the specification cannot be proved in a 1st order theory.

Proof: In order to obtain a contradiction, suppose that it is possible to prove in a 1st order theory $\exists[i:\mathcal{N}] \text{ResponseBefore}[i]$. Therefore the infinite set of propositions $\{\neg \text{ResponseBefore}[i] \mid i:\mathcal{N}\}$ is inconsistent. By the compactness theorem of 1st order theory, it follows that there is finite subset of the set of propositions that is inconsistent. But this is a contradiction, because all the finite subsets are consistent since the amount of time before a server responds is unbounded, that is,

$\nexists[i:\mathcal{N}] \vdash \text{ResponseBefore}[i]$.

However, the above specification axiom does *not* compute any actual output! Instead the above axiom simply asserts the *existence* of unbounded outputs for **Unbounded**. \blacksquare

Theorem. The nondeterministic function defined by Unbounded (earlier in this article) cannot be implemented by a nondeterministic Logic Program⁷⁴ or a nondeterministic Turing Machine:

*Proof.*⁷⁵

The task of a nondeterministic Logic Program P is to start with an initial set of axioms and prove $\text{Output}=n$ for some numeral n . Now the set of proofs of P starting from initial axioms will form a tree. The branching points will correspond to the nondeterministic choice points in the program and the choices as to which rules of inference to apply. Since there are always only finitely many alternatives at each choice point, the branching factor of the tree is always finite. Now König's lemma says that if every branch of a finitary tree is finite, then so is the tree itself. In the present case this means that if every proof of P proves $\text{Output}=n$ for some numeral n , then there are only finitely many proofs. So if P nondeterministically proves $\text{Output}=n$ for every numeral n , it must contain a nonterminating computation in which it does not prove $\text{Output}=n$ for some numeral n .

The following arguments support unbounded nondeterminism in the Actor model [Hewitt 1985]:

- There is no bound that can be placed on how long it takes a computational circuit called an *arbiter* to settle. Arbiters are used in computers to deal with the circumstance that computer clocks operate asynchronously with input from outside, *e.g.*, keyboard input, disk access, network input, *etc.* So it could take an unbounded time for a message sent to a computer to be received and in the meantime the computer could traverse an unbounded number of states.
- Electronic mail enables unbounded nondeterminism since mail can be stored on servers indefinitely before being delivered.
- Communication links to servers on the Internet can be out of service indefinitely

1st order theory is **not** a suitable mathematical foundation for Intelligent Applications for the Internet of Things.

As a foundation of mathematics for Computer Science, Classical Direct Logic provides categorical⁷⁶ numbers (integer and real), sets, lists, trees, graphs, *etc.* which can be used in arbitrary mathematical theories including theories for

categories, large cardinals, etc. These various theories might have “monsters” of various kinds. However, these monsters should not be imported into models of computation used in Computer Science.

Computer Science needs *stronger* systems than provided by 1st order theory in order to weed out unwanted models. In this regard, Computer Science doesn't have a problem computing with “infinite” objects (*i.e.* Actors) such as π and uncountable sets such as the set of real numbers $\text{Set}\langle\mathcal{R}\rangle$. However, the mathematical foundation of Computer Science is very different from the general philosophy of mathematics in which the infinite integers and infinitesimal reals allowed by models of 1st order theories may be of some interest. Of course, it is always possible to have special theories that are *not* part of the foundations with infinite integers, infinitesimal reals, unicorns, *etc.*⁷⁷

Of course some problems are theoretically not computable. However, even in these cases, it is often possible to compute approximations and cases of practical interest.⁷⁸

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

Appendix 2. Appendix 1. More Categorical Mathematical Theories

Theory of Nondeterministic Lambda Calculus (\mathbf{Lam}_\diamond)

Definition: $\mathit{Functional}\langle t_1, t_2 \rangle \equiv [([t_1] \rightarrow t_2)] \rightarrow ([t_1] \rightarrow t_2)$

Theory \mathbf{Lam}_t

In addition to Lambda Induction (above), the theory \mathbf{Lam}_t has the following axioms:⁷⁹

- **Base**
 $\forall [x:t] x:\Lambda\langle t \rangle$
- **Composition**
 $\forall [x:t_1, f:([t_1] \rightarrow t_2)] f \bullet [x]:t_2$
 $\forall [x_1:t_1, x_2:t_2] [x_1, x_2]:[t_1, t_2]$
 $\forall [x_1, x_2:\Lambda\langle t \rangle] [x_1, x_2]:\Lambda\langle t \rangle$
- **Identity** $\langle t \rangle:([\Lambda\langle t \rangle] \rightarrow \Lambda\langle t \rangle)$
 $\mathit{Identity}\langle t \rangle \bullet [x] = x$
- **1st** $\langle t_1, t_2 \rangle:([([t_1, t_2]) \rightarrow t_1])$
 $\mathit{1st}\langle t_1, t_2 \rangle \bullet [x_1, x_2] = x_1$
- **2nd** $\langle t_1, t_2 \rangle:([([t_1, t_2]) \rightarrow t_2)$
 $\mathit{2nd}\langle t_1, t_2 \rangle \bullet [x_1, x_2] = x_2$
- **Fix** $\langle t_1, t_2 \rangle:([\mathit{Functional}\langle t_1, t_2 \rangle] \rightarrow \mathit{Functional}\langle t_1, t_2 \rangle)$ ⁸⁰
 $\mathit{Fix}\langle t_1, t_2 \rangle \bullet [F] = F \bullet [\mathit{Fix}\langle t_1, t_2 \rangle \bullet [F]]$
- **Either** $\langle t \rangle:([\Lambda\langle t \rangle, \Lambda\langle t \rangle] \rightarrow \Lambda\langle t \rangle)$
 $\mathit{Either}\langle t \rangle \bullet [x_1, x_2] = x_1 \vee \mathit{Either}\langle t \rangle \bullet [x_1, x_2] = x_2$
- **Equality**
 $\forall [f_1, f_2:([t_1] \rightarrow t_2)] f_1 = f_2 \Leftrightarrow \forall [x:t_1] f_1 \bullet [x] = f_2 \bullet [x]$
- **Lambda Equality**⁸¹
 $\forall [f:([t_1] \rightarrow t_2)] f = [x:t_1] \rightsquigarrow f \bullet [x]$

Theorem. Computational Inadequacy of Nondeterministic Lambda Calculus.
The nondeterministic lambda calculus is inadequate to implement all computable procedures.

Proof. $F_i[j:\mathbf{N}] \equiv j > i \text{ ? } \mathbf{True} \text{ then } \text{InfiniteLoop} \bullet [] \text{ False then } i \text{ either } F_i[j+1]$
For each $i:\mathbf{N}$, F_i is a nondeterministic λ expression but $(\text{limit}_{i:\mathbf{N}} F_i)$ **cannot** be implemented as a nondeterministic λ expression. However $(\text{limit}_{i:\mathbf{N}} F_i):\text{Actor}$

Lambda Induction

The theorem of Lambda Induction is as follows:⁸²

$$\begin{aligned} & \forall [P:\text{Proposition} \langle 1 \triangleright \Lambda \langle t \triangleright \rangle] \\ & (P[\text{Identity} \langle t \triangleright \rangle] \wedge P[\text{1st} \langle t_1, t_2 \triangleright \rangle] \wedge P[\text{2nd} \langle t_1, t_2 \triangleright \rangle] \\ & \wedge P[\text{Fix} \langle t_1, t_2 \triangleright \rangle] \wedge P[\text{Either} \langle t \triangleright \rangle] \\ & \wedge \forall [x:t] P[x] \\ & \wedge \forall [x_1:t_1, x_2:t_2] P[x_1] \wedge P[x_2] \Rightarrow P[[x_1, x_2]]) \\ & \wedge \forall [F:\text{Functional} \langle t_1, t_2 \triangleright \rangle] P[F] \Rightarrow P[\text{Fix} \langle t_1, t_2 \triangleright \bullet [F]]) \\ & \wedge \forall [x:t_1, f:(t_1 \rightarrow t_2)] P[x] \wedge P[f] \Rightarrow P[f \bullet [x]]) \\ & \Rightarrow \forall [x:\Lambda \langle t \triangleright \rangle] P[x] \end{aligned}$$

Convergence: $\forall [f:(t_1 \rightarrow t_2), x:t_1] f \bullet [x] \downarrow \Leftrightarrow \exists [y:t_2] f \bullet [x] = y$

Approximation: $\forall [f_1, f_2:(t_1 \rightarrow_1 t_2)] f_1 \leq f_2 \Leftrightarrow \forall [x:t_1] f_1 \bullet [x] \downarrow \Rightarrow f_1 \bullet [x] = f_2 \bullet [x]$

Bottom: $\forall [f:t] \neg (\perp \langle t \triangleright \bullet [f]) \downarrow$

Note: $\forall [f:(t \rightarrow t)] \perp \langle t \triangleright \leq f$

Monotone:

$$F:\text{Monotone} \langle t_1, t_2 \triangleright \Leftrightarrow F:\text{Functional} \langle t_1, t_2 \triangleright \wedge \forall [g:(t_1 \rightarrow t_2)] g \leq F \bullet [g]$$

Limit Theorem: $\forall [F:\text{Monotone} \langle t \triangleright \rangle] F = \text{limit}_{i:\mathbf{N}_+} F^i \bullet [\perp \langle t \triangleright]$ ⁸³

Theorem: Some nondeterministic procedures have unbounded nondeterminism⁸⁴

$$\exists [f:(t \rightarrow t)] \neg f:\Lambda \langle t \triangleright$$

Theorem.⁸⁵ \mathbf{Lam}_t is categorical with a unique isomorphism.

Proof: Suppose that \mathcal{X} satisfies the axioms for \mathbf{Lam}_t .

By lambda induction, the isomorphism $I: \mathcal{X}^{\Lambda\langle t \rangle}$ is defined as follows:⁸⁶

- $I[\mathbf{Identity}\langle t \rangle] \equiv \mathbf{Identity}_{\mathcal{X}\langle t \rangle}$
- $I[\mathbf{1st}\langle t_1, t_2 \rangle] \equiv \mathbf{1st}_{\mathcal{X}\langle t_1, t_2 \rangle}$
- $I[\mathbf{2nd}\langle t_1, t_2 \rangle] \equiv \mathbf{2nd}_{\mathcal{X}\langle t_1, t_2 \rangle}$
- $I[\mathbf{Fix}\langle t_1, t_2 \rangle] \equiv \mathbf{Fix}_{\mathcal{X}\langle t_1, t_2 \rangle}$
- $I[\mathbf{Either}\langle t \rangle] \equiv \mathbf{Either}_{\mathcal{X}\langle t \rangle}$
- $I[[x_1, x_2]] \equiv [I[x_1], I[x_2]]_{\mathcal{X}}$
- $\forall [x: t_1, f: (t_1 \rightarrow t_2)] I[f \cdot x] \equiv I[f] \cdot_{\mathcal{X}} [I[x]]$

I is the unique isomorphism:

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

Theorem (Model Soundness of \mathbf{Lam}_t): $(\vdash_{\mathbf{Lam}_t} \Psi) \Leftrightarrow \models \Psi$

Proof: Suppose $\vdash_{\mathbf{Lam}_t} \Psi$. The theorem immediately follows because the axioms for the theory \mathbf{Lam}_t hold in the type $\Lambda\langle t \rangle$.

Theorem (Indiscernibility for \mathbf{Lam}_t):⁸⁷

$$\forall [f, g: \Lambda\langle t \rangle] f=g \Leftrightarrow \forall [P: \mathbf{Proposition}^{\Lambda\langle t \rangle}] P[f] \Leftrightarrow P[g]$$

Theorem: Inferential completeness of \mathbf{Lam}_t

$$\forall [\Psi: \mathbf{Proposition}\langle \mathbf{Lam}_t \rangle] (\models \Psi) \Rightarrow \vdash_{\mathbf{Lam}_t} \Psi$$

Corollary. Equivalence of satisfiability and provability in \mathbf{Lam}_t , i.e.,

$$\forall [\Psi: \mathbf{Proposition}\langle \mathbf{Lam}_t \rangle] (\models \Psi) \Leftrightarrow \vdash_{\mathbf{Lam}_t} \Psi$$

Theorem. Inferential Decidability of \mathbf{Lam}_t , i.e.,

$$\forall [P: \textit{Proposition} \langle \mathbf{Lam}_t \rangle^{\Lambda \langle \Phi \rangle}] \\ (\vdash_{\mathbf{Lam}_t} \forall [f: \Lambda \langle t \rangle] P[f]) \vee \vdash_{\mathbf{Lam}_t} \exists [f: \Lambda \langle t \rangle] \neg P[f]$$

Weakest Preconditions

$$\text{WeakestPrecondition}[\Phi, f] \equiv \lambda [i: \mathbb{N}] \Phi[f.[i]]$$

Theorem Weakest Preconditions are monotonic in both arguments, i.e.,⁸⁸

- $\forall [\Phi_1, \Phi_2: \textit{Proposition}^{\mathbb{N}}; f: ([\mathbb{N}] \rightarrow \mathbb{N})]$
 $(\Phi_1 \Leftrightarrow \Phi_2) \Leftrightarrow (\text{WeakestPrecondition}[\Phi_1, f] \Leftrightarrow \text{WeakestPrecondition}[\Phi_2, f])$
- $\forall [\Phi: \textit{Proposition}^{\mathbb{N}}; f_1, f_2: ([\mathbb{N}] \rightarrow \mathbb{N})]$
 $f_1 \sqsupseteq f_2 \Leftrightarrow (\text{WeakestPrecondition}[\Phi, f_1] \Leftrightarrow \text{WeakestPrecondition}[\Phi, f_2])$

Theory of Reals (*Reals*)

Reals is strictly more powerful than the relational 1st order theory of *RealClosedFields*.⁸⁹

Theorem (Categoricity of *Reals*):⁹⁰

If \mathbf{X} is a type satisfying the axioms⁹¹ for the real numbers *Reals*, then there is a unique isomorphism of \mathbf{X} with \mathcal{R} .

Theory of Ordinals (Ord)

A theory of the ordinals can be axiomatized⁹² using a 2nd order ordinal induction axiom as follows: For each order: \mathcal{N}_* and P : *Proposition*,

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

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

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

In order to guarantee that there are enough ordinals, the following axioms are also included in **Ord**:

- $\forall[\alpha: \mathcal{O}] \omega_{\alpha}: \mathcal{O}$
- Definition by ordinal induction:
 1. $\omega_0 = \mathcal{N}$
 2. $\omega_{\alpha+1} = \text{Boolean}^{\{\beta: \mathcal{O} \mid \beta < \omega_{\alpha}\}}$
 3. $\omega_{\alpha} = \cup_{\alpha} (\lambda[\beta: \mathcal{O}] \omega_{\beta})$

Theorem Ordinals have the following properties:

- Ordinals are well-ordered:
 - Least: $\mathcal{O} \uparrow (\text{Boolean}^{\mathcal{O}})$
 - Least[{}] = $0_{\mathcal{O}}$
 - $\forall[S: \text{Boolean}^{\mathcal{O}}] S \neq \{\} \Rightarrow \text{Least}[S] \in S$
 - $\forall[S: \text{Boolean}^{\mathcal{O}}] S \neq \{\} \Rightarrow \forall[\alpha \in S] \text{Least}[S] \leq \alpha$
- Reals can be well-ordered because $\omega_1 = \mathcal{R}$

Proof Checkers in Ord

$$\forall[\Psi_1, \Psi_2: \text{Proposition} \langle \text{Ord} \rangle]$$

$$(\Psi_1 \vdash_{\text{Ord}} \Psi_2) \Leftrightarrow \exists[p: \text{ProofChecker} \langle \text{Ord} \rangle] \Psi_1 \vdash_{\text{Ord}}^p \Psi_2$$

Proof checking in **Ord** is **computationally decidable**.

$$\forall[\Psi_1, \Psi_2: \text{Proposition} \langle \text{Ord} \rangle, p: \text{ProofChecker} \langle \text{Ord} \rangle]$$

$$(\Psi_1 \vdash_{\text{Ord}}^p \Psi_2) \Leftrightarrow p \cdot [\Psi_1, \Psi_2] = \text{True}$$

Example of a proof checker with a nondenumerable domain:

The $+_1$ of Ord checker below has a nondenumerable domain (including $\text{Proposition} \langle \text{Nat} \rangle$ and \mathcal{O}) even though $+_1$ of Ord is an effectively computable procedure.

$$+_1 \text{ of Ord}: \langle [\text{Proposition} \langle \text{Ord} \rangle, \text{Proposition} \langle \text{Ord} \rangle] \rightarrow_1 \text{Boolean}$$

$$+_1 \text{ of Ord} \cdot [\Psi_1, \Psi_2] \equiv \Psi_1[?] (\alpha: \mathcal{O}) \text{ then } s_2 = (\alpha + 1: \mathcal{O}), \text{ else False}$$

// If s_1 is $\alpha: \mathcal{O}$, then $s_2 = (\alpha + 1: \mathcal{O})$, otherwise False

$$+_1 \text{ of Ord}: \text{ProofChecker} \langle \text{Ord} \rangle$$

$$\forall[\alpha: \mathcal{O}] (\alpha: \mathcal{O}) \vdash_{\text{Ord}}^{+_1 \text{ of Ord}} (\alpha + 1: \mathcal{O})$$

Theorem (Categoricity of Ord):

If \mathbf{X} be a type satisfying the axioms the theory of the ordinals **Ord**, then there is a unique isomorphism with \mathcal{O} .⁹³

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

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

Theorem (Indiscernibility for Ord):⁹⁴

$$\forall[\alpha, \beta: \mathcal{O}] \alpha = \beta \Leftrightarrow \forall[P: \text{Proposition} \langle \text{Ord} \rangle^{\mathcal{O}}] P[\alpha] \Leftrightarrow P[\beta]$$

Theorem: Inferential completeness of Ord

$$\forall[\Psi: \text{Proposition} \langle \text{Ord} \rangle] (\models \Psi) \Rightarrow \vdash_{\text{Ord}} \Psi$$

Corollary. Equivalence of satisfiability and provability in **Ord**, i.e.,

$$\forall[\Psi: \text{Proposition} \langle \text{Ord} \rangle] (\models \Psi) \Leftrightarrow \vdash_{\text{Ord}} \Psi$$

Theorem. Inferential Decidability of **Ord**, i.e.,

$$\forall[\Psi: \text{Proposition} \langle \text{Ord} \rangle^{\circ}] (\vdash_{\text{Ord}} \Psi) \vee \vdash_{\text{Ord}} \Psi$$

Proof. $\forall[\Psi: \text{Proposition} \langle \text{Ord} \rangle] (\models \Psi) \vee \models \Psi$

Theorem follows from Equivalence of satisfiability and provability in **Ord**.

Type Choice

$$\forall[f: (\text{Boolean}^{t_1})^{t_2}] \exists[\text{choice}: t_1^{t_2}] \forall[x: t_1] f[x] \neq \{\} \Rightarrow \text{choice}[x] \in f[x]$$

Theory **Sets_t** defined using strong parameterized types

Set Theory

A theory of the ordinals can be axiomatized using a 2nd order set induction axiom

as follows: For each order: N_+ and $P: \text{Proposition} \langle \text{order} \rangle^{\circ}$:

$$\begin{aligned} (\forall[S: \text{Set} \langle t \rangle, \alpha: \circ] (S \doteq \alpha \Rightarrow \forall[X: \text{Set} \langle t \rangle, \beta < \alpha: \circ] P[X] \wedge X \doteq \beta \Rightarrow P[X]) \\ \Rightarrow \forall[S: \text{Set} \langle t \rangle] P[S]) \end{aligned}$$

The type $\text{Set} \langle t \rangle$ can be characterized as follows:

$$\text{Set} \langle t \rangle \equiv \text{Boolean}^t$$

Of course set membership is defined as follows:

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

Inductive definition:

1. $\text{Sets}_0 \langle t \rangle \equiv \text{Boolean}^t$
 2. $\text{Sets}_{\alpha+1} \langle t \rangle \equiv \text{Set} \langle \text{Sets}_{\alpha} \langle t \rangle \rangle$
 3. $\alpha: \text{Limit} \langle \circ \rangle \Rightarrow (S: \text{Sets}_{\alpha} \langle t \rangle \Leftrightarrow \forall[X \in S] \exists[\beta < \alpha: \circ, Y: \text{Sets}_{\beta} \langle t \rangle] X \in Y)$
- $S: \text{Sets} \langle t \rangle \Leftrightarrow \exists[\alpha: \circ] S: \text{Sets}_{\alpha} \langle t \rangle$

The properties below mean that $\text{Sets}_{\langle t \rangle}$ is a "universe" of mathematical discourse.⁹⁵

- Foundation: There are no downward infinite membership chains.⁹⁶
- Transitivity of \in :⁹⁷ $\forall [S: \text{Sets}_{\langle t \rangle}] \forall [X \in S] X: \text{Sets}_{\langle t \rangle}$
- Powerset:⁹⁸ $\forall [S: \text{Sets}_{\langle t \rangle}] \text{Boolean}^S: \text{Sets}_{\langle t \rangle}$
- Union:⁹⁹

$$\forall [S: \text{Sets}_{\langle t \rangle}] \text{US}: \text{Sets}_{\langle t \rangle}$$

$$\forall [S: \text{Sets}_{\langle t \rangle}] \forall [X: \text{Sets}_{\langle t \rangle}] X \in \text{US} \Leftrightarrow \exists [Y \in S] X \in Y$$
- Replacement:¹⁰⁰ The function image of any set is also a set, *i.e.*:
$$\forall [f: \text{Sets}_{\langle t \rangle}^{\text{Sets}_{\langle t \rangle}}, S: \text{Sets}_{\langle t \rangle}]$$

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

Sets_t is much stronger than relational 1st order ZFC.¹⁰¹

Theorem. Theory \mathbf{Sets}_t is categorical for $\mathbf{Sets}\langle t \rangle$ with a unique isomorphism.

Proof:¹⁰² Suppose that \mathcal{X} satisfies the axioms for \mathbf{Sets}_t

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

1. $S: \mathbf{Sets}_{0\langle t \rangle}$
 $I[S] \equiv S$
2. $S: \mathbf{Sets}_{\alpha+1\langle t \rangle}$
 $Z \in_{\mathcal{X}} I[S] \Leftrightarrow \exists [Y: \mathbf{Sets}_{\alpha\langle t \rangle}] I[Y] \in_{\mathcal{X}} Z$
3. $S: \mathbf{Sets}_{\alpha\langle t \rangle}$ and $\alpha: \mathbf{Limit}\langle \mathcal{O} \rangle$
 $Z \in_{\mathcal{X}} I[S] \Leftrightarrow \exists [\beta < \alpha: \mathcal{O}, Y: \mathbf{Sets}_{\beta\langle t \rangle}] I[Y] \in_{\mathcal{X}} Z$
4. I is a unique isomorphism:
 - I is one to one
 - The range of I is \mathcal{X}
 - I is a homomorphism:
 - $I[\{ \}_{{\mathbf{Sets}\langle t \rangle}}] = \{ \}_{\mathcal{X}}$
 - $\forall [S1, S2: \mathbf{Sets}\langle t \rangle] I[S1 \cup S2] = I[S1] \cup_{\mathcal{X}} I[S2]$
 - $\forall [S1, S2: \mathbf{Sets}\langle t \rangle] I[S1 \cap S2] = I[S1] \cap_{\mathcal{X}} I[S2]$
 - $\forall [S1, S2: \mathbf{Sets}\langle t \rangle] I[S1 - S2] = I[S1] -_{\mathcal{X}} I[S2]$
 - $\forall [S: \mathbf{Sets}\langle t \rangle] I[US] = \cup_{\mathcal{X}} \{ I[x] \mid x \in S \}$
 - $I^{-1}: \mathbf{Sets}\langle t \rangle^{\mathcal{X}}$ is a homomorphism
 - I is the unique isomorphism: If $g: \mathcal{X}^{\mathbf{Sets}\langle t \rangle}$ is an isomorphism, then $g = I$

Theorem (Model Soundness of \mathbf{Sets}_t): $(\vdash_{\mathbf{Sets}_t} \Psi) \Leftrightarrow \models \Psi$

Proof: Suppose $\vdash_{\mathbf{Sets}_t} \Psi$. The theorem immediately follows because the axioms for the theory \mathbf{Sets}_t hold in the type $\mathbf{Sets}\langle t \rangle$.

Theorem: Indiscernibility for \mathbf{Sets}_t ¹⁰³

$$\forall [S_1, S_2: \mathbf{Sets}_{\langle t \rangle}] S_1 = S_2 \Leftrightarrow \forall [P: \mathbf{Proposition}_{\langle \mathbf{Sets}_t \rangle}^{\mathbf{Sets}_{\langle t \rangle}}] P[S_1] \Leftrightarrow P[S_2]$$

Theorem: Inferential completeness of \mathbf{Sets}_t

$$\forall [\Psi: \mathbf{Proposition}_{\langle \mathbf{Sets}_t \rangle}^{\mathbf{Sets}_{\langle t \rangle}}] (\models \Psi) \Rightarrow \vdash_{\mathbf{Sets}_t} \Psi$$

Corollary. Equivalence of satisfiability and provability in \mathbf{Sets}_t i.e.,

$$\forall [\Psi: \mathbf{Proposition}_{\langle \mathbf{Sets}_t \rangle}] (\models \Psi) \Leftrightarrow \vdash_{\mathbf{Sets}_t} \Psi$$

Theorem. Inferential Decidability of \mathbf{Sets}_t i.e.,

$$\forall [\Psi: \mathbf{Proposition}_{\langle \mathbf{Sets}_t \rangle}] (\vdash_{\mathbf{Sets}_t} \Psi) \vee (\vdash_{\mathbf{Sets}_t} \neg \Psi)$$

Appendix 3: Blocking Known Paradoxes

Strong types block all known paradoxes including the following:

- Liar [Eubulides of Miletus] is an example of using untyped propositions to derive an inconsistency:

$$F\langle n \rangle \equiv \lambda[p:\text{Proposition}\langle n \rangle] \neg p$$

// above definition has no fixed point because $\neg p$ has

// order one greater than the order of 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)

- Burali-Forti [Burali-Forti 1897] The set of all ordinals Ω is *Boolean*^o so that: $\forall[\alpha:\text{O}] \alpha \in \Omega \Leftrightarrow \alpha:\text{O}$. Note that it is **not** the case that Ω is of type O , thereby blocking the paradox.

- Russell [Russell 1902]. Transformed to types, Russell's paradoxical (*Sets* $\langle N \rangle \ni \lambda[s:\text{Sets}\langle N \rangle] s \notin s = \text{Sets}\langle N \rangle$ because no set is an element of itself.

- Curry [Curry 1941] Suppose $\Psi:\text{Proposition}\langle \text{anOrder} \rangle$ and

$$\text{Curry}\langle n \rangle:\text{Proposition}\langle \text{Max}[n+1, \text{anOrder}+1] \rangle \text{Proposition}\langle n \rangle \text{ where}$$

$$\text{Curry}\langle n \rangle \equiv \lambda[p:\text{Proposition}\langle n \rangle] p \Rightarrow \Psi$$

Curry's Paradox is blocked because the procedure Curry does **not** have a fixed point.

- Löb [Löb 1955] Suppose $\Psi:\text{Proposition}\langle \text{anOrder} \rangle$.

$$\text{Löb}\langle n \rangle:\text{Proposition}\langle \text{Max}[n+1, \text{anOrder}+1] \rangle \text{Proposition}\langle n \rangle \text{ where}$$

$$\text{Löb}\langle n \rangle \equiv \lambda[p:\text{Proposition}\langle n \rangle] (\vdash p) \Rightarrow \Psi$$

Löb's Paradox is blocked because the procedure Löb does **not** have a fixed point.

- Girard [Girard 1972] There is no type *Type* thereby blocking the paradox. However, the type of a type is a type, e.g. *Type* $\langle N \rangle$ is the type of N .

- Berry [Russell 1906] can be formalized using the proposition Characterize[s, k] meaning that the string s characterizes the integer k as follows:

Characterize \equiv

$$\lambda[s: \text{String} \langle \text{Proposition} \langle \text{anOrder} \rangle^N, k: N] \forall [x: N] \lfloor s[x] \rfloor \Leftrightarrow x=k$$

The Berry Paradox is to construct a string BString for the string for the proposition that holds for integer n if and only if every string with length less than 100 does not characterize n using the following definition:¹⁰⁴

StringSmallerThan100DoesNotCharacterize: $\text{Proposition} \langle \text{anOrder}+1 \rangle^N$

StringSmallerThan100DoesNotCharacterize \equiv

$$\lambda[n: N] \forall [s: \text{String} \langle \text{Proposition} \langle \text{anOrder} \rangle] \text{Length}[s] < 100 \Leftrightarrow \neg \text{Characterize}[s, n]$$

BString: $\text{String} \langle \text{Proposition} \langle \text{anOrder}+1 \rangle^N \rangle$

BString \equiv “StringForStringSmallerThan100DoesNotCharacterize”

Note that

- Length[BString] < 100.
- $\{s: \text{String} \langle \text{Proposition} \langle \text{anOrder} \rangle \mid \text{Length}[s] < 100\}$ is finite.
- Therefore, the following set is finite:

$$\text{BSet: Set} \langle N \rangle \equiv \{n: N_+ \mid \exists [s: \text{String} \langle \text{Proposition} \langle \text{anOrder} \rangle^N] \text{Length}[s] < 100 \wedge \text{Characterize}[s, n]\}$$

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

BNumber: $N \equiv \text{Least}[\text{BSet}]$

[BString][BNumber]

$$= \forall [s: \text{String} \langle \text{Proposition} \langle \text{anOrder} \rangle^N] \text{Length}[s] < 100 \Leftrightarrow \neg \text{Characterize}[s, \text{BNumber}]^{105}$$

However BString: $\text{String} \langle \text{Proposition} \langle \text{anOrder}+1 \rangle^N \rangle$ cannot be

substituted for s: $\text{String} \langle \text{Proposition} \langle \text{anOrder} \rangle^N \rangle$

Appendix 4: Notation of Direct Logic

Types i.e., a type is a *discrimination* of the following:¹⁰⁶

- *Boolean*::¹⁰⁷, *N*::¹⁰⁸, *O*::¹⁰⁹ and *Actor*::¹¹⁰
- *Term* $\langle t \rangle$::¹¹¹, *Expression* $\langle t \rangle$::¹¹², $\Lambda \langle t \rangle$ ¹¹³,
String $\langle t \rangle$::¹¹⁴, and *Type* $\langle t \rangle$::¹¹⁵, where t ::¹¹⁶
- *Proposition* $\langle \text{anOrder} \rangle$::¹¹⁷ and *Sentence* $\langle \text{anOrder} \rangle$::¹¹⁸ where
anOrder: N_+
- $(t_1 \oplus t_2)$::¹¹⁹, $[t_1, t_2]$::¹²⁰, $([t_1] \rightarrow t_2)$::¹²¹, $([t_1] \rightarrow_1 t_2)$::¹²² and $t_2^{t_1}$::¹²³ where
 t_1, t_2 ::
- $(t \nexists P)$:: where t :: and P :*Proposition*^t ¹²⁴

Propositions, i.e., a *Proposition* is a *discrimination* of the following:

- $(\neg\Psi_1), \Psi_1\wedge\Psi_2, \Psi_1\vee\Psi_2, \Psi_1\Rightarrow\Psi_2, (\Psi_1\Leftrightarrow\Psi_2):t$ where $\Psi_1, \Psi_2:t$ and $t::$
- $(x_1=x_2):Proposition\langle 1\rangle$ where $x_1, x_2:t$ and $t::$
- $(t_1\sqsubseteq t_2):Proposition\langle 1\rangle^{125}$ where $t_1, t_2::$
- $(x:t):Proposition\langle 1\rangle$ where $t::$
- $P[x]:Proposition\langle anOrder+1\rangle$ where $x:t, P:Proposition\langle anOrder\rangle^t$
and $anOrder:N_+$
- $P.[x]:Proposition\langle anOrder+1\rangle$ where $P:[t]\rightarrow Proposition\langle anOrder\rangle,$
 $x:t, t::$ and $anOrder:N_+$
- $(\Psi_1\vdash_T \Psi_2):t$ where $T:Theory^{126}, \Psi_1, \Psi_2:Proposition\langle anOrder\rangle$ and
 $t\sqsubseteq Proposition\langle anOrder\rangle$ and $anOrder:N_+$
- $(\Psi_1\vdash_{\frac{P}{T}} \Psi_2):t$ where $p:ProofChecker\langle T\rangle, T:Theory,$
 $\Psi_1, \Psi_2:Proposition\langle anOrder\rangle,$ and $t\sqsubseteq Proposition\langle anOrder\rangle$ and
 $anOrder:N_+$
- $(\forall t P)^{127}, (\exists t P)^{128}:Proposition\langle anOrder+1\rangle$ where
 $P:Proposition\langle anOrder\rangle^t, t\sqsubseteq Proposition$ and $anOrder:N_+$
- $(\lambda t P)^{129}:Proposition\langle anOrder+1\rangle$ where
 $P:Proposition\langle anOrder\rangle^t, t\sqsubseteq Proposition$ and $anOrder:N_+$
- $(\models \Phi):t$ where $\Phi:t, t\sqsubseteq Proposition\langle anOrder\rangle$ and $anOrder:N_+$
- $\lfloor s\rfloor:Proposition\langle anOrder\rangle$ where $s:Sentence\langle anOrder\rangle$ with no free
variables $T:Theory,$ and $anOrder:N_+$

Grammar (syntax) trees (*i.e.* terms, expressions and sentences) are defined below.

Terms, *i.e.*, a $Term\langle t \rangle$ is a *discrimination* of the following:

- $(Boolean): Constant\langle Type\langle Boolean \rangle \rangle$,
- $(N): Constant\langle Type\langle N \rangle \rangle$, $(O): Constant\langle Type\langle O \rangle \rangle$ and
- $(Actor): Constant\langle Type\langle Actor \rangle \rangle$
- $x: Term\langle t \rangle$ where $x: Constant\langle t \rangle$ and $t::$
- $(x_1 \oplus x_2): Term\langle x_1 \oplus x_2 \rangle$, $([x_1, x_2]): Term\langle [x_1, x_2] \rangle$,
- $([x_1] \rightarrow x_2): Term\langle [x_1] \rightarrow x_2 \rangle$, $([x_1] \rightarrow_1 x_2): Term\langle [x_1] \rightarrow_1 x_2 \rangle$ and
- $(x_2^{x_1}): Term\langle t_2^{t_1} \rangle$ where $x_1: Term\langle t_1 \rangle$, $x_2: Term\langle t_2 \rangle$, $t_1, t_2::$
- $(\lambda t_1 x): Term\langle t_2^{t_1} \rangle$ where $x: Term\langle t_2^{t_1} \rangle$, and $t_1, t_2::$
- $(f[x]): Term\langle t_2 \rangle$ where $f: Term\langle t_2^{t_1} \rangle$, $x: Term\langle t_1 \rangle$, $t_1, t_2::$
- $[x]: t$ where $x: Term\langle t \rangle$ with **no** free variables and $t::$

Expressions, i.e., an $\text{Expression} \langle t \rangle$ is a *discrimination* of the following:

- $x:\text{Expression} \langle t \rangle$ where $x:\text{Constant} \langle t \rangle$ and $t::$
- $x:\text{Expression} \langle t \rangle$ where $x:\text{Identifier} \langle t \rangle$ and $t::$
- $(e_1 \oplus e_2):\text{Expression} \langle t_1 \oplus t_2 \rangle, ([e_1, e_2]):\text{Expression} \langle [t_1, t_2] \rangle,$
 $([e_1] \rightarrow e_2):\text{Expression} \langle [t_1] \rightarrow t_2 \rangle,$ where $e_1:\text{Expression} \langle t_1 \rangle,$
 $e_2:\text{Expression} \langle t_2 \rangle,$ and $t_1, t_2::$
- $(e_1 \text{ ? True then } e_2, \text{ False then } e_3):\text{Expression} \langle t \rangle^{130}$ where
 $e_1:\text{Expression} \langle \text{Boolean} \rangle, e_2, e_3:\text{Expression} \langle t \rangle$ and $t::$
- $(e_1 \text{ either } e_2):\text{Expression} \langle t \rangle$ where $e_1, e_2:\text{Expression} \langle t \rangle$ and $t::$
- $(t_1 \rightsquigarrow E):\text{Expression} \langle [t_1] \rightarrow t_2 \rangle^{131}$ where
 $E:[\text{Expression} \langle t_1 \rangle] \rightarrow \text{Expression} \langle t_2 \rangle,$ and $t_1, t_2::$
- $(e.[x]):\text{Expression} \langle t_2 \rangle$ where $e:\text{Expression} \langle [t_1] \rightarrow t_2 \rangle,$
 $x:\text{Expression} \langle t_1 \rangle,$ and $t_1, t_2::$
- $\lfloor e \rfloor : t$ where $e:\text{Expression} \langle t \rangle$ with **no** free identifiers and $t::$

Every nondeterministic computable procedure can be obtained by abstraction from an expression:

$$\forall [f:([t_1] \rightarrow t_2)] \text{Nondeterministic}[f] \Rightarrow \exists [e:\text{Expression} \langle [t_1] \rightarrow t_2 \rangle] f = \lfloor e \rfloor$$

However as explained in this article, there are computations that cannot be implemented in the nondeterministic lambda calculus and therefore require Actors for their implementation.

Sentences, i.e., a *Sentence* is a *discrimination* of the following:

- $(\neg s_1), (s_1 \wedge s_2), (s_1 \vee s_2), (s_1 \Rightarrow s_2), (s_1 \Leftrightarrow s_2): t$ where $s_1, s_2: t$ and $t::$
- $(e_1 = e_2): \text{Sentence} \langle 1 \rangle$ where $e_1, e_2: \text{Expression} \langle t \rangle$ and $t::$
- $(e_1 \sqsubseteq e_2): \text{Sentence} \langle 1 \rangle$ where $e_1, e_2: \text{Expression} \langle t_1 \rangle$, $t_1: t_2$ and $t_2::$
- $(e_1: e_2): \text{Sentence} \langle 1 \rangle$ where $e_1: \text{Expression} \langle t_1 \rangle$, $e_2: \text{Expression} \langle t_2 \rangle$ and $t_1, t_2::$
- $(e::): \text{Sentence} \langle 1 \rangle$ where $e: \text{Expression} \langle t \rangle$ and $t::$
- $(\forall t P), (\exists t P): \text{Expression} \langle \text{Sentence} \langle \text{anOrder} + 1 \rangle \rangle$ where,
 $P: \text{Expression} \langle \text{Sentence} \langle \text{anOrder} \rangle \text{Expression} \langle t \rangle \rangle$,
 $t \sqsubseteq \text{Expression} \langle \text{Sentence} \rangle$, and $t::$
- $(\lambda t P): \text{Expression} \langle \text{Sentence} \langle \text{anOrder} + 1 \rangle \rangle$ where
 $P: \text{Expression} \langle \text{Sentence} \langle \text{anOrder} \rangle \text{Expression} \langle t \rangle \rangle$,
 $t \sqsubseteq \text{Expression} \langle \text{Sentence} \rangle$, and $t::$
- $(t \rightsquigarrow P): \text{Expression} \langle \text{Sentence} \langle \text{anOrder} + 1 \rangle \rangle^{132}$ where,
 $P: \text{Expression} \langle \text{Sentence} \langle \text{anOrder} \rangle \text{Expression} \langle t \rangle \rangle$,
 $t \sqsubseteq \text{Expression} \langle \text{Sentence} \rangle$, and $t::$
- $(P[x]): \text{Sentence} \langle \text{anOrder} + 1 \rangle^{133}$ where $x: \text{Expression} \langle t \rangle$,
 $P: \text{Expression} \langle \text{Sentence} \langle \text{anOrder} \rangle \text{Expression} \langle t \rangle \rangle$, $t::$ and $\text{anOrder}: N_+$
- $(P.[x]): \text{Sentence} \langle \text{anOrder} + 1 \rangle^{134}$ where $x: \text{Expression} \langle t \rangle$,
 $P: [\text{Expression} \langle t \rangle] \rightarrow \text{Expression} \langle \text{Sentence} \langle \text{anOrder} \rangle \rangle$, $t::$ and
 $\text{anOrder}: N_+$
- $(s_1 \vdash_T s_2): t$ where $T: \text{Expression} \langle \text{Theory} \rangle$, $s_1, s_2: t \sqsubseteq \text{Sentence}$ and
 $\text{anOrder}: N_+$
- $(s_1 \vdash_T^p s_2): t$ where $c: \text{Expression} \langle \text{ProofChecker} \langle T \rangle \rangle$,
 $T: \text{Expression} \langle \text{Theory} \rangle$, $s_1, s_2: \text{Expression} \langle \text{Sentence} \langle \text{anOrder} \rangle \rangle$
and $t \sqsubseteq \text{Proposition} \langle \text{anOrder} \rangle$ and $\text{anOrder}: N_+$
- $(\models s): t$ where $s: t$ and $t::$
- $\lfloor s \rfloor_T: \text{Proposition} \langle \text{anOrder} \rangle$ where $s: \text{Sentence} \langle \text{anOrder} \rangle$, $\text{anOrder}: N_+$,
 $T: \text{Expression} \langle \text{Theory} \rangle$, and there are **no** free variables in s .¹³⁵

Appendix 5: Rules of Direct Logic

The following are adequate for \wedge , \neg , \vdash , \vee , and \Rightarrow :

- \wedge (**Conjunction**)
 - \wedge **Introduction**
 $\Psi, \Phi \vdash (\Psi \wedge \Phi)$
 - \wedge **Elimination**
 $\Psi \wedge \Phi \vdash \Psi$
 - **Proves Both**
 $(\Theta \vdash (\Psi \wedge \Phi)) \dashv\vdash ((\Theta \vdash \Psi) \wedge (\Theta \vdash \Phi))$
- \neg (**Negation**)
 - **Double Negation**
 $\neg\neg\Psi \dashv\vdash \Psi$
 - **Proof by Contradiction**
 $(\Psi \vdash \Phi \wedge \neg\Phi) \vdash \neg\Psi$
- \vdash (**Inference**)
 - **Idempotency**
 $\Psi \vdash \Psi$
 - **Theorems can be used in proofs**
 $(\vdash\Psi) \vdash \Psi$
 - **Chaining**
 $\Psi, (\Psi \vdash \Phi) \vdash \Phi$
 - **Monotonicity**
 $(\Phi \vdash \Psi) \vdash (\Phi, \Theta \vdash \Psi)$
 - **Subproof**
 $(\Psi, \Phi \vdash \Theta) \dashv\vdash (\Psi \vdash (\Phi \vdash \Theta))$
 - **Theorems Prove Theorems**
 $((\vdash\Psi) \vdash \Phi) \vdash (\vdash\Phi)$
 - **Adequacy**
 $(\Psi \vdash \Phi) \dashv\vdash (\vdash(\Psi \vdash \Phi))$
- \vee (**Disjunction** defined)
 $\Psi \vee \Phi \equiv \neg(\neg\Psi \wedge \neg\Phi)$
- \Rightarrow (**Implication** defined)
 $\Psi \Rightarrow \Phi \equiv (\Psi \vdash \Phi)$

Theorem (Soundness). $(\vdash\Psi) \Rightarrow (\models\Psi)$

Proof: Axioms above are true and truth is preserved.¹³⁶

Theorem (Inferential Completeness). $(\vdash \Psi) \Leftrightarrow (\models \Psi)$

Proof: It is sufficient to prove $(\models \Psi) \Rightarrow (\vdash \Psi)$. $(\models \Psi) \vdash (\neg \Psi \vdash (\models \Psi) \wedge \neg \Psi)$.
Hence $(\models \Psi) \vdash (\neg \Psi \vdash \Psi \wedge \neg \Psi)$ because $(\models \Psi) \Rightarrow \Psi$. The theorem follows by proof by contradiction.

Theorem (Formal Consistency). $\not\vdash \Psi \wedge \neg \Psi$

Proof: Follows immediately from proof by contradiction because
 $(\vdash \Psi \wedge \neg \Psi) \vdash \Psi \wedge \neg \Psi$

Theorem (Step Introduction). $(\Psi \vdash \Phi), (\Psi, \Phi \vdash \Theta) \vdash (\Psi \vdash \Theta)$

Proof: Follows immediately from $(\Psi \vdash \Phi), (\Psi, \Phi \vdash \Theta), \Psi \vdash \Theta$

Theorem (Transitivity). $(\Psi \vdash \Theta), (\Theta \vdash \Phi) \vdash (\Psi \vdash \Phi)$

Proof: Follows immediately from $(\Psi \vdash \Theta), (\Theta \vdash \Phi), \Psi \vdash \Phi$ which follows from
 $(\Psi \vdash \Theta), (\Theta \vdash \Phi), \Psi, \Theta \vdash \Phi$

Theorem (Contrapositive). $(\Psi \vdash \Phi) \vdash (\neg \Phi \vdash \neg \Psi)$

Proof: Using proof by contradiction, follows immediately from
 $(\Psi \vdash \Phi) \vdash (\neg \Phi \vdash (\Psi \vdash \Phi \wedge \neg \Phi))$, which follows from $(\Psi \vdash \Phi), \neg \Phi \vdash (\Psi \vdash \Phi)$ and
 $(\Psi \vdash \Phi), \neg \Phi \vdash (\Psi \vdash \neg \Phi)$, the latter of which follows from $(\Psi \vdash \Phi), \neg \Phi, \Psi \vdash \neg \Phi$

Index

\neg , 7, 59, 61, 62, 63, 64
 \oplus , 60, 62, 63, 71
 \therefore , 61, 64

\therefore , 64
 $=$, 61, 64
 \forall , 7, 61, 64

\exists , 61, 64
 \in , 55
 \blacksquare , 61, 63, 64
 \doteq , 55
 \sqsubseteq , 61, 64
 \vdash , 61, 64
 \models , 9, 54, 56, 57, 61, 64
 \cup , 55
 $\boxed{?}$, 63
 \rightarrow , 60, 62, 63, 71
 \rightsquigarrow , 64
 \Leftrightarrow , 61, 64
 \rightarrow_1 , 60, 62
 \Leftrightarrow , 61, 64
Act, 18
Act₁, 38
 Actor, 60, 62
 Unbounded Nondeterminacy, 19
 Actors, 63
 Computational Adequacy, 19
 argumentation, 15
 Barwise, J., 44
 Beeson, M., 26
 Berry Paradox, 59
 Boolean, 4, 60, 62
 Boom, H., 26
 Burali-Forti, C., 58
 categorical, 6, 46
 categoricity
 natural numbers, 9
 ordinals, 54
 reals, 51
 Chaitin, G., 35
ProofChecker, 8, 53, 61, 64
 choice
 type, 54
 Church, A., 1, 7, 23, 24, 35, 37
 Church's Paradox, 26, 40
 Classical Direct Logic, 2
 Cohen, P., 32
 Computational Adequacy
 Actors, 19
 computational inadequacy
 nondeterministic lambda
 calculus, 49
 consistency
 formal, 14
 Consistency of Mathematics, 20
Constant, 62, 63
 Coq, 23
 Coquand, T., 23
 Curry, H., 1, 58
 Dawson, J., 24
 decidability
 inferential, 23
 Dedekind, R., 1
 Direct Logic
 sentence, 64
 Direct Logic
 expression, 63
 proposition, 61
 term, 62
 Eubulides of Miletus, 58
 expression
 Direct Logic, 63
Expression, 63
Expression, 60
 False, 4
 First-Order Thesis, 44
 Foundation
 sets, 55
 Galbraith, J. K., 14
 Girard, J., 58
 Gödel
 validity of incompleteness
 arguments, 33
 Gödel, K., 32
 Halt, 7

- halting problem, 7
- higher order logic, 6
- Hobbes, T., 43
- Hodges, W., 43
- Huet, G., 23
- Identifier*, 63
- ImUnprovable*, 23, 24, 33
- Inconsistency Robustness, 1
- indiscernibility, 8, 16, 25, 50, 54, 57
- inexhaustibility
 - Mathematics, 23
- inexpressibility, 25
- inexpressible proofs, 25
- inferability
 - computationally undecidable, 7
- inferability
 - computationally undecidable, 24
- Instance_i*, 40
- Isabelle, 23
- Kleene, S., 23
- Kuhn, T., 14
- Lakatos, I., 32
- lambda calculus
 - computational inadequacy, 49
- Lambda Calculus
 - Unbounded Nondeterminacy, 19
- Lambda Induction, 49
- Lam_e*, 48
- Law, J., 32
- Liar Paradox, 58
- Löb, M., 1, 58
- Logic Program
 - nondeterministic, 46
 - pure, 17, 19, 38
- Logic Programs
 - Unbounded Nondeterminacy, 19
- Maddy, P., 3
- Mathematics
 - inexhaustibility, 23
- Monk, R., 33
- N*, 7, 44, 60, 62
- Nat*, 6, 7, 14, 42, 45
 - proposition, 7
- Nat₁*, 36
- Natural Deduction, 20
- nondeterministic computable
 - procedure, 63
- nondeterministic lambda calculus, 17, 19, 38
- Nondeterministic Lambda Calculus, 48
- nondeterministic Logic Program, 46
- nondeterministic Turing Machine, 46
- O*, 60, 62
- Ord*, 54
- paradox, 1
- Paulson, L., 23
- Planck, M., 14
- Plotkin, G., 26
- Powerset
 - sets, 55
- Principle of Excluded Middle, 14
- Proof by Contradiction, 10
- Proof Checker, 8
- proof checking
 - computational decidability, 53
 - computationally decidable, 24
- proof inexpressibility, 25
- Proofs, 8, 53
- proposition
 - Direct Logic, 61
- Proposition, 4, 60, 61
- Pythagoreans, 2
- RealClosedFields*, 51
- reals
 - categoricity, 51
- Reals*, 51

Replacement
 sets, 55
 Rosser, J. B., 23
 Russell, B., 1, 23, 24
 sentence
 Direct Logic, 64
Sentence, 60, 64
 sets
 Axiom of Foundation, 55
 Axiom of Replacement, 55
 axiom of Union, 55
 powerset, 55
 Transitivity of \in , 55
 ZFC, 55
Sets $\langle t \rangle$, 55
*Sets*_o, 55, 56
String, 60
 Tarski, A., 4
 term
 Direct Logic, 62
Term, 60, 62
Theory, 61, 64
 Transitivity of \in
 sets, 55

 True, 4
 truth, 15
 Turing Machine
 nondeterministic, 46
 Turing, A., 7, 35
Type, 60, 71
 types
 choice, 54
 Unbounded Nondeterminacy, 19
 Unbounded Nondeterminism, 45
 Union
 sets, 55
 universe
 sets, 55
 von Neumann, J., 32
 Wittgenstein
 validity of Gödel incompleteness
 arguments, 33
 Wittgenstein, L., 1
 Woods, J., 26
 Zermelo, E., 1, 43
 ZFC
 sets, 55
 λ , 61, 62, 64
 Λ , 60

End Notes

¹ As in programming language integrated development environments, color has no semantics significance.

Double colons are used to express something is a type. For example $t::$ expresses that t is a type.

Types can be composed using the following:

- All functions from t_1 into t_2 is $t_2^{t_1}::$ where $t_1, t_2::$
A function is total and may be uncomputable.
- Tuple of types is $[t_1, t_2]::$ where $t_1, t_2::$
For example, $[Nat, Nat]$ is the type of pairs of Nat
- Discriminated union of types is $(t_1 \oplus t_2)::$ where $t_1, t_2::$
- Predicate restriction of a type is $(t \dot{\neq} P)::$ where $t::$ and $P: Proposition^t$
For example replacement for types (range of a function $f: t_2^{t_1}$)
is $t_2 \dot{\neq} \lambda[y:t_2] \exists[x:t_1] y=f[x]$
- Computable procedures from t_1 into t_2 is $([t_1] \rightarrow t_2)::$ where $t_1, t_2::$
A computable procedure can be partial and can be indeterminate in its outcome.
- Type of a type is $Type \langle t \rangle::$ where $t::$ and $Type$ is a parametrized type with parameter t . Parametrized types have become popular in programming languages where in Java they are called “generics.”
There is **no** unparameterized type $Type$ in order to block Girard’s paradox.

² Blocking *all* the known paradoxes is necessary to defeat hackers.

³ Inference rules for Direct Logic are presented in an appendix.

⁴ [Verbrugge 2010]

-
- ⁵ Axioms and rules of Direct Logic are presented in appendices of this article.
- ⁶ Performance of computer information systems is measured in consumption of processing cycles and storage space as well as latency for response. Pervasive inconsistency for information systems means that there are numerous inconsistencies that cannot be readily found and that many of the ones that are found cannot be easily removed.
- ⁷ In this sense, Mathematics is “incomplete”, but not in the sense of Gödel’s results. See below on the inferential completeness of standard theories of mathematics, such as natural numbers, real numbers, ordinal numbers, set theory, lambda calculus, and Actors.
- ⁸ Which is not the same as proving the much *stronger* proposition that no contradiction can be derived from the exact axioms and inference rules of Direct Logic.
- ⁹ Furthermore, theorems of these theories are not enumerable by a provably total procedure.
- ¹⁰ cf. [Wigner 1960]
- ¹¹ [Nielsen 2014]
- ¹² parametrized types have become very popular in programming languages, e.g., Java where they are called “generics.”
- ¹³ Classical results can be embedded in intuitionistic logic as follows:
- $\neg\neg\neg\Psi \Leftrightarrow \neg\Psi$
 - \exists can be transformed to $\neg\forall\neg$
 - $\Psi_1\vee\Psi_2$ can be transformed to $\neg(\neg\Psi_1\wedge\neg\Psi_2)$
- ¹⁴ Some of the outputs are conventionally identified as being the same real number, e.g., .0111111... and .1000000000
- ¹⁵ For example (p[3])[y] holds if and only if y=3.
- ¹⁶ [Isaacson 2007]
- ¹⁷ A theory is defined by a set of propositions in Direct Logic that are taken to be axioms of the theory.
- ¹⁸ There are **no** sets in the induction axiom. Quine famously criticized 2nd order theory as nothing more than “*set theory in sheep’s clothing*” [Quine 1970, pg. 66].

¹⁹ $Type \langle Nat \rangle ::$ and

$$\begin{aligned}
& \forall [P: Proposition \langle Nat \rangle Type \langle Nat \rangle] \\
& ((\forall [t_1, t_2: Type \langle Nat \rangle] P[t_1] \wedge P[t_2] \Rightarrow P[t_2^{t_1}]) \\
& \hspace{15em} // \text{all functions from } t_1 \text{ into } t_2 \\
& (\forall [t_1, t_2: Type \langle Nat \rangle] P[t_1] \wedge P[t_2] \Rightarrow P[t_1 \oplus t_2]) \\
& \hspace{15em} // \text{discriminated union of } t_1 \text{ and } t_2 \\
& (\forall [t: Type \langle Nat \rangle Q: Proposition \langle Nat \rangle Type \langle Nat \rangle] P[t] \Rightarrow P[t \# Q]) \\
& \hspace{15em} // \text{restriction of } t \text{ by } Q \\
& (\forall [t: Type \langle Nat \rangle] P[t] \Rightarrow P[Type \langle t \rangle]) \quad // \text{type of } t \\
& (\forall [t_1, t_2: Type \langle Nat \rangle] P[t_1] \wedge P[t_2] \Rightarrow P[[t_1] \rightarrow t_2])) \\
& \hspace{15em} // \text{computable procedures from } t_1 \text{ into } t_2 \\
& \Rightarrow (\forall [t: Type \langle Nat \rangle] P[t])
\end{aligned}$$

Above characterizes $Type \langle Nat \rangle$ up to a unique isomorphism.

²⁰ Suppose that \mathcal{X} satisfies the axioms for $Proposition \langle Nat \rangle$. Inductively

define $I: \mathcal{X}^{Proposition \langle Nat \rangle}$ as follows:

- $I[x_1 = x_2] \equiv (x_1 = x_2)$
- $I[\neg \Psi] \equiv \neg I[\Psi]$
- $I[\Psi_1 \wedge \Psi_2] \equiv I[\Psi_1] \wedge I[\Psi_2]$
- $I[\forall [x: t] P[x]] \equiv \forall [x: t] I[P[x]]$ where $t: Type \langle Nat \rangle$ and $P: Proposition \langle Nat \rangle^t$

Theorem. I is a unique isomorphism between $Proposition \langle Nat \rangle$ and \mathcal{X}

Theorem. $\forall[\Psi: \textit{Proposition} \langle \text{Nat} \rangle] \exists[s: \textit{Sentence} \langle \text{Nat} \rangle] \Psi = \llbracket s \rrbracket$

Proof. Induction on *Proposition* $\langle \text{Nat} \rangle$

For example, inductively suppose that

$$\forall[t: \textit{Type} \langle \text{Nat} \rangle, x: t, P: \textit{Proposition} \langle \text{Nat} \rangle^t] \\ \exists[s: \textit{Sentence} \langle \text{Nat} \rangle] \llbracket s \rrbracket = P[x]$$

which can be rephrased as follows:

$$\forall[t: \textit{Type} \langle \text{Nat} \rangle, x: t, P: \textit{Proposition} \langle \text{Nat} \rangle^t] \\ \exists[E: \textit{Term} \langle \textit{Proposition} \langle \text{Nat} \rangle^t \rangle] \llbracket E \rrbracket [x] = P[x]$$

which means that

$$\forall[t: \textit{Type} \langle \text{Nat} \rangle, P: \textit{Proposition} \langle \text{Nat} \rangle^t] \\ \exists[E: \textit{Term} \langle \textit{Proposition} \langle \text{Nat} \rangle^t \rangle] \llbracket E \rrbracket = P$$

Let f be an existential function such that

$$\forall[t: \textit{Type} \langle \text{Nat} \rangle, P: \textit{Proposition} \langle \text{Nat} \rangle^t] \llbracket f[t, P] \rrbracket = P.$$

Therefore because $\llbracket (\forall t f[t, P]) \rrbracket = \forall t \llbracket f[t, P] \rrbracket$

$$\forall[t: \textit{Type} \langle \text{Nat} \rangle, P: \textit{Proposition} \langle \text{Nat} \rangle^t] \llbracket (\forall t f[t, P]) \rrbracket = \forall t P$$

Consequently,

$$\forall[t: \textit{Type} \langle \text{Nat} \rangle, P: \textit{Proposition} \langle \text{Nat} \rangle^t] \\ \exists[s: \textit{Sentence} \langle \text{Nat} \rangle] \llbracket s \rrbracket = \forall t P$$

²¹ See the section on ordinals for a more convincing example of a procedure with a nondenumerable domain.

²² [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.”

²³ [Hardy 1992] page 19.

²⁴ highlighted below

²⁵ cf. [Zermelo 1932] pp. 6-7.

²⁶ Examples:

- $\forall [P:\textit{Proposition} \langle 1 \rangle^{\mathbb{N}}] (\models \forall [i:\mathbb{N}] P[i]) \Rightarrow \vdash_{\textit{Nat}} \forall [i:\mathbb{N}] P[i]$

Suppose in **Nat**, $P:\textit{Proposition} \langle 1 \rangle^{\mathbb{N}}$ and $\models \forall [i:\mathbb{N}] P[i]$. Further suppose to obtain a contradiction that $\neg \forall [i:\mathbb{N}] P[i]$. Therefore $\exists [i:\mathbb{N}] \neg P[i]$ and by Existential Elimination $\neg P[i_0]$ where $i_0:\mathbb{N}$, which contradicts $\models P[i_0]$, from the hypothesis of the theorem. Therefore $\vdash_{\textit{Nat}} \forall [i:\mathbb{N}] P[i]$ using proof by contradiction in **Nat**.

- $\forall [P:\textit{Proposition} \langle 1 \rangle^{\mathbb{N}}] (\models \exists [i:\mathbb{N}] P[i]) \Rightarrow \vdash_{\textit{Nat}} \exists [i:\mathbb{N}] P[i]$

Suppose in **Nat**, $P:\textit{Proposition} \langle 1 \rangle^{\mathbb{N}}$ and $\models \exists [i:\mathbb{N}] P[i]$. Further suppose to obtain a contradiction that $\neg \exists [i:\mathbb{N}] P[i]$ and therefore $\forall [i:\mathbb{N}] \neg P[i]$. However, $\models P[i_0]$ where $i_0:\mathbb{N}$, which contradicts $\neg P[i_0]$. Therefore $\vdash_{\textit{Nat}} \exists [i:\mathbb{N}] P[i]$ using proof by contradiction in **Nat**.

²⁷ often misleading called ω -consistency [Gödel 1931]

²⁸ 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.”

²⁹ The theorems themselves can be represented as character strings because totality can be expressed as the abstraction of a character string and each procedure can be represented as the abstraction of a character string.

³⁰ Theorem. There are uncountably many countable ordinals (order types). Totality proofs have countable ordinals of arbitrarily high degree.

³¹ Note that the results in [Gödel 1931] **do not apply** because propositions in Mathematics are strongly typed and consequently the fixed point used construct Gödel’s proposition *It’s Unprovable* **does not exist** in Mathematics. See the critique of Gödel’s results in this article.

³² 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.]”

³³ for discussion see [Hewitt 2015d]

³⁴ 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

³⁵ in an unlawful way (Einstein, a member of the editorial board, refused to support Hilbert's action)

³⁶ Hilbert letter to Brouwer, October 1928

³⁷ Gödel said "Has Wittgenstein lost his mind?"

³⁸ However, Actors cannot implement any pure functions that cannot be implemented in the nondeterministic lambda calculus.

³⁹ of type `[Com]→`

```
Outcome[created= FiniteSet⟨Actor⟩, // new Actors
sent= FiniteSet⟨Com⟩, // new Communications
next= Behavior]
```

⁴⁰ 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].

⁴¹ Note that the results in [Löb 1955] **do not apply** because propositions in Mathematics are strongly typed and consequently the fixed point used to establish his result does not exist. See discussion of Löb's Paradox in this article.

⁴² Note that the results in [Gödel 1931] **do not apply** because propositions in Mathematics are strongly typed and consequently the fixed point used to construct Gödel's proposition *I'mUnprovable* **does not exist** in Mathematics. See the critique of Gödel's results in this article.

⁴³ As shown above, there is a simple proof in Classical Direct Logic that Mathematics (\vdash) is formally consistent. If the stated axioms and rules of inference of Classical Direct Logic have 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 formally consistent is very robust. One explanation is that formal 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 formal consistency of Mathematics (\vdash) that does not use all of the machinery of Classical Direct Logic.

The usefulness of Classical Direct Logic depends crucially on the much *stronger* proposition that Mathematics is operationally consistent, i.e., that there is no proof of contradiction from the stated axioms and inference rules of Direct Logic. Good evidence for the inferential consistency of Mathematics comes from the way that Classical Direct Logic avoids the known paradoxes. Humans have spent millennia devising paradoxes.

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 Ψ 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.

⁴⁴ [Gödel 1931] based 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 definition of NotProvable could be attempted as follows:

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

With strong types, the attempted definition becomes:

$$\text{NotProvable}\langle n:N_n \rangle [\text{Proposition} \langle n \rangle] \rightarrow_1 \text{Proposition} \langle n+1 \rangle$$

$$\text{NotProvable}\langle n:N_n \rangle \equiv \lambda[p:\text{Proposition} \langle n \rangle] \not\vdash p$$

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

$$\text{NotProvable}\langle n:N_n \rangle [\text{I'mUnprovable}] \Leftrightarrow \text{I'mUnprovable}$$

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

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

$$\text{NotProvable}\langle n:N_n \rangle [\text{String} \langle \text{Proposition} \langle n \rangle \rangle] \rightarrow_1 \text{String} \langle \text{Proposition} \langle n+1 \rangle \rangle$$

$$\text{NotProvable}\langle n:N_n \rangle \equiv \lambda[s:\text{String} \langle \text{Proposition} \langle n \rangle \rangle] \text{ "'} \not\vdash \text{ " s"}$$

Consequently, there is no fixed point *I'mUnprovableString* for the procedure $\text{NotProvable}\langle n:N_n \rangle$ such that the following holds (where [s] is the proposition for well-formed string s):

$$[\text{NotProvable}\langle n:N_n \rangle [\text{I'mUnprovableString}]] \Leftrightarrow [\text{I'mUnprovableString}]$$

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.

⁴⁵ 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.”

⁴⁶ Consequently, there can cannot be any escape hatch into an unformalized “meta-theory.”

⁴⁷ sometimes called logical “incompleteness”

⁴⁸ The claim also relied on Gödel's proposition *I'mUnprovable*.

⁴⁹ Fixed points exist for types other than propositions.

⁵⁰ emphasis in original

⁵¹ [Gödel 1931] was accepted doctrine by mainstream logicians for over eight decades.

⁵² Of course, Direct Logic must preserve as much previous learning as possible.

⁵³ According to Solomon 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]

⁵⁴ [Feferman 1986, pg. 1 and 8]

⁵⁵ Wittgenstein in 1937 published in Wittgenstein 1956, p. 50e and p. 51e]

⁵⁶ Wittgenstein was granting the supposition that [Gödel 1931] had proved inferential undecidability (sometimes called “incompleteness”) of Russell's system, that is., $\vdash \neg P$. However, inferential undecidability is easy to prove using the proposition P where $P \Leftrightarrow \neg P$:

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

1) $\vdash \neg P$ from the hypothesis because $P \Leftrightarrow \neg P$

2) $\vdash \vdash P$ from the hypothesis by Adequacy.

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

⁵⁷ [Wang 1972]

⁵⁸ The Liar Paradox [Eubulides of Miletus] is an example of using untyped propositions to derive an inconsistency. See appendix on paradoxes.

⁵⁹ According to [Church 1956 page 329]: “completeness [as to provability] is unattainable, as is shown in the incompleteness theorems of Gödel.”

⁶⁰ [Church 1935] correctly proved computational undecidability without using Gödel's *I'mUnprovable*. The Church theorem and its proof are very robust.

⁶¹ $\models \forall [P: \text{String} \langle \text{Proposition} \rangle]$

$\vdash_{\text{Nat}_1} (\forall [i: \mathbb{N}] \vdash_{\text{Nat}_1} [P[i]]) \Leftrightarrow \forall [i: \mathbb{N}] [P[i]]$

⁶² 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!

⁶³ Which is **not** the same as proving the much *stronger* proposition that Mathematics is operationally consistent, i.e., that there is no proof of contradiction from the stated axioms and inference rules of Direct Logic.

⁶⁴ **Theorem:** $\vdash_{\text{Nat}} \forall [P: \text{String} \langle 1 \rangle^{\mathbb{N}}]$
 $(\lfloor P[0] \rfloor \wedge \forall [i: \mathbb{N}] \lfloor P[i] \rfloor \Rightarrow \lfloor P[i+1] \rfloor) \Rightarrow \forall [i: \mathbb{N}] \lfloor P[i] \rfloor$

⁶⁵ with the consequence that the argument in Church's Paradox is blocked in the theory **Nat** because theorems are not enumerable by a provably total procedure

⁶⁶ In 1666, England's House of Commons introduced a bill against atheism and blasphemy, singling out Hobbes' Leviathan. Oxford university condemned and burnt Leviathan four years after the death of Hobbes in 1679.

⁶⁷ Likewise, relational 1st order set theory (e.g. ZFC) is very weak. See discussion in this article.

⁶⁸ ContinuumForReals is defined as follows:

$$\text{ContinuumForReals} \Leftrightarrow \nexists [S: \text{Boolean}^{\mathbb{N}}] \mathbb{N} \leq S \leq \text{Boolean}^{\mathbb{N}}$$

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

$$\text{ContinuumForBorelSets} \Leftrightarrow \nexists [S: \text{BorelSet}] \mathbb{N} \leq S \leq \text{Boolean}^{\mathbb{N}}$$

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

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

For less well behaved subset of \mathcal{R} , ContinuumForReals remains an open problem.

Note that it is important not to confuse ContinuumForReals with ContinuumForRelational1stOrderZFC. **Relational1stOrderZFC** has countably many 1st order propositions as axioms. [Cohen 1963] proved the following theorem which is much weaker than ContinuumForReals because sets in the models of **Relational1stOrderZFC** do **not** include all of *Proposition* $\langle 1 \rangle^{\mathbb{N}}$ and the theory **Relational1stOrderZFC** is much weaker than the theory **Sets_N**:

- $\nVdash_{\text{Relational1stOrderZFC}} \text{ContinuumForRelational1stOrderZFC}$
- $\nVdash_{\text{Relational1stOrderZFC}} \neg \text{ContinuumForRelational1stOrderZFC}$

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

$$\vdash_{\text{Sets}_N} \text{ContinuumForReals}$$

-
- ⁶⁹ [Zermelo 1930, van Dalen 1998, Ebbinghaus 2007]
- ⁷⁰ 1st order theories fall prey to paradoxes like the Löwenheim–Skolem theorems (e.g. any 1st order theory of the real numbers has a countable model). Theorists have used the weakness of 1st order theory to prove results that do not hold in stronger formalisms such as Direct Logic [Cohen 1963, Barwise 1985].
- ⁷¹ a restricted form of Model Checking in which the properties checked are limited to those that can be expressed in Linear-time Temporal Logic has been studied [Clarke, Emerson, Sifakis, *etc.* ACM 2007 Turing Award].
- ⁷² proving that software developers and computer systems are using the same structures
- ⁷³ An implementation of such a system is given below in this article.
- ⁷⁴ the lambda calculus is a special case of Logic Programs
- ⁷⁵ cf. Plotkin [1976]
- ⁷⁶ up to a unique isomorphism
- ⁷⁷ Rejection of the 1st Order Thesis resolves the seeming paradox between the formal proof in this article that Mathematics formally proves its own formal consistency and the proof that ‘Every “strong enough” formal system that admits a proof of its own consistency is actually inconsistent.’ [Paulson 2014]. Although Mathematics is “strong enough,” the absence of “self-referential” propositions (constructed using the **Y** untyped fixed point operator on propositions) blocks the proof of formal inconsistency to which Paulson referred.
- ⁷⁸ e.g. see Terminator [Knies 2006], which practically solves the halting problem for device drivers
- ⁷⁹ In order to show that the listed primitives are an adequate basis for **Lam**_τ, it is sufficient to show that the S (Substitution) combinator can be implemented. The following definition suffices:
Substitute_{⟨t₁, t₂, t₃⟩}:([t₁, [[t₁→t₂, [t₂→t₃]] → t₃])
Substitute_{⟨t₁, t₂, t₃⟩}.[x, pair] ≡ ((2nd.[pair]).[x]).[1st.[pair]].[x]]
- ⁸⁰ **Fix** implements recursion.
For example:
F:([N]→N)→([N]→N)
F.[g] ≡ [i:N] ~ i=1 [?] **True then 1, False then i*g.[i-1]**
Therefore by the **Fix** axiom, **Fix**_{⟨N, N⟩}.[F]=F.[**Fix**_{⟨N, N⟩}.[F]] and
Fix_{⟨N, N⟩}.[F] = F.[Factorial] = Factorial where
Factorial ≡ [i:N] ~ i=1 [?] **True then 1, False then i*Factorial.[i-1]**
- ⁸¹ Because of Lambda Equality, the domain of [Scott 2015] is *not* a valid model of **Lam**_τ

⁸² $t, t_1, t_2: \text{Type} \langle \Lambda \langle t \rangle \rangle$

⁸³ where $F^1 \cdot [x] \equiv F \cdot [x]$

$$F^{n+1} \cdot [x] \equiv F^n \cdot [F \cdot [x]]$$

⁸⁴ e.g., ones using concurrent Actors. See discussion in this article.

⁸⁵ cf. [Engeler 1981; Hindley, and Seldin 2008]

⁸⁶ $t, t_1, t_2: \text{Type} \langle \Lambda \langle t \rangle \rangle$

⁸⁷ Prove by induction on $f, g: \Lambda \langle t \rangle$

⁸⁸ $\Phi_1 \Rightarrow \Phi_2$ means $\forall [i: N] \Phi_1[i] \Rightarrow \Phi_2[i]$

Weakest precondition because:

$$\forall [\Psi: \text{Proposition}^N]$$

$$(\Psi \Rightarrow \lambda [i: N] \Phi[f \cdot [i]]) \Leftrightarrow (\Psi \Rightarrow \text{WeakestPrecondition}[\Phi, f])$$

⁸⁹ Robinson [1961]

⁹⁰ [Dedekind 1888]

⁹¹ The following can be used to characterize the real numbers (\mathcal{R}) up to a unique isomorphism:

$$\forall [S: \text{Set} \langle \mathcal{R} \rangle] S \neq \{\} \wedge \text{Bounded}[S] \Leftrightarrow \text{HasLeastUpperBound}[S]$$

where

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

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

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

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

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

⁹² The theory of the ordinals **Ord** is axiomatised as follows:

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

$\omega_0 = \mathcal{N}$

$\forall[\alpha: \mathcal{O}] \alpha > 0_{\mathcal{O}} \Rightarrow \omega_{\alpha} \doteq \text{Boolean}^{\{\beta: \mathcal{O} \mid \beta < \omega_{\alpha}\}}$

$\forall[\alpha, \beta: \mathcal{O}] \beta \doteq \omega_{\alpha} \Leftrightarrow \omega_{\alpha} = \beta \vee \omega_{\alpha} \in \beta$

where $t_1 \doteq t_2 \Leftrightarrow \exists[f: t_2^{t_1}] 1 \text{to} 1 \text{onto} \langle t_1, t_2 \rangle [f]$

$1 \text{to} 1 \langle t_1, t_2 \rangle [f: t_2^{t_1}] \Leftrightarrow \forall[x_1, x_2: t_1] f[x_1] = f[x_2] \Leftrightarrow x_1 = x_2$

$1 \text{to} 1 \text{onto} \langle t_1, t_2 \rangle [f: t_2^{t_1}]$

$\Leftrightarrow 1 \text{to} 1 \langle t_1, t_2 \rangle [f: t_2^{t_1}] \wedge \forall[y: t_2] \exists[x: t_1] f[x] = y$
- Transitivity of <

$\forall[\alpha, \beta < \alpha, \delta < \beta: \mathcal{O}] \alpha < \delta$
- $\forall[\alpha, \beta: \mathcal{O}] \alpha < \beta \vee \alpha = \beta \vee \beta < \alpha$
- $\forall[\alpha, \beta: \mathcal{O}] \alpha < \beta \Leftrightarrow \neg \beta < \alpha$
- The following ordinal induction axiom holds:

$\forall[P: \text{Proposition} \langle \text{order} \rangle^{\mathcal{O}}]$

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

⁹³ For each type \mathcal{X} that satisfies the theory **Ord** there is a unique isomorphism $I: \mathcal{X}^{\mathcal{O}}$ inductively defined as follows:

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

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

$$\forall[\alpha: \text{Limit}\langle \mathcal{O} \rangle] I[\alpha] \equiv y$$

$$\text{where } y: \mathcal{X} \wedge \forall[\beta < \alpha] y \preceq_{\mathcal{X}} I[\beta]$$

$$\wedge \forall[z: \mathcal{X}] (\forall[\beta < \alpha] z \preceq_{\mathcal{X}} I[\beta]) \Rightarrow y \preceq_{\mathcal{X}} z$$

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

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

⁹⁴ Prove by ordinal induction on $\alpha, \beta: \mathcal{O}$

⁹⁵ [Bourbaki 1972; Fantechi, et. al. 2005]

⁹⁶ This implies, for example, that no set is an element of itself.

⁹⁷ Proof: Suppose $S: \text{Sets}_{\alpha} \langle t \rangle$ and therefore $\exists[\alpha: \mathcal{O}] S: \text{Sets}_{\alpha} \langle t \rangle$

Proof by ordinal induction on

$$P[\beta: \mathcal{O}] \equiv \forall[X \in S] X: \text{Sets}_{\beta} \langle t \rangle$$

$$\text{Assume: } (\forall[\beta < \alpha: \mathcal{O}] \forall[X \in S] X: \text{Sets}_{\beta} \langle t \rangle) \Leftrightarrow \forall[X \in S] X: \text{Sets}_{\alpha} \langle t \rangle$$

Show: $\forall[X \in S] X: \text{Sets}_{\alpha} \langle t \rangle$

Assume: $X \in S$

Show $X: \text{Sets}_{\alpha} \langle t \rangle$

Proof by cases on α

1. $X: \text{Sets}_0 \langle t \rangle$

$X: \text{Boolean}^t$

2. $\forall[\alpha: \mathcal{O}] \text{Sets}_{\alpha} \langle t \rangle = \text{Set} \langle \text{Sets}_{\alpha-1} \langle t \rangle \rangle$

$X: \text{Sets}_{\alpha-1} \langle t \rangle$ QED by induction hypothesis

3. $\forall[\alpha: \text{Limit} \langle \mathcal{O} \rangle] \exists[\beta < \alpha, Y: \text{Sets}_{\beta} \langle t \rangle] X \in Y$

QED by induction hypothesis

⁹⁸ Proof: Suppose $S: \text{Sets}_{\alpha} \langle t \rangle$ and therefore $\exists[\alpha: \mathcal{O}] S: \text{Sets}_{\alpha} \langle t \rangle$

$S: \text{Sets}_{\alpha} \langle t \rangle$

Show: $\text{Boolean}^S: \text{Sets}_{\alpha} \langle t \rangle$

$\text{Boolean}^S: \text{Sets}_{\alpha+1} \langle t \rangle$ QED

⁹⁹ Proof by ordinal induction on

$$P[\alpha: \mathcal{O}] \equiv \forall[S: \text{Sets}_{\alpha} \langle t \rangle] US: \text{Sets}_{\alpha} \langle t \rangle$$

$$\text{Assume: } \forall[\beta < \alpha: \mathcal{O}] \forall[S: \text{Sets}_{\beta} \langle t \rangle] US: \text{Sets}_{\beta} \langle t \rangle$$

Show: $\forall[S: \text{Sets}_{\alpha} \langle t \rangle] US: \text{Sets}_{\alpha} \langle t \rangle$

Assume: $S: \text{Sets}_{\alpha} \langle t \rangle$

Show: $US: \text{Sets}_{\alpha} \langle t \rangle$

$$\forall[X: \text{Sets}_{\alpha} \langle t \rangle] X \in US \Leftrightarrow \exists[Y \in S] X \in Y$$

$$\forall[X: \text{Sets}_{\alpha} \langle t \rangle] X \in US \Leftrightarrow \exists[\beta < \alpha: \mathcal{O}, Y: \text{Sets}_{\beta} \langle t \rangle] X \in Y$$

$$\forall[X: \text{Sets}_{\alpha} \langle t \rangle] X \in US \Rightarrow X: \text{Sets}_{\alpha} \langle t \rangle$$

QED by definition of $\text{Sets}_{\alpha} \langle t \rangle$

¹⁰⁰ Suppose $f: \text{Sets}_{\langle t \rangle}^{\text{Sets}_{\langle t \rangle}}$ and $S: \text{Sets}_{\langle t \rangle}$
 Show $\text{Image}_{\langle t \rangle}[f, S]: \text{Sets}_{\langle t \rangle}$
 Proof by ordinal induction on
 $P[\alpha: \mathbf{O}] \Leftrightarrow S: \text{Sets}_{\alpha \langle t \rangle} \Rightarrow \text{Image}_{\langle t \rangle}[f, S]: \text{Sets}_{\langle t \rangle}$
 Suppose $\forall [\beta < \alpha: \mathbf{O}] S: \text{Sets}_{\beta \langle t \rangle} \Rightarrow \text{Image}_{\langle t \rangle}[f, S]: \text{Sets}_{\langle t \rangle}$
 Show $S: \text{Sets}_{\alpha \langle t \rangle} \Rightarrow \text{Image}_{\langle t \rangle}[f, S]: \text{Sets}_{\langle t \rangle}$
 Suppose $S: \text{Sets}_{\alpha \langle t \rangle}$
 Show $\text{Image}_{\langle t \rangle}[f, S]: \text{Sets}_{\langle t \rangle}$
 $\forall [y: \text{Sets}_{\langle t \rangle}] y: \text{Image}_{\langle t \rangle}[f, S] \Leftrightarrow \exists [x \in S] f[x]=y$
 Show $\forall [y: \text{Sets}_{\langle t \rangle}] y \in \text{Image}_{\langle t \rangle}[f, S] \Rightarrow y: \text{Sets}_{\langle t \rangle}$
 Suppose $y: \text{Sets}_{\langle t \rangle} \wedge y \in \text{Image}_{\langle t \rangle}[f, S]$
 Show $y: \text{Sets}_{\langle t \rangle}$
 $\exists [x \in S] f[x]=y$ because $y \in \text{Image}_{\langle t \rangle}[f, S]$
 $\exists [\beta < \alpha: \mathbf{O}] x: \text{Sets}_{\beta \langle t \rangle}$ because $x \in S$ and $S: \text{Sets}_{\alpha \langle t \rangle}$
 $\text{Image}_{\langle t \rangle}[f, x]: \text{Sets}_{\langle t \rangle}$ by induction hypothesis
 Show $f[x]: \text{Sets}_{\langle t \rangle}$
 Suppose $z \in f[x]$
 Show $z: \text{Sets}_{\langle t \rangle}$
 $z \in \text{Sets}_{\langle t \rangle}$ because $z \in f[x]$ and $\text{Image}_{\langle t \rangle}[f, x]: \text{Sets}_{\langle t \rangle}$
 $f[x]: \text{Sets}_{\langle t \rangle}$
 $y: \text{Sets}_{\langle t \rangle}$ because $f[x]=y$

¹⁰¹ [Mizar; Matuszewski1 and Rudnicki: 2005; Naumowicz and Artur Kornilowicz 2009; Naumowicz 2009]

¹⁰² Note that this proof is fundamentally different from the categoricity proof in [Martin 2015].

¹⁰³ Prove by ordinal rank on $s_1, s_2: \text{Sets}_{\langle t \rangle}$

¹⁰⁴ Note that the Berry paradox is blocked using strong types because BString is a string for a term of a proposition of anOrder+1 thereby preventing it from being substituted for a string for a term of a proposition of anOrder.

¹⁰⁵ substituting BNumber for n

¹⁰⁶ There is no universal type. Instead, *Type* is parameterized, e.g.,

$\text{Boolean}: \text{Type}_{\langle \text{Boolean} \rangle}$ and $N: \text{Type}_{\langle N \rangle}$

¹⁰⁷ **True** \neq **False**, **True**: *Boolean*, and **False**: *Boolean*

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

¹⁰⁸ *N* is the type of the natural numbers

-
- ¹⁰⁹ \mathcal{O} is the type of ordinals
- ¹¹⁰ $Actor$ is the type of Actors
- ¹¹¹ term of type t . The following axiom holds:
 $\forall [t::, x: Term \langle t \rangle] [x]::t$
- ¹¹² expression of type t . The following axiom holds:
 $\forall [t::, e: Expression \langle t \rangle] [e]::t$
- ¹¹³ $\Lambda \langle t \rangle$ is the type of lambda procedures over t
- ¹¹⁴ string of type t . The following axiom holds:
 $\forall [t::, s: String \langle t \rangle] [s]::t$
- ¹¹⁵ type of type t
- ¹¹⁶ $\forall [t::] t: Type \langle t \rangle$
- ¹¹⁷ $Proposition \langle anOrder \rangle$ is the parametrized type consisting of type $Proposition$ parametrized by $anOrder$.
- ¹¹⁸ $Sentence \langle anOrder \rangle$ is the parametrized type consisting of type $Sentence$ parametrized by $anOrder$.
- ¹¹⁹ Discrimination of t_1 and t_2
 For $i=1,2$
- If $x:t_i$, then $((t_1 \oplus t_2)[x]):(t_1 \oplus t_2)$ and $x = ((t_1 \oplus t_2)[x]) \downarrow t_i$.
 - $\forall [z:t] z:t_1 \oplus t_2 \Leftrightarrow \exists [x:t_i] z = (t_1 \oplus t_2)[x]$
- ¹²⁰ type of 2-element list with first element of type t_1 and with second element of type t_2
- ¹²¹ Type of computable *nondeterministic* procedures from t_1 into t_2 .
 If $f:([t_1] \rightarrow t_2)$ and $x:t_1$, then $f.[x]:t_2$
- ¹²² Type of computable *deterministic* procedures from t_1 into t_2 .
 If $f:([t_1] \rightarrow_1 t_2)$ and $x:t_1$, then $f.[x]:t_2$.
- ¹²³ Type of functions from t_1 into t_2 . If $f:t_2^{t_1}$ and $x:t_1$, then $f[x]:t_2$.
- ¹²⁴ $\forall [x:t] x:t \# P \Leftrightarrow P[x]$
 For example,
 $\forall [t::, X: Set \langle Set \langle t \rangle \rangle] UX \equiv t \# \lambda [y:t] \exists [Z: Set \langle t \rangle] Z \in X \wedge y \in Z$

¹²⁵ t_1 is a subtype of t_2 , i.e., $\forall[x:t_1] x:t_2$

¹²⁶ *Theory* \equiv *Boolean Proposition*, i.e., sets of propositions

¹²⁷ meaning $\forall[x:t] P[x]$

¹²⁸ meaning $\exists[x:t] P[x]$

¹²⁹ meaning $\lambda[x:t] P[x]$

¹³⁰ *(if e₁ then e₂ else e₃)*

¹³¹ i.e. $(\lambda[x:t] \sim E.[x])$

¹³² i.e. $(\lambda[x:t] \sim P[x])$

¹³³ The type of $(p[x])$ means that the untyped **Y** fixed point construction cannot be used to construct sentences for “self-referential” propositions in Direct Logic.

¹³⁴ The type of $(p[x])$ means that the untyped **Y** fixed point construction cannot be used to construct sentences for “self-referential” propositions in Direct Logic.

¹³⁵ Constructing propositions from sentences achieves much of what Russel sought to achieve in the ramified theory of types. [Russell and Whitehead 1910-1913]

¹³⁶ The following are axioms for \models :

- $\models \mathbf{True}$
- $\neg \models \mathbf{False}$
- $(\models \Phi \wedge \Psi) \Leftrightarrow (\models \Phi) \wedge (\models \Psi)$
- $(\models \Phi \vee \Psi) \Leftrightarrow (\models \Phi) \vee (\models \Psi)$
- $(\models \neg \Phi) \Leftrightarrow \neg \models \Phi$
- $(\models (\Phi \vdash \Psi)) \Leftrightarrow ((\models \Phi) \Rightarrow (\models \Psi))$