



HAL
open science

Formalizing common sense reasoning for scalable inconsistency-robust information coordination using Direct Logic™ Reasoning and the Actor Model

Carl Hewitt

► **To cite this version:**

Carl Hewitt. Formalizing common sense reasoning for scalable inconsistency-robust information coordination using Direct Logic™ Reasoning and the Actor Model. Inconsistency Robustness, 2015, 978-1-84890-159-9. hal-01148501v2

HAL Id: hal-01148501

<https://hal.science/hal-01148501v2>

Submitted on 13 Jul 2015 (v2), last revised 16 Jan 2017 (v12)

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

Formalizing common sense reasoning for scalable inconsistency-robust information coordination using Direct Logic™ Reasoning and the Actor Model

Carl Hewitt

*This article is dedicated to Stanislaw Jaskowski,
John McCarthy, Marvin Minsky and Ludwig Wittgenstein.*

Abstract

People use common sense in their interactions with large information systems. This common sense needs to be formalized so that it can be used by computer systems. Unfortunately, previous formalizations have been inadequate. For example, classical logic is not safe for use with pervasively inconsistent information. The goal is to develop a standard foundation for reasoning in large-scale Internet applications (including sense making for natural language).

Inconsistency Robust Direct Logic is a minimal fix to Classical Logic without the rule of Classical Proof by Contradiction

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

Addition of the above transforms Inconsistency Robust Direct Logic into Classical Logic. Inconsistency Robust Direct Logic makes the following contributions over previous work:

- *Direct Inference*¹
- *Direct Argumentation* (argumentation directly expressed)
- *Inconsistency-robust Natural Deduction* that doesn't require artifices such as indices (labels) on propositions or restrictions on reiteration
- Intuitive inferences hold including the following:
 - Propositional Equivalences (except absorption) including Double Negation and De Morgan
 - \vee -Elimination (Disjunctive Syllogism), *i.e.*, $\neg \Phi, (\Phi \vee \Psi) \vdash \Psi$
 - Reasoning by disjunctive cases, *i.e.*, $(\Psi \vee \Phi), (\Psi \vdash \Theta), (\Phi \vdash \Omega) \vdash \Theta \vee \Omega$
 - Contrapositive for implication *i.e.*, $\Psi \Rightarrow \Phi$ if and only if $\neg \Phi \Rightarrow \neg \Psi$
 - Soundness (a theorem can be used in a proof), *i.e.*, $\vdash ((\vdash \Psi) \Rightarrow \Psi)$
 - Inconsistency Robust Proof by Contradiction, *i.e.*, $\vdash (\Psi \Rightarrow (\Phi \wedge \neg \Phi)) \Rightarrow \neg \Psi$

A fundamental goal of Inconsistency Robust Direct Logic is to effectively reason about large amounts of pervasively inconsistent information using computer information systems.

Jaśkowski [1948] stated the following initial goal:

To find a system [for inconsistency robust inference] ... which:

- 1) *when applied to contradictory [information] would not always entail overcompleteness [i.e. infer every proposition]*
- 2) *would be rich enough for practical inference*
- 3) *would have an intuitive justification*

According to Feferman [2008]: *So far as I know, it has not been determined whether such [inconsistency robust] logics account for “sustained ordinary reasoning”, not only in everyday discourse but also in mathematics and the sciences.* Direct Logic is put forward as an improvement over classical logic with respect to Feferman’s desideratum above using the following:

- **Inconsistency Robust Direct Logic** for pervasively inconsistent theories of practiceⁱ
- **Classical Direct Logic** for use of consistent mathematical theories in inconsistency robust theories

Direct Logic is an improvement over classical logic with respect to Feferman’s desideratum above for today’s information systems that are perpetually, pervasively inconsistent. Information technology needs an all-embracing system of inconsistency-robust reasoning to support practical information coordination. Having such a system is important in computer science because computers must be able to carry out all inferences (including inferences about their own inference processes) without relying on humans

Consequently, Direct Logic is proposed as a standard to replace classical logic as a mathematical foundation for Computer Science.

Since the global state space model of computation (first formalized by Turing) is inadequate to the needs of modern large-scale Internet applications the Actor Model was developed to meet this need.

Hypothesis:ⁱⁱ **All physically possible computation can be directly implemented using Actors.**

ⁱ e.g., theories for climate modeling and for modeling the human brain

ⁱⁱ This hypothesis is an update to [Church 1936] that all physically computable functions can be implemented using the lambda calculus. It is a consequence of the Actor Model that there are some computations that *cannot* be implemented in the lambda calculus.

Using, the Actor Model, this paper proves that Logic Programs are not computationally universal in that there are computations that cannot be implemented using logical inference. Consequently the Logic Program paradigm is strictly less general than the Embedding of Knowledge paradigm.

Introduction

Beneath the surface of the world are the rules of science. But beneath them there is a far deeper set of rules: a matrix of pure mathematics, which explains the nature of the rules of science and how it is that we can understand them in the first place.

Malone [2007]

Our lives are changing: *soon we will always be online*. People use their common sense interacting with large information systems. This common sense needs to be formalized.ⁱ

Large-scale Internet software systems present the following challenges:

1. **Pervasive inconsistency is the norm** and consequently classical logic infers too much, i.e., anything and everything. Inconsistencies (e.g. that can be derived from implementations, documentation, and use cases) in large software systems are pervasive and despite enormous expense have not been eliminated.
2. **Concurrency is the norm**. Logic Programs based on the inference rules of mathematical logic are not computationally universal because the message order reception indeterminate computations of concurrent programs in open systems cannot be deduced using mathematical logic from propositions about pre-existing conditions. The fact that computation is not reducible to logical inference has important practical consequences. For example, reasoning used in information coordination cannot be implemented using logical inference [Hewitt 2008a].

This paper suggests some principles and practices formalizing common sense approaches to addressing the above issues.

Interaction *creates Reality*²

[We] cannot think of any object apart from the possibility of its connection with other things.

Wittgenstein, *Tractatus*

ⁱ Eventually, computer systems need to be able to address issues like the following:

- What will be the effects of increasing greenhouse gasses?
- What is the future of mass cyber surveillance?
- What can be done about the increasing prevalence of metabolic syndrome?

According to [Rovelli 2008]:

*a pen on my table has information because it points in this or that direction. We do not need a human being, a cat, or a computer, to make use of this notion of information.*ⁱ

Relational physics takes the following view [Laudisa and Rovelli 2008]:

- Relational physics discards the notions of absolute state of a system and absolute properties and values of its physical quantities.
- State and physical quantities refer always to the interaction, or the relation, among multiple systems.ⁱⁱ
- Nevertheless, relational physics is a complete description of reality.ⁱⁱⁱ

According to this view, **Interaction creates reality.**³

Information is a generalization of physical information in Relational Physics

Information, as used in this article, is a generalization of the physical information of Relational Physics.^{iv} Information systems participate in reality and thus are both consequence and cause. Science is a large information system that investigates and theorizes about interactions. So how does Science work?

ⁱ Rovelli added: *This [concept of information] is very weak; it does not require [consideration of] information storage, thermodynamics, complex systems, meaning, or anything of the sort. In particular:*

- Information can be lost dynamically ([correlated systems can become uncorrelated]);*
- [It does] not distinguish between correlation obtained on purpose and accidental correlation;*
- Most important: any physical system may contain information about another physical system.*

Also, Information is exchanged via physical interactions. and furthermore, It is always possible to acquire new information about a system.

ⁱⁱ *In place of the notion of state, which refers solely to the system, [use] the notion of the information that a system has about another system.*

ⁱⁱⁱ Furthermore, according to [Rovelli 2008], *quantum mechanics indicates that the notion of a universal description of the state of the world, shared by all observers, is a concept which is physically untenable, on experimental grounds. In this regard, [Feynman 1965] offered the following advice: Do not keep saying to yourself, if you can possibly avoid it, "But how can it be like that?" because you will go "down the drain," into a blind alley from which nobody has yet escaped.*

^{iv} Unlike physical information in Relational Physics [Rovelli 2008, page 10], this paper does *not* make the assumption that information is necessarily a discrete quantity or that it must be consistent.

According to [Law 2004, emphasis added]:

*... scientific routinisation, produced with immense difficulty and at immense cost, that secures the general continued stability of natural (and social) scientific reality. Elements within [this routinisation] may be overturned... But overall and most of the time, ... it is the expense [and other difficulties] of doing otherwise that allows [scientific routinisation] to achieve relative stability. So it is that a scientific reality is produced that holds together more or less.*⁴

He added that we can respond as follows:

That we refuse the distinction between the literal and the metaphorical (as various philosophers of science have noted, the literal is always 'dead' metaphor, a metaphor that is no longer seen as such). ... That we work allegorically. That we imagine coherence without consistency. [emphasis added]

The coherence envisaged by Law (above) is a dynamic interactive ongoing process among humans and other objects.

Pervasive Inconsistency is the Norm in Large Software Systems

"... find bugs faster than developers can fix them and each fix leads to another bug"

Cusumano & Selby, 1995, p. 40

The development of large software systems and the extreme dependence of our society on these systems have introduced new phenomena. These systems have pervasive inconsistencies among and within the following:⁵

- *Use cases* that express how systems can be used and tested in practice.⁶
- *Documentation* that expresses over-arching justification for systems and their technologies.⁷
- *Code* that expresses implementations of systems

Adapting a metaphor used by Popper⁸ for science, the bold structure of a large software system rises, as it were, above a swamp. It is like a building erected on piles. The piles are driven down from above into the swamp, but not down to any natural or given base; and when we cease our attempts to drive our piles into a deeper layer, it is not because we have reached bedrock. We simply pause when we are satisfied that they are firm enough to carry the structure, at least for the time being. Or perhaps we do something else more pressing. Under some piles there is no rock. Also some rock does not hold.

Different communities are responsible for constructing, evolving, justifying and maintaining documentation, use cases, and code for large, software systems. In specific cases any one consideration can trump the others.

Sometimes debates over inconsistencies among the parts can become quite heated, e.g., between vendors. ***In the long run, after difficult negotiations, in large software systems, use cases, documentation, and code all change to produce systems with new inconsistencies. However, no one knows what they are or where they are located!***

A large software system is never done [Rosenberg 2007].⁹

With respect to *detected* contradictions in large information systems, according to [Russo, Nuseibeh, and Easterbrook 2000]:

The choice of an inconsistency handling strategy depends on the context and the impact it has on other aspects of the development process. Resolving the inconsistency may be as simple as adding or deleting information from a software description. However, it often relies on resolving fundamental conflicts, or taking important design decisions. In such cases, immediate resolution is not the best option, and a number of choices are available:

- ***Ignore*** - it is sometimes the case that the effort of fixing an inconsistency is too great relative to the (low) risk that the inconsistency will have any adverse consequences. In such cases, developers may choose to ignore the existence of the inconsistency in their descriptions. Good practice dictates that such decisions should be revisited as a project progresses or as a system evolves.
- ***Defer*** - this may provide developers with more time to elicit further information to facilitate resolution or to render the inconsistency unimportant. In such cases, it is important to flag the parts of the descriptions that are affected, as development will continue while the inconsistency is tolerated.
- ***Circumvent*** - in some cases, what appears to be an inconsistency according to the consistency rules is not regarded as such by the software developers. This may be because the rule is wrong, or because the inconsistency represents an exception to the rule that had not been captured. In these cases, the inconsistency can be circumvented by modifying the rule, or by disabling it for a specific context.
- ***Ameliorate*** - it may be more cost-effective to 'improve' a description containing inconsistencies without necessarily resolving them all. This may include adding information to the description that alleviates some adverse effects of an inconsistency and/or resolves other inconsistencies as a side effect. In such cases, amelioration can be a useful inconsistency handling strategy in that it moves the development process in a 'desirable' direction in which inconsistencies and their adverse impact are reduced.

Inconsistency Robustness

*You cannot be confident about applying your calculus until you know that there are no hidden contradictions in it.*ⁱ

Turing *circa* 1930. [Wittgenstein 1933-1935]

Indeed, even at this stage, I predict a time when there will be mathematical investigations of calculi containing contradictions, and people will actually be proud of having emancipated themselves from consistency.

Wittgenstein *circa* 1930. [Wittgenstein 1933-1935]¹⁰

Inconsistency robustness is information system performance in the face of continually pervasive inconsistencies--- a shift from the previously dominant paradigms of *inconsistency denial* and *inconsistency elimination* attempting to sweep them under the rug.ⁱⁱ

In fact, inconsistencies are pervasive throughout our information infrastructure and they affect one another. Consequently, an interdisciplinary approach is needed.

Inconsistency robustness differs from previous paradigms based on belief revision, probability, and uncertainty as follows:

- *Belief revision*: Large information systems are continually, pervasively inconsistent and there is no way to revise them to attain consistency.
- *Probability and fuzzy logic*: In large information systems, there are typically several ways to calculate probability. Often the result is that the probability is both close to 0% and close to 100%!
- *Uncertainty*: Resolving uncertainty to determine truth is not realistic in large information systems.

There are many examples of inconsistency robustness in practice including the following:

- Our economy relies on large software systems that have tens of thousands of known inconsistencies (often called “bugs”) along with tens of thousands more that have yet to be pinned down even though their symptoms are sometimes obvious.

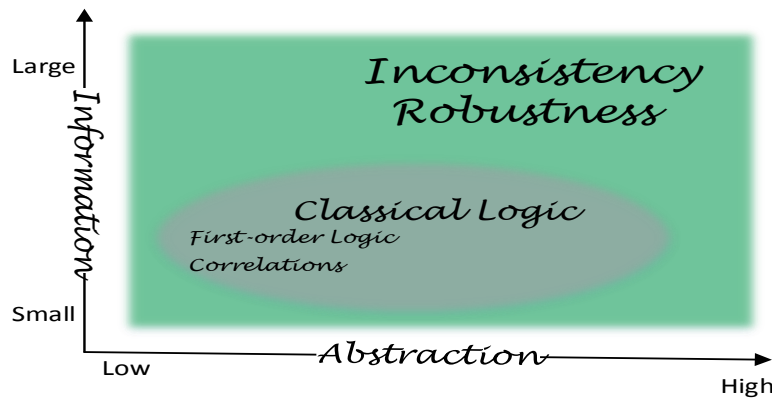
ⁱ Turing was correct that it is unsafe to use classical logic to reason about inconsistent information. Church and Turing later proved that determining whether there are hidden inconsistencies in a mathematical theory is computationally undecidable.

ⁱⁱ Inconsistency robustness builds on previous work on inconsistency tolerance, *e.g.*, [Bertossi, Hunter and Schaub 2004; Gabbay and Hunter 1991-1992; Béziau, Carnielli and Gabbay 2007].

- Physics has progressed for centuries in the face of numerous inconsistencies including the ongoing decades-long inconsistency between its two most fundamental theories (general relativity and quantum mechanics).
- Decision makers commonly ask for the case against as well as the case for proposed findings and action plans in corporations, governments, and judicial systems.

Inconsistency robustness stands to become a more central theme for computation. The basic argument is that because inconsistency is continually pervasive in large information systems, the issue of inconsistency robustness must be addressed!

A fundamental goal of Inconsistency Robustness is to effectively reason about large amounts of information at high degrees of abstraction:



Classical logic is safe only for theories for which there is strong evidence of consistency.

A little inaccuracy sometimes saves tons of explanation.
Saki in "The Square Egg"

Inconsistency robust theories can be easier to develop than classical theories because perfect absence of inconsistency is not required. In case of inconsistency, there will be some propositions that can be both proved and disproved, *i.e.*, there will be arguments both for and against the propositions.

A classic case of inconsistency occurs in the novel *Catch-22* [Heller 1961] which states that a person "*would be crazy to fly more missions and sane if he didn't, but if he was sane he had to fly them. If he flew them he was crazy and didn't have to; but if he didn't want to he was sane and had to.* Yossarian was

moved very deeply by the absolute simplicity of this clause of Catch-22 and let out a respectful whistle. 'That's some catch, that Catch-22,' he observed."

Consider the follow formalization of the above in classical logic:ⁱ

Policy₁[x] \equiv Sane[x] \Rightarrow Obligated[x, Fly]

Policy₂[x] \equiv Obligated[x, Fly] \Rightarrow Fly[x]

Policy₃[x] \equiv Crazy[x] \Rightarrow \neg Obligated[x, Fly]

Observe₁[x] \equiv \neg Obligated[x, Fly] \wedge \neg Fly[x] \Rightarrow Sane[x]

Observe₂[x] \equiv Fly[x] \Rightarrow Crazy[x]

Observe₃[x] \equiv Sane[x] \wedge \neg Obligated[x, Fly] \Rightarrow \neg Fly[x]

Observe₄ \equiv Sane[Yossarian]

In addition, there is the following background material:

Background₂ \equiv \neg Obligated[Moon, Fly]

Using classical logic, the following rather surprising conclusion can be inferred:

Fly[Moon]

i.e., the moon flies an aircraft!

Classical logic is not safe for theories not know to be consistent.ⁱⁱ

Inconsistency robustness facilitates formalization

Inconsistency Robust Direct logic facilitates common sense reasoning by formalizing inconsistency robust inference.ⁱⁱⁱ

ⁱ This is a very simple example of how classical logic can infer absurd conclusions from inconsistent information. More generally, classical inferences using inconsistent information can be arbitrarily convoluted and there is no practical way to test if inconsistent information has been used in a derivation.

ⁱⁱ It turns out that there is a hidden inconsistency in the theory *Catch22*:

Inference₁ \equiv \vdash_{Catch22} Fly[Yossarian]

Inference₂ \equiv \vdash_{Catch22} \neg Fly[Yossarian]

Thus there is an inconsistency in the theory *Catch22* concerning whether Yossarian flies.

ⁱⁱⁱ According to [Minsky 1974]:

The consistency that [classical] logic absolutely demands is not otherwise usually available – and probably not even desirable! – because consistent systems are likely to be too “weak”.

In Direct Logic, the above can be formulated using a very strong form of implication in Inconsistency Robust Direct Logic as follows in the theory *Catch22*:¹¹

Policy₁[x] $\equiv \vdash_{\text{Catch22}} \text{Sane}[x] \Rightarrow \text{Obligated}[x, \text{Fly}]$

Policy₂[x] $\equiv \vdash_{\text{Catch22}} \text{Obligated}[x, \text{Fly}] \Rightarrow \text{Fly}[x]$

Policy₃[x] $\equiv \vdash_{\text{Catch22}} \text{Crazy}[x] \Rightarrow \neg \text{Obligated}[x, \text{Fly}]$

Observe₁[x] $\equiv \vdash_{\text{Catch22}} \neg \text{Obligated}[x, \text{Fly}] \wedge \neg \text{Fly}[x] \Rightarrow \text{Sane}[x]$

Observe₂[x] $\equiv \vdash_{\text{Catch22}} \text{Fly}[x] \Rightarrow \text{Crazy}[x]$

Observe₃[x] $\equiv \vdash_{\text{Catch22}} \text{Sane}[x] \wedge \neg \text{Obligated}[x, \text{Fly}] \Rightarrow \neg \text{Fly}[x]$

Observe₄ $\equiv \vdash_{\text{Catch22}} \text{Sane}[\text{Yossarian}]$

Background₂ $\equiv \vdash_{\text{Catch22}} \neg \text{Obligated}[\text{Moon}, \text{Fly}]$

Unlike Classical Logic, in Direct Logic:

$\not\vdash_{\text{Catch22}} \text{Fly}[\text{Moon}]$

It turns out that the following can be inferred:¹²

$\vdash_{\text{Catch22}} \text{Fly}[\text{Yossarian}]$

$\vdash_{\text{Catch22}} \neg \text{Fly}[\text{Yossarian}]$

However, instead of being able to infer everythingⁱ, once the above contradiction been noticed, question answering can be improved using the “**but**” construct of Inconsistency Robust Direct Logic as follows:

$\vdash_{\text{Catch22}} \text{Fly}[\text{Yossarian}]$ **but** $\vdash_{\text{Catch22}} \neg \text{Fly}[\text{Yossarian}]$

$\vdash_{\text{Catch22}} \neg \text{Fly}[\text{Yossarian}]$ **but** $\vdash_{\text{Catch22}} \text{Fly}[\text{Yossarian}]$

Contradictions can facilitate Argumentation

[I] *emphasize that contradictions are not always an entirely bad thing. I think we have all found in our googling that it is often better to find contradictory information on a search topic rather than finding no information at all. I explore some of the various reasons this may arise, which include finding that there is at least active interest in the topic, appraising the credentials of the informants, counting their relative number, assessing their arguments, trying to reproduce their experimental results, discovering their authoritative sources, etc.*

[Dunn 2014]

ⁱ which is the case in classical logic from a contradiction

Using Direct Logic, various arguments can be made in *Catch22*. For example:

$$\text{Sane}[x] \vdash_{\text{Catch22}}^{\text{Argument1}} \text{Crazy}[x]$$

i.e. “The sane ones are thereby crazy because they fly.”

$$\text{Crazy}[x], \neg\text{Fly}[x] \vdash_{\text{Catch22}}^{\text{Argument2}} \text{Sane}[x]$$

i.e. “The crazy ones who don’t fly are thereby sane.”

However, neither of the above arguments is absolute because there might be arguments against the above arguments. Also, the following axiom can be added to the mix:

$$\text{Observes}[x] \equiv \vdash_{\text{Catch22}} \text{Crazy}[x] \Rightarrow \neg\text{Sane}[x]$$

Once, the above axiom is added we have:

$$\vdash_{\text{Catch22}} \text{Fly}[\text{Yossarian}] \text{ but } \vdash_{\text{Catch22}} \neg\text{Sane}[\text{Yossarian}]$$

although $\text{Sane}[\text{Yossarian}]$ is used in the argument for $\text{Fly}[\text{Yossarian}]$.

The theory *Catch22* illustrates the following points:

- *Inconsistency robustness facilitates theory development because a single inconsistency is not disastrous.*
- *Even though the theory Catch22 is inconsistent, it is not meaningless.*
- *Queries can be given sensible answers in the presence of inconsistent information.*

Inconsistent probabilities

You can use all the quantitative data you can get, but you still have to distrust it and use your own intelligence and judgment.

Alvin Toffler

it would be better to ... eschew all talk of probability in favor of talk about correlation.

N. David Mermin [1998]

Inconsistency is built into the very foundations of probability theory:¹³

- $\mathbb{P}\text{PresentMoment} \cong 0$
Because of cumulative contingencies to get here.ⁱ
- $\mathbb{P}\text{PresentMoment} \cong 1$
Because it's reality.

ⁱ For example, suppose that we have just flipped a coin a large number of times producing a long sequence of heads and tails. The exact sequence that has been produced is extremely unlikely.

The above problem is not easily fixed because of the following:

- Indeterminacies are omnipresent/
- Interdependencies¹⁴ are pervasive thereby calling to question probabilistic calculations that assume independence.

The above points about the perils of correlation were largely missed in [Anderson 2008]. which stated

“Correlation is enough.” We can stop looking for models. We can analyze the data without hypotheses about what it might show. We can throw the numbers into the biggest computing clusters the world has ever seen and let statistical algorithms find patterns where science cannot.
(emphasis added)

Of course, Anderson missed the whole point that causality is about *affecting* correlations through interaction. Statistical algorithms can always find meaningless correlations. Models (*i.e.* theories) are used to create interventions to test which correlations are causal.

Theorem. $(\Psi \vdash \Phi) \Rightarrow \mathbb{P}\Psi \leq \mathbb{P}\Phi$

Proof: Suppose $\Psi \vdash \Phi$.

$$1 \cong^i \mathbb{P}\Phi|\Psi \equiv \frac{\mathbb{P}\Phi \wedge \Psi}{\mathbb{P}\Psi}$$

$$\mathbb{P}\Psi \cong \mathbb{P}\Phi \wedge \Psi \leq \mathbb{P}\Phi$$

Thus probabilities for the theory Catch22 obey the following:

P1. $\vdash_{\text{Catch22}} \mathbb{P}\text{Sane}[x] \leq \mathbb{P}\text{Obligated}[x, \text{Fly}]$

P2. $\vdash_{\text{Catch22}} \mathbb{P}\text{Obligated}[x, \text{Fly}] \leq \mathbb{P}\text{Fly}[x]$

P3. $\vdash_{\text{Catch22}} \mathbb{P}\text{Crazy}[x] \leq \mathbb{P}\neg\text{Obligated}[x, \text{Fly}]$]

S1. $\vdash_{\text{Catch22}} \mathbb{P}\neg\text{Obligated}[x, \text{Fly}] \wedge \neg\text{Fly}[x] \leq \mathbb{P}\text{Sane}[x]$

S2. $\vdash_{\text{Catch22}} \mathbb{P}\text{Fly}[x] \leq \mathbb{P}\text{Crazy}[x]$

S3. $\vdash_{\text{Catch22}} \mathbb{P}\text{Sane}[x] \wedge \neg\text{Obligated}[x, \text{Fly}] \leq \mathbb{P}\neg\text{Fly}[x]$

S4. $\vdash_{\text{Catch22}} \mathbb{P}\text{Sane}[\text{Yossarian}] \cong 1$

Consequently, the following inferences hold

- | | |
|--|---------------------------------|
| I1. $\vdash_{\text{Catch22}} 1 \cong \mathbb{P}\text{Obligated}[\text{Yossarian}, \text{Fly}]$ | ⓐ using P1 and S4 |
| I2. $\vdash_{\text{Catch22}} 1 \cong \mathbb{P}\text{Fly}[\text{Yossarian}]$ | ⓐ using P2 and I1 |
| I3. $\vdash_{\text{Catch22}} 1 \cong \mathbb{P}\text{Crazy}[\text{Yossarian}]$ | ⓐ using S2 and I2 |
| I4. $\vdash_{\text{Catch22}} 1 \lesssim \mathbb{P}\neg\text{Obligated}[\text{Yossarian}, \text{Fly}]$ | ⓐ using P3 and I3 |
| I5. $\vdash_{\text{Catch22}} \mathbb{P}\neg\text{Fly}[\text{Yossarian}] \cong 0$ | ⓐ using I4 and S3 |
| I6. $\vdash_{\text{Catch22}} \mathbb{P}\text{Fly}[\text{Yossarian}] \cong 1$ | ⓐ reformulation of I5 |

ⁱ This conclusion is not accepted by all. See [Lewis 1976].

Thus there is an inconsistency in Catch22 in that both of the following hold in the above:

I2. $\vdash_{\text{Catch22}} \mathbb{P}\text{Fly}[\text{Yossarian}] \cong 1$

I6. $\vdash_{\text{Catch22}} \mathbb{P}\text{Fly}[\text{Yossarian}] \cong 0$

Inconsistent probabilities are potentially a much more serious problem than logical inconsistencies because they have unfortunate consequences like the following: $\vdash_{\text{Catch22}} 1 \cong 0$.¹⁵

In addition to inconsistency non-robustness, probability models are limited by the following:

- ✗ Limited expressiveness (avoidance of non-numerical reasoning)
- ✗ Limited scalability
- ✗ Fragile independence assumptions
- ✗ Markovian ahistoricity
- ✗ Bayes rule (very conservative) versus general reasoning
- ✗ Contrafactuals (contra scientific knowledge)

Nevertheless, probabilities have important uses in physics, *e.g.* quantum systems.

However, statistical reasoning is enormously important in practice including the following:

- Aggregation and Correlation
- Interpolation and Extrapolation
- Classification and Simulation

Circular information

How can inconsistencies such as the one above be understood?

Assigning truth values to propositions is an attempt to characterize whether or not a proposition holds in a theory. Of course, this cannot be done consistently if the theory is inconsistent. Likewise, assigning probabilities to propositions is an attempt to characterize the likelihood that a proposition holds in a theory. Similar to assigning truth values, assigning probabilities cannot be done consistently if the theory is inconsistent.

The process of theory development can generate circularities that are an underlying source of inconsistency:

*Mol shows that clinical diagnoses often depend on collective and statistically generated norms. What counts as a 'normal' haemoglobin level in blood is a function of measurements of a whole population. She is saying, then, that **individual diagnoses include collective norms though***

*they cannot be reduced to these (Mol and Berg 1994). At the same time, however, the collective norms depend on a sample of clinical measurements which may be influenced by assumptions about the distribution of anaemia—though it is not, of course, reducible to any individual measurement. The lesson is that **the individual is included in the collective, and the collective is included in the individual—but neither is reducible to the other.***¹⁶

Classical logic is unsafe for use with potentially inconsistent information

Irony is about contradictions that do not resolve into larger wholes even dialectically, about the tension of holding incompatible things together because all are necessary and true.

Haraway [1991]

An important limitation of classical logicⁱ for inconsistent information is that it supports the principle that from an inconsistency anything and everything can be inferred, e.g. “*The moon is made of green cheese.*”

For convenience, I have given the above principle the name IGOR¹⁷ for **Inconsistency in Garbage Out Redux**. IGOR can be formalized as follows in which a contradiction about a proposition Ω infers any proposition Θ ,ⁱⁱ i.e., $\Omega, \neg \Omega \vdash \Theta$.

Of course, IGOR cannot be part of Inconsistency Robust Direct Logic because it allows every proposition to be inferred from a contradiction.

The IGOR principle of classical logic may not seem very intuitive! So why is it included in classical logic?

- **Classical Proof by Contradiction:** $(\Psi \vdash \Phi, \neg \Phi) \Rightarrow (\vdash \neg \Psi)$, which can be justified in classical logic on the grounds that if Ψ infers a contradiction in a consistent theory then Ψ must be false. In an inconsistent theory. Classical Proof by Contradiction leads to explosion by the following derivation in classical logic by which a contradiction about P infers any proposition Θ :

$$P, \neg P \vdash \neg \Theta \vdash P, \neg P \vdash (\neg \neg \Theta) \vdash \Theta$$
- **Classical Contrapositive for Inference:** $(\Psi \vdash \Phi) \Rightarrow (\neg \Phi \vdash \neg \Psi)$, which can be justified in classical logic on the grounds that if $\Psi \vdash \Phi$, then if Φ

ⁱ A very similar limitation holds for intuitionistic logic.

ⁱⁱ Using the symbol \vdash to mean “infers in classical mathematical logic.” The symbol was first published in [Frege 1879].

is false then Ψ must be false. In an inconsistent theory. Classical Contrapositive for Inference leads to explosion by the following derivation in classical logic by which a contradiction about P (i.e., $\vdash P, \neg P$) infers any proposition Θ by the following proof:

Since $\vdash P, \neg\Theta \vdash P$ by monotonicity. Therefore $\neg P \vdash \Theta$ by Classical Contrapositive for Inference. Consequently $P, \neg P \vdash \Theta$.

- **Classical Extraneous \vee Introduction:**¹⁸ $\Psi \vdash (\Psi \vee \Phi)$, which in classical logic says that if Ψ is true then $\Psi \vee \Phi$ is true regardless of whether Φ is true.¹⁹ In an inconsistent theory, Extraneous \vee introduction leads to explosion via the following derivation in classical logic in which a contraction about P infers any proposition Θ :

$$P, \neg P \vdash (P \vee \Theta), \neg P \vdash \Theta$$

- **Classical Excluded Middle:** $\vdash (\Psi \vee \neg\Psi)$, which in classical logic says that $\Psi \vee \neg\Psi$ is true regardless of whether Ψ is true. *Excluded Middle* is the principle of Classical Logic that for every proposition X the following holds: $\text{ExcludedMiddle}[X] \equiv X \vee \neg X$. However, Excluded Middle is not suitable for inconsistency-robust logic because it is equivalentⁱ to saying that there are no inconsistencies, i.e., for every proposition X,

$$\text{Noncontradiction}[X] \equiv \neg(X \wedge \neg X)$$

Using propositional equivalences, note that

$$\text{ExcludedMiddle}[\Phi \vee \Psi] \Leftrightarrow (\Psi \vee \neg\Psi \vee \Phi) \wedge (\Phi \vee \neg\Phi \vee \Psi)$$

Consequently, $\text{ExcludedMiddle}[\Phi \vee \Psi] \Rightarrow (\Psi \vee \neg\Psi \vee \Phi)$, which means that the principle of Excluded Middle implies $\Psi \vee \neg\Psi \vee \Phi$ for all propositions Ψ and Φ . Thus the principle of Excluded Middle is not inconsistency robust because it implies every proposition Φ can be provedⁱⁱ given any contradiction Ψ . [Kao 2011]

Classical Logic is unsafe for inference using potentially inconsistent information.ⁱⁱⁱ

Direct Logic

“But if the general truths of Logic are of such a nature that when presented to the mind they at once command assent, wherein consists the difficulty of constructing the Science of Logic?” [Boole, 1853 pg. 3]

ⁱ using propositional equivalences

ⁱⁱ using \vee -Elimination, i.e., $\neg\Phi, (\Phi \vee \Psi) \vdash \Psi$

ⁱⁱⁱ Turing noted that classical logic can be used to make invalid inferences using inconsistent information “without actually going through [an explicit] contradiction.” [Diamond 1976] Furthermore, [Church 1935, Turing 1936] proved that it is computationally undecidable whether a mathematical theory of practice is inconsistent.

Direct Logic²⁰ is a framework: propositions have arguments for and against. Inference rules provide arguments that let you infer more propositions. Direct Logic is just a bookkeeping system that helps you keep track. It doesn't tell you what to do when an inconsistency is derived. But it does have the great virtue that it doesn't make the mistakes of classical logic when reasoning about inconsistent information.

The semantics of Direct Logic are based on argumentation. Arguments can be inferred for and against propositions. Furthermore, additional arguments can be inferred for and against these *arguments*, e.g., supporting and counter arguments.²¹

Direct Logic must meet the following challenges:

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

Inconsistency Robust Direct Logic is for reasoning about pervasively-inconsistent large software systems with the following goals:

- Provide a foundation for reasoning about the mutually inconsistent implementation, specifications, and use cases large software systems.
- Formalize a notion of “direct” inference for reasoning about inconsistent information
- Support “natural” deduction [Jaśkowski 1934]ⁱ inference rulesⁱⁱ
- Support the usual propositional equivalencesⁱⁱⁱ

ⁱ See discussion in [Pelletier 1999].

ⁱⁱ with the *exception* of the following:

- *Classical Proof by Contradiction* i.e., $(\Psi \vdash_{\tau} \neg\Phi, \Phi) \vdash_{\tau} \neg\Psi$
- *Extraneous \vee Introduction*, i.e., $\Psi \vdash_{\tau} (\Phi \vee \Psi)$
- *Excluded Middle*, i.e., $\vdash_{\tau} (\Phi \vee \neg\Phi)$

ⁱⁱⁱ with exception of absorption, which must be restricted to avoid IGOR

- \vee -Elimination , i.e., $\neg\Phi, (\Phi\vee\Psi) \vdash_T \Psi$
- Reasoning by disjunctive cases, i.e., $(\Psi\vee\Phi), (\Psi \vdash_T \Theta), (\Phi \vdash_T \Omega) \vdash_T \Theta\vee\Omega$
- Inconsistency Robust Proof by Contradiction, i.e., $\vdash_T (\Psi\Rightarrow(\neg\Phi\wedge\Phi)) \Rightarrow \neg\Psi$
- Support abstraction among code, documentation, and use cases of large software systems. (See discussion below.)
- Provide increased safety in reasoning using inconsistent information.ⁱ

Consequently, Inconsistency Robust Direct Logic is well suited in practice for reasoning about large software systems.ⁱⁱ

Adding just Classical Proof by Contradiction to Inconsistency Robust Direct Logic transforms it into a classical logic.

The theories of Direct Logic are “open” in the sense of open-ended schematic axiomatic systems [Feferman 2007b]. The language of a theory can include any vocabulary in which its axioms may be applied, i.e., it is not restricted to a specific vocabulary fixed in advance (or at any other time). Indeed a theory can be an open system can receive new information at any time [Hewitt 1991, Cellucci 1992].

In the argumentation lies the knowledge

You don't understand anything until you learn it more than one way.
[Minsky 2005]²²

Partly in reaction to Popperⁱⁱⁱ, Lakatos [1967, §2]) calls the view below *Euclidean*.²³

“Classical epistemology has for two thousand years modeled its ideal of a theory, whether scientific or mathematical, on its conception of Euclidean geometry. The ideal theory is a deductive system with an indubitable truth-injection at the top (a finite conjunction of axioms)—so that truth, flowing

ⁱ by comparison with classical logic

ⁱⁱ In this respect, Direct Logic differs from previous inconsistency tolerant logics, which had inference rules that made them intractable for use with large software systems.

ⁱⁱⁱ Proof by contradiction has played an important role in science (emphasized by Karl Popper [1962]) as formulated in his principle of refutation which in its most stark form is as follows:

If $\vdash_T \neg Ob$ for some observation Ob , then it can be concluded that T is refuted (in a theory called *Popper*), i.e., $\vdash_{\text{Popper}} \neg T$

See Suppe [1977] for further discussion.

down from the top through the safe truth-preserving channels of valid inferences, inundates the whole system.”

Since truth is out the window for inconsistent theories, we need a reformulation in terms of argumentation.

Direct Argumentation

Inference in a theory \mathcal{T} ($\vdash_{\mathcal{T}}$) carries chains of argument from antecedents to consequents.

Direct Argumentation means that $\vdash_{\mathcal{T}}$ in a proposition actually means inference in the theory \mathcal{T} .²⁴ For example, together $\vdash_{\mathcal{T}}\Psi$ and $\Psi \vdash_{\mathcal{T}}\Phi$ infer $\vdash_{\mathcal{T}}\Phi$, which in Inconsistency Robust Direct Logic can be expressed as follows by *Direct Argumentation*: $\Psi, (\Psi \vdash_{\mathcal{T}}\Phi) \vdash_{\mathcal{T}}\Phi$

Theory Dependence

Inference in Inconsistency Robust Direct Logic is theory dependent. For example [Latour 2010]:

*“Are these stone, clay, and wood idols true divinitiesⁱ?” [The Africans] answered “Yes!” with utmost innocence: yes, of course, otherwise we would not have made them with our own handsⁱⁱ! The Portuguese, shocked but scrupulous, not want to condemn without proof, gave the Africans one last chance: “You can’t say both that you’ve made your own [idols] and that they are true divinitiesⁱⁱⁱ; **you have to choose**: it’s either one or the other. Unless,” they went on indignantly, “you really have no brains, and you’re as oblivious to the principle of contraction^{iv} as you are to the sin of idolatry.” Stunned silence from the [Africans] who failed to see any contradiction.^v*

As stated, there is no inconsistency in either the theory Africans or the theory Portuguese. But there is an inconsistency in the join of these theories, namely, Africans+Portuguese.

In general, the theories of Inconsistency Robust Direct Logic are inconsistent and therefore propositions cannot be consistently labeled with truth values.

ⁱ $\vdash_{\text{Africans}} \text{Divine}[\text{idols}]$

ⁱⁱ $\vdash_{\text{Africans}} \text{Fabricated}[\text{idols}]$

ⁱⁱⁱ $\vdash_{\text{Portuguese}} \neg(\text{Fabricated}[\text{idols}] \wedge \text{Divine}[\text{idols}])$

^{iv} in *Africans+Portuguese*

^v in *Africans*

Information Invariance

Become a student of change. It is the only thing that will remain constant.
Anthony D'Angelo, The College Blue Book

Invarianceⁱ is a fundamental technical goal of Direct Logic.

Invariance: Principles of Direct Logic are invariant as follows:

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

Semantics of Direct Logic

The semantics of Direct Logic is the semantics of argumentation. Arguments can be made in favor of against propositions. And, in turn, arguments can be made in favor and against arguments. The notation $\vdash_T^A \Psi$ is used to express that A is an argument for Ψ in T.

The semantics of Direct Logic are grounded in the principle that every proposition that holds in a theory must have argument in its favor which can be expressed as follows:

The principle **Inferences have Arguments** says that $\vdash_T \Psi$ if and only if there is an argument A for Ψ in T, *i.e.*, $\vdash_T^A \Psi$ ⁱⁱ

For example, there is a controversy in biochemistry as to whether or not it has been shown that arsenic can support life with published arguments by Redfield²⁵ and NASA²⁶ to the following effect:

$\frac{\text{Redfield}}{\text{Biochemistry}} (\not\frac{\text{NASA}}{\text{Biochemistry}} \text{SupportsLife}[\text{Arsenic}])$

[Rovelli 2011] has commented on this general situation:

There is a widely used notion that does plenty of damage: the notion of "scientifically proven". Nearly an oxymoron. The very foundation of science is to keep the door open to doubt. Precisely because we keep

ⁱ Closely related to conservation laws in physics

ⁱⁱ There is a computational decision deterministic procedure Checker_T running in linear time such that:

$$\forall [a:\text{Argument}, s:\text{Sentence}] \rightarrow \text{Checker}_T[a, s] = \text{True} \Leftrightarrow \vdash_T^a [s]_T$$

questioning everything, especially our own premises, we are always ready to improve our knowledge. Therefore a good scientist is never 'certain'. Lack of certainty is precisely what makes conclusions more reliable than the conclusions of those who are certain: because the good scientist will be ready to shift to a different point of view if better elements of evidence, or novel arguments emerge. Therefore certainty is not only something of no use, but is in fact damaging, if we value reliability.

A fanciful example of argumentation comes from the famous story “*What the Tortoise Said to Achilles*” [Carroll 1895].

Applied to example of the Tortoise in the stony, we have

$$\frac{\text{ProofOfZ(Axiom1, Axiom2)}}{\text{Achilles}} Z^{27}$$

where

A \equiv “*Things that are equal to the same are equal to each other.*”

B \equiv “*The two sides of this Triangle are things that are equal to the same.*”

Z \equiv “*The two sides of this Triangle are equal to each other.*”

Axiom₁ \equiv \vdash A, B

Axiom₂ \equiv A, B \vdash Z

The above proposition fulfills the demand of the Tortoise that

*Whatever Logic is good enough to tell me is worth **writing down**.*

Inference in Argumentation

Scientist and engineers speak in the name of new allies that they have shaped and enrolled; representatives among other representatives, they add these unexpected resources to tip the balance of force in their favor.

Latour [1987] Second Principle

“ \vdash Elimination” (Chaining) is a fundamental principle of inference: ²⁸

\vdash Elimination (Chaining): $\Psi, (\Psi \vdash_{\mathcal{T}} \Phi) \vdash_{\mathcal{T}} \Phi$ <i>① Φ inferred in \mathcal{T} from $\vdash_{\mathcal{T}} \Psi$ and $\Psi \vdash_{\mathcal{T}} \Phi$</i>

SubArguments is another fundamental principle of inference:

\vdash Introduction (SubArguments): $(\vdash_{\mathcal{T} \wedge \Psi} \Phi) \vdash_{\mathcal{T}} (\Psi \vdash_{\mathcal{T}} \Phi)$ <i>① In \mathcal{T}, Ψ infers Φ when Φ is inferred in $\mathcal{T} \wedge \Psi$</i>

Please see the appendix “Detail of Direct Logic” for more information.

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 provably computationally enumerated:²⁹

Theorem \vdash Mathematics is Open

Proof.ⁱ Suppose to obtain a contradiction that it is possible to prove closure, *i.e.*, there is a provably computable total deterministic procedure Theorem such that it is provable that

$$\forall[\Psi:\text{Proposition}] \rightarrow (\vdash \Psi) \Leftrightarrow \exists[i:\mathbb{N}] \rightarrow \text{Theorem}.[i]=p$$

As a consequence of the above, there is a provably total procedure ProvableComputableTotal that enumerates the provably total computable procedures that can be used in the implementation of the following procedure: $\text{Diagonal}[i] \equiv (\text{ProvableComputableTotal}[i])[i]+1$

However,

- $\text{ProvableComputableTotal}[\text{Diagonal}]$ because Diagonal is implemented using provably computable total procedures.
- $\neg \text{ProvableComputableTotal}[\text{Diagonal}]$ because Diagonal is a provably computable total procedure that differs from every other provably computable total procedure.

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

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

ⁱⁱ See section immediately below.

Contributions of Direct Logic

Inconsistency Robust Direct Logic aims to be a minimal fix to classical logic to meet the needs of information coordination. (Addition of just the rule of Classical Proof by Contradiction by Inference, transforms Direct Logic into Classical Logic.) Direct Logic makes the following contributions over previous work:

- *Direct Inference*³⁰
- *Direct Argumentation* (inference directly expressed)
- *Inconsistency Robustness*
- *Inconsistency-robust Natural Deduction*³¹
- Intuitive inferences hold including the following:
 - *Propositional equivalences*ⁱ
 - *Reasoning by disjunctive cases*, i.e.,
 $(\Psi \vee \Phi), (\Psi \vdash_{\tau} \Theta), (\Phi \vdash_{\tau} \Omega) \vdash_{\tau} \Theta \vee \Omega$
 - *\vee -Elimination*, i.e., $\neg\Phi, (\Phi \vee \Psi) \vdash_{\tau} \Psi$
 - *Contrapositive for implication*: A proposition implies another if and only if negation of the latter implies negation of the former, i.e., $\Psi \Rightarrow \Phi$ if and only if $\neg\Phi \Rightarrow \neg\Psi$
 - *Soundness*: A theorem can be used in a proof, i.e.,
 $\vdash_{\tau} ((\vdash_{\tau} \Psi) \Rightarrow \Psi)$
 - *Inconsistency Robust Proof by Contradiction*: A hypothesis can be refuted by showing that it implies a contradiction, i.e.,
 $\vdash_{\tau} (\Phi \Rightarrow (\Psi \wedge \neg\Psi)) \Rightarrow \neg\Phi$

Actor Model of Computation³²

The distinction between past, present and future is only a stubbornly persistent illusion.

Einstein

Concurrency has now become the norm. However nondeterminism came first. See [Hewitt 2010b] for a history of models of nondeterministic computation.

What is Computation?

Any problem in computer science can be solved by introducing another level of abstraction.

paraphrase of Alan Perlis

Turing's model of computation was intensely psychological.³³ He proposed the thesis that it included all of purely mechanical computation.³⁴

ⁱ except absorption

Gödel declared that

It is “*absolutely impossible that anybody who understands the question [What is computation?] and knows Turing’s definition should decide for a different concept.*”³⁵

By contrast, in the Actor model [Hewitt, Bishop and Steiger 1973; Hewitt 2010b], computation is conceived as distributed in space where computational devices called Actors communicate asynchronously using addresses of Actors and the entire computation is not in any well-defined state. The behavior of an Actor is defined when it receives a message and at other times may be indeterminate.

Axioms of locality including *Organizational* and *Operational* hold as follows:

- *Organization*: The local storage of an Actor can include *addresses* only
 1. that were provided when it was created or of Actors that it has created
 2. that have been received in messages
- *Operation*: In response to a message received, an Actor can
 1. create more Actors
 2. send messagesⁱ to addresses in the following:
 - the message it has just received
 - its local storage
 3. for an exclusiveⁱⁱ Actor, designate how to process the next message receivedⁱⁱⁱ

The Actor Model differs from its predecessors and most current models of computation in that the Actor model assumes the following:

- Concurrent execution in processing a message.
- The following are *not* required by an Actor: a thread, a mailbox, a message queue, its own operating system process, *etc.*
- Message passing has the same overhead as looping and procedure calling.

Configurations versus Global State Spaces

Computations are represented differently in Turing Machines and Actors:

1. *Turing Machine*: a computation can be represented as a global state that determines all information about the computation. It can be nondeterministic as to which will be the next global state, *e.g.*, in

ⁱ Likewise the messages sent can contain addresses only

1. that were provided when the Actor was created
2. that have been received in messages that are for Actors created here

ⁱⁱ An *exclusive* Actor can perform at most one activity at a time.

ⁱⁱⁱ An Actor that will never update its local storage can be freely replicated and cached.

simulations where the global state can transition nondeterministically to the next state as a global clock advances in time, e.g., Simula [Dahl and Nygaard 1967].³⁶

1. *Actors*: a computation can be represented as a configuration. Information about a configuration can be indeterminate.¹

Functions defined by lambda expressions [Church 1941] are special case Actors that never change.

That Actors which behave like mathematical functions exactly correspond with those definable in the lambda calculus provides an intuitive justification for the rules of the lambda calculus:

- *Lambda identifiers*: each identifier is bound to the address of an Actor. The rules for free and bound identifiers correspond to the Actor rules for addresses.
- *Beta reduction*: each beta reduction corresponds to an Actor receiving a message. Instead of performing substitution, an Actor receives addresses of its arguments.

The lambda calculus can be implemented in ActorScript as follows:

```

Actor Identifier<aType>[]
  implements Expression<aType> using
    anEnvironment.lookup[[:Expression<aType>]]
    // lookup this identifier in anEnvironment

Actor ProcedureCall<aType, AnotherType>
  [operator:([aType]→ anotherType), operand:aType]
  implements Expression<anotherType> using
    eval[anEnvironment]→
      (operator.eval[anEnvironment]).[operand.eval[environment]]

Actor Lambda<aType, AnotherType>
  [anIdentifier:Identifier<aType>, body:anotherType]
  implements Expression<[aType]→ anotherType> using
    eval[anEnvironment]→
      [anArgument:aType]→
        body.eval[anEnvironment.bind[anIdentifier, anEnvironment]]
        // create a new environment with anIdentifier
        // bound to anArgument

```

¹ For example, there can be messages in transit that will be delivered at some indefinite time.

Note that in the above:

- All operations are local.
- The definition is modular in that each lambda calculus programming language construct is an Actor.
- The definition is easily extensible since it is easy to add additional programming language constructs.
- The definition is easily operationalized into efficient concurrent implementations.
- The definition easily fits into more general concurrent computational frameworks for many-core and distributed computation.

However, there are *nondeterministic* computable functions on integers that cannot be implemented using the nondeterministic lambda calculus. Furthermore, the lambda calculus can be very inefficient as illustrate by the theorem below:

Theorem: In systems of practiceⁱ, simulating an Actor system using a the parallel lambda calculus (*i.e.* using purely functional programming) can be exponentially slower.

The lambda calculus can express parallelism but not general concurrency (see discussion below).

Actors generalize Turing Machines

Actor systems can perform computations that are impossible by Turing Machines as illustrated by the following example:

There is a bound on the size of integer that can be computed by an *always halting* nondeterministic Turing Machine starting on a blank tape.³⁷

Plotkin [1976] gave an informal proof as follows:³⁸

Now the set of initial segments of execution sequences of a given nondeterministic program P, starting from a given state, will form a tree. The branching points will correspond to the choice points in the program. Since there are always only finitely many alternatives at each choice point, the branching factor of the tree is always finite.³⁹ That is, the tree is finitary. Now König's lemma says that if every branch of a finitary tree is finite, then so is the tree itself. In the present case this means that if every execution sequence of P terminates, then there are only finitely many execution sequences. So if an output set of P is infinite, it must contain a nonterminating computation.⁴⁰

ⁱ Examples include climate models and medical diagnosis and treatment systems for cancer. A software system of practice typically has tens of millions of lines of code.

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

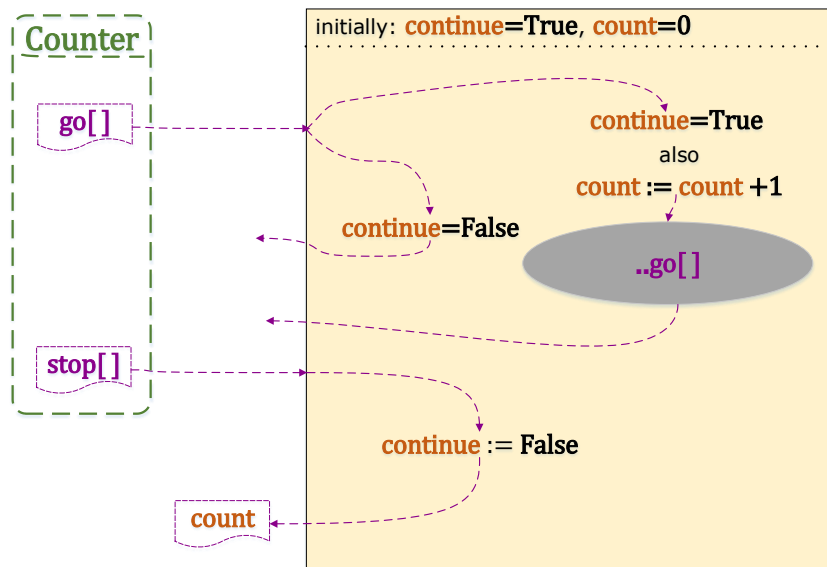
The above Actor system can be implemented as follows using ActorScript™:

Unbounded ≡

```

start[] → // a start message is implemented by
Let aCounter ← SimpleCounter.[]. // let aCounter be a new Counter
  ① aCounter.go[], // send aCounter a go message and concurrently
  ② aCounter.stop[]
  // return the value of sending aCounter a stop message

```



```

Actor SimpleCounter[]
  count := 0 // the variable count is initially 0
  continue := True // the variable continue is initially True
  implements Counter using
  stop[] →
    count // return count
    afterward continue := false
    // continue is false for the next message received
  go[] → continue
    True: // if continue is True,
      Hole ..go[] // send go[] to this counter after
      after count := count+1 // incrementing count
    False: Void // if continue is False, return Void

```

By the semantics of the Actor model of computation [Clinger 1981; Hewitt 2006], sending Unbounded a **start** message will result in sending an integer of unbounded size to the return address that was received with the **start** message.

The nondeterministic procedure Unbounded above can be axiomatized as follows:

$$\begin{aligned} & \forall [n:\text{Integer}] \rightarrow \\ & \quad \exists [aRequest:\text{Request}, anInteger:\text{Integer}] \rightarrow \\ & \quad \text{Unbounded sent}_{aRequest} \text{ start}[] \\ & \quad \Rightarrow \text{Sent}_{\text{Response}_{aRequest}} \text{ returned}[anInteger] \wedge anInteger > n \end{aligned}$$

However, the above axiom does *not* compute any actual output! Instead the above axiom simply asserts the *existence* of unbounded outputs for **start** messages.

Theorem. There are *nondeterministic* computable functions on integers that cannot be implemented by a nondeterministic Turing machine.

Proof. The above Actor system implements a nondeterministic functionⁱ that cannot be implemented by a nondeterministic Turing machine.

The following arguments support unbounded nondeterminism in the Actor model [Hewitt 1985, 2006]:

- 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.

Reception order indeterminacy

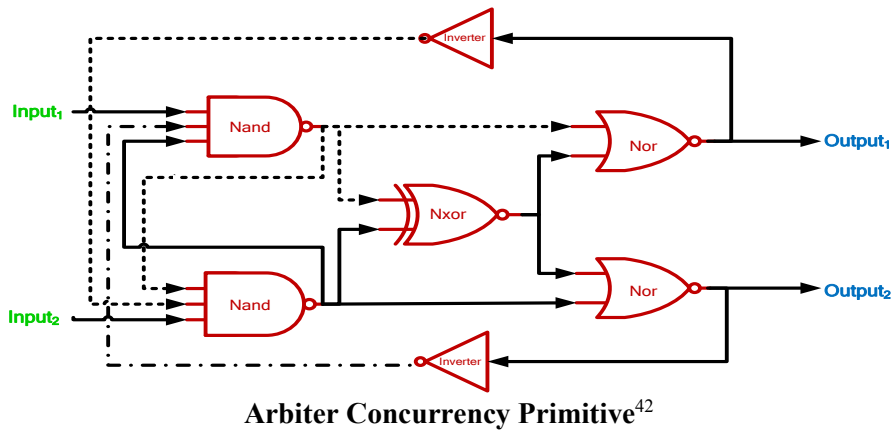
Hewitt and Agha [1991] and other published work argued that mathematical models of concurrency did not determine particular concurrent computations as follows: The Actor Modelⁱⁱ makes use of arbitration for implementing the order in which Actors process message. Since these orders are in general

ⁱ with graph { **start**[] \rightsquigarrow 0, **start**[] \rightsquigarrow 1, **start**[] \rightsquigarrow 2, ... }

ⁱⁱ Actors are the universal conceptual primitives of concurrent computation.

indeterminate, they cannot be deduced from prior information by mathematical logic alone. Therefore mathematical logic cannot implement concurrent computation in open systems.

In concrete terms for Actor systems, typically we cannot observe the details by which the order in which an Actor processes messages has been determined. Attempting to do so affects the results. Instead of observing the internals of arbitration processes of Actor computations, we await outcomes.⁴¹ Indeterminacy in arbiters produces indeterminacy in Actors.ⁱ



The reason that we await outcomes is that we have no realistic alternative.

Actor Physics

The Actor model makes use of two fundamental orders on events [Baker and Hewitt 1977; Clinger 1981, Hewitt 2006]:

1. The *activation order* (\rightsquigarrow) is a fundamental order that models one event activating another (there is energy flow from an event to an event which it activates). The activation order is discrete:

$$\forall [e_1, e_2 \in \text{Events}] \rightarrow \text{Finite}[\{e \in \text{Events} \mid e_1 \rightsquigarrow e \rightsquigarrow e_2\}]$$

There are two kinds of events involved in the activation order: reception and transmission. Reception events can activate transmission events and transmission events can activate reception events.

2. The *reception order* of an exclusive Actor \mathbf{x} ($\overset{\mathbf{x}}{\Rightarrow}$) models the (total) order of events in which a message is received at \mathbf{x} . The reception order of each \mathbf{x} is discrete:

$$\forall [r_1, r_2 \in \text{ReceptionEvents}_{\mathbf{x}}] \rightarrow \text{Finite}[\{r \in \text{ReceptionEvents}_{\mathbf{x}} \mid r_1 \overset{\mathbf{x}}{\Rightarrow} r \overset{\mathbf{x}}{\Rightarrow} r_2\}]$$

ⁱ dashes are used solely to delineate crossing wires

The *combined order* (denoted by \rightsquigarrow) is defined to be the transitive closure of the activation order and the reception orders of all Actors. So the following question arose in the early history of the Actor model: “*Is the combined order discrete?*” Discreteness of the combined order captures an important intuition about computation because it rules out counterintuitive computations in which an infinite number of computational events occur between two events (*à la* Zeno).

Hewitt conjectured that the discreteness of the activation order together with the discreteness of all reception orders implies that the combined order is discrete. Surprisingly [Clinger 1981; later generalized in Hewitt 2006] answered the question in the negative by giving a counterexample.

The counterexample is remarkable in that it violates the compactness theorem for 1st order logic:

Any finite set of propositions is consistent (the activation order and all reception orders are discrete) and represents a potentially physically realizable situation. But there is an infinite set of propositions that is inconsistent with the discreteness of the combined order and does not represent a physically realizable situation.

The counterexample is not a problem for Direct Logic because the compactness theorem does not hold.

The resolution of the problem is to take discreteness of the combined order as an axiom of the Actor model:

$$\forall[e_1, e_2 \in \text{Events}] \rightarrow \text{Finite}[\{e \in \text{Events} \mid e_1 \rightsquigarrow e \rightsquigarrow e_2\}]$$

Computational Representation Theorem

*a philosophical shift in which knowledge is no longer treated primarily as referential, as a set of statements **about** reality, but as a practice that interferes with other practices. It therefore participates **in** reality.*

Annemarie Mol [2002]

What does the mathematical theory of Actors have to say about the relationship between logic and computation? A closed system is defined to be one which does not communicate with the outside. Actor model theory provides the means to characterize all the possible computations of a closed system in terms of the Computational Representation Theorem [Clinger 1982; Hewitt 2006]:⁴³

The denotation Denote_S of a closed system S represents all the possible behaviors of S as $\text{Denote}_S = \text{limit } \text{Progression}_S^i$ where Progression_S takes a set of partial behaviors \vec{t}^i to their next stage, i.e., $\text{Progression}_S \vec{t}^i \rightarrow \vec{t}^{i+1}$

ⁱ read as “*can evolve to*”

In this way, S can be mathematically characterized in terms of all its possible behaviors (including those involving unbounded nondeterminism).ⁱ

The denotations form the basis of constructively checking programs against all their possible executions,ⁱⁱ

A consequence of the Computational Representation system is that there are uncountably many different Actors.

For example, `CreateReal.[]` can produce any real numberⁱⁱⁱ between 0 and 1 where

$\text{CreateReal}.\text{[]} \equiv [(0 \text{ either } 1), \forall \text{Postpone CreateReal}.\text{[]}]$

where

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

The upshot is that **concurrent systems can be represented and characterized by logical deduction but cannot be implemented.**

Thus, the following problem arose:

How can programming languages be rigorously defined since the proposal by Scott and Strachey [1971] to define them in terms lambda calculus failed because the lambda calculus cannot implement concurrency?

One solution is to develop a concurrent interpreter using **eval** messages in which **eval[anEnvironment]** is a message that can be sent to an expression to cause it be evaluated using the environment `anEnvironment`. Using such messages, modular meta-circular definitions can be concisely expressed in the Actor model for universal concurrent programming languages [Hewitt 2010a].

ⁱ There are no messages in transit in `Denotes`

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

ⁱⁱⁱ using binary representation. See [Feferman 2012] for more on computation over the reals.

Computation is not subsumed by logical deduction

The gauntlet was officially thrown in *The Challenge of Open Systems* [Hewitt 1985] to which [Kowalski 1988b] replied in *Logic-Based Open Systems*. [Hewitt and Agha 1988] followed up in the context of the Japanese Fifth Generation Project.

Kowalski claims that “*computation could be subsumed by deduction*”ⁱ His claim has been valuable in that it has motivated further research to characterize exactly which computations could be performed by Logic Programs. *However, contrary to Kowalski, computation in general is not subsumed by deduction.*

Bounded Nondeterminism of Direct Logic

Since it includes the nondeterministic λ calculus, direct inference, and categorical induction in addition to its other inference capabilities, Direct Logic is a very powerful foundation for Logic Program languages.

But there is no Direct Logic expression that is equivalent to sending Unbounded a **start** message for the following reason:

An expression ε will be said to always converge (written as `AlwaysConverges[ε]`) if and only if every reduction path terminates. *I.e.*, there is no function f such that $f[0] = \varepsilon$ and $\forall [i:\mathbb{N}] \rightarrow [f[i]] \rightarrow [f[i+1]]$ where the symbol \rightarrow is used for reduction (see the appendix of this paper on classical mathematics in Direct Logic). For example, $\neg \text{AlwaysConverges}[(x \rightarrow (0 \text{ either } x.[x]))].[x \rightarrow (0 \text{ either } x.[x])]$ ⁱⁱ because there is a nonterminating path.

Theorem: Bounded Nondeterminism of Direct Logic. If an expression in Direct Logic always converges, then there is a bound Bound_ε on the number to which it can converge. *I.e.*,

$$\forall [i:\mathbb{N}] \rightarrow (\varepsilon \text{ AlwaysConvergesTo } n) \Rightarrow i \leq \text{Bound}_\varepsilon$$

Consequently there is no Direct Logic program equivalent to sending Unbounded a **start** message because it has unbounded nondeterminism whereas every Direct Logic program has bounded nondeterminism.

ⁱ In fact, [Kowalski 1980] forcefully stated:

There is only one language suitable for representing information -- whether declarative or procedural -- and that is first-order predicate logic. There is only one intelligent way to process information -- and that is by applying deductive inference methods.

ⁱⁱ Note that there are two expressions (separated by “**either**”) in the bodies which provides for nondeterminism.

In this way, we have proved that the Procedural Embedding of Knowledge paradigm is strictly more general than the Logic Program paradigm.

Computational Undecidability

Some questions cannot be uniformly answered computationally.

The halting problem is to computationally decide whether a program halts on a given inputⁱ *i.e.*, there is a total computational deterministic predicate Halt such that the following 3 properties hold for any program p and input x:

1. $\text{Halt}_\bullet[p, x] \rightarrow_1 \text{True} \Leftrightarrow \text{Converges}[(\lfloor p \rfloor_\bullet[x])]$
2. $\text{Halt}_\bullet[p, x] \rightarrow_1 \text{False} \Leftrightarrow \neg \text{Converges}[(\lfloor p \rfloor_\bullet[x])]$
3. $\text{Halt}_\bullet[p, x] \rightarrow_1 \text{True} \vee \text{Halt}_\bullet[p, x] \rightarrow_1 \text{False}$

[Church 1935 and later Turing 1936] published proofs that the halting problem is computationally undecidable for computable deterministic procedures.⁴⁴ In other words, there is no such procedure Halt for computable procedures.

Theorem: $\vdash \neg \text{ComputationallyDecidable}[\text{Halt}]^{\text{ii}}$

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

The following rules are fundamental to classical mathematics:

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

Theorem:⁴⁵ Mathematics self proves its own consistency.

Formal Proof. By definition,

$\neg \text{Consistent} \Leftrightarrow \exists[\Psi:\text{Proposition}] \rightarrow \vdash(\Psi \wedge \neg\Psi)$. By the rule of Existential Elimination, there is some proposition Ψ_0 such that $\neg \text{Consistent} \Rightarrow \vdash(\Psi_0 \wedge \neg\Psi_0)$ which by the rule of Soundness and transitivity of implication means $\neg \text{Consistent} \Rightarrow (\Psi_0 \wedge \neg\Psi_0)$.

ⁱ Adapted from [Church 1936]. Normal forms were discovered for the lambda calculus, which is the way that they “halt.” [Church 1936] proved the halting problem computationally undecidable. Having done considerable work, Turing was disappointed to learn of Church’s publication. The month after Church’s article was published, [Turing 1936] was hurriedly submitted for publication.

ⁱⁱ The fact that the halting problem is computationally undecidable does not mean that proving that programs halt cannot be done in practice [Cook, Podelski, and Rybalchenko 2006].

Substituting for Φ and Θ , in the rule for Proof by Contradiction, we have $(\neg\text{Consistent} \Rightarrow (\Psi_0 \wedge \neg\Psi_0)) \vdash \text{Consistent}$. Thus, $\vdash \text{Consistent}$.

1) $\neg\text{Consistent}$ // hypothesis to derive a contradiction just in this subargument	
2) $\exists[\Psi:\text{Proposition}] \rightarrow \vdash(\Psi \wedge \neg\Psi)$ // definition of inconsistency using 1)	
3) $\vdash(\Psi_0 \wedge \neg\Psi_0)$ // rule of Existential Elimination using 2)	
4) $\Psi_0 \wedge \neg\Psi_0$ // rule of Soundness using 3)	
$\vdash \text{Consistent}$	// rule of Proof by Contradiction using 1) and 4)

Natural Deductionⁱ Proof of Consistency of Mathematics

Please note the following points:

- The above argument formally mathematically proves that mathematics is consistent and that **it is not a premise of the theorem that mathematics is consistent.**⁴⁶
- Classical mathematics was designed for consistent axioms and consequently the rules of classical mathematics can be used to prove consistency regardless of other axioms.⁴⁷

The above proof means that “Mathematics is consistent” is a theorem in Classical Direct Logic. This means that the usefulness of Classical Direct Logic depends crucially on the consistency of Mathematics.⁴⁸ Good evidence for the consistency of Mathematics comes from the way that Classical Direct Logic avoids the known paradoxes. Humans have spent millennia devising paradoxes.

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

Long ago, Wittgenstein showed that contradiction in mathematics results from the kind of “self-referential”ⁱ sentence that Gödel used in his proof. However, using a typed notation for mathematical sentences, it can be proved that the kind “self-referential” sentence that Gödel used in his proof cannot be constructed because required fixed points do not exist. In this way, consistency of mathematics is preserved without giving up power.

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

Completeness versus Inferential Undecidability

“In mathematics, there is no *ignorabimus*.”

Hilbert, 1902

A mathematical theory is an extension of mathematics whose proofs are computationally enumerable. For example, group theory is obtained by adding the axioms of groups along with the provision that theorems are computationally enumerable.

By definition, if T is a mathematical theory, there is a total deterministic procedure Proof_T such that:

$$\vdash_T^p \psi \Leftrightarrow \exists [i:\mathbb{N}] \rightarrow \text{Proof}_T [i]=p$$

Theorem: If \mathcal{T} is a consistent mathematical theory, there is a proposition $\Psi_{\text{ChurchTuring}}$, such that both of the following hold:ⁱ

- $\vdash_{\mathcal{T}} \Psi_{\text{ChurchTuring}}$
- $\vdash_{\mathcal{T}} \neg \Psi_{\text{ChurchTuring}}$

Note the following important ingredients for the proof of inferential undecidabilityⁱⁱ of mathematical theories:

- Closure (computational enumerability) of the theorems of a mathematical theory to carry through the proof.
- Consistency (nontriviality) to prevent everything from being provable.

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

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

ⁱ Otherwise, provability in classical logic would be computationally decidable because

$$\forall [p:\text{Program}, x:\mathbb{N}] \rightarrow (\text{Halt}[p, x] \Leftrightarrow \vdash_{\mathcal{T}} \text{Halt}[p, x])$$

where $\text{Halt}[p, x]$ if and only if program p halts on input x . If such a $\Psi_{\text{ChurchTuring}}$ did not exist, then provability could be decided by a computable procedure $\text{Decide}_{\mathcal{T}}: [\text{Sentence}] \rightarrow \text{Boolean}$ enumerating theorems of \mathcal{T} until the proposition in question or its negation is encountered:

$$\text{Decide}_{\mathcal{T}}[s] \rightarrow \text{True} \Leftrightarrow (\vdash_{\mathcal{T}}[s]) \vee \vdash_{\mathcal{T}} \neg[s]$$

Of course, $\text{Decide}_{\mathcal{T}}$ is a partial procedure and does not always converge.

ⁱⁱ sometimes called “incompleteness”

ⁱⁱⁱ related to conservation laws in physics

^{iv} *E.g.* inconsistent information does not infer nonsense.

Note that that a closed mathematical theory T is inferentially undecidable with respect to $\Psi_{\text{ChurchTuring}}$ does not mean “incompleteness” with respect to the information that can be inferred because

$$\vdash (\not\vdash_T \Psi_{\text{ChurchTuring}}), (\not\vdash_T \neg \Psi_{\text{ChurchTuring}}).^i$$

Information Coordination

Technology now at hand can coordinate all kinds of digital information for individuals, groups, and organizations so their information usefully links together.⁴⁹ Information coordination needs to make use of the following information system principles:

- **Persistence.** *Information is collected and indexed.*
- **Concurrency:** *Work proceeds interactively and concurrently, overlapping in time.*
- **Quasi-commutativity:** *Information can be used regardless of whether it initiates new work or become relevant to ongoing work.*
- **Sponsorship:** *Sponsors provide resources for computation, i.e., processing, storage, and communications.*
- **Pluralism:** *Information is heterogeneous, overlapping and often inconsistent.*
- **Provenance:** *The provenance of information is carefully tracked and recorded.*
- **Lossless :** *Once a system has some information, then it has it thereafter.*

Opposition of Philosophers

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]

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]

ⁱ by construction

A number of philosophers have opposed the results in this paper:

- Some would like to stick with just classical logic and not consider inconsistency robustness.⁵⁰
- Some would like to stick with the first-order theories and not consider direct inference.
- Some would like to stick with just Logic Programs (*e.g.* nondeterministic Turing Machines, λ -calculus, *etc.*) and not consider concurrency.

*And some would like to have nothing to do with any of the above!*⁵¹ However, the results in this paper (and the driving technological and economic forces behind them) tend to push towards inconsistency robustness, direct inference, and concurrency. [Hewitt 2008a]

Philosophers are now challenged as to whether they agree that

- *Inconsistency is the norm.*
- *Direct inference is the norm.*
- *Logic Programs are **not** computationally universal.*

Scalable Information Coordination

Information coordination works by making connections including examples like the following:

- A statistical connection between “being in a traffic jam” and “driving in downtown Trenton between 5PM and 6PM on a weekday.”
- A terminological connection between “MSR” and “Microsoft Research.”
- A causal connection between “joining a group” and “being a member of the group.”
- A syntactic connection between “a pin dropped” and “a dropped pin.”
- A biological connection between “a dolphin” and “a mammal”.
- A demographic connection between “undocumented residents of California” and “7% of the population of California.”
- A geographical connection between “Leeds” and “England.”
- A temporal connection between “turning on a computer” and “joining an on-line discussion.”

By making these connections, iInfo™ information coordination offers tremendous value for individuals, families, groups, and organizations in making more effective use of information technology.

In practice coordinated information is invariably inconsistent.⁵² Therefore iInfo must be able to make connections even in the face of inconsistency.⁵³ The business of iInfo is not to make difficult decisions like deciding the ultimate truth or probability of propositions. Instead it provides means for

processing information and carefully recording its provenance including arguments (including arguments about arguments) for and against propositions.

Work to be done

The best way to predict the future is to invent it. Alan Kay

There is much work to be done including the following:

Invariance

Invariance should be precisely formulated and proved. This bears on the issue of how it can be known that all the principles of Direct Logic have been discovered.

Consistency

The following conjectures for Direct Logic need to be convincingly proved:

- Consistency of Inconsistency Robust Direct Logicⁱ relative to the consistency of classical mathematics. In this regard Direct Logic is consonant with Bourbaki:

*Absence of contradiction, in mathematics as a whole or in any given branch of it, ... appears as an empirical fact, rather than as a metaphysical principle. The more a given branch has been developed, the less likely it becomes that contradictions may be met with in its farther development.*ⁱⁱ

Thus the long historical failure to find an explosion in the methods used by Direct Logic can be considered to be strong evidence of its nontriviality.

- Constructive proof of consistency of Classical Direct Logic

Inconsistency Robustness

Inconsistency robustness of theories of Direct Logic needs to be formally defined and proved. Church remarked as follows concerning a *Foundation of Logic* that he was developing:

Our present project is to develop the consequences of the foregoing set of postulates until a contradiction is obtained from them, or until the development has been carried so far consistently as to make it empirically probable that no contradiction can be obtained from them. And in this connection it is to be remembered that just such empirical evidence, although admittedly inconclusive, is the only existing evidence

ⁱ *i.e.* consistency of \vdash

ⁱⁱ [André Weil 1949] speaking as a representative of Bourbaki

of the freedom from contradiction of any system of mathematical logic which has a claim to adequacy. [Church 1933]ⁱ

Direct Logic is in a similar position except that the task is to demonstrate inconsistency robustness of inconsistent theories. This means that the exact boundaries of Inconsistency Robust Direct Logic as a minimal fix to classical logic need to be established.

Argumentation

Argumentation is fundamental to inconsistency robustness.

- Further work is need on fundamental principles of argumentation for large-scale information coordination. See [Hewitt 2008a, 2008b].
- Tooling for Direct Logic needs to be developed to support large software systems. See [Hewitt 2008a].

Inferential Explosion

Inconsistencies such as the one about whether Yossarian flies are relatively *benign* in the sense that they lack significant consequences to software engineering. Other propositions (such as $\vdash_{\mathcal{T}} 1=0$ in a theory \mathcal{T}) are more *malignant* because they can be used to infer that all integers are equal to 0 using mathematical induction. To address malignant propositions, deeper investigations of argumentation using must be undertaken in which the provenance of information will play a central role. See [Hewitt 2008a].

Robustness, Soundness, and Coherence

Fundamental concepts such as *robustness*, *soundness*, and *coherence* need to be rigorously characterized and further developed. Inconsistency-robust reasoning beyond the inference that can be accomplished in Direct Logic needs to be developed, e.g., analogy, metaphor, discourse, debate, and collaboration.

Evolution of Mathematics

In the relation between mathematics and computing science, the latter has been far many years at the receiving end, and I have often asked myself if, when, and how computing would ever be able to repay the debt. [Dijkstra 1986]

We argue that mathematics will become more like programming. [Asperti, Geuvers and Natrajan 2009]

ⁱ The difference between the time that Church wrote the above and today is that the standards for adequacy have gone up dramatically. Direct Logic must be adequate to the needs of reasoning about large software systems.

Mathematical foundations are thought to be consistent by an overwhelming consensus of working professional mathematicians, e.g., mathematical theories of real numbers, integers, *etc.*

In practice, mathematical theories that are thought to be consistency by an overwhelming consensus of working mathematicians play an important supporting role for inconsistency-robust theories, *e.g.*, theories of the Liver, Diabetes, Human Behavior, *etc.*

Conclusion

“The problem is that today some knowledge still feels too dangerous because our times are not so different to Cantor or Boltzmann or Gödel’s time. We too feel things we thought were solid being challenged; feel our certainties slipping away. And so, as then, we still desperately want to cling to a belief in certainty. It makes us feel safe. ... Are we grown up enough to live with uncertainties or will we repeat the mistakes of the twentieth century and pledge blind allegiance to another certainty?”

Malone [2007]

Inconsistency robustness builds on the following principles:

- We know only a little, but it affects us *enormously*ⁱ
- At any point in time, much is wrongⁱⁱ with the consensus of leading scientists but it is not known how or which parts.
- Science is never certain; it is continually (re-)made

Software engineers for large software systems often have good arguments for some proposition P and also good arguments for its negation of P. So what do large software manufacturers do? If the problem is serious, they bring it before a committee of stakeholders to try and sort it out. In many particularly difficult cases the resulting decision has been to simply live with the problem for an indefinite period. Consequently, large software systems are shipped to customers with thousands of known inconsistencies of varying severity where

- Even relatively simple subsystems can be subtly inconsistent.
- There is no practical way to test for inconsistency.
- Even though a system is inconsistent, it is not meaningless.

Inconsistency Robust Direct Logic is a minimal fix to Classical Logic without the rule of Classical Proof by Contradictionⁱⁱⁱ, *the addition of which transforms*

ⁱ for better or worse

ⁱⁱ *e.g.*, misleading, inconsistent, wrong-headed, ambiguous, contra best-practices, *etc.*

ⁱⁱⁱ *i.e.*, $(\Psi \vdash (\Phi \wedge \neg \Phi)) \vdash \neg \Psi$

Inconsistency Robust Direct Logic into Classical Logic. A big advantage of inconsistency robust logic is that it makes it practical for computer systems to reason about theories of practice (e.g. for macroeconomics, human history, etc.) that are pervasively inconsistent. Since software engineers have to deal with theories chock full of inconsistencies, Inconsistency Robust Direct Logic should be attractive. However, to make it relevant we need to provide them with tools that are cost effective.

Our everyday life is becoming increasingly dependent on large software systems. And these systems are becoming increasingly permeated with inconsistency and concurrency.

As pervasively inconsistent concurrent systems become a major part of the environment in which we live, it becomes an issue of common sense to use them effectively. We will need sophisticated software systems that formalize this common sense to help people understand and apply the principles and practices suggested in this paper.

Creating this software is not a trivial undertaking!

There is much work to be done!

Acknowledgements

Science and politics and aesthetics, these do not inhabit different domains. Instead they interweave. Their relations intersect and resonate together in unexpected ways.

Law [2004 pg. 156]

Sol Feferman, Mike Genesereth, David Israel, Bill Jarrold, Ben Kuipers, Pat Langley, Vladimir Lifschitz, Frank McCabe, John McCarthy, Fanya S. Montalvo, Peter Neumann, Ray Perrault, Natarajan Shankar, Mark Stickel, Richard Waldinger, and others provided valuable feedback at seminars at Stanford, SRI, and UT Austin to an earlier version of the material in this paper. For the AAI Spring Symposium'06, Ed Feigenbaum, Mehmet Göker, David Lavery, Doug Lenat, Dan Shapiro, and others provided valuable feedback. At MIT Henry Lieberman, Ted Selker, Gerry Sussman and the members of Common Sense Research Group made valuable comments. Reviewers for AAMAS '06 and '07, KR'06, COIN@AAMAS'06 and IJCAR'06 made suggestions for improvement.

In the logic community, Mike Dunn, Sol Feferman, Mike Genesereth, Tim Hinrichs, Mike Kassoff, John McCarthy, Chris Mortensen, Graham Priest, Dana Scott, Richard Weyhrauch and Ed Zalta provided valuable feedback

Dana Scott made helpful suggestions concerning inferential undecidability. Richard Waldinger provided extensive suggestions that resulted in better focusing a previous version of this paper and increasing its readability. Discussion with Pat Hayes and Bob Kowalski provided insight into the early history of Prolog. Communications from John McCarthy and Marvin Minsky suggested making common sense a focus. Mike Dunn collaborated on looking at the relationship of the Boolean Fragment of Inconsistency Robust Direct Logic to R-Mingle. Greg Restall pointed out that Inconsistency Robust Direct Logic does not satisfy some Relevantist principles. Gerry Allwein and Jeremy Forth made detailed comments and suggestions for improvement. Bob Kowalski and Erik Sandewall provided helpful pointers and discussion of the relationship with their work. Discussions with Ian Mason and Tim Hinrichs helped me develop Löb's theorem for Direct Logic. Scott Fahlman suggested introducing the roadmap in the introduction of the paper. At CMU, Wilfried Sieg introduced me to his very interesting work with Clinton Field on automating the search for proofs of the Gödel/Rosser inferential undecidability theorems. Also at CMU, I had productive discussions with Jeremy Avigad, Randy Bryant, John Reynolds, Katia Sycara, and Jeannette Wing. At my MIT seminar and afterwards, Marvin Minsky, Ted Selker, Gerry Sussman, and Pete Szolovits made helpful comments. Les Gasser, Mike Huhns, Victor Lesser, Pablo Noriega, Sascha Ossowski, Jaime Sichman, Munindar Singh, *etc.* provided valuable suggestions at AAMAS'07. I had a very pleasant dinner with Harvey Friedman at Chez Panisse after his 2nd Tarski lecture.

Jeremy Forth, Tim Hinrichs, Fanya S. Montalvo, and Richard Waldinger provided helpful comments and suggestions on the logically necessary inconsistencies in theories of Direct Logic. Rineke Verbrugge provided valuable comments and suggestions at MALLOW'07. Mike Genesereth and Gordon Plotkin kindly hosted my lectures at Stanford and Edinburgh, respectively, on "*The Logical Necessity of Inconsistency*". Inclusion of Cantor's diagonal argument as motivation was suggested by Jeremy Forth. John McCarthy pointed to the distinction between Logic Programs and the Logician Programme for Artificial Intelligence. Reviewers at JAIR made useful suggestions. Mark S. Miller made important suggestions for improving the meta-circular definition of ActorScript. Comments by Michael Beeson helped make the presentation of Direct Logic more rigorous. Conversations with Jim Larson helped clarify the relationship between classical logic and the inconsistency robust logic. An anonymous referee of the Journal of Logic and Computation made a useful comment. John-Jules Meyer and Albert Visser provided helpful advice and suggestions. Comments by Mike Genesereth, Eric Kao, and Mary-Anne Williams at my Stanford Logic Group seminar "*Inference in Boolean Direct Logic is Computationally Decidable*" on 18

November 2009 greatly improved the explanation of direct inference. Discussions at my seminar “Direct Inference for Direct Logic™ Reasoning” at SRI hosted by Richard Waldinger on 7 January 2010 helped improve the presentation of Direct Logic. Helpful comments by Emily Bender, Richard Waldinger and Jeannette Wing improved the section on Inconsistency Robustness.

Eric Kao provided numerous helpful comments and discovered bugs in the principles of Self-refutation and Excluded Middle that were part of a previous version of Inconsistency Robust Direct Logic [Kao 2011]. Self-refutation has been replaced by Self-annihilation in the current version. Stuart Shapiro provided helpful information on why SNePS [Shapiro 2000] was based on Relevance Logic. Discussions with Dennis Allison, Eugene Miya, Vaughan Pratt and others were helpful in improving this article.

Make Travers made suggestions and comments that greatly improved the overall organization of the paper. Richard Waldinger provided guidance on classical automatic theorem provers. Illuminating conversations with Patrick Suppes provided additional ideas for improvement.

Bibliography

- Hal Abelson and Gerry Sussman *Structure and Interpretation of Computer Programs* 1984.
- Luca Aceto and Andrew D. Gordon (editors). *Algebraic Process Calculi: The First Twenty Five Years and Beyond* Bertinoro, Italy, August, 2005.
- Sanjaya Addanki, Roberto Cremonini, and J. Scott Penberthy. “Reasoning about assumptions in graphs of models” *Readings in Qualitative Reasoning about Physical Systems*. Kaufman. 1989.
- Gul Agha. *Actors: A Model of Concurrent Computation in Distributed Systems* Doctoral Dissertation. 1986.
- Gul Agha, Ian Mason, Scott Smith, and Carolyn Talcott. “A foundation for Actor computation.” *Journal of Functional Programming*. 1997.
- Allen, L. E., and Saxon, C. S. “More is needed in AI: Interpretation assistance for coping with the problem of multiple structural interpretations” *ICAIL* 1991.
- Bruce Anderson. “Documentation for LIB PICO-PLANNER” School of Artificial Intelligence, Edinburgh University. 1972.
- Chris Anderson. “The End of Theory: The Data Deluge Makes the Scientific Method Obsolete” *Wired*. June 23, 2009.
- Alan Anderson and Nuel Belnap, Jr. (1975) *Entailment: The Logic of Relevance and Necessity* Princeton University Press.

- Robert Anderson and Woody Bledsoe (1970) “A Linear Format for Resolution with Merging and a New Technique for Establishing Completeness” *JACM* 17.
- Aldo Antonelli (2006). “Non-monotonic Logic” *Stanford Encyclopedia of Philosophy*. March 2006.
- A. I. Arruda. “Aspects of the historical development of paraconsistent logic” In *Paraconsistent Logic: Essays on the Inconsistent* Philosophia Verlag. 1989
- William Aspray “Interview with J. Barkley Rosser and Stephen C. Kleene” *The Princeton Mathematics Community in the 1930s* Transcript PMC23 1985.
- William Athas and Nanette Boden “Cantor: An Actor Programming System for Scientific Computing” *Proceedings of the NSF Workshop on Object-Based Concurrent Programming*. 1988. Special Issue of SIGPLAN Notices.
- Henry Baker. *Actor Systems for Real-Time Computation* MIT EECS Doctoral Dissertation. January 1978.
- Henry Baker and Carl Hewitt: *Laws for Communicating Parallel Processes* IFIP. August 1977.
- Henry Baker and Carl Hewitt “The Incremental Garbage Collection of Processes.” Symposium on Artificial Intelligence Programming Languages. SIGPLAN Notices. August 1977. “
- Bob Balzer. “Tolerating Inconsistency” *13th International Conference on Software Engineering*. 1991.
- Marcel Barzin 1940.
- Bruce Baumgart. “Micro-Planner Alternate Reference Manual” Stanford AI Lab Operating Note No. 67, April 1972.
- JC Beall and Greg Restall. *Logical Pluralism* Oxford University Press. 2006.
- Michael Beeson. “Lambda Logic” Lecture Notes in Artificial Intelligence 3097. Springer. 2004.
- Nuel Belnap. “A useful four-valued logic” in *Modern uses of multiple valued logics*. D. Reidel, Dordrecht, 1977.
- Francesco Berto *The Gödel Paradox and Wittgenstein’s Reasons* Philosophia Mathematica (III) 17. 2009.
- Francesco Berto. *There’s Something about Gödel: The Complete Guide to the Incompleteness Theorem* John Wiley and Sons. 2010.
- Francesco Berto. “Representing Inconsistency” *Inconsistency Robustness* 2011.
- Leopoldo Bertossi, Anthony Hunter, and Torsten Schaub eds. *Inconsistency Tolerance* Springer. 2004.
- Philippe Besnard and Anthony Hunter. “Quasi-classical Logic: Non-trivializable classical reasoning from inconsistent information” *Symbolic and Quantitative Approaches to Reasoning and Uncertainty* Springer LNCS 1995.

- Philippe Besnard and Torsten Schaub. "Significant Inferences: Preliminary Report. 2000.
- Jean-Yves Bèziau. "The future of paraconsistent logic" *Logical Studies* 2. 1999.
- Jean-Yves Bèziau, Walter Carnielli, and Dov Gabbay. Ed. *Handbook of Paraconsistency* College Publications Kings College London. 2007
- S. V. Bhave. "Situations in Which Disjunctive Syllogism Can Lead from True Premises to a False Conclusion" *Notre Dame Journal of Formal Logic* Vol. 38, No. 3. 1997.
- Fisher Black. *A deductive question answering system*, Harvard University Thesis. 1964.
- Simon Blackburn and Keith Simmons (1999) *Truth* Oxford University Press.
- H. Blair and V. S. Subrahmanian. "Paraconsistent Logic Programming". *Theoretical Computer Science*, 68(2) 1989.
- Patricia Blanchette "The Frege-Hilbert Controversy" *The Stanford Encyclopedia of Philosophy* December 7, 2007.
- Andreas Blass, Yuri Gurevich, Dean Rosenzweig, and Benjamin Rossman (2007a) *Interactive small-step algorithms I: Axiomatization* Logical Methods in Computer Science. 2007.
- Andreas Blass, Yuri Gurevich, Dean Rosenzweig, and Benjamin Rossman (2007b) *Interactive small-step algorithms II: Abstract state machines and the characterization theorem*. Logical Methods in Computer Science. 2007.
- George Boole. *An Investigation of the Laws of Thought* 1853. <http://www.gutenberg.org/etext/15114>
- Geof Bowker, Susan L. Star, W. Turner, and Les Gasser, (Eds.) *Social Science Research, Technical Systems and Cooperative Work* Lawrence Erlbaum. 1997.
- Robert Boyer (1971) *Locking: A Restriction of Resolution* Ph. D. University of Texas at Austin.
- Fisher Black. *A Deductive Question Answering System* Harvard University. Thesis. 1964.
- Daniel Bobrow and Bertram Raphael. "New programming languages for Artificial Intelligence research" *ACM Computing Surveys*. 1974.
- Jean-Pierre Briot. *From objects to actors: Study of a limited symbiosis in Smalltalk-80* Rapport de Recherche 88-58, RXF-LITP. Paris, France. September 1988.
- Stephen Brookes, Tony Hoare, and Bill Roscoe. *A theory of communicating sequential processes* JACM. July 1984.
- Maurice Bruynooghe, Luís Moniz Pereira, Jörg Siekmann, Maarten van Emden. "A Portrait of a Scientist as a Computational Logician" *Computational Logic: Logic Programming and Beyond: Essays in Honour of Robert A. Kowalski, Part I* Springer. 2004.
- Martin Caminda. "On the Issue of Contraposition of Defeasible Rules" *COMMA '08*.

- Andrea Cantini “Paradoxes and Contemporary Logic” *The Stanford Encyclopedia of Philosophy* October 16, 2007.
- George Cantor. “Diagonal Argument” German Mathematical Union (*Deutsche Mathematiker-Vereinigung*) (Bd. I, S. 75-78) 1890-1.
- Rudolph Carnap. *Logische Syntax der Sprache. (The Logical Syntax of Language* Open Court Publishing 2003) 1934.
- Luca Cardelli and Andrew Gordon. “Mobile Ambients” *Foundations of Software Science and Computational Structures* Springer, 1998.
- Lewis Carroll “What the Tortoise Said to Achilles” *Mind* 4. No. 14. 1895.
- Lewis Carroll. *Through the Looking-Glass* Macmillan. 1871.
- 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>
- Carlo Cellucci “The Growth of Mathematical Knowledge: An Open World View” *The growth of mathematical knowledge* Kluwer. 2000.
- Aziem Chawdhary, Byron Cook, Sumit Gulwani, Mooly Sagiv, and Hongseok Yang *Ranking Abstractions* ESOP’08.
- Alonzo Church “A Set of postulates for the foundation of logic (1)” *Annals of Mathematics*. Vol. 33, 1932.
- Alonzo Church “A Set of postulates for the foundation of logic (2)” *Annals of Mathematics*. Vol. 34, 1933.
- Alonzo Church. *An unsolvable problem of elementary number theory* Bulletin of the American Mathematical Society 19, May, 1935. *American Journal of Mathematics*, 58 (1936),
- Alonzo Church *The Calculi of Lambda-Conversion* Princeton University Press. 1941.
- Will Clinger. *Foundations of Actor Semantics* MIT Mathematics Doctoral Dissertation. June 1981.
- Paul Cohen “My Interaction with Kurt Gödel; the man and his work” *Gödel Centenary: An International Symposium Celebrating the 100th Birthday of Kurt Gödel* April 27–29, 2006.
- Alain Colmerauer and Philippe Roussel. “The birth of Prolog” *History of Programming Languages* ACM Press. 1996
- Melvin Conway. *Design of a separable transition-diagram compiler* CACM. 1963.
- F. S. Correa da Silva, J. M. Abe, and M. Rillo. “Modeling Paraconsistent Knowledge in Distributed Systems”. Technical Report RT-MAC-9414, Instituto de Matematica e Estatística, Universidade de Sao Paulo, 1994.
- James Crawford and Ben Kuipers. “Negation and proof by contradiction in access-limited logic.” *AAAI-91*.
- Haskell Curry “Some Aspects of the Problem of Mathematical Rigor” *Bulletin of the American Mathematical Society* Vol. 4. 1941.

- Haskell Curry. "The combinatory foundations of mathematics" *Journal of Symbolic Logic*. 1942.
- Haskell Curry. *Foundations of Mathematical Logic*. McGraw-Hill. 1963.
- Michael Cusumano and Richard Selby, R. *Microsoft Secrets: How the World's Most Powerful Software Company Creates Technology, Shapes Markets, and Manages People*. Free Press. 1995
- Newton da Costa *Inconsistent Formal Systems* (Sistemas Formais Inconsistentes in Portuguese) Doctoral dissertation. University of Paraná. 1963.
- Newton da Costa. "On the Theory of Inconsistent Formal Systems" *Notre Dame Journal of Formal Logic* October 1974.
- Ole-Johan Dahl and Kristen Nygaard. "Class and subclass declarations" *IFIP TC2 Conference on Simulation Programming Languages*. May 1967.
- Ole-Johan Dahl and Tony Hoare. *Hierarchical Program Structures* in "Structured Programming" Prentice Hall. 1972.
- Carlos Damásio and Luís Pereira. "A Model Theory for Paraconsistent Logic Programming" *Portuguese Conference on Artificial Intelligence* 1995.
- Giacomo Mauro D'Ariano and Alessandro Tosini. "Space-time and special relativity from causal networks" *ArXiv*. 1008.4805. August 2010.
- Julian Davies. "Popler 1.5 Reference Manual" University of Edinburgh, TPU Report No. 1, May 1973.
- Ernest Davis. "The Naïve Physics Perplex" *AI Magazine*. Winter 1998.
- Ernest Davis and Leora Morgenstern. "A First-Order Theory of Communication and Multi-Agent Plans" *Journal of Logic and Computation*, Vol. 15, No. 5, 2005.
- John Dawson *Logical Dilemmas. The Life and Work of Kurt Gödel* AK Peters. 1997
- John Dawson. "What Hath Gödel Wrought?" *Synthese*. Jan. 1998.
- John Dawson. "Shaken Foundations or Groundbreaking Realignment? A Centennial Assessment of Kurt Gödel's Impact on Logic, Mathematics, and Computer Science" *FLOC'06*.
- Walter Dean and Hdenori Kurokawa. "Knowledge, proof, and the Knower" TARK'09,
- 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.
- Hendrik Decker. *A Case for Paraconsistent Logic as a Foundation of Future Information Systems*. CAiSE'05 Workshop PHISE'05. 2005.
- Hendrik Decker. *Historical and Computational Aspects of Paraconsistency in View of the Logic Foundation of Databases*. Semantics in Databases. Springer. 2003.
- David Deutsch. "Quantum theory, the Church-Turing principle and the universal quantum computer" *Proceedings of the Royal Society of London*. 1985.

- Richard De Millo, Richard Lipton and Alan Perlis “Social Processes and Proofs of Theorems and Programs” CACM. May 1979.
- René Descartes. *Principles of Philosophy* (English translation in *The Philosophical Writings of Descartes* Cambridge University Press 1985). 1644.
- Harry Deutsch “A Note on the Decidability of a Strong Relevant Logic” *Studia Logica* Vol. 44. No. 2. 1985.
- Cora Diamond. *Wittgenstein's Lectures on the Foundations of Mathematics, Cambridge, 1939* Cornell University Press. 1976.
- Edsger Dijkstra. *A Discipline of Programming*. Prentice Hall. 1976.
- Edsger Dijkstra and A.J.M. Gasteren. “A Simple Fixpoint Argument Without the Restriction of Continuity” *Acta Informatica*. Vol. 23. 1986.
- Kosta Dōzen. “Logical Constants as Punctuation Marks” *Notre Dame Journal of Formal Logic*. Summer 1989.
- Paul du Bois-Reymond-1880 “Der Beweis des Fundamentalsatzes der Integralrechnung” *Mathematische Annalen* Vol. 16. 1880.
- Michael Dummett (1973). “The Justification of Deduction” in *Truth and other Enigmas* Duckworth. 1978.
- Michael Dunn and Greg Restall. “Relevance Logic” in *The Handbook of Philosophical Logic, second edition*. Dov Gabbay and Franz Guenther (editors), Kluwer. 2002.
- Michael Dunn. *Contradictory Information: Better than the Nothing* CMU Philosophy Colloquium. April 10, 2014.
- T. S. Eliot. *Four Quartets*. Harcourt. 1943.
- Ralph Waldo Emerson. “Self Reliance “ Essays—First Series. 1841.
- Pascal Engel. “Dummett, Achilles and the Tortoise” *The philosophy of Michael Dummett* Open Court. 2007.
- Euclid. *The Thirteen Books of Euclid's Elements*. (3 Vol. translated by Thomas Heath. Cambridge University Press. 1925). Circa 300BC.
- Scott Fahlman. *A Planning System for Robot Construction Tasks* MIT AI TR-283. June 1973.
- Adam Farquhar, Anglela Dappert, Richard Fikes, and Wanda Pratt. “Integrating Information Sources Using Context” Logic Knowledge Systems Laboratory. KSL-95-12. January, 1995.
- Anita Feferman and Solomon Feferman *Alfred Tarski: Life and Logic*. Cambridge University Press. 2004.
- Solomon Feferman (1984a) “Toward Useful Type-Free Theories, I” in *Recent Essays on Truth and the Liar Paradox*. Ed. Robert Martin (1991) Clarendon Press.
- Solomon Feferman (1984b) “Kurt Gödel: Conviction and Caution” *Philosophia Naturalis* Vol. 21.
- Solomon Feferman. “Reflecting on incompleteness” *Journal of Symbolic Logic* 1991
- Solomon Feferman *In the Light of Logic* Oxford University Press. 1998.

- Solomon Feferman. "Logic, Logics, and Logicism" *Notre Dame Journal of Formal Logic*. V 40. 1999.
- Solomon Feferman "Does reductive proof theory have a viable rationale?" *Erkenntnis* 53. 2000.
- Solomon Feferman "Tarski's Conceptual Analysis for Semantical Notions" *Sémantique et Épistémologie* 2004.
- Solomon Feferman "Predicativity" in *The Oxford Handbook of Philosophy of Mathematics and Logic* Oxford University Press. 2005.
- Solomon Feferman (2006a) "The nature and significance of Gödel's incompleteness theorems" lecture for the Princeton Institute for Advanced Study Gödel Centenary Program, Nov. 17, 2006.
- Solomon Feferman (2006b) "Lieber Herr Bernays! Lieber Herr Gödel! Gödel on finitism, constructivity and Hilbert's program" submitted version of lecture for the Gödel centenary conference, *Horizons of Truth*, Vienna, 27-29 April 2006.
- Solomon Feferman (2006c) "Are there absolutely unsolvable problems? Gödel's dichotomy" *Philosophia Mathematica* Series III vol. 14.
- Solomon Feferman (2007a) "Axioms for determinateness and truth"
- Solomon Feferman (2007b) "Gödel, Nagel, minds and machines" October 25, 2007.
- Solomon Feferman "Axioms for determinateness and truth" *Review of Symbolic Logic*. 2008.
- Solomon Feferman. "About and around computing over the reals" *Computability: Gödel, Church, Turing and Beyond* MIT Press. forthcoming 2012.
- Dieter Fensel and Frank van Harmelen. "Unifying Reasoning and Search to Web Scale" *IEEE Internet Computing*. March/April 2007.
- James Fetzer. "Program Verification: The Very Idea" *CACM* September 1988.
- Paul Feyerabend. *Killing Time: The Autobiography of Paul Feyerabend*. University Of Chicago Press. 1995.
- Richard Feynman. "Lecture 6: Probability and Uncertainty — the Quantum Mechanical view of Nature" *The Character of Physical Law*. MIT Press. 1965.
- Hartry Field. "A Revenge-Immune Solution to the Semantic Paradoxes." *Journal of Philosophical Logic*, April 2003
- Kit Fine. "Analytic Implication" *Notre Dame Journal of Formal Logic*. April 1986.
- A. C. W. Finkelstein, D. Gabbay, A. Hunter, J. Kramer, and B. Nuseibeh, "Inconsistency Handling in Multi-Perspective Specifications" *Transactions on Software Engineering*, August 1994.
- Frederic Fitch. *Symbolic Logic: an Introduction*. Ronald Press. 1952.
- Juliet Floyd and Hilary Putnam. "Wittgenstein's 'Notorious' Paragraph About the Gödel Theorem: Recent Discussions" ("Wittgenstein's 'berühmter' Paragraph über das Gödel-Theorem: Neuere Diskussionen") in *Prosa oder*

- Besweis? Wittgenstein's ›berüchtigte‹ Bemerkungen zu Gödel, Texte und Dokumente* Parerga Verlag. 2008.
- J.M. Foster and E.W. Elcock. (1969) “ABSYS: An Incremental Compiler for Assertions” *Machine Intelligence 4*. Edinburgh University Press.
- Nissim Francez, Tony Hoare, Daniel Lehmann, and Willem-Paul de Roever. “Semantics of nondeterminism, concurrency, and communication” *Journal of Computer and System Sciences*. December 1979.
- Torkel Franzén. *Inexhaustibility* AK Peters. 2004
- Torkel Franzén. *Gödel's Theorem: an incomplete guide to its use and abuse*. A K Peters. 2005.
- Gottlob Frege. *Begriffsschrift: eine der arithmetischen nachgebildete Formelsprache des reinen Denkens* Halle, 1879.
- Gottlob Frege (1915) “My Basic Logical Insights” *Posthumous Writings* University of Chicago Press. 1979.
- Kazuhiro Fuchi, Robert Kowalski, Kazunori Ueda, Ken Kahn, Takashi Chikayama, and Evan Tick. “Launching the new era”. *CACM*. 1993.
- Dov Gabbay (ed.) *What is a Logical System?* Oxford. 1994.
- Dov Gabbay and Anthony Hunter. “Making inconsistency respectable: A logical framework for inconsistency in reasoning (Part 1). *Fundamentals of Artificial Intelligence Research '91*, Springer-Verlag.. 1991.
- Dov Gabbay and Anthony Hunter. “Making inconsistency respectable: A logical framework of r inconsistency in reasoning (Part 2). *Symbolic and Quantitative Approaches to Reasoning and Uncertainty LNCS*, Springer-Verlag, 1992.
- John Kenneth Galbraith. *Economics, Peace and Laughter*. New American Library. 1971.
- Robin Gandy. “Church’s Thesis and Principles of Mechanisms” *The Kleene Symposium* North–Holland. 1980.
- John Gay. “The Elephant and the Bookseller” *Fifty-one Fables in Verse* 1727
- Michael Gelfond and Vladimir Lifschitz. “Logic programs with classical negation” *International Conference on Logic Programming*. MIT Press. 1990.
- Gerhard Gentzen. “Provability and nonprovability of restricted transfinite induction in elementary number theory” (*Collected Papers of Gerhard Gentzen*. North-Holland. 1969) Habilitation thesis. Göttingen. 1942.
- Gerhard Gentzen (1935) “Investigations into Logical Deduction.” (*Collected Papers of Gerhard Gentzen*. North-Holland. 1969)
- Steve Gerrard “Wittgenstein's Philosophies of Mathematics” *Synthese* 87. 1991.
- Matt Ginsberg. “AI and nonmonotonic reasoning” in *Handbook of Logic in Artificial Intelligence and Logic Programming* Clarendon Press. 1994.
- Jean-Yves Girard. *The Blind Spot: Lectures on proof-theory* Roma Tre. 2004.

- Andreas Glausch and Wolfgang Reisig. *Distributed Abstract State Machines and Their Expressive Power* Informatik-Berichte 196. Humboldt University of Berlin. January 2006.
- Kurt Gödel (1930) “The completeness of the axioms of the functional calculus of logic” (translated in *A Source Book in Mathematical Logic, 1879-1931*. Harvard Univ. Press. 1967)
- Kurt Gödel (1931) “On formally undecidable propositions of *Principia Mathematica*” in *A Source Book in Mathematical Logic, 1879-1931*. Translated by Jean van Heijenoort. Harvard Univ. Press. 1967.
- Kurt Gödel (1933) “An Interpretation of the Intuitionistic Propositional Calculus,” in *Collected Works of Kurt Gödel*, Oxford University Press, Volume 3, 1995, pp. 296-302.
- Kurt Gödel (1944) “Russell’s Mathematical Logic” in *Philosophy of Mathematics* (2nd ed.) Cambridge University Press.
- Kurt Gödel (1951) “Some basic theorems on the foundations of mathematics and their implications” in *Collected Works of Kurt Gödel*, Oxford University Press, Volume 3, 1995.
- Kurt Gödel (1965) “On Undecidable Propositions of Formal Mathematical Systems” (a copy of Gödel’s 1931 paper with his corrections of errata and added notes) in *The Undecidable: Basic Papers on Undecidable Propositions, Unsolvability Problems and Computable Functions* Martin Davis editor. Raven Press 1965.
- Kurt Gödel (1972), “Some Remarks on the Undecidability Results” in *Kurt Gödel Collected Works, II*. Oxford University Press. 2001.
- Dina Goldin and Peter Wegner. “The Interactive Nature of Computing: Refuting the Strong Church-Turing Thesis” *Minds and Machines* March 2008.
- Solomon Golomb and Leonard Baumert. (1965) “Backtrack Programming” JACM. Vol. 12 No. 4.
- Thomas Gordon. *Foundations of Argumentation Technology: Summary of Habilitation Thesis* Technische Universität Berlin. 2009.
- C. Cordell Green: “Application of Theorem Proving to Problem Solving” IJCAI 1969.
- Steve Gregory. “Concurrent Logic Programming Before ICOT: A Personal Perspective” August 15, 2007.
<http://www.cs.bris.ac.uk/~steve/papers/ALP/CLPbeforeICOT.pdf>
- Irene Greif. *Semantics of Communicating Parallel Processes* MIT EECS Doctoral Dissertation. August 1975
- Ramanathan Guha. *Contexts: Formalization and Some Applications* PhD thesis, Stanford University, 1991.
- Robert Hadley. “Consistency, Turing Computability and Gödel’s First Incompleteness Theorem” *Minds and Machines* 18. 2008.
- Volker Halbach “Axiomatic theories of truth” *Stanford Encyclopedia of Philosophy*. 2007.

- Ronald Harrop. "Some structure results for propositional calculi" *Journal of Symbolic Logic*, 30. 1965.
- W. D. Hart. "Skolem Redux" *Notre Dame Journal of Formal Logic*. 41, no. 4. 2000.
- Donna Haraway. "Situated Knowledge: the Science Question in Feminism and the Privilege of Partial Perspective" in *Simians, Cyborgs, and Women: the Reinvention of Nature*. Free Association Books. 1991.
- Pat Hayes. "Computation and Deduction" *Mathematical Foundations of Computer Science: Proceedings of Symposium and Summer School, Štrbské Pleso, High Tatras, Czechoslovakia*. September 1973.
- Pat Hayes "Some Problems and Non-Problems in Representation Theory" *AISB*. Sussex. July, 1974.
- Pat Hayes. "The Naïve Physics Manifesto". *Expert Systems in the Microelectronic Age*. Edinburgh University Pres. 1979.
- Pat Hayes. 1985a. "The Second Naïve Physics Manifesto" *Formal Theories of the Commonsense World*. Ablex. 1985.
- Pat Hayes. 1985b. "Naïve Physics 1: Ontology for Liquids" *Formal Theories of the Commonsense World*. Ablex. 1985.
- Pat Hayes. "Contexts in context." *Contexts in Knowledge Representation and Natural Language*. AAAI. 1997.
- Pat Hayes. "Context Mereology." *Commonsense 2007*.
- Jean van Heijenoort (1967) *From Frege to Gödel. A Source Book in Mathematical Logic, 1897-1931*, Harvard University Press.
- Joseph Heller. *Catch-22*. Simon & Schuster. 1961.
- Leon Henkin "A Problem Concerning Provability" *Journal of Symbolic Logic*, Vol. 17 (1952).
- Carl Hewitt. "Planner: A Language for Proving Theorems in Robots" *IJCAI* 1969.
- Carl Hewitt. "Procedural Embedding of Knowledge In Planner" *IJCAI* 1971.
- Carl Hewitt, Peter Bishop and Richard Steiger. "A Universal Modular Actor Formalism for Artificial Intelligence" *IJCAI* 1973.
- Carl Hewitt and Henry Baker Laws for Communicating Parallel Processes *IFIP*. August 1977.
- Carl Hewitt. "Viewing Control Structures as Patterns of Passing Messages" *Journal of Artificial Intelligence*. June 1977.
- Carl Hewitt and Peter de Jong. "Open Systems" *Perspectives on Conceptual Modeling*, Brodie, Mylopoulos, and Schmidt (eds.), Springer-Verlag, 1983.
- Carl Hewitt. "The Challenge of Open Systems" *Byte Magazine*. April 1985.
- Carl Hewitt (1986). "Offices Are Open Systems" *ACM Transactions on Information Systems* 4(3)
- Carl Hewitt (1990). "Towards Open Information Systems Semantics" *International Workshop on Distributed Artificial Intelligence*
- Carl Hewitt (1991). "Open Information Systems Semantics" *Journal of Artificial Intelligence*. January 1991.

- Carl Hewitt and Jeff Inman. "DAI Betwixt and Between: From 'Intelligent Agents' to Open Systems Science" *IEEE Transactions on Systems, Man, and Cybernetics*. Nov. /Dec. 1991.
- Carl Hewitt and Gul Agha. "Guarded Horn clause languages: are they deductive and Logical?" *International Conference on Fifth Generation Computer Systems*. Ohmsha 1988.
- Carl Hewitt. (2006). "What is Commitment? Physical, Organizational, and Social" *COIN@AAMAS'06*. (Revised version to be published in Springer Verlag Lecture Notes in Artificial Intelligence. Edited by Javier Vázquez-Salceda and Pablo Noriega. 2007) April 2006.
- Carl Hewitt (2007a). "Organizational Computing Requires Unstratified Paraconsistency and Reflection" *COIN@AAMAS*. 2007.
- Carl Hewitt (2008a) "A historical perspective on developing foundations for privacy-friendly client cloud computing: iConsult™ & iEntertain™ Apps using iInfo™ Information Integration for iOrgs™ Information Systems" (Revised version of "Development of Logic Programming: What went wrong, What was done about it, and What it might mean for the future" in Proceedings of *What Went Wrong and Why* edited by Mehmet Göker and Daniel Shapiro, AAAI Press. 2008 pp. 1-11) ArXiv. 0901.4934
- Carl Hewitt (2008b). "Norms and Commitment for iOrgs™ Information Systems: Direct Logic™ and Participatory Grounding Checking" ArXiv 0906.2756
- Carl Hewitt (2008c) "Large-scale Organizational Computing requires Unstratified Reflection and Strong Paraconsistency" *Coordination, Organizations, Institutions, and Norms in Agent Systems III* Jaime Sichman, Pablo Noriega, Julian Padget and Sascha Ossowski (ed.). Springer-Verlag. <http://organizational.carlhewitt.info/>
- Carl Hewitt (2008d) "*Middle History of Logic Programming: Resolution, Planner, Edinburgh Logic for Computable Functions, Prolog and the Japanese Fifth Generation Project*" ArXiv 0904.3036
- Carl Hewitt (2008e). *ORGs for Scalable, Robust, Privacy-Friendly Client Cloud Computing* IEEE Internet Computing September/October 2008.
- Carl Hewitt (2009a) *Perfect Disruption: The Paradigm Shift from Mental Agents to ORGs* IEEE Internet Computing. Jan/Feb 2009.
- Carl Hewitt (2010a) ActorScript™ extension of C#®, Java®, and Objective C®: iAdaptive™ concurrency for antiCloud™ privacy-friendly computing in Inconsistency Robustness. College Publications. 2015.
- Carl Hewitt (2010b) "Actor Model of Computation: Scalable Robust Information Systems" in Inconsistency Robustness. College Publications. 2015.
- Carl Hewitt (2010c) Wittgenstein versus Gödel on the Foundations of Logic Stanford Media X Logic Colloquium video recording. April 23, 2010.
- Carl Hewitt. Looming private information fiasco versus the new cloud business model: The next generation will ask, "*Where were you when this was going down?*" Risks Digest. Vol. 26: Issue 37. March 9. 2011.

- Carl Hewitt (editor). Inconsistency Robustness 1011 Stanford University. 2011.
- Carl Hewitt. *What is computation? Actor Model versus Turing's Model* in “A Computable Universe: Understanding Computation & Exploring Nature as Computation” Edited by Hector Zenil. World Scientific Publishing Company. 2012.
- David Hilbert. 1900. in “Mathematical Developments Arising From Hilbert Problems” *Proceedings of Symposia in Pure Mathematics*, Vol. 28. American Mathematical Society. 1976
- David Hilbert (1926) “Über das Unendliche” *Mathematische Annalen*, 95: 161-90. (“On the Infinite” English translation in van Heijenoort. 1967).
- David Hilbert and Paul Bernays. *Grundlagen der Mathematik I*. (L'Harmattan edition 2001) 1934.
- David Hilbert and Paul Bernays. *Grundlagen der Mathematik II*. (L'Harmattan edition 2001) 1939.
- Tony Hoare. “Communicating Sequential Processes” *CACM* August, 1978.
- Tony Hoare. *Communicating Sequential Processes*. Prentice Hall. 1985.
- Tony Hoare. “The verifying compiler: A grand challenge for computing research” *JACM*. January 2003.
- Tony Hoare. *Retrospective: An Axiomatic Basis for Computer Programming* CACM 2009.
- Wilfrid Hodges (2006) “Tarski’s Truth Definitions” *Stanford Encyclopedia of Philosophy*.
- Douglas Hofstadter. *Godel, Escher, Bach: An Eternal Golden Braid*. Random House. 1980.
- Douglas Hofstadter. *I am a Strange Loop* Basic Books. 2007.
- Jim Holt. “Code-Breaker” *The New Yorker* February 6, 2006.
- Leon Horsten “Philosophy of Mathematics” *The Stanford Encyclopedia of Philosophy* September 27, 2007.
- Matthew Huntbach and Graem Ringwood. *Agent-Oriented Programming: From Prolog to Guarded Definite Clauses* Springer. 1999.
- Anthony Hunter. *Reasoning with Contradictory Information using Quasi-classical Logic* Journal of Logic and Computation. Vol. 10 No. 5. 2000.
- Daniel Ingalls. “The Evolution of the Smalltalk Virtual Machine” *Smalltalk-80: Bits of History, Words of Advice*. Addison Wesley. 1983.
- Daniel Isaacson. “The reality of mathematics and the case of set theory” *Truth, Reference, and Realism* Central European University Press, 2008.
- Stanisław Jaśkowski “On the Rules of Suppositions in Formal Logic” *Studia Logica* 1, 1934. (reprinted in: *Polish logic 1920-1939*, Oxford University Press, 1967.
- Stanisław Jaśkowski. *Propositional calculus for contradictory deductive systems* *Studia Logica*. 24 (1969) *Rachunek zdań dla systemów dedukcyjnych sprzecznych* in: *Studia Societatis Scientiarum Torunensis, Sectio A*, Vol. I, No. 5, Toruń 1948.

- Eric Kao. "Proof by self-refutation and excluded middle lead to explosion" *Inconsistency Robustness 2011* Stanford. August 16-18, 2011.
- Michael Kassoff, Lee-Ming Zen, Ankit Garg, and Michael Genesereth. *PrediCalc: A Logical Spreadsheet Management System* 31st International Conference on Very Large Databases (VLDB). 2005.
- Alan Kay. "Personal Computing" in *Meeting on 20 Years of Computing Science* Istituto di Elaborazione della Informazione, Pisa, Italy. 1975. <http://www.mprove.de/diplom/gui/Kay75.pdf>
- Jussi Ketonen and Richard Weyhrauch. "A decidable fragment of Predicate Calculus" *Theoretical Computer Science*. 1984.
- Thomas Kida. *Don't Believe Everything You Think: The 6 Basic Mistakes We Make in Thinking* Prometheus Books. 2006.
- Stephen Kleene and John Barkley Rosser "The inconsistency of certain formal logics" *Annals of Mathematics* Vol. 36. 1935.
- Stephen Kleene *General recursive functions and natural numbers* Mathematical Annals. 1936.
- Stephen Kleene *Recursive Predicates and Quantifiers* American Mathematical Society Transactions. 1943
- Stephen Kleene "Reflections on Church's Thesis" *Notre Dame Journal of Formal Logic* 1987.
- Morris Kline. *Mathematical thought from ancient to modern times* Oxford University Press. 1972.
- Frederick Knabe. "A Distributed Protocol for Channel-Based Communication with Choice" *PARLE* 1992.
- Bill Kornfeld and Carl Hewitt. "The Scientific Community Metaphor" *IEEE Transactions on Systems, Man, and Cybernetics*. January 1981.
- Bill Kornfeld. *Parallelism in Problem Solving* MIT EECS Doctoral Dissertation. August 1981.
- Robert Kowalski "Predicate Logic as Programming Language" Memo 70, Department of Artificial Intelligence, Edinburgh University. 1973
- Robert Kowalski. "A proof procedure using connection graphs" *JACM*. October 1975.
- Robert Kowalski (1979) "Algorithm = Logic + Control" *CACM*. July 1979.
- Robert Kowalski (1986). "The limitation of logic" *ACM Annual Conference on Computer Science*.
- Robert Kowalski 1988a. "The Early Years of Logic Programming" *CACM*. January 1988.
- Robert Kowalski (1988b). "Logic-based Open Systems" *Representation and Reasoning*. Stuttgart Conference Workshop on Discourse Representation, Dialogue tableaux and Logic Programming. 1988.
- Robert. Kowalski and Francesca Toni. "Abstract Argumentation" *Artificial Intelligence and Law*. 1996
- Robert Kowalski (2006) "The Logical Way to be Artificially Intelligent." *CLIMA VI*. Springer Verlag.

- Robert Kowalski (2007) "What is Logic Programming?"
http://en.wikipedia.org/wiki/Talk:Logic_programming#What_is_Logic_Programming.3F
- S. Kraus, D. Lehmann and M. Magidor. "Non-monotonic reasoning, preferential models and cumulative logics" *Artificial Intelligence* 44:167–207. 1990.
- Richard Kraut. "Plato" *Stanford Encyclopedia of Philosophy*. 2004.
- Georg Kreisel. "Wittgenstein's Remarks on the Foundations of Mathematics" *British Journal for the Philosophy of Science* 1958.
- Thomas Kuhn. *The Structure of Scientific Revolutions* University of Chicago Press. 1962.
- Ernest Kurtz and Katherine Ketcham. *The Spirituality of Imperfection: Storytelling and the Search for Meaning* Bantam 1993.
- Henry Kyburg and Choh Teng) *Uncertain Inference*, Cambridge University Press. 2001
- Imre Lakatos. "A renaissance of empiricism in the recent philosophy of mathematics?" *Mathematics, Science and Epistemology*. 1978.
- Imre Lakatos. *Proofs and Refutations* Cambridge University Press. 1976
- Imre Lakatos. *Mathematics, Science and Epistemology* edited by J. Worrall and G. Currie. Cambridge University Press. 1978.
- Hélène Landemore. " 'Talking it Out': Deliberation with Others Versus Deliberation Within" *Political Psychology*. forthcoming 2011.
- Peter Landin. "A Generalization of Jumps and Labels" UNIVAC Systems Programming Research Report. August 1965. (Reprinted in *Higher Order and Symbolic Computation*. 1998)
- Bruno Latour *Science in Action: How to Follow Scientists and Engineers Through Society* Harvard University Press. 1987.
- Bruno Latour. *The Making of Law* Polity Press. 2010.
- Bruno Latour. *On the Modern Cult of the Factish Gods* Duke University Press. 2010.
- John Law. *After Method: mess in social science research* Routledge. 2004.
- Federico Laudisa and Carlo Rovelli. "Relational Quantum Mechanics" *Stanford Encyclopedia of Philosophy* 2008.
- Hannes Leitgeb. "What theories of truth should be like (but cannot be)" *Philosophy Compass* 2 (2). 2007.
- Doug Lenat "CYC: Lessons Learned in Large-Scale Ontological Engineering" November 17, 2005.
- Isaac Levi. *Direct Inference* Journal of Philosophy. Jan. 1977.
- Steven Levy *Hackers: Heroes of the Computer Revolution* Doubleday. 1984.
- Clarence Lewis and Cooper Langford. *Symbolic Logic* Century-Croft, 1932.
- David Lewis. "Probabilities of Conditionals, and Conditional Probabilities" *Philosophical Review*. 1976.
- Philip Lewis. "Jonathon von Neumann and EDVAC" Nov. 8. 2004.
www.cs.berkeley.edu/~christos/classics/paper.pdf
- Henry Lieberman. "A Preview of Act 1" MIT AI memo 625. June 1981.

- James Lighthill. "Artificial Intelligence: A General Survey" *Artificial Intelligence: a paper symposium*. UK Science Research Council. 1973
- Martin Löb. "Solution of a problem of Leon Henkin." *Journal of Symbolic Logic*. Vol. 20. 1955.
- Per Martin-Löf "Verificationism then and now" *The Foundational Debate*. Kluwer. 1995.
- Van McGee "Counterexample to Modus Ponens" *The Journal of Philosophy* 82. 1985.
- Eckart Menzler-Trott. *Logic's Lost Genius: The Life of Gerhard Gentzen* American Mathematical Society. 2007.
- Donald Loveland. *Report of a Workshop on the Future Directions of Automated Deduction* NSF 1997.
<http://www.cs.duke.edu/AutoDedFD/report/>
- Leopold Löwenheim (1915) "Über Möglichkeiten im Relativkalkül" *Mathematische Annalen* 76. (Translated as "On possibilities in the calculus of relatives" in Jean van Heijenoort, 1967. *From Frege to Gödel: A Source Book in Mathematical Logic, 1879-1931*. Harvard Univ. Press)
- Michael Lynch *The Nature of Truth* MIT Press. 2001.
- Donald MacKenzie. *Mechanizing Proof*. MIT Press. 2001.
- Edwin Mares. "Relevance Logic" *Stanford Encyclopedia of Philosophy*. Jan. 2006.
- Roger Maddux *Relevance Logic and the calculus of relations* International Conference on. Order, Algebra and Logics. Vanderbilt. 2007.
- Frederick Maier, Yu Ma, and Pascal Hitzler. "Paraconsistent OWL and Related Logics" *Semantic Web Journal*. 2011.
- David Malone. *Dangerous Knowledge* BBC4 documentary. 2007.
http://www.dailymotion.com/playlist/x1cbyd_xSilverPhinx_bbc-dangerous-knowledge/1
- Edwin Mares. *Relevant Logic* Cambridge University Press. 2007
- Per Martin-Löf. "Verificationism then and now" in W. De Pauli-Schimanovich, *et al.*, eds. *The Foundational Debate* Kluwer. 1995.
- John McCarthy. "Programs with common sense" *Symposium on Mechanization of Thought Processes*. National Physical Laboratory. Teddington, England. 1958.
- John McCarthy. "Situations, actions and causal laws" Stanford Artificial Intelligence Project: Memo 2. 1963
- John McCarthy and Pat Hayes. "Some Philosophical Problems from the Standpoint of Artificial Intelligence" *Machine Intelligence* 4. 1969
- John McCarthy, Paul Abrahams, Daniel Edwards, Timothy Hart, and Michael Levin. *Lisp 1.5 Programmer's Manual* MIT Computation Center and Research Laboratory of Electronics. 1962.
- John McCarthy. "Review of 'Artificial Intelligence: A General Survey'" *Artificial Intelligence: a paper symposium*. UK Science Research Council. 1973.

- John McCarthy. "Circumscription—a form of nonmonotonic reasoning." *Artificial Intelligence*. 1980.
- John McCarthy. "Applications of circumscription to formalizing common sense knowledge" *Artificial Intelligence*. 1986.
- John McCarthy. "Generality in Artificial Intelligence" *CACM*. December 1987.
- John McCarthy. "A logical AI Approach to Context" Technical note, Stanford Computer Science Department, 1996.
- John McCarthy. *Sterile Containers* September 8, 2000. <http://www.ai.sri.com/~rkf/designdoc/sterile.ps>
- John McCarthy. "What is Artificial Intelligence" September 1, 2007. <http://www-formal.stanford.edu/jmc/whatisai/whatisai.html>
- L. Thorne McCarty. "Reflections on TAXMAN: An Experiment on Artificial Intelligence and Legal Reasoning" *Harvard Law Review* Vol. 90, No. 5, March 1977.
- Drew McDermott and Gerry Sussman. "The Conniver Reference Manual" MIT AI Memo 259. May 1972.
- Drew McDermott. *The Prolog Phenomenon* ACM SIGART Bulletin. Issue 72. July, 1980.
- Vann McGee "In Praise of the Free Lunch: Why Disquotationalists Should Embrace Compositional Semantics" *Self-Reference* CSLI Publications. 2006.
- Casey McGinnis "Paraconsistency and logical hypocrisy" *The Logica Yearbook Praha*. <http://www.geocities.com/cnmcginnis/ParaLogHyp.pdf>
- Hugo Mercier and Dan Sperber. "Why Do Humans Reason? Arguments for an Argumentative Theory" *Behavioral and Brain Sciences*. 34. 2011.
- Hugo Mercier and Hélène Landemore. "Reasoning is for Arguing: Understanding the Successes and Failures of Deliberation" *Political Psychology*. forthcoming 2011.
- N. David Mermin. "What is Quantum Mechanics Trying to Tell us?" arXiv:quant-ph/9801057. 1998.
- George Milne and Robin Milner. "Concurrent processes and their syntax" *JACM*. April, 1979.
- Robert Milne and Christopher Strachey. *A Theory of Programming Language Semantics* Chapman and Hall. 1976.
- Robin Milner. *Logic for Computable Functions: description of a machine implementation*. Stanford AI Memo 169. May 1972
- Robin Milner "Elements of interaction: Turing award lecture" *CACM*. January 1993.
- Marvin Minsky (ed.) *Semantic Information Processing* MIT Press. 1968.
- Marvin Minsky and Seymour Papert. "Progress Report on Artificial Intelligence" MIT AI Memo 252. 1971.
- Marvin Minsky. *A Framework for Representing Knowledge*. MIT AI Lab Memo 306. 1974.

- Marvin Minsky, Push Singh, and Aaron Sloman. "The St. Thomas Common Sense Symposium: Designing Architectures for Human-Level Intelligence" *AI Magazine*. Summer 2004.
- Annemarie Mol and Marc Berg. "Principles and Practices of Medicine: the Coexistence of various Anaemias" *Culture, Medicine, and Psychiatry* 1994.
- Annemarie Mol. *The Body Multiple: ontology in medical practice* Duke University Press. 2002
- Ray Monk. "Bourgeois, Boshevist or anarchist? The Reception of Wittgenstein's Philosophy of Mathematics" in *Wittgenstein and his interpreters* Blackwell. 2007.
- Charles Morgan. "The Nature of Nonmonotonic Reasoning" *Minds and Machines* 2000
- Chris Mortensen. "The Validity of Disjunctive Syllogism is Not So Easily Proved." *Notre Dame Journal of Formal Logic* January 1983.
- Chris Mortensen. *Inconsistent Mathematics* Kluwer Academic Publishers. 1995.
- Alexander Nekham. *De Naturis Rerum* Thomas Wright, editor. London: Longman, 1863.
- Allen Newell and Herbert Simon. "The logic theory machine: A complex information processing system" *IRE Transactions on Information Theory* IT-2:61-79. 1956.
- Bashar Nuseibeh "To Be and Not to Be: On Managing Inconsistency in Software Development" *IWSSD-8*. March 1996,
- Kristen Nygaard. *SIMULA: An Extension of ALGOL to the Description of Discrete-Event Networks* IFIP'62.
- David Park. "Concurrency and Automata on Infinite Sequences" Lecture Notes in Computer Science, Vol 104. Springer. 1980
- Peter Patel-Schneider *A decidable first-order logic for knowledge representation* IJCAI'85.
- Mike Paterson and Carl Hewitt. "Comparative Schematology" MIT AI Memo 201. August 1970.
- Giuseppe Peano *Arithmetices principia, nova methodo exposita* (The principles of arithmetic, presented by a new method) 1889.
- Judea Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* Morgan Kaufmann. 1988
- Francis Pelletier "A Brief History of Natural Deduction" *History and Philosophy of Logic* Vol. 20, Issue. 1, 1999.
- Carl Petri. *Kommunikation mit Automate*. Ph. D. Thesis. University of Bonn. 1962.
- Andrew Pitts. "Categorical Logic" in *Algebraic and Logical Structures* Oxford University Press. 2000.
- Max Planck *Scientific Autobiography and Other Papers* 1949.
- Gordon Plotkin. "A powerdomain construction" *SIAM Journal of Computing* September 1976.

- Henri Poincaré. "La grandeur mathématiques et l'expérience, *La Science et l'Hypothèse*," Bibliothèque de Philosophie Scientifique *Ernest Flammarion* 1902; English translation "Mathematical magnitude and experiment" *Science and Hypothesis* Walter Scott Publishing Co, 1905
- George Polya (1957) *Mathematical Discovery: On Understanding, Learning and Teaching Problem Solving Combined Edition* Wiley. 1981.
- Karl Popper(1962). *Conjectures and Refutations* Basic Books.
- Karl Popper. (1934) *Logik der Forschung*, Springer. (*Logic of Scientific Discovery* Routledge 2002).
- Howard Pospel. *Propositional Logic* Prentice Hall. 2000
- H. Prakken "A tool in modeling disagreement in law: Preferring the most specific argument" *ICAIL'91*.
- H. Prakken and G. Sartor. "A dialectical model of assessing conflicting arguments in legal reasoning" *Artificial Intelligence and Law* 1996.
- Graham Priest. "Dialetheism" *The Stanford Encyclopedia of Philosophy* (Winter 2004 Edition)
- Graham Priest, and Richard Routley "The History of Paraconsistent Logic" in *Paraconsistent Logic: Essays on the Inconsistent* Philosophia Verlag. 1989.
- Graham Priest. "Paraconsistent Logic" *Handbook of Philosophical Logic* Volume 6, 2nd ed. Kluwer. 2002
- Graham Priest and Koji Tanaka. "Paraconsistent Logic" *The Stanford Encyclopedia of Philosophy*. Winter 2004.
- Graham Priest. "Wittgenstein's Remarks on Gödel's Theorem" in *Wittgenstein's Lasting Significance* Routledge. 2004.
- Graham Priest (2006a) "60% Proof: Lakatos, Proof, and Paraconsistency" 2006.
- Graham Priest (2006b) *In Contradiction 2nd Edition* Clarendon Press. 2006.
- Michael Rathjen. "The art of ordinal analysis" *Proceedings of the International Congress of Mathematicians* 2006
- Willard Quine "Review of Charles Parsons' Mathematics in Philosophy" *Journal of Philosophy* 1984.,
- Miklós Rédei "John von Neumann 1903-1957" *European Mathematical Society Newsletter* March 2004.
- Stephen Reed and Doug Lenat. "Mapping Ontologies into Cyc" *AAAI 2002 Conference Workshop on Ontologies for the Semantic Web* July 2002.
- Ray Reiter. "A logic for default reasoning" *Artificial Intelligence* 13:81. 1980.
- Ray Reiter. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press, 2001.
- Greg Restall "Curry's Revenge: the costs of non-classical solutions to the paradoxes of self-reference" (to appear in *The Revenge of the Liar* ed. J.C. Beall. Oxford University Press. 2007) July 12, 2006.
<http://consequently.org/papers/costing.pdf>
- Greg Restall "Proof Theory and Meaning: on Second Order Logic" *Logica 2007 Yearbook*, 2007.

- Edwina Rissland. "The Ubiquitous Dialectic" ECAI'84.
- Abraham Robinson. "Model theory and non-standard arithmetic" in *Infinitistic Methods*. Proceedings of the Symposium on Foundations of Mathematics. September 2-9, 1959. Pergamon Press.
- John Alan Robinson, "A Machine-Oriented Logic Based on the Resolution Principle." CACM. 1965.
- Victor Rodych. "Wittgenstein on Mathematical Meaningfulness, Decidability, and Application" *Notre Dame Journal on Formal Logic* Vol. 38. No. 2. 1997.
- Victor Rodych. "Wittgenstein's Inversion of Gödel's Theorem" *Erkenntnis* 51. 1999.
- Victor Rodych. "Wittgenstein on Gödel: The Newly Published Remarks" *Erkenntnis* 56. 2002.
- Victor Rodych. "Misunderstanding Gödel: New Arguments about Wittgenstein and New Remarks by Wittgenstein" *Dialectica* Vol. 57. No. 3. 2003.
- Bill Roscoe. *The Theory and Practice of Concurrency* Prentice-Hall. Revised 2005.
- Scott Rosenberg. *Dreaming in Code*. Crown Publishers. 2007.
- Marcus Rossberg. "Second-Order Logic" Socrates Teaching Mobility Intensive Seminar, University of Helsinki, 16-19 May, 2005. <http://www.st-andrews.ac.uk/~mr30/SOL/SOL3.pdf>
- John Barkley Rosser. "Extensions of Some Theorems of Gödel and Church" *Journal of Symbolic Logic*. 1(3) 1936.
- Philippe Rouchy (2006). "Aspects of PROLOG History: Logic Programming and Professional Dynamics" *TeamEthno-Online Issue 2*, June 2006.
- Richard Routley "Dialectical Logic, Semantics and Metamathematics" *Erkenntnis* 14. 1979.
- Richard Routley *Relevant Logics and Their Rivals I* Ridgeview. 2003.
- Carlo Rovelli "Relational quantum mechanics" *International Journal of Theoretical Physics*, 1996.
- Carlo Rovelli. "The Uselessness of Certainty" *Edge* 2011.
- Jeff Rulifson, Jan Derksen, and Richard Waldinger. "QA4, A Procedural Calculus for Intuitive Reasoning" SRI AI Center Technical Note 73. November 1973.
- Bertrand Russell. *Principles of Mathematics* Norton. 1903.
- Bertrand Russell. *Principia Mathematica 2nd Edition* 1925.
- Alessandra Russo, Bashar Nuseibeh, and Steve Easterbrook. "Making Inconsistency Respectable in Software Development" *Journal of Systems and Software*. Vol. 56. No. 58. 2000.
- Earl Sacerdoti, *et. al.*, "QLISP A Language for the Interactive Development of Complex Systems" *AFIPS*. 1976.
- Eric Sandewall. "A functional approach to non-monotonic logic" *Computational Intelligence*. Vol. 1. 1985.

- Eric Sandewall. "From Systems to Logic in the Early Development of Nonmonotonic Reasoning" CAISOR. July, 2006.
- Davide Sangiorgi and David Walker. *The Pi-Calculus: A Theory of Mobile Processes* Cambridge University Press. 2001.
- Marek Sergot. "Bob Kowalski: A Portrait" *Computational Logic: Logic Programming and Beyond: Essays in Honour of Robert A. Kowalski, Part I* Springer. 2004.
- R. W. Schwanke and G. E. Kaiser, "Living With Inconsistency in Large Systems" International Workshop on Software Version and Configuration Control. January 1988.
- Dana Scott "Data Types as Lattices". *SIAM Journal on computing*. 1976.
- Dana Scott. "The Future of Proof" LICS 2006.
- Dana Scott and Christopher Strachey. *Toward a mathematical semantics for computer languages* Oxford Programming Research Group Technical Monograph. PRG-6. 1971
- Thoralf Skolem (1920) "Logico-combinatorial investigations on the satisfiability or provability of mathematical propositions: A simplified proof of a theorem by Löwenheim" (English translation in Jean van Heijenoort, 1967. *From Frege to Gödel: A Source Book in Mathematical Logic, 1879-1931*. Harvard Univ. Press)
- Oron Shagrir "Gödel on Turing on Computability" *Church's Thesis after 70 years* Ontos-Verlag. 2006.
- Natarajan Shankar. *Metamathematics, Machines, and Gödel's Proof* Cambridge University Press. 1994.
- Ehud Shapiro. "The family of concurrent logic programming languages" *ACM Computing Surveys*. September 1989
- Stewart Shapiro. *Thinking About Mathematics*. Oxford University Press. 2000.
- Stewart Shapiro. "Lakatos and logic Comments on Graham Priest's '60% proof: Lakatos, proof, and paraconsistency'" Preprint 2006
- Stewart Shapiro *Foundations without Foundationalism: A Case for Second-Order Logic* Oxford. 2002.
- Stewart Shapiro. "Do Not Claim Too Much: Second-order Logic and First-order Logic" *Philosophia Mathematica* (3) Vol. 7. 1999.
- Stuart. Shapiro. "Relevance logic in computer science" in *Entailment, Volume II* pg. 553-563. Princeton University Press. 1992.
- Stuart Shapiro. "SNePS: A Logic for Natural Language Understanding and Commonsense Reasoning" in *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*, AAAI Press. 2000.
- Wilfried Sieg and Clinton Field. "Automated search for Gödel proofs." *Annals of Pure and Applied Logic*. 2005.
- Wilfried Sieg and J. Byrnes "An Abstract Model for Parallel Computations: Gandy's Thesis" *Monist* 1999.

- Wilfried Sieg. "Gödel on Computability" *Philosophia Mathematica* 2006.
- Wilfried Sieg "Church Without Dogma – axioms for computability" *New Computational Paradigms* Springer Verlag. 2008.
- G. R. Simari and R. P. Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence* Vol. 53 No. 2-3 1992.
- John Slaney. "Relevant Logic and Paraconsistency" in *Inconsistency Tolerance* Springer 2004.
- Aaron Sloman. "Must Intelligent Systems Be Scruffy?" *Evolving Knowledge in Natural Science and Artificial Intelligence*. Pitman. 1990.
- Timothy Smiley. "The Logical Basis of Ethics," *Acta Philosophica Fennica*, 16: 1963.
- Peter Smith. *An Introduction to Gödel's Theorems*. Draft. 2006. <http://www.godelbook.net/>
- Lee Smolin. *The Trouble with Physics: The Rise of String Theory, the Fall of a Science, and What Comes Next* Houghton Mifflin. 2006
- Craig Smorynski. "The Incompleteness Theorems" *Handbook of Mathematical Logic*. North Holland. 1977.
- Raymond Smullyan *Gödel's Incompleteness Theorems* Oxford Univ. Press. 1991.
- Michael Smyth. *Power domains* Journal of Computer and System Sciences. 1978.
- Gerry Sussman, Terry Winograd and Eugene Charniak. "Micro-Planner Reference Manual (Update)" AI Memo 203A, MIT AI Lab, December 1971.
- Gerry Sussman and Guy Steele *Scheme: An Interpreter for Extended Lambda Calculus* AI Memo 349. December, 1975. University of Illinois Press. 1977.
- Frederick Suppe, ed. "The Structure of Scientific Theories" University of Illinois Press. 1977.
- Alfred Tarski *Introduction to Logic* Oxford University Press. 1940 (and many subsequent editions).
- Alfred Tarski (1944) "The semantic conception of truth and the foundations of semantics" *Philosophy and Phenomenological Research* 4 (Reprinted in *Readings in Philosophical Analysis*, Appleton-1944)
- Alfred Tarski and Robert Vaught (1957). "Arithmetical extensions of relational systems" *Compositio Mathematica* 13.
- Paul Tillich. "Courage to be" *Yale University Press*. 2000.
- Stephen Toulmin *The Uses of Argument* Cambridge University Press. 1958.
- Alan Turing. "On computable numbers, with an application to the Entscheidungsproblem." *Proceedings London Math Society*. 1936.
- Alan Turing. "Intelligent Machinery". National Physical Laboratory Report. 1948. Also in *Machine Intelligence 5*. Edinburgh: Edinburgh University Press. (Digital facsimile viewable at http://www.AlanTuring.net/intelligent_machinery)

- Shunichi Uchida and Kazuhiro Fuchi (1992). *Proceedings of the FGCS Project Evaluation Workshop* Institute for New Generation Computer Technology (ICOT)
- Moshe Vardi “More Debate, Please!” CACM. Jan. 2010.
- Rineke Verbrugge “Provability Logic” *The Stanford Encyclopedia of Philosophy* 2010.
- John von Neumann. “The role of mathematics in the sciences and in society” *John von Neumann Collected Works Vol. VI*. Pergamon. 1961.
- John von Neumann. “The Mathematician” *John von Neumann Collected Works Vol. I*. Pergamon. 1962.
- Richard Waldinger and R. Lee (1969) “PROW: a step toward automatic program writing” IJCAI’69.
- Douglas Walton *Fundamentals of Critical Argumentation* Cambridge University Press. 2006.
- Hao Wang *A Logical Journey, From Gödel to Philosophy* MIT Press. 1974.
- André Weil, In letter to Fréchet, January 31, 1927..
- Peter Whalley. “Modifying the metaphor in order to improve understanding of control languages—the little-person becomes a cast of actors.” *British Journal of Educational Technology*. 2006.
- John Wheeler. “It from Bit” in *Complexity, Entropy, and the Physics of Information* Addison-Wesley. 1990
- Eugene Wigner. “The Unreasonable Effectiveness of Mathematics in the Natural Sciences” *Communications in Pure and Applied Mathematics* February 1960.
- Bill Wilson. *Twelve Steps and Twelve Traditions* Alcoholics Anonymous. 1952
- Terry Winograd. *Procedures as a Representation for Data in a Computer Program for Understanding Natural Language*. MIT AI TR-235. January 1971.
- Ludwig Wittgenstein. 1956. Bemerkungen über die Grundlagen der Mathematik/*Remarks on the Foundations of Mathematics, Revised Edition* Basil Blackwell. 1978
- Ludwig Wittgenstein. *Philosophische Grammatik* Basil Blackwell. 1969.
- Ludwig Wittgenstein. (1933-1935) *Blue and Brown Books*. Harper. 1965.
- Ludwig Wittgenstein *Philosophical Investigations* Blackwell. 1953/2001.
- John Woods, *Paradox and Paraconsistency* Cambridge University Press. 2003
- Larry Wos, George Robinson, Daniel Carson (1965) “Efficiency and Completeness of the Set of Support Strategy in Theorem Proving” *JACM* 12(4).
- Noson Yanofsky. “A universal approach to self-referential paradoxes, incompleteness and fixed points” *Bulletin of Symbolic Logic* 9 No. 3. 2003.
- Noson Yanofsky. *The Outer Limits of Reason: What Science, Mathematics, and Logic Cannot Tell Us* MIT Press. 2013.

Aki Yonezawa. *Specification and Verification Techniques for Parallel Programs Based on Message Passing Semantics* MIT EECS Ph. D. December 1977.

Ernst Zermelo. "Investigations in the foundations of set theory" (English translation in *From Frege to Gödel: A Source Book in Mathematical Logic, 1879-1931* Ed. Jean van Heijenoort 1967). 1908.

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

APPENDIX 1: DETAILS OF DIRECT LOGIC

Notation of Direct Logic

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

[Tarski 1940]

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

- There are uncountably many Actors.⁵⁴ For example, $\text{Real}_{\bullet}[\]$ can output any real numberⁱ between 0 and 1 where
$$\text{Real}_{\bullet}[\] \equiv [(0 \text{ either } 1), \forall \text{Postpone Real}_{\bullet}[\]]$$
where
 - $(0 \text{ either } 1)$ is the nondeterministic choice of 0 or 1,
 - $[\textit{first}, \forall \textit{rest}]$ is the list that begins with *first* and whose remainder is *rest*, and
 - **Postpone** *expression* delays execution of *expression* until the value is needed.
- There are uncountably many propositions (because there is a different proposition for every real number). Consequently, there are propositions that are not the abstraction of any element of a denumerable set of sentences. For example,
$$p \equiv [x \in \mathbb{R}] \rightarrow ([y \in \mathbb{R}] \rightarrow (y=x))$$
defines a different predicate $p[x]$ for each real number x , which holds for only one real number, namely x .ⁱⁱ

It is important to distinguish between strings, sentences, and propositions. Some strings can be parsed into sentencesⁱⁱⁱ, which can be abstracted into propositions that can be asserted. Furthermore, expressions^{iv} can be abstracted into Actors (*e.g.*, objects in mathematics).

ⁱ using binary representation.

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

ⁱⁱⁱ which are grammar tree structures

^{iv} which are grammar tree structures

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

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

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

For example:

Proposition

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

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

Sentence

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

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

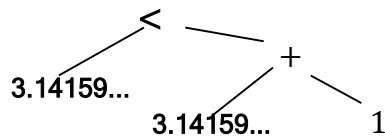
String<Sentence>

e.g. “ $\forall [n:\mathbb{N}] \rightarrow \exists [m:\mathbb{N}] \rightarrow m > n$ ”

*which is a string that for **Sentence** that begins with the symbol “ \forall ”*

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

The sentence $((3.14159... < ((3.14159... + 1)))$ is impossible to obtain by parsing a string (where $3.14159...$ is an Actorⁱ for the transcendental real number that is the ratio of a circle's circumference to its diameter). The issue is that there is no string which when parsed is $((3.14159... < ((3.14159... + 1)))$. Of course, because the digits of $3.14159...$ are computable, there is a expression_1 such that $\lfloor \text{expression}_1 \rfloor = 3.14159...$ that can be used to create the sentence $((\text{expression}_1 < ((\text{expression}_1 + 1)))$. However the sentence $((\text{expression}_1 < ((\text{expression}_1 + 1)))$ is not the same as $((3.14159... < ((3.14159... + 1)))$ because it does not have the same vocabulary and it is a much larger sentence that has many terminals whereas $((3.14159... < ((3.14159... + 1)))$ has just 3 terminals:



Consequently, sentences *cannot* be enumerated and there are some sentences that *cannot* be obtained by parsing strings. These arrangements exclude known paradoxes from Classical Direct Logic.ⁱⁱ

Note: type theory of Classical Direct Logic is much stronger than constructive type theory with constructive logic⁵⁵ because Classical Direct Logic has all of the power of Classical Mathematics.

ⁱ whose digits are incrementally computable

ⁱⁱ Please see historical appendix of this article.

Types and Propositions are defined as follows:

- **Types** *i.e.*, a **Type** is *only* by the rules below:
 - **Boolean**, \mathbb{N} ⁵⁶, **Sentence**, **Proposition**, **Proof**, **Theory**: **Type**
 - If σ_1, σ_2 : **Type**, then $\sigma_1 \oplus \sigma_2$ ⁱ, $[\sigma_1, \sigma_2]$ ⁵⁷, $[\sigma_1] \mapsto \sigma_2$ ⁱⁱ, $\sigma_2^{\sigma_1}$ ⁱⁱⁱ: **Type**
 - If f : **Type**^N, then $(\oplus_{\mathbb{N}} f)$: **Type**^{iv}
 - If σ : **Type**, then **Term** $\langle \sigma \rangle$ ⁵⁸: **Type**
 - If σ : **Type** and e : **Expression** $\langle \sigma \rangle$ with no free variables, then $\lfloor e \rfloor$: σ .
- **Propositions**, *i.e.*, a **Proposition** is *only* by the rules below:
 - If σ : **Type**, Π : **Boolean** ^{σ} and x : σ , then $\Pi[x]$: **Proposition**^v.
 - If Φ : **Proposition**, then $\neg \Phi$: **Proposition**.
 - If Φ, Ψ : **Proposition**, then $\Phi \wedge \Psi$, $\Phi \vee \Psi$, $\Phi \Rightarrow \Psi$, $\Phi \Leftrightarrow \Psi$: **Proposition**.
 - If p : **Boolean** and Φ, Ψ : **Proposition**, then $(p \diamond \text{True} \circlearrowleft \Phi_1 \square \text{False} \circlearrowright \Phi_2)$: **Proposition**⁵⁹.
 - If σ_1, σ_2 : **Type**, x_1 : σ_1 and x_2 : σ_2 , then $x_1 = x_2, x_1 \in x_2, x_1 \sqsubseteq x_2, x_1 \subseteq x_2, x_1 : x_2$: **Proposition**.
 - If T : **Theory**, p : **Proof** and Φ_1 to Φ_n : **Proposition**, then $(\Phi_1, \dots, \Phi_k \vdash_T^p \Phi_{k+1}, \dots, \Phi_n)$: **Proposition**⁶⁰.
 - If s : **Sentence** with no free variables, then $\lfloor s \rfloor$: **Proposition**.

ⁱ For $i=1,2$

- If x : σ_1 , then $(x \odot (\sigma_1 \oplus \sigma_2)) : (\sigma_1 \oplus \sigma_2)$ and $x = (x \odot (\sigma_1 \oplus \sigma_2)) \odot \sigma_1$.
- $\forall [\tau$: **Type**, z : $\tau] \rightarrow z : \sigma_1 \oplus \sigma_2 \Leftrightarrow \exists [x$: $\sigma_1] \rightarrow z = x \odot (\sigma_1 \oplus \sigma_2)$

ⁱⁱ Type of computable procedures from σ_1 into σ_2 .

If σ_1, σ_2 : **Type**, f : $([\sigma_1] \mapsto \sigma_2)$ and x : σ_1 , then $f[x]$: σ_2 .

ⁱⁱⁱ If σ_1, σ_2 : **Type**, f : $\sigma_2^{\sigma_1}$ and x : σ_1 , then $f[x]$: σ_2 .

^{iv} $\forall [i$: \mathbb{N} , x : $f[i]] \rightarrow (x \odot \oplus_{\mathbb{N}} f) : \oplus_{\mathbb{N}} f \wedge x = (x \odot \oplus_{\mathbb{N}} f) \odot f[i]$
 $\forall [\tau$: **Type**, z : $\tau] \rightarrow z : \oplus_{\mathbb{N}} f \Leftrightarrow \exists [i$: \mathbb{N} , x : $f[i]] \rightarrow z = x \odot \oplus_{\mathbb{N}} f$

^v $\forall [\sigma$: **Type**, x : $\sigma] \rightarrow \Pi[x] \Leftrightarrow (\Pi[x] = \text{True})$

Note that σ : **Strict**, Π : **Boolean** ^{σ} means that there are no fixed points for propositions.

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

- **Expressions**, *i.e.*, an **Expression** $\langle\sigma\rangle$ is *only* by the rules below:
 - **True, False:Constant** $\langle\text{Boolean}\rangle$ and **0,1:Constant** $\langle\mathbb{N}\rangle$.
 - **Boolean, N, Sentence, Proposition, Proof, Theory:Constant** $\langle\text{Type}\rangle$.
 - If $\sigma:\text{Type}$ and $x:\text{Constant}\langle\sigma\rangle$, then $x:\text{Expression}\langle\sigma\rangle$.
 - If $\sigma:\text{Type}$ and $x:\text{Variable}\langle\sigma\rangle$, then $x:\text{Expression}\langle\sigma\rangle$.
 - If $\sigma, \sigma_1 \text{ to } n, \tau_1 \text{ to } n:\text{Type}$, $x_1 \text{ to } n:\text{Expression}\langle\sigma_1 \text{ to } n\rangle$ and $y:\text{Expression}\langle\sigma\rangle$, then
 $(\text{Let } v_1 \langle \tau_1 \rangle \equiv x_1, \dots, v_n \langle \tau_n \rangle \equiv x_n^{61} \circ y):\text{Expression}\langle\sigma\rangle$ and
 $v_1 \text{ to } n:\text{Variable}\langle\sigma_1 \text{ to } n\rangle$ in y and in each $x_1 \text{ to } n$.
 - If $e_1, e_2:\text{Expression}\langle\text{Type}\rangle$, then
 $(e_1 \oplus e_2), ([e_1, e_2]), ([e_1] \mapsto e_2), (e_2^{e_1}):\text{Expression}\langle\text{Type}\rangle$.
 - If $\sigma:\text{Type}$, $t_1:\text{Expression}\langle\text{Boolean}\rangle$, $t_2, t_3:\text{Expression}\langle\sigma\rangle$, then
 $(t_1 \diamond \text{True} \circ t_2 \square \text{False} \circ t_3):\text{Expression}\langle\sigma\rangle$.⁶²
 - If $\sigma_1, \sigma_2:\text{Type}$, $t:\text{Expression}\langle\sigma_2\rangle$, then
 $([x:\sigma_1] \rightarrow t):\text{Expression}\langle[\sigma_1] \mapsto \sigma_2\rangle$ and $x:\text{Variable}\langle\sigma_1\rangle$ in t .⁶³
 - If $\sigma_1, \sigma_2:\text{Type}$, $p:\text{Expression}\langle[\sigma_1] \mapsto \sigma_2\rangle$ and $x:\text{Expression}\langle\sigma_1\rangle$, then
 $(p \cdot [x]):\text{Expression}\langle\sigma_2\rangle$.
 - If $\sigma:\text{Type}$ and $s:\text{String}\langle\sigma\rangle$, then $[s]:\text{Expression}\langle\sigma\rangle$.
- **Sentences**, *i.e.*, a **Sentence** is *only* by the rules below:
 - If $s_1:\text{Sentence}$ then, $(\neg s_1):\text{Sentence}$.
 - If $s_1:\text{Sentence}$ and $s_2:\text{Sentence}$ then
 $(s_1 \wedge s_2), (s_1 \vee s_2), (s_1 \Rightarrow s_2), (s_1 \Leftrightarrow s_2):\text{Sentence}$.
 - If $\sigma:\text{Type}$, $t_1:\text{Expression}\langle\text{Boolean}^\sigma\rangle$ and $t_2:\text{Expression}\langle\sigma\rangle$, then
 $(t_1[t_2]):\text{Sentence}$
 - If $t:\text{Expression}\langle\text{Boolean}\rangle$, $s_1, s_2:\text{Sentence}$, then
 $(t \diamond \text{True} \circ s_1 \square \text{False} \circ s_2):\text{Sentence}$.⁶⁴
 - If $\sigma_1, \sigma_2:\text{Type}$, $t_1:\text{Expression}\langle\sigma_1\rangle$ and $t_2:\text{Expression}\langle\sigma_2\rangle$, then
 $(t_1 = t_2), (t_1 \in t_2), (t_1 \sqsubseteq t_2), (t_1 \sqsubseteq t_2), (t_1 : ? t_2):\text{Sentence}$.
 - If $\sigma:\text{Type}$ and $s:\text{Sentence}$, then $(\forall [x:\sigma] \rightarrow s), (\exists [x:\sigma] \rightarrow s):\text{Sentence}$ and
 $x:\text{Variable}\langle\sigma\rangle$ in s .
 - If $T:\text{Expression}\langle\text{Theory}\rangle$, $s_1 \text{ to } n:\text{Sentence}$ and $p:\text{Expression}\langle\text{Proof}\rangle$,
then $(s_1, \dots, s_k \stackrel{p}{\vdash}_T s_{k+1}, \dots, s_n):\text{Sentence}$
 - If $s:\text{String}\langle\text{Sentence}\rangle$, then $[s]:\text{Sentence}$.

ⁱ Because expressions are typed, fixed points do *not* exist. Parameterized mutually recursive definitions are used instead.

Inconsistency Robust Implication

Whether a deductive system is Euclidean or quasi-empirical is decided by the pattern of truth value flow in the system. The system is Euclidean if the characteristic flow is the transmission of truth from the set of axioms ‘downwards’ to the rest of the system—logic here is an organon of proof; it is quasi-empirical if the characteristic flow is retransmission of falsity from the false basic statements ‘upwards’ towards the ‘hypothesis’—logic here is an organon of criticism. [Lakatos 1967]

Inconsistency-robust bi-implication is denoted by \Leftrightarrow .

Logical Equivalence: $(\Psi \Leftrightarrow \Phi) = (\Psi \Rightarrow \Phi) \wedge (\Phi \Rightarrow \Psi)$

Direct Logic has the following rules for inconsistency robust implicationⁱ in theory \mathcal{T} :ⁱⁱ

Reiteration: $\vdash_{\mathcal{T}} \Psi \Rightarrow \Psi$
Exchange: $(\vdash_{\mathcal{T}} \Psi \wedge \Phi \Rightarrow \Theta) \Leftrightarrow \vdash_{\mathcal{T}} \Phi \wedge \Psi \Rightarrow \Theta$
 $(\vdash_{\mathcal{T}} \Theta \Rightarrow \Psi \wedge \Phi) \Leftrightarrow \vdash_{\mathcal{T}} \Theta \Rightarrow \Phi \wedge \Psi$
Dropping: $(\vdash_{\mathcal{T}} \Psi \Rightarrow \Phi \wedge \Theta) \Rightarrow \vdash_{\mathcal{T}} \Psi \Rightarrow \Phi$
 \textcircled{i} *an implication holds if extra conclusions are dropped*
Accumulation: $(\vdash_{\mathcal{T}} \Psi \Rightarrow \Phi, \Psi \Rightarrow \Theta) \Rightarrow \vdash_{\mathcal{T}} \Psi \Rightarrow \Phi \wedge \Theta$
 $(\vdash_{\mathcal{T}} \Phi \Rightarrow \Psi, \Theta \Rightarrow \Psi) \Rightarrow \vdash_{\mathcal{T}} \Phi \vee \Theta \Rightarrow \Psi$
Implication implies inference: $(\vdash_{\mathcal{T}} \Phi \Rightarrow \Psi) \Rightarrow \Phi \vdash_{\mathcal{T}} \Psi$
Transitivity: $(\vdash_{\mathcal{T}} \Psi \Rightarrow \Phi, \Phi \Rightarrow \Theta) \Rightarrow \vdash_{\mathcal{T}} \Psi \Rightarrow \Theta$
 \textcircled{i} *implication in a theory is transitive*
Contrapositive: $(\Psi \Rightarrow \Phi) \Leftrightarrow \neg \Phi \Rightarrow \neg \Psi$
 \textcircled{i} *contrapositive holds for implication*
Implication infers disjunction: $(\Phi \Rightarrow \Psi) \vdash_{\mathcal{T}} \Psi \vee \neg \Phi$

ⁱ denoted by \Rightarrow . Inconsistency-robust implication is different from the much *weaker* concept of non-monotonic consequence [e.g. Kraus, *et. al.* 1990] which has axioms that are *not* valid for inconsistency-robust implication.

ⁱⁱ Inconsistency-robust implication is a very strong relationship. For example, monotonicity does not hold for implication although it does hold for inference. See section on Inconsistency Robust Inference below.

The \vee -rule for **Accumulation** is due to Eric Kao [private communication].

Substitution of Equivalent Propositions

Logical equivalence is defined for propositions for which the usual substitution rules apply:ⁱ

Substitution of equivalent propositions:	
$(\Psi \Leftrightarrow \Phi) \Rightarrow$	$(\neg\Psi) \Leftrightarrow (\neg\Phi)$
$(\Psi \Leftrightarrow \Phi) \Rightarrow$	$((\Psi \vee \Theta) \Leftrightarrow (\Phi \vee \Theta))$
$(\Psi \Leftrightarrow \Phi) \Rightarrow$	$((\Psi \wedge \Theta) \Leftrightarrow (\Phi \wedge \Theta))$
$(\Psi \Leftrightarrow \Phi) \Rightarrow$	$((\Psi \wedge \Theta) \Leftrightarrow (\Phi \wedge \Theta))$
$(\Psi \Leftrightarrow \Phi) \Rightarrow$	$((\Psi \vdash_T \Theta) \Leftrightarrow (\Phi \vdash_T \Theta))$
$(\Psi \Leftrightarrow \Phi) \Rightarrow$	$((\Theta \vdash_T \Psi) \Leftrightarrow (\Theta \vdash_T \Phi))$
$(\Psi \Leftrightarrow \Phi) \Rightarrow$	$((\Psi \Rightarrow \Theta) \Leftrightarrow (\Phi \Rightarrow \Theta))$
$(\Psi \Leftrightarrow \Phi) \Rightarrow$	$((\Theta \Rightarrow \Psi) \Leftrightarrow (\Theta \Rightarrow \Phi))$
$(F \Leftrightarrow G) \Rightarrow$	$(\forall F \Leftrightarrow \forall G)$

Propositional Equivalences

Theorem: The following usual propositional equivalences hold:

- Self Equivalence:** $\Psi \Leftrightarrow \Psi$
- Double Negation:** $\neg\neg\Psi \Leftrightarrow \Psi$
- Idempotence of \wedge :** $\Psi \wedge \Psi \Leftrightarrow \Psi$
- Commutativity of \wedge :** $\Psi \wedge \Phi \Leftrightarrow \Phi \wedge \Psi$
- Associativity of \wedge :** $\Psi \wedge (\Phi \wedge \Theta) \Leftrightarrow (\Psi \wedge \Phi) \wedge \Theta$
- Distributivity of \wedge over \vee :** $\Psi \wedge (\Phi \vee \Theta) \Leftrightarrow (\Psi \wedge \Phi) \vee (\Psi \wedge \Theta)$
- De Morgan for \wedge :** $\neg(\Psi \wedge \Phi) \Leftrightarrow \neg\Psi \vee \neg\Phi$
- Idempotence of \vee :** $\Psi \vee \Psi \Leftrightarrow \Psi$
- Commutativity of \vee :** $\Psi \vee \Phi \Leftrightarrow \Phi \vee \Psi$
- Associativity of \vee :** $\Psi \vee (\Phi \vee \Theta) \Leftrightarrow (\Psi \vee \Phi) \vee \Theta$
- Distributivity of \vee over \wedge :** $\Psi \vee (\Phi \wedge \Theta) \Leftrightarrow (\Psi \vee \Phi) \wedge (\Psi \vee \Theta)$
- De Morgan for \vee :** $\neg(\Psi \vee \Phi) \Leftrightarrow \neg\Psi \wedge \neg\Phi$
- Contrapositive for \Rightarrow :** $(\Psi \Rightarrow \Phi) \Leftrightarrow \neg\Phi \Rightarrow \neg\Psi$

ⁱ Classical implication (denoted by \Rightarrow) is logical implication for classical mathematics. (See the appendix on classical mathematics in Direct Logic.) Likewise classical bi-implication is denoted by \Leftrightarrow .

Direct Logic has the following usual principles for equality:

- $E_1 = E_1$
- $E_1 = E_2 \Rightarrow E_2 = E_1$
- $(E_1 = E_2 \wedge E_2 = E_3) \Rightarrow E_1 = E_3$

Also, the following usual propositional inferences hold:

$$\begin{array}{ll} \text{Absorption of } \wedge: & \Psi \wedge (\Phi \vee \Psi) \vdash_{\mathcal{T}} \Psi \\ \text{Absorption of } \vee: & \Psi \vee (\Phi \wedge \Psi) \vdash_{\mathcal{T}} \Psi^{65} \end{array}$$

Conjunction, i.e., comma

$$\wedge\text{-Elimination: } (\vdash_{\mathcal{T}} \Psi \wedge \Phi) \Rightarrow \vdash_{\mathcal{T}} \Psi, \Phi$$

$$\wedge\text{-Introduction: } (\vdash_{\mathcal{T}} \Psi, \Phi) \Rightarrow \vdash_{\mathcal{T}} \Psi \wedge \Phi$$

Disjunction

$$\vee\text{-Elimination:}^i \vdash_{\mathcal{T}} \neg \Psi \wedge (\Psi \vee \Phi) \Rightarrow \Phi$$

$$\vee\text{-Introduction: } \vdash_{\mathcal{T}} \Psi \wedge \Phi \Rightarrow \Psi \vee \Phi$$

$$\text{Disjunctive Cases: } \vdash_{\mathcal{T}} (\Psi \vee \Phi) \wedge (\Psi \Rightarrow \Theta) \wedge (\Phi \Rightarrow \Omega) \Rightarrow \Theta \vee \Omega$$

Theorem: *Inconsistency Robust Resolution*ⁱⁱ

$$\vdash_{\mathcal{T}} (\Psi \vee \neg \Psi) \wedge (\Psi \vee \Theta) \wedge (\Phi \vee \Omega) \Rightarrow \Theta \vee \Omega$$

Proof: Immediate from Disjunctive Cases and \vee -Elimination.

Inconsistency Robust Inference

Logic merely sanctions the conquests of the intuition.
Jacques Hadamard (quoted in Kline [1972])

Inference in theory \mathcal{T} (denoted by $\vdash_{\mathcal{T}}$) is characterized by the following additional axioms:ⁱⁱⁱ

ⁱ i.e. Disjunctive Syllogism

ⁱⁱ Joint work with Eric Kao

ⁱⁱⁱ Half of the Classical Deduction Theorem holds for Inconsistency Direct Logic. That one proposition infers another in a theory does not in general imply that the first proposition implies the second because Inconsistency Robust Implication is a *very* strong relationship.

Soundness

Soundness: $\vdash_T ((\vdash_T \Psi) \Rightarrow \Psi)$

① *a proposition inferred in a theory implies the proposition in the theory*

Inconsistency Robust Proof by Contradiction

Inconsistency Robust Proof by Contradiction:

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

Quantifiers

Direct Logic makes use of functions for quantification.⁶⁶ For example following expresses commutativity for natural numbers:

$\forall [x,y:\mathbb{N}] \rightarrow x+y=y+x$

Variable Elimination: $\forall F \Rightarrow F[E]$

① *a universally quantified variable of a statement can be instantiated with any expression E (taking care that none of the variables in E are captured).*

Variable Introduction: *Let Z be a new constant, $F[Z] \Leftrightarrow \forall F$*

① *inferring a statement with a universally quantified variable is equivalent to inferring the statement with a newly introduced constant substituted for the variable*

Existential quantification: $\exists F = \neg \forall \neg F$

Appendix 2. Foundations of Classical Mathematics beyond Logicism

Mathematicians do not study objects, but the relations between objects; to them it is a matter of indifference if these objects are replaced by others, provided that the relations do not change. Matter does not engage their attention, they are interested by form alone.

Poincaré [1902]

This appendix presents foundations for mathematics that goes beyond logicism in that it does not attempt to reduce mathematics solely to logic, solely to types, or solely to sets in a way that encompasses all of standard mathematics including the integers, reals, analysis, geometry, etc.⁶⁷

Consistency has been the bedrock of classical mathematics.

Platonic Ideals were to be perfect, unchanging, and eternal.⁶⁸ Beginning with the Hellenistic mathematician Euclid [*circa* 300BC] in Alexandria, theories were intuitively supposed to be consistent.⁶⁹ Wilhelm Leibniz, Giuseppe Peano, George Boole, Augustus De Morgan, Richard Dedekind, Gottlob Frege, Charles Peirce, David Hilbert, *etc.* developed mathematical logic. However, a crisis occurred with the discovery of the logical paradoxes based on self-reference by Burali-Forti [1897], Cantor [1899], Russell [1903], *etc.* In response Russell [1925] stratified types, [Zermelo 1905, Fränkel 1922, Skolem 1922] stratified sets and [Tarski and Vaught 1957] stratified logical theories to limit self-reference. [Church 1935, Turing 1936] proved that closed mathematical theories are inferentially undecidableⁱ, *i.e.*, there are propositions which can neither be proved nor disproved. However, the bedrock of consistency remained.

This appendix present classical mathematics in Direct Logic using \vdash .ⁱⁱ

The following additional principles are available because \vdash is thought to be consistent by an overwhelming consensus of working professional mathematicians:

Classical Proof by Contradiction: $(\Phi \vdash \Psi, \neg \Psi) \vdash \neg \Phi$
i.e., the negation of a proposition can be inferred from inferring a contradiction

Classical Deduction Theorem: $(\Psi \vdash \Phi) \Leftrightarrow \vdash (\Psi \Rightarrow \Phi)$
i.e., an implication can be proved by inference

Classical Proof by Contradiction: $(\Phi \vdash \Psi, \neg \Psi) \vdash \neg \Phi$
i.e., the negation of a proposition can be inferred from inferring a contradiction

Inheritance from classical mathematics

Theorems of mathematics hold in every theory:

If Φ is a proposition of mathematics, $(\vdash \Phi) \Rightarrow (\vdash_T \Phi)$

ⁱ sometimes called “incomplete”

ⁱⁱ with no subscripted inconsistency robust theory, *i.e.*, \vdash is used for classical mathematics whereas \vdash_T is used for inconsistency-robust inference in theory T.

Nondeterministic Execution

Direct Logic makes use of the nondeterministic execution as follows:⁷⁰

- If E_1 and E_2 are *expressions*, then $E_1 \rightarrow E_2$ (E_1 can nondeterministically evolve to E_2) is a *proposition*.
- If E is an *expression*, then $\text{Converges}[E]$ (E always converges) is a *proposition*.

Foundations with both Types and Sets

Classical Direct Logic develops foundations for mathematics using *both*ⁱ typesⁱⁱ and setsⁱⁱⁱ encompassing all of standard mathematics including the integers, reals, analysis, geometry, *etc.*⁷¹

Combining types and sets as the foundation has the advantage of using the strengths of each without the limitations of trying to use just one because each can be used to make up for the limitations of the other. The key idea is compositionality, *i.e.*, composing new entities from others. Types can be composed from other types and sets can be composed from other sets.^{iv}

Functions are fundamental to Computer Science. Consequently, graphs of functions and sets are fundamental collections.⁷² $\text{SetFunctions}\langle\sigma\rangle$ (type of set functions based on type σ) that can be defined inductively as follows:

$$\text{SetFunctionsOfOrder}\langle\sigma\rangle[1] \equiv \sigma^\sigma$$

$$\text{SetFunctionsOfOrder}\langle\sigma\rangle[n+1] \equiv$$

$$(\sigma \oplus \text{SetFunctionsOfOrder}\langle\sigma\rangle[n]) \sigma \oplus \text{SetFunctionsOfOrder}\langle\sigma\rangle[n]$$

ⁱ Past attempts to reduce mathematics to logic alone, to sets alone, or to types alone have not been very successful.

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

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

^{iv} Compositionality avoids standard foundational paradoxes. For example, Direct Logic composes propositions from others using strict types so there are no “self-referential” propositions.

Furthermore the process of constructing orders of $\text{SetFunctionsOfOrder}\langle\sigma\rangle$ is exhaustive for $\text{SetFunctions}\langle\sigma\rangle$:ⁱ

$$\text{SetFunctions}\langle\sigma\rangle \equiv \coprod_{i:\mathbb{N}} \text{SetFunctionsOfOrder}\langle\sigma\rangle[i]$$

Sets (along with lists) provide a convenient way to collect together elements.⁷³ For example, sets (of sets of sets of ...) of σ can be axiomatized as follows:

$$\forall[s:\text{Sets}\langle\sigma\rangle] \rightarrow \exists[f:\text{SetFunctions}\langle\sigma\rangle] \rightarrow \text{CharacteristicFunction}[f, s]$$

$$\text{where } \forall[s:\text{Sets}\langle\sigma\rangle, f:\text{Boolean}^{\text{SetFunctions}\langle\sigma\rangle}] \rightarrow$$

$$\text{CharacteristicFunction}[f, s]$$

$$\Leftrightarrow \forall[e:\sigma \oplus \text{Sets}\langle\sigma\rangle] \rightarrow e \in s \Leftrightarrow f[e]=\text{True}$$

i.e. every set of type $\text{Sets}\langle\sigma\rangle$ is defined by a characteristic function of $\text{SetFunctions}\langle\sigma\rangle$

Note that there is no set corresponding to the *type* $\text{Sets}\langle\mathbb{N}\rangle$ which is an example of how types extend the capabilities of sets.⁷⁴

XML

We speak in strings, but think in trees.

---Nicolaas de Bruijn⁷⁵

The base domain of Direct Logic is XMLⁱⁱ. In Direct Logic, a dog is an XML dog, *e.g.*, $\langle\text{Dog}\rangle\langle\text{Name}\rangle\text{Fido}\langle\text{Name}\rangle\langle\text{Dog}\rangle \in \text{Dogs} \subseteq \text{XML}$

Unlike First Order Logic, there is no unrestricted quantification in Direct Logic. So the proposition $\forall d \in \text{Dogs} \rightarrow \text{Mammal}[d]$ is about dogs in XML. *The base equality built into Direct Logic is equality for XML, not equality in some abstract "domain".* In this way Direct Logic does not have to take a stand on the various ways that dogs, photons, quarks and everything else can be considered "equal"!

This axiomization omits certain aspects of standard XML, *e.g.*, attributes, namespaces, *etc.*

Two XML expressions are equal if and only if they are both atomic and are identical or are both elements and have the same tag and the same number of children such that the corresponding children are equal.

ⁱ The closure property below is used to guarantee that there is just one model of $\text{SetFunctions}\langle\mathbb{N}\rangle$ up to isomorphism using a unique isomorphism.

ⁱⁱ Lisp was an important precursor of XML. The Atomics axiomatised below correspond roughly to atoms and the Elements to lists.

The following are axioms for XML:

$$(\text{Atomics} \cup \text{Elements}) = \text{XML}$$

$$(\text{Atomics} \cap \text{Elements}) = \{\}$$
⁷⁶

$$\text{Tags} \subseteq \text{Atomics}$$

$$\forall [x] \rightarrow x \in \text{Elements} \Leftrightarrow x = \langle \text{Tag}(x) \rangle x_1 \dots x_{\text{Length}(x)} \langle / \text{Tag}(x) \rangle$$

where x_i is the i th subelement of x and

$\text{Tag}(x)$ is the tag of x

$\text{Length}(x)$ is the number of subelements of x

A set $p \subseteq \text{XML}$ is defined to be *inductive* (written $\text{Inductive}[p]$) if and only if it contains the atomics and for all elements that it contains, it also contains every element with those sub-elements:

$$(\forall [p \subseteq \text{XML}; x_1 \dots x_n \in p; t \in \text{Tags}] \rightarrow$$

$$\text{Inductive}[p] \Rightarrow (\text{Atomics} \subseteq p \wedge \langle t \rangle x_1 \dots x_n \langle / t \rangle \in p)$$

The Strong Principle of Induction for XML is as follows:

$$\forall [p \subseteq \text{XML}] \rightarrow \text{Inductive}[p] \Rightarrow p = \text{XML}$$

The reason that induction is called “*strong*” is that there are no restrictions on inductive predicates.⁷⁷

Natural Numbers, Real Numbers, and their Sets are Unique up to Isomorphism

The following question arises: What mathematics have been captured in the above foundations?

Theoremⁱ (Categoricity of \mathbb{N}):⁷⁸ $\forall [M: \text{Model} \langle \mathbb{N} \rangle] \rightarrow M \approx \mathbb{N}$, *i.e.*, models of the natural numbers \mathbb{N} are isomorphic by a unique isomorphism.ⁱⁱ

The following categorical induction axiom⁷⁹ can be used to characterize the natural numbers (\mathbb{N}^{80}) up to isomorphism with a unique isomorphism:

$$\forall [P: \text{Boolean}^{\mathbb{N}}] \rightarrow \text{Inductive}[P] \Leftrightarrow \forall [i: \mathbb{N}] \rightarrow P[i]$$

where $\forall [P: \text{Boolean}^{\mathbb{N}}] \rightarrow$

$$\text{Inductive}[P] \Leftrightarrow P[0] \wedge \forall [i: \mathbb{N}] \rightarrow P[i] \Leftrightarrow P[i+1]^{\text{iii}}$$

ⁱ [Dedekind 1888, Peano 1889]

ⁱⁱ Consequently, the type \mathbb{N} is unique up to isomorphism and the type \mathbb{R} is unique up to isomorphism.

ⁱⁱⁱ which can be equivalently expressed as:

$$\forall [P: \text{Boolean}^{\mathbb{N}}] \rightarrow \text{Inductive}[P] \Leftrightarrow \forall [i: \mathbb{N}] \rightarrow P[i] = \text{True}$$

where $\forall [P: \text{Boolean}^{\mathbb{N}}] \rightarrow$

$$\text{Inductive}[P] \Leftrightarrow (P[0] = \text{True} \wedge \forall [i: \mathbb{N}] \rightarrow P[i] = \text{True} \Leftrightarrow P[i+1] = \text{True})$$

Theoremⁱ (Categoricity of \mathbb{R}):⁸¹ $\forall[M:\mathbf{Model}\langle\mathbb{R}\rangle]\rightarrow M\approx\mathbb{R}$, *i.e.*, models of the real numbers \mathbb{R} are isomorphic by a unique isomorphism.ⁱⁱ

The following can be used to characterize the real numbers (\mathbb{R} ⁸²) up to isomorphism with a unique isomorphism:

$$\begin{aligned} \forall[S:\mathbf{Set}\langle\mathbb{R}\rangle]\rightarrow S\neq\{\} \wedge \text{Bounded}[S] &\Leftrightarrow \text{HasLeastUpperBound}[S] \\ \text{where} \\ \text{UpperBound}[b:\mathbb{R}, S:\mathbf{Set}\langle\mathbb{R}\rangle] &\Leftrightarrow b\in S \wedge \forall[x\in S]\rightarrow x\leq b \\ \text{HasLeastUpperBound}[S:\mathbf{Set}\langle\mathbb{R}\rangle] &\Leftrightarrow \exists[b:\mathbb{R}]\rightarrow \text{LeastUpperBound}[b, S] \\ \text{LeastUpperBound}[b:\mathbb{R}, S:\mathbf{Set}\langle\mathbb{R}\rangle] &\Leftrightarrow \text{UpperBound}[b, S] \wedge \forall[x\in S]\rightarrow \text{UpperBound}[x, S] \Leftrightarrow x\leq b \end{aligned}$$

Theorem (Categoricity of Sets $\langle\mathbb{N}\rangle$):⁸³

$$\forall[M:\mathbf{Model}\langle\mathbf{Sets}\langle\mathbb{N}\rangle\rangle]\rightarrow M\approx\mathbf{Sets}\langle\mathbb{N}\rangle$$

i.e., models of $\mathbf{Sets}\langle\mathbb{N}\rangle$ are isomorphic by a unique isomorphism.ⁱⁱⁱ

$\mathbf{Sets}\langle\mathbb{N}\rangle$ (which is a fundamental type of mathematics) is exactly characterized axiomatically, which is what is required for Computer Science.

Proof: By above, $\forall[M:\mathbf{Model}\langle\mathbb{N}\rangle]\rightarrow M\approx\mathbb{N}$, *i.e.*, models of \mathbb{N} are isomorphic by a unique isomorphism. Unique isomorphism of higher order sets can be proved using induction from the following closure property for SetFunctions (see above):

$$\begin{aligned} \mathbf{SetFunctionsOfOrder}\langle\sigma\rangle[n+1] &\equiv \\ (\sigma\oplus\mathbf{SetFunctionsOfOrder}\langle\sigma\rangle[n])^{\sigma\oplus\mathbf{SetFunctionsOfOrder}\langle\sigma\rangle[n]} & \end{aligned}$$

Unique isomorphism for $\mathbf{SetFunctions}\langle\mathbb{N}\rangle$ can be extended $\mathbf{Sets}\langle\mathbb{N}\rangle$ because every set in $\mathbf{Sets}\langle\mathbb{N}\rangle$ is defined by a characteristic function of $\mathbf{SetFunctions}\langle\mathbb{N}\rangle$ (see above):

Classical Direct Logic is much stronger than first-order axiomatizations of set theory.⁸⁴ Also, the semantics of Classical Direct Logic cannot be characterized using Tarskian Set Models [Tarski and Vaught 1957].^{iv}

ⁱ [Dedekind 1888]

ⁱⁱ Consequently, the type of natural numbers \mathbb{N} is unique up to isomorphism and the type of reals \mathbb{R} is unique up to isomorphism.

ⁱⁱⁱ Consequently, the set of natural numbers \mathbb{N} is unique up to isomorphism and is contained in the set of reals \mathbb{R} that is unique up to isomorphism.

^{iv} See section on “Inadequacy of Tarskian Set Models.”

Theorem (Set Theory Model Soundness): $\vdash_{\text{Sets}\langle\mathbb{N}\rangle}\Psi$ implies $\models_{\text{Sets}\langle\mathbb{N}\rangle}\Psi$

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

Appendix 3. Historical development of inferential undecidability (“incompleteness”)

Truth versus Argumentation

[Peano 1889, Dedekind 1888] made fundamental contributions to the foundations of mathematics with the following theorems:

- *Full Peano Integers:* Let \mathbf{X} be the structure $\langle X, 0_x, S_x \rangle$, then $\text{Peano}[\mathbf{X}] \Rightarrow \mathbf{X} \approx \langle \mathbb{N}, 0, S \rangle$ ⁸⁵ The theory Peano is the full theory of natural numbers with categorical induction that is strictly more powerful than cut-down first-order theory. Perhaps of greater import, there are nondeterministic Turing machines that Peano proves always halt that cannot be proved to halt in the cut-down first-order theory.
- *Full Dedekind Reals:* Let \mathbf{X} be the structure $\langle X, \leq_x, 0_x, 1_x, +_x, *_x \rangle$, then $\text{Dedekind}[\mathbf{X}] \Rightarrow \mathbf{X} \approx \langle \mathbb{R}^1, \leq, 0, 1, +, * \rangle$ ⁸⁶ The theory Dedekind is the full theory of real numbers that is strictly more powerful than cut-down first-order theory.⁸⁷

The above results categorically characterize the natural numbers (integers) and the real numbers up to isomorphism based on *argumentation*. There is no way to go beyond argumentation to get at some special added insight called “*truth*.” Argumentation is all that we have.

ⁱ \mathbb{R} is the set of real numbers

Turing versus Gödel

*You shall not cease from exploration
And the end of all our journeying
Will be to arrive where we started
And know the place for the first time.*
T.S. Eliot [1942]

Turing recognized that proving that inference in mathematics is computationally undecidable is quite different than proving that there is a proposition of mathematics that is inferentially undecidable.ⁱ [Turing 1936, page 259]:

It should perhaps be remarked what I shall prove is quite different from the well-known results of Gödel [1931]. Gödel has shown that there are propositions U such that neither U nor $\neg U$ is provable. ... On the other hand, I shall show that there is no general method which tells whether a given formula U is provable.⁸⁸

Although they share some similar underlying ideas, the method of proving computational undecidability developed by Church and Turing is much more robust than the one previously developed by Gödel that relies on “self-referential” propositions.

The difference can be explicated as follows:

- Actors: an Actor that has an address for itself can be used to generate infinite computations.
- Propositions: “self-referential” propositions can be used to infer inconsistencies in mathematics.

As Wittgenstein pointed out, the following “self-referential” proposition leads an inconsistency in the foundations of mathematics: This proposition is not provable. If the inconsistencies of “self-referential” propositions stopped with this example, then it would be somewhat tolerable for an inconsistency-robust theory. However, other “self-referential” propositions (constructed in a similar way) can be used to prove every proposition thereby rendering inference useless.

This is why Direct Logic does not support “self-referential” propositions.ⁱⁱ

ⁱ sometimes called “incompleteness.”

ⁱⁱ There It seems that are no practical uses for “self-referential” propositions in the mathematical foundations of Computer Science.

Contra Gödel et. al

The proof of the consistency of mathematics in this article contradicts the result [Gödel 1931] using “self-referential” propositions that mathematics cannot prove its own consistency.

One resolution is not to have “self-referential” propositions, which is contra Gödel *et. al.* Direct Logic aims to not have “self-referential” propositions by carefully arranging the rules so that “self-referential” propositions cannot be constructed. The basic idea is to use typed functions [Russell 1908, Church 1940] to construct propositions so that fixed points do not exist and consequently cannot be used to construct “self-referential” propositions.

How the self-proof of consistency of mathematics was overlooked and then discovered

Before the paradoxes were discovered, not much attention was paid to proving consistency. Hilbert *et. al.* undertook to find a *convincing* proof of consistency. Gentzen found a consistency proof for the first-order Peano theory but many did not find it convincing because the proof was not elementary. Then following Carnap and Gödel, philosophers blindly accepted the necessity of “self-referential” propositions in mathematics. And none of them seemed to understand Wittgenstein's critique. (Gödel insinuated that Wittgenstein was “*crazy*.”) Instead, philosophers turned their attention to exploring the question of which is the weakest theory in which Gödel's proof can be carried out. They were prisoners of the existing paradigm.

Computer scientists brought different concerns and a new perspective. They wanted foundations with the following characteristics:

- powerful so that arguments (proofs) are short and understandable and all logical inferences can be formalized
- standard so they can join forces and develop common techniques and technology
- inconsistency robust because computers deal in pervasively inconsistent information.

The results of [Gödel 1931], [Curry 1941], and [Löb 1955] played an important role in the development of Direct Logic:

- Direct Logic easily formalized Wittgenstein's proof that Gödel's “self-referential” proposition leads to contradiction. So the consistency of mathematics had to be rescued against Gödel's “self-referential” proposition. The “self-referential” propositions used in results of [Curry 1941] and [Löb 1955] led to inconsistency in mathematics. So the consistency of mathematics had to be rescued against these “self-referential” propositions as well.
- Direct Logic easily proves the consistency of mathematics. So the consistency of mathematics had to be rescued against Gödel's “2nd incompleteness theorem.”
- Direct Logic easily proves Church's Paradox. So the consistency of mathematics had to be rescued against the assumption that the theorems of mathematics can be computationally enumerated.

In summary, computer science advanced to a point where it caused the development of Direct Logic.

Inconsistency-robust Logic Programs

Logic Programsⁱ can logically infer computational steps.

Forward Chaining

Forward chaining is performed using \vdash

$((\vdash_{aTheory} PropositionExpression): Continuation)$
Assert *PropositionExpression* for *aTheory*.

$((When \vdash_{aTheory} PropositionPattern \rightarrow Expression): Continuation)$
When *PropositionPattern* holds for *aTheory*, evaluate *Expression*.

ⁱ [Church 1932; McCarthy 1963; Hewitt 1969, 1971, 2010; Milner 1972, Hayes 1973; Kowalski 1973]. Note that this definition of Logic Programs does *not* follow the proposal in [Kowalski 1973, 2011] that Logic Programs be restricted only to backward chaining, *e.g.*, to the exclusion of forward chaining, *etc.*

Illustration of forward chaining:

$\vdash_t \text{Human}[\text{Socrates}] \mathbf{!}$

When $\vdash_t \text{Human}[x] \rightarrow \vdash_t \text{Mortal}[x] \mathbf{!}$

will result in asserting $\text{Mortal}[\text{Socrates}]$ for theory t

Backward Chaining

Backward chaining is performed using \Vdash

$((\Vdash_{a\text{Theory}} \text{GoalPattern} \rightarrow \text{Expression})) : \text{Continuation}$
 Set GoalPattern for Theory and when established evaluate Expression .

$(\Vdash_{a\text{Theory}} \text{GoalPattern}) : \text{Expression}$
 Set GoalPattern for Theory and return a list of assertions that satisfy the goal.

(When $\Vdash_{a\text{Theory}} \text{GoalPattern} \rightarrow \text{Expression}) : \text{Continuation}$
 When there is a goal that matches GoalPattern for Theory , evaluate Expression .

Illustration of backward chaining:

$\vdash_t \text{Human}[\text{Socrates}] \mathbf{!}$

When $\Vdash_t \text{Mortal}[x] \rightarrow (\Vdash_t \text{Human}[x] \rightarrow \vdash_t \text{Mortal}[x]) \mathbf{!}$

$\Vdash_t \text{Mortal}[\text{Socrates}] \mathbf{!}$

will result in asserting $\text{Mortal}[\text{Socrates}]$ for theory t .

SubArguments

This section explains how subargumentsⁱ can be implemented in natural deduction.

When $\Vdash_s (\psi \vdash_t \phi) \rightarrow$

Let $t' \leftarrow \text{extension}(t),$

Do $\vdash_{t'} \psi,$

$\Vdash_{t'} \phi \rightarrow \Vdash_s (\psi \vdash_t \phi) \mathbf{!}$

Note that the following hold for t' because it is an extension of t :

- **When** $\vdash_t \theta \rightarrow \vdash_{t'} \theta \mathbf{!}$
- **When** $\Vdash_{t'} \theta \rightarrow \Vdash_t \theta \mathbf{!}$

ⁱ See appendix on Inconsistency Robust Natural Deduction.

End Notes

- ¹ Inference is direct when it does not involve unnecessary circumlocutions, *e.g.*, coding sentences as Gödel numbers. In Direct Logic, it is possible to speak directly about inference relationships.
- ² This section shares history with [Hewitt 2010b]
- ³ D’Ariano and Tosini [2010] showed how the Minkowskian space-time emerges from a topologically homogeneous causal network, presenting a simple analytical derivation of the Lorentz transformations, with metric as pure event-counting.
Do events happen in space-time or is space-time that is made up of events? This question may be considered a “which came first, the chicken or the egg?” dilemma, but the answer may contain the solution of the main problem of contemporary physics: the reconciliation of quantum theory (QT) with general relativity (GR). Why? Because “events” are central to QT and “space-time” is central to GR. Therefore, the question practically means: which comes first, QT or GR? In spite of the evidence of the first position—“events happen in space-time”—the second standpoint—“space-time is made up of events”—is more concrete, if we believe à la Copenhagen that whatever is not “measured” is only in our imagination: space-time too must be measured, and measurements are always made-up of events. Thus QT comes first. How? Space-time emerges from the tapestry of events that are connected by quantum interactions, as in a huge quantum computer: this is the Wheeler’s “It from bit.” [Wheeler 1990].
- ⁴ According to [Law 2006], a classical realism (to which he does not subscribe) is:
Scientific experiments make no sense if there is no reality independent of the actions of scientists: an independent reality is one of conditions of possibility for experimentation. The job of the investigator is to experiment in order to make and test hypotheses about the mechanisms that underlie or make up reality. Since science is conducted within specific social and cultural circumstances, the models and metaphors used to generate fallible claims are, of course, socially contextualized, and always revisable...Different ‘paradigms’ relate to (possibly different parts of) the same world.
- ⁵ Vardi [2010] has defended the traditional paradigm of proving that programs meet specifications and attacked an early critical analysis as follows: “With hindsight of 30 years, it seems that De Millo, Lipton, and Perlis’ [1979] article has proven to be rather misguided.” However, contrary to Vardi, limitations of the traditional paradigm of proving that programs meet specifications have become much more apparent in the last 30 years—as admitted even by some who had been the most prominent proponents, *e.g.*, [Hoare 2003, 2009].

⁶ According to [Hoare 2009]: *One thing I got spectacularly wrong. I could see that programs were getting larger, and I thought that testing would be an increasingly ineffective way of removing errors from them. I did not realize that the success of tests is that they test the programmer, not the program. Rigorous testing regimes rapidly persuade error-prone programmers (like me) to remove themselves from the profession. Failure in test immediately punishes any lapse in programming concentration, and (just as important) the failure count enables implementers to resist management pressure for premature delivery of unreliable code. The experience, judgment, and intuition of programmers who have survived the rigors of testing are what make programs of the present day useful, efficient, and (nearly) correct.*

⁷ According to [Hoare 2009]: *Verification [proving that programs meet specifications] technology can only work against errors that have been accurately specified, with as much accuracy and attention to detail as all other aspects of the programming task. There will always be a limit at which the engineer judges that the cost of such specification is greater than the benefit that could be obtained from it; and that testing will be adequate for the purpose, and cheaper. Finally, verification [proving that programs meet specifications] cannot protect against errors in the specification itself.*

⁸ Popper [1934] section 30.

⁹ The thinking in almost all scientific and engineering work has been that models (also called theories or microtheories) should be internally consistent, although they could be inconsistent with each other.

Indeed some researchers have even gone so far as to construct consistency proofs for some small software systems, *e.g.*, [Davis and Morgenstern 2005] in their system for deriving plausible conclusions using classical logical inference for Multi-Agent Systems. In order to carry out the consistency proof of their system, Davis and Morgenstern make some simplifying assumptions:

- No two agents can simultaneously make a choice (following [Reiter 2001]).
- No two agents can simultaneously send each other inconsistent information.
- Each agent is individually serial, *i.e.*, each agent can execute only one primitive action at a time.
 - There is a global clock time.
 - Agents use classical Speech Acts (see [Hewitt 2006b 2007a, 2007c, 2008c]).
 - Knowledge is expressed in first-order logic.

The above assumptions are not particularly good ones for modern systems (e.g., using Web Services and many-core computer architectures). [Hewitt 2007a]

The following conclusions can be drawn for documentation, use cases, and code of large software systems for human-computer interaction:

- Consistency proofs are impossible for whole systems.
- There are some consistent subtheories but they are typically mathematical. There are some other consistent microtheories as well, but they are small, make simplistic assumptions, and typically are inconsistent with other such microtheories [Addanki, Cremonini and Penberthy 1989].

Nevertheless, the Davis and Morgenstern research programme to prove consistency of microtheories can be valuable for the theories to which it can be applied. Also some of the techniques that they have developed may be able to be used to prove the consistency of the mathematical fragment of Direct Logic and to prove inconsistency robustness (see below in this article).

¹⁰ Turing differed fundamentally on the question of inconsistency from Wittgenstein when he attended Wittgenstein's seminar on the Foundations of Mathematics [Diamond 1976]:

Wittgenstein: ... Think of the case of the Liar. It is very queer in a way that this should have puzzled anyone — much more extraordinary than you might think... Because the thing works like this: if a man says 'I am lying' we say that it follows that he is not lying, from which it follows that he is lying and so on. Well, so what? You can go on like that until you are black in the face. Why not? It doesn't matter. ...it is just a useless language-game, and why should anyone be excited?

Turing: What puzzles one is that one usually uses a contradiction as a criterion for having done something wrong. But in this case one cannot find anything done wrong.

Wittgenstein: Yes — and more: nothing has been done wrong, ... where will the harm come?

Turing: The real harm will not come in unless there is an application, in which a bridge may fall down or something of that sort.... You cannot be confident about applying your calculus until you know that there are no hidden contradictions in it.... Although you do not know that the bridge will fall if there are no contradictions, yet it is almost certain that if there are contradictions it will go wrong somewhere.

Wittgenstein followed this up with [Wittgenstein 1956, pp. 104e–106e]:
Can we say: 'Contradiction is harmless if it can be sealed off'? But what prevents us from sealing it off?.

¹¹ A more conservative axiomatization in Direct Logic is the following:

Policy₁[x] ≡ Sane[x] ⊢_{Catch22} Obligated[x, Fly]

Policy₂[x] ≡ Obligated[x, Fly] ⊢_{Catch22} Fly[x]

Policy₃[x] ≡ Crazy[x] ⊢_{Catch22} ¬Obligated[x, Fly]

Observe₁[x] ≡ ¬Obligated[x, Fly].¬Fly[x] ⊢_{Catch22} Sane[x]

Observe₂[x] ≡ Fly[x] ⊢_{Catch22} Crazy[x]

Observe₃[x] ≡ Sane[x], ¬Obligated[x, Fly] ⊢_{Catch22} ¬Fly[x]

Observe₄ ≡ ⊢_{Catch22} Sane[Yossarian]

Background₂ ≡ ⊢_{Catch22} ¬Obligated[Moon, Fly]

For the more conservative axiomatization above:

⊢_{Catch22} Fly[Yossarian] **but** ⊢_{Catch22} ¬Fly[Yossarian]

⊢_{Catch22} ¬Fly[Yossarian] **but** ⊢_{Catch22} Fly[Yossarian]

But, unlike for the stronger axiomatization using strong implication:

⊄_{Catch22} ¬Obligated[Yossarian, Fly]

⊄_{Catch22} ¬Sane[Yossarian]

¹² Because of the use of a very strong form of implication in the axiomatization, the following can also be inferred:

⊢_{Catch22} ¬Obligated[Yossarian, Fly]

⊢_{Catch22} ¬Sane[Yossarian]

¹³ Philosophers have given the name *a priori* and *a posteriori* to the inconsistency

¹⁴ including entanglement

¹⁵ One possible approach towards developing inconsistency robust probabilities is to attach directionality to the calculations as follows:

$$\mathbf{P1.} \vdash_{\text{Catch22}} \mathbb{P}\text{Sane}[x] \overset{\leq}{\rightarrow} \mathbb{P}\text{Obligated}[x, \text{Fly}]$$

$$\mathbf{P2.} \vdash_{\text{Catch22}} \mathbb{P}\text{Obligated}[x, \text{Fly}] \overset{\leq}{\rightarrow} \mathbb{P}\text{Fly}[x]$$

$$\mathbf{P3.} \vdash_{\text{Catch22}} \mathbb{P}\text{Crazy}[x] \overset{\leq}{\rightarrow} \mathbb{P}\neg\text{Obligated}[x, \text{Fly}]$$

$$\mathbf{S1.} \vdash_{\text{Catch22}} \mathbb{P}\neg\text{Obligated}[x, \text{Fly}] \wedge \neg\text{Fly}[x] \overset{\leq}{\rightarrow} \mathbb{P}\text{Sane}[x]$$

$$\mathbf{S2.} \vdash_{\text{Catch22}} \mathbb{P}\text{Fly}[x] \overset{\leq}{\rightarrow} \mathbb{P}\text{Crazy}[x]$$

$$\mathbf{S3.} \vdash_{\text{Catch22}} \mathbb{P}\text{Sane}[x] \wedge \neg\text{Obligated}[x, \text{Fly}] \overset{\leq}{\rightarrow} \mathbb{P}\neg\text{Fly}[x]$$

$$\mathbf{S4.} \vdash_{\text{Catch22}} \mathbb{P}\text{Sane}[\text{Yossarian}] \Rightarrow 1$$

Consequently, the following inferences hold

$$\mathbf{I1.} \vdash_{\text{Catch22}} \mathbb{P}\text{Obligated}[\text{Yossarian}, \text{Fly}] \Rightarrow 1 \quad \textcircled{i} \mathbf{P1} \text{ and } \mathbf{S4}$$

$$\mathbf{I2.} \vdash_{\text{Catch22}} \mathbb{P}\text{Fly}[\text{Yossarian}] \Rightarrow 1 \quad \textcircled{i} \text{ using } \mathbf{P2} \text{ and } \mathbf{I1}$$

$$\mathbf{I3.} \vdash_{\text{Catch22}} \mathbb{P}\text{Crazy}[\text{Yossarian}] \Rightarrow 1 \quad \textcircled{i} \text{ using } \mathbf{S2} \text{ and } \mathbf{I2}$$

$$\mathbf{I4.} \vdash_{\text{Catch22}} \mathbb{P}\neg\text{Obligated}[\text{Yossarian}, \text{Fly}] \Rightarrow 1 \quad \textcircled{i} \mathbf{P3} \text{ and } \mathbf{I3}$$

$$\mathbf{I5.} \vdash_{\text{Catch22}} \mathbb{P}\neg\text{Fly}[\text{Yossarian}] \Rightarrow 0 \quad \textcircled{i} \mathbf{I4} \text{ and } \mathbf{S3}$$

$$\mathbf{I6.} \vdash_{\text{Catch22}} \mathbb{P}\text{Fly}[\text{Yossarian}] \Rightarrow 1 \quad \textcircled{i} \text{ reformulation of } \mathbf{I5}$$

Thus there is a contradiction in *Catch22* in that both of the following hold in the above:

$$\mathbf{I2.} \vdash_{\text{Catch22}} \mathbb{P}\text{Fly}[\text{Yossarian}] \Rightarrow 1$$

$$\mathbf{I6.} \vdash_{\text{Catch22}} \mathbb{P}\text{Fly}[\text{Yossarian}] \Rightarrow 0$$

However, it is not possible to immediately conclude that $1 \approx 0$ because of the directionality.

¹⁶ In [Law 2006]. Emphases added.

¹⁷ In Latin, the principle is called *ex falso quodlibet* which means that from falsity anything follows.

¹⁸ [Nekham 1200, pp. 288-289]; later rediscovered and published in [Lewis and Langford 1932]

¹⁹ [Pospesel 2000] has discussed extraneous \vee introduction on in terms of the following principle: $\Psi, (\Psi \vee \Phi \vdash \Theta) \vdash \Theta$

However, the above principle immediately derives extraneous \vee introduction when Θ is $\Psi \vee \Phi$. In Direct Logic, argumentation of the above form would often be reformulated as follows to eliminate the spurious Φ middle proposition: $\Psi, (\Psi \vdash \Theta) \vdash \Theta$

²⁰ Direct Logic is distinct from the Direct Predicate Calculus [Ketonen and Weyhrauch 1984].

²¹ The importance of (counter) examples in reasoning was emphasized in [Rissland 1984] citing mathematics, law, linguistics and computer science. According to [Gordon 2009]:

[Toulmin 1958] was one of the first to reflect on the limitations of mathematical logic as a model of rationality in the context of everyday discourse and practical problems. By the 1950s, logic had become more or less synonymous with mathematical logic, as invented by Boole, De Morgan, Pierce, Frege, Hilbert and others, starting in the middle of the nineteenth century. Interestingly, Toulmin proposed legal argumentation as a model for practical reasoning, claiming that normative models of practical reasoning should be measured by the ideals of jurisprudence. [Walton 2006] is a good starting point for getting an overview of the modern philosophy of argumentation.

²² in Rebecca Herold *Managing an Information Security and Privacy Awareness and Training Program* 2005. p. 101.

²³ although there is no claim concerning Euclid's own orientation

²⁴ Cf. "on the ordinary notion of proof, it is compelling just because, presented with it, we cannot resist the passage from premises to conclusion without being unfaithful to the meanings we have already given to the expressions employed in it." [Dummett 1973]

²⁵ Rosemary Redfield. *Arsenic associated bacteria (NASA's claims)* RR Research blog. Dec. 6, 2010.

²⁶ Felisa Wolfe-Simon, et. al. *A bacterium that can grow by using arsenic instead of phosphorus* Science. Dec. 2, 2010.

²⁷ $\text{Consequence}_1 \equiv \text{NaturalDeduction}(\text{Axiom}_2)$

$$= \vdash_{\text{Achilles}}(A, B \vdash_{\text{Achilles}} Z)$$

$$\text{Consequence}_2 \equiv \text{Combination}(\text{Axiom}_1, \text{Consequence}_1)$$

$$= \vdash_{\text{Achilles}} A, B, (A, B \vdash_{\text{Achilles}} Z)$$

$$\text{Consequence}_3 \equiv \text{ForwardChaining}(\text{Consequence}_2)$$

$$= \vdash_{\text{Achilles}} Z$$

$$\text{ProofOfZ}[a_1, a_2] \equiv$$

$$\text{ForwardChaining}[\text{Combination}[a_1, \text{NaturalDeduction}[a_2]]]$$

²⁸ McGee [1985] has challenged modus ponens using an example that can be most simply formalized in Direct Logic as follows:

RepublicanWillWin $\vdash_{McGee} (\neg \text{ReaganWillWin} \vdash_{McGee} \text{AndersonWillWin})$

and $\vdash_{McGee} \text{RepublicanWillWin}$

From the above, in Direct Logic it follows that:

$\neg \text{ReaganWillWin} \vdash_{McGee} \text{AndersonWillWin}$

McGee challenged the reasonableness of the above conclusion on the grounds that, intuitively, the proper inference is that if Reagan will not win, then $\neg \text{AndersonWillWin}$ because Carter (the Democratic candidate) will win. However, in theory *McGee*, it is reasonable to infer *AndersonWillWin* from $\neg \text{ReaganWillWin}$ because *RepublicanWillWin* holds in *McGee*.

McGee phrased his argument in terms of implication which in Direct Logic (see following discussion in this paper) would be as follows:

$\vdash_{McGee} \text{RepublicanWillWin} \Rightarrow (\neg \text{ReaganWillWin} \Rightarrow \text{AndersonWillWin})$

However, this makes no essential difference because, in Direct Logic, it still follows that $\vdash_{McGee} (\neg \text{ReaganWillWin} \Rightarrow \text{AndersonWillWin})$

²⁹ [cf. Church 1934, Kleene 1936]

³⁰ Direct inference is defined differently in this paper from probability theory [Levy 1977, Kyburg and Teng 2001], which refers to “*direct inference*” of frequency in a reference class (the most specific class with suitable frequency knowledge) from which other probabilities are derived.

³¹ [Jaśkowski 1934]³¹ that doesn’t require artifices such as indices (labels) on propositions or restrictions on reiteration

³² This section of the paper shares some history with [Hewitt 2010b].

³³ Turing [1936] stated:

- *the behavior of the computer at any moment is determined by the symbols which he [the computer] is observing, and his ‘state of mind’ at that moment*
- *there is a bound B to the number of symbols or squares which the computer can observe at one moment. If he wishes to observe more, he must use successive observations.*

Gödel's conception of computation was formally the same as Turing but more reductionist in motivation:

There is a major difference between the historical contexts in which Turing and Gödel worked. Turing tackled the Entscheidungsproblem [computational decidability of provability] as an interesting mathematical problem worth solving; he was hardly aware of the fierce foundational debates. Gödel on the other hand, was passionately interested in the foundations of mathematics. Though not a student of Hilbert, his work was nonetheless deeply entrenched in the framework of Hilbert's finitistic program, whose main goal was to provide a meta-theoretic finitary proof of the consistency of a formal system "containing a certain amount of finitary number theory." Shagrir [2006]

³⁴ According to [Turing 1948]:

LCMs [Logical Computing Machines: Turing's expression for Turing machines] can do anything that could be described as ... "purely mechanical"... This is sufficiently well established that it is now agreed amongst logicians that "calculable by means of an LCM" is the correct accurate rendering [of phrases like "purely mechanical"]

³⁵ [Wang 1974, p. 84]

³⁶ An example of the global state space model is the Abstract State Machine (ASM) model [Blass, Gurevich, Rosenzweig, and Rossman 2007a, 2007b; Glausch and Reisig 2006].

³⁷ This result is very old. It was known by Dijkstra motivating his belief that it is impossible to implement unbounded nondeterminism. Also the result played a crucial role in the invention of the Actor Model in 1972.

Consider the following Nondeterministic Turing Machine:

Step 1: Next do either *Step 2* or *Step 3*.

Step 2: Next do *Step 1*.

Step 3: Halt.

It is possible that the above program does not halt. It is also possible that the above program halts.

Note that above program is not equivalent to the one below in which it is not possible to halt:

Step 1: Next do *Step 1*.

³⁸ The below proof is quite general and applies to the Abstract State Machine (ASM) model [Blass, Gurevich, Rosenzweig, and Rossman 2007a, 2007b; Glausch and Reisig 2006], which consequently are not really models of concurrency. It also applies to the parallel lambda calculus, which includes all the capabilities of the nondeterministic lambda calculus. Researchers (before the Actor Model was invented) hypothesized that the parallel lambda calculus naturally modeled all of computation and their research programme was to reduce all computation to the parallel lambda calculus [Scott and Strachey 1971, Milne and Strachey 1976].

³⁹ This proof does not apply to extensions of Nondeterministic Turing Machines that are provided with a new primitive instruction `NoLargest` which is defined to write an unbounded large number on the tape. Since executing `NoLargest` can write an unbounded amount of tape in a single instruction, executing it can take an unbounded time during which the machine cannot read input.

Also, the `NoLargest` primitive is of limited practical use. Consider a Nondeterministic Turing Machine with two input-only tapes that can be read nondeterministically and one standard working tape.

It is possible for the following program to copy both of its input tapes onto its working tape:

Step 1: Either

1. copy the current input from the 1st input tape onto the working tape and next do *Step 2*,

or

2. copy the current input from the 2nd input tape onto the working tape and next do *Step 3*.

Step 2: Next do Step 1.

Step 3: Next do Step 1.

It is also possible that the above program does not read any input from the 1st input tape (*cf.* [Knabe 1993]) and the use of `NoLargest` is of no use in alleviating this problem. Bounded nondeterminism is a symptom of deeper underlying issues with Nondeterministic Turing Machines.

⁴⁰ Consequently,

- The tree has an infinite path. \Leftrightarrow The tree is infinite. \Leftrightarrow It is possible that P does not halt. If it is possible that P does not halt, then it is possible that the set of outputs with which P halts is infinite.
- The tree does not have an infinite path. \Leftrightarrow The tree is finite. \Leftrightarrow P always halts. If P always halts, then the tree is finite and the set of outputs with which P halts is finite.

⁴¹ Arbiters render meaningless the states in the Abstract State Machine (ASM) model [Blass, Gurevich, Rosenzweig, and Rossman 2007a, 2007b; Glausch and Reisig 2006].

⁴² The logic gates require suitable thresholds and other characteristics.

⁴³ *cf.* denotational semantics of the lambda calculus [Scott 1976]

⁴⁴ Proof: Suppose to obtain a contraction that
 ComputationallyDecidable[HaltingProblem].

Define the program Diagonal as follows:

Diagonal $\equiv [x] \rightarrow \text{Halt}_\bullet[x, x] \diamond \text{True} \text{;} \text{InfiniteLoop}_\bullet[] \boxtimes \text{False} \text{;} \text{True}$
 where $\text{InfiniteLoop} \equiv [] \rightarrow \text{InfiniteLoop}_\bullet[]$

Poof of inconsistency: By the definition of Diagonal:

$[\text{Diagonal}]_\bullet[\text{Diagonal}] \rightarrow_1 \text{Halt}_\bullet[\text{Diagonal}, \text{Diagonal}] \diamond$
 $\text{True} \text{;} \text{InfiniteLoop}_\bullet[] \boxtimes$
 $\text{False} \text{;} \text{True}$

Consider the following 2 cases:

1. $\text{Halt}_\bullet[\text{Diagonal}, \text{Diagonal}] \rightarrow_1 \text{True}$
 Converges $[\text{Diagonal}]_\bullet[\text{Diagonal}]$ by the axioms for Halt
 $\neg \text{Converges}[\text{Diagonal}]_\bullet[\text{Diagonal}]$ by the definition of Diagonal
2. $\text{Halt}_\bullet[\text{Diagonal}, \text{Diagonal}] \rightarrow_1 \text{False}$
 $\neg \text{Converges}[\text{Diagonal}]_\bullet[\text{Diagonal}]$ by the axioms for Halt
 Converges $[\text{Diagonal}]_\bullet[\text{Diagonal}]$ by the definition of Diagonal

Consequently, $\neg \text{ComputationallyDecidable}[\text{HaltingProblem}]$

⁴⁵ Note that this theorem is very different from the result [Kleene 1938], that mathematics can be extended with a proposition asserting its own consistency.

⁴⁶ A prominent logician referee of this article suggested that if the proof is accepted then consistency should be made an explicit premise of every theorem of classical mathematics!

⁴⁷ As shown above, there is a simple proof in Classical Direct Logic that Mathematics (\vdash) is consistent. If Classical Direct Logic has a bug, then there might also be a proof that Mathematics is inconsistent. Of course, if a such a bug is found, then it must be repaired.

Fortunately, Classical Direct Logic is simple in the sense that it has just *one* fundamental axiom:

$\forall [P:\text{Boolean}^{\mathbb{N}}] \rightarrow \text{Inductive}[P] \Leftrightarrow \forall [i:\mathbb{N}] \rightarrow P[i]$
where $\forall [P:\text{Boolean}^{\mathbb{N}}] \rightarrow$

$\text{Inductive}[P] \Leftrightarrow (P[0] \wedge \forall [i:\mathbb{N}] \rightarrow P[i] \Leftrightarrow P[i+1])$

Of course, Classical Direct Logic has machinery in addition the above axiom that could also have bugs.

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

In reaction to paradoxes, philosophers developed the dogma of the necessity of strict separation of “object theories” (theories about basic

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

⁴⁸ As shown above, there is a simple proof in Classical Direct Logic that Mathematics (\vdash) is consistent. If Classical Direct Logic has a bug, then there might also be a proof that Mathematics is inconsistent. Of course, if a such a bug is found, then it must be repaired.

Fortunately, Classical Direct Logic is simple in the sense that it has one fundamental axiom:

$$\forall [P:\mathbf{Boolean}^{\mathbf{N}}] \rightarrow \text{Inductive}[P] \Leftrightarrow \forall [i:\mathbf{N}] \rightarrow P[i]$$

where $\forall [P:\mathbf{Boolean}^{\mathbf{N}}] \rightarrow$

$$\text{Inductive}[P] \Leftrightarrow P[0] \wedge \forall [i:\mathbf{N}] \rightarrow P[i] \Leftrightarrow P[i+1]$$

Of course, Classical Direct Logic has machinery in addition the above axiom that could also have bugs.

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

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

⁴⁹ This coordination can include calendars and to-do lists, communications (including email, SMS, Twitter, Facebook), presence information (including who else is in the neighborhood), physical (including GPS recordings), psychological (including facial expression, heart rate, voice

stress) and social (including family, friends, team mates, and colleagues), maps (including firms, points of interest, traffic, parking, and weather), events (including alerts and status), documents (including presentations, spreadsheets, proposals, job applications, health records, photos, videos, gift lists, memos, purchasing, contracts, articles), contacts (including social graphs and reputation), purchasing information (including store purchases, web purchases, GPS and phone records, and buying and travel habits), government information (including licenses, taxes, and rulings), and search results (including rankings and rating).

⁵⁰ In 1994, Alan Robinson noted that he has “*always been a little quick to make adverse judgments about what I like to call ‘wacko logics’ especially in Australia...I conduct my affairs as though I believe ... that there is only one logic. All the rest is variation in what you’re reasoning about, not in how you’re reasoning ... [Logic] is immutable.*” (quoted in Mackenzie [2001] page 286)

On the other hand Richard Routley noted:

... classical logic bears a large measure of responsibility for the growing separation between philosophy and logic which there is today... If classical logic is a modern tool inadequate for its job, modern philosophers have shown a classically stoic resignation in the face of this inadequacy. They have behaved like people who, faced with a device, designed to lift stream water, but which is so badly designed that it spills most of its freight, do not set themselves to the design of a better model, but rather devote much of their energy to constructing ingenious arguments to convince themselves that the device is admirable, that they do not need or want the device to deliver more water; that there is nothing wrong with wasting water and that it may even be desirable; and that in order to “improve” the device they would have to change some features of the design, a thing which goes totally against their engineering intuitions and which they could not possibly consider doing. [Routley 2003]

⁵¹ According to [Kuhn 1962 page 151]

And 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.”

⁵² It is not possible to guarantee the consistency of information because consistency testing is computationally undecidable even in logics much weaker than first order logic. Because of this difficulty, it is impractical to test whether information is consistent.

⁵³ Consequently iDescriber makes use of direct inference in Direct Logic to reason more safely about inconsistent information because it omits the rules of classical logic that enable every proposition to be inferred from a single inconsistency.

⁵⁴ By the *Computational Representation Theorem* [Clinger 1981; Hewitt 2006], which can define all the possible executions of a procedure.

⁵⁵ e.g. [Shulman 2012, nLab 2014]

⁵⁶ \mathbf{N} is the type of Natural Number axiomatized in this article.

⁵⁷ type of 2-element list with first element of type σ_1 and with second element of type σ_2

⁵⁸ type of term of type σ

⁵⁹ *if* \mathbf{t} *then* Φ_1 *else* Φ_2

⁶⁰ Φ_1, \dots and Φ_k infer $\Psi_1, \dots,$ and Ψ_n

⁶¹ parameterized mutually recursive definitions of $v_{1 \text{ to } n} \langle \tau_{1 \text{ to } n} \rangle$

⁶² *if* \mathbf{t}_1 *then* \mathbf{t}_2 *else* \mathbf{t}_3

⁶³ Because there is no type restriction, fixed points may be freely used to define recursive procedures on expressions.

⁶⁴ *if* \mathbf{t} *then* \mathbf{S}_1 *else* \mathbf{S}_1

⁶⁵ Proof: $(\Psi \vee (\Phi \wedge \Psi)) \Leftrightarrow (\Psi \vee \Phi) \wedge (\Psi \vee \Psi) \Leftrightarrow (\Psi \vee \Phi) \wedge \Psi$

⁶⁶ Direct Logic uses the full meaning of quantification as opposed to a cut down syntactic variant, e.g., [Henken 1950]. Disadvantages of the Henkin approach are explained in [Restall 2007].

⁶⁷ [Church 1956; Concoran 1973, 1980; Boulos 1975; Shapiro 2002]

⁶⁸ *“The world that appears to our senses is in some way defective and filled with error, but there is a more real and perfect realm, populated by entities [called “ideals” or “forms”] that are eternal, changeless, and in some sense paradigmatic for the structure and character of our world. Among the most important of these [ideals] (as they are now called, because they are not located in space or time) are Goodness, Beauty, Equality, Bigness, Likeness, Unity, Being, Sameness, Difference, Change, and Changelessness. (These terms — “Goodness”, “Beauty”, and so on — are often capitalized by those who write about Plato, in order to call attention to their exalted status; ...) The most fundamental distinction in Plato’s philosophy is between the many observable objects that appear beautiful (good, just, unified, equal, big) and the one object that is what Beauty (Goodness, Justice, Unity) really is, from which those many beautiful (good, just, unified, equal, big) things receive their names and their corresponding characteristics. Nearly every major work of Plato is, in some way, devoted to or dependent on this distinction.*

Many of them explore the ethical and practical consequences of conceiving of reality in this bifurcated way. We are urged to transform our values by taking to heart the greater reality of the [ideals] and the defectiveness of the corporeal world.” [Kraut 2004]

⁶⁹ Structuralism takes a different view of mathematics:

The structuralist vigorously rejects any sort of ontological independence among the natural numbers. The essence of a natural number is its relations to other natural numbers. The subject matter of arithmetic is a single abstract structure, the pattern common to any infinite collection of objects that has a successor relation, a unique initial object, and satisfies the induction principle. The number 2 is no more and no less than the second position in the natural number structure; and 6 is the sixth position. Neither of them has any independence from the structure in which they are positions, and as positions in this structure, neither number is independent of the other. [Shapiro 2000]

⁷⁰ Basic axioms are as follows:

$$\mathbf{True} \diamond \mathbf{True} \circledast E_1 \boxtimes \mathbf{False} \circledast E_2 \rightarrow E_1$$

$$\mathbf{False} \diamond \mathbf{False} \circledast E_1 \boxtimes \mathbf{True} \circledast E_2 \rightarrow E_1$$

$$\mathbf{False} \diamond \mathbf{True} \circledast E_1 \boxtimes \mathbf{False} \circledast E_2 \rightarrow E_2$$

$$\mathbf{True} \diamond \mathbf{False} \circledast E_1 \boxtimes \mathbf{True} \circledast E_2 \rightarrow E_2$$

$$(E_1 \rightarrow E_2) \wedge (E_2 \rightarrow E_3) \Rightarrow (E_1 \rightarrow E_3)$$

$$([x] \rightarrow F[x])[E] \rightarrow F[E]$$

$$(E_1 \mathbf{either} E_2) \rightarrow E_1^{70}$$

$$(E_1 \mathbf{either} E_2) \rightarrow E_2^{70}$$

$$F_1 \rightarrow F_2 \Rightarrow F_1(E) \rightarrow F_2(E)$$

① *an application evolves if its operator evolves*

$$E_1 \rightarrow E_2 \Rightarrow F(E_1) \rightarrow F(E_2)$$

① *an application evolves if its operand evolves*

$$E_1 \rightarrow E_2 \Rightarrow (\downarrow E_2 \Rightarrow \downarrow E_1)$$

$$E_1 \downarrow E_2 \Leftrightarrow ((E_1 \rightarrow E_2 \wedge \downarrow E_2) \vee (\downarrow E_1 \wedge E_1 = E_2))$$

$$E \downarrow_1 \Leftrightarrow (E \downarrow \wedge (E \downarrow E_1 \wedge E \downarrow E_2) \Rightarrow E_1 = E_1)$$

$$\downarrow E_1 \Rightarrow \neg (E_1 \rightarrow E_2)$$

⁷¹ [Church 1956; Boolos 1975; Corcoran 1973, 1980]

⁷² along with lists

⁷³ $\text{Set}\langle\sigma\rangle$ is the type of a set of type σ , $\text{Sets}\langle\sigma\rangle$ is the type all sets of sets over type σ , and $\text{Domain}\langle\sigma\rangle = \sigma \oplus \text{Sets}\langle\sigma\rangle$ with the following axioms:

- $\{\}: \text{Set}\langle\sigma\rangle$ ① the empty set $\{\}$ is a set
- $\forall[x:\sigma] \rightarrow \{x\}: \text{Set}\langle\sigma\rangle$ ① a singleton set is a set
- $\forall[s:\text{Sets}\langle\sigma\rangle] \rightarrow \cup s: \text{Sets}\langle\sigma\rangle$ ① all elements of the subsets of a set is a set
- $\forall[x:\sigma] \rightarrow x \notin \{\}$ ① the empty set $\{\}$ has no elements
- $\forall[s:\text{Set}\langle\sigma\rangle, f:\sigma^\sigma] \rightarrow (\text{Elementwise}[f])[s]: \text{Set}\langle\sigma\rangle$
① the function image of a set is a set
- $\forall[s:\text{Set}\langle\sigma\rangle, p:\text{Boolean}^\sigma] \rightarrow s \upharpoonright p: \text{Set}\langle\sigma\rangle$
① a predicate restriction of a set is a set
- $\forall[s:\text{Set}\langle\sigma\rangle] \rightarrow \{\} \subseteq s$ ① $\{\}$ is a subset of every set

$$\forall[s_1, s_2: \text{Set}\langle\sigma\rangle] \rightarrow s_1 = s_2 \Leftrightarrow (\forall[x:\sigma] \rightarrow x \in s_1 \Leftrightarrow x \in s_2)$$

$$\forall[x, y: \sigma] \rightarrow x \in \{y\} \Leftrightarrow x = y$$

$$\forall[s_1, s_2: \text{Set}\langle\sigma\rangle] \rightarrow s_1 \subseteq s_2 \Leftrightarrow \forall[x:\sigma] \rightarrow x \in s_1 \Rightarrow x \in s_2$$

$$\forall[x:\sigma; s_1, s_2: \text{Set}\langle\sigma\rangle] \rightarrow x \in s_1 \cup s_2 \Leftrightarrow (x \in s_1 \vee x \in s_2)$$

$$\forall[x:\sigma; s_1, s_2: \text{Set}\langle\sigma\rangle] \rightarrow x \in s_1 \cap s_2 \Leftrightarrow (x \in s_1 \wedge x \in s_2)$$

$$\forall[x:\text{Domain}\langle\sigma\rangle; s:\text{Sets}\langle\sigma\rangle] \rightarrow x \in \cup s \Leftrightarrow \exists[s_1:\text{Sets}\langle\sigma\rangle] \rightarrow x \in s_1 \wedge s_1 \in s$$

$$\forall[y:\sigma; s:\text{Set}\langle\sigma\rangle, f:\sigma^\sigma] \rightarrow$$

$$y \in (\text{Elementwise}[f])[s] \Leftrightarrow \exists[x \in s] \rightarrow f[x] = y$$

$$\forall[y:\sigma; s:\text{Set}\langle\sigma\rangle, p:\text{Boolean}^\sigma] \rightarrow y \in s \upharpoonright p \Leftrightarrow y \in s \wedge p[y]$$

The natural numbers are axiomatised as follows where Successor is the successor function:

- $0:\mathbb{N}$
- Successor: $\mathbb{N}^{\mathbb{N}}$
- $\forall[i:\mathbb{N}] \rightarrow \text{Successor}[i] \neq 0$
- $\forall[i, j:\mathbb{N}] \rightarrow \text{Successor}[i] = \text{Successor}[j] \Rightarrow i = j$
- $\forall[P:\text{Boolean}^{\mathbb{N}}] \rightarrow \text{Inductive}[P] \Rightarrow \forall[i:\mathbb{N}] \rightarrow P[i]$

where

$$\forall[P:\text{Boolean}^{\mathbb{N}}] \rightarrow \text{Inductive}[P]$$

$$\Leftrightarrow P[0] \wedge \forall[i:\mathbb{N}] \rightarrow P[i] \Rightarrow P[\text{Successor}[i]]$$

⁷⁴ I.e., $\nexists[s:\text{Sets}\langle\mathbb{N}\rangle, \sigma:\text{Type}] \rightarrow \forall[x:\sigma] \rightarrow x \in s \Leftrightarrow x:\text{Sets}\langle\mathbb{N}\rangle$

⁷⁵ Quoted by Bob Boyer [personal communication 12 Jan. 2006].

⁷⁶ *Atoms and Elements are disjoint*

⁷⁷ For example, there is no restriction that an inductive predicate must be defined by a first order proposition.

⁷⁸ [Dedekind 1888], [Peano 1889], and [Zermelo 1930].

⁷⁹ [Dedekind 1888, Peano 1889]

⁸⁰ \mathbb{N} is identified with the type of natural numbers

⁸¹ [Dedekind 1888], [Peano 1889], and [Zermelo 1930].

⁸² \mathbb{R} is identified with the type of real numbers

⁸³ *cf.* [Zermelo 1930].

⁸⁴ The Continuum Hypothesis remains an open problem for Direct Logic because its set theory is very powerful. The forcing technique used to prove the independence of the Continuum Hypothesis for first-order set theory [Cohen 1963] does not apply to Direct Logic because of the categorical induction axiom [Dedekind 1888, Peano 1889] used in formalizing the natural numbers \mathbb{N} , which is the foundation of set theory. Of course, trivially,

$(\models_{\text{Domain}\langle\mathbb{N}\rangle} \text{ContinuumHypothesis}) \vee (\models_{\text{Domain}\langle\mathbb{N}\rangle} \neg \text{ContinuumHypothesis})$

where $\text{Domain}\langle\sigma\rangle = \sigma \oplus \text{Sets}\langle\sigma\rangle$.

⁸⁵ $\text{Peano}[\mathbf{X}]$, means that \mathbf{X} satisfies the full Peano axioms for the non-negative integers, \mathbb{N} is the type of non-negative integers, s is the successor function, and \approx means isomorphism.

The isomorphism is proved by defining a function f from \mathbb{N} to \mathbf{X} by:

1. $f[0] = 0_{\mathbf{X}}$
2. $f[S[n]] = S_{\mathbf{X}}[f[n]]$

Using proof by induction, the following follow:

1. f is defined for every element of \mathbb{N}
2. f is one-to-one

Proof:

First prove $\forall [n \in X] \rightarrow f[n] = 0_x \Rightarrow n = 0$

Base: Trivial.

Induction: Suppose $f[n] = 0_x \Rightarrow n = 0$

$f[S[n]] = S_x[f[n]]$ Therefore if $f[S[n]] = 0_x$ then $0_x = S_x[f[n]]$
which is an inconsistency

Suppose $f[n] = f[m]$. To prove: $n = m$

Proof: By induction on n :

Base: Suppose $f[0] = f[m]$. Then $f[m] = 0_x$ and $m = 0$ by above

Induction: Suppose $\forall [m \in \mathbb{N}] \rightarrow f[n] = f[m] \Rightarrow n = m$

Proof: By induction on m :

Base: Suppose $f[n] = f[0]$. Then $n = m = 0$

Induction: Suppose $f[n] = f[m] \Rightarrow n = m$

$f[S[n]] = S_x[f[n]]$ and $f[S[m]] = S_x[f[m]]$

Therefore $f[S[n]] = f[S[m]] \Rightarrow S[n] = S[m]$

3. the range of f is all of X .

Proof: To show: $\text{Inductive}[\text{Range}[f]]$

Base: To show $0_x \in \text{Range}[f]$. Clearly $f[0] = 0_x$

Induction: To show $\forall [n \in \text{Range}[f]] \rightarrow S_x[n] \in \text{Range}[f]$.

Suppose that $n \in \text{Range}[f]$. Then there is some m such that $f[m] = n$.

To prove: $\forall [k \in \mathbb{N}] \rightarrow f[k] = n \Rightarrow S_x[n] \in \text{Range}[f]$

Proof: By induction on k :

Base: Suppose $f[0] = n$. Then $n = 0_x = f[0]$ and

$S_x[n] = f[S[0]] \in \text{Range}[f]$

Induction: Suppose $f[k] = n \Rightarrow S_x[n] \in \text{Range}[f]$

Suppose $f[S[k]] = n$. Then $n = S_x[f[k]]$ and

$S_x[n] = S_x[S_x[f[k]]] = S_x[f[S[k]]] = f[S[S[k]]] \in \text{Range}[f]$

⁸⁶ Dedekind[X], means that X satisfies the Dedekind axioms for the real numbers

⁸⁷ Robinson [1961]

⁸⁸ The inferability problem is to computationally decide whether a proposition defined by sentence is inferable.

Theorem [Church 1935 followed by Turing 1936]:

$\text{Consistent}_{\mathcal{T}} \Rightarrow \neg \text{ComputationallyDecidable}[\text{InferenceProblem}_{\mathcal{T}}]$

Proof. Suppose to obtain a contradiction that

$\text{ComputationallyDecidable}[\text{InferenceProblem}_{\mathcal{T}}]$.

This means that there is a total computational deterministic predicate $\text{Inferable}_{\mathcal{T}}$ such that the following 3 properties hold

1. $\text{Inferable}_{\mathcal{T}}[\Psi] \rightarrow_1 \text{True} \Leftrightarrow \vdash_{\mathcal{T}} \Psi$
2. $\text{Inferable}_{\mathcal{T}}[\Psi] \rightarrow_1 \text{False} \Leftrightarrow \not\vdash_{\mathcal{T}} \Psi$
3. $\text{Inferable}_{\mathcal{T}}[\Psi] \rightarrow_1 \text{True} \vee \text{Inferable}_{\mathcal{T}}[\Psi] \rightarrow_1 \text{False}$

The proof proceeds by showing that if inference is computationally decidable, the halting problem is computationally decidable.

Consider proposition of the form $\text{Converges}[\text{Lp}_{\bullet}[x]]$, which is the proposition that the program p halts on input x .

Lemma: $\text{Consistent}_{\mathcal{T}} \Rightarrow \text{Inferable}_{\mathcal{T}}[\text{Converges}[\text{Lp}_{\bullet}[x]]] \rightarrow_1 \text{True}$
if and only if $\text{Converges}[\text{Lp}_{\bullet}[x]]$

Proof of lemma: Suppose $\text{Consistent}_{\mathcal{T}}$

1. Suppose $\text{Inferable}_{\mathcal{T}}[\text{Converges}[\text{Lp}_{\bullet}[x]]] \rightarrow_1 \text{True}$. Then $\vdash_{\mathcal{T}} \text{Converges}[\text{Lp}_{\bullet}[x]]$ by definition of $\text{Inferable}_{\mathcal{T}}$. Suppose to obtain a contradiction that $\neg \text{Converges}[\text{Lp}_{\bullet}[x]]$. The contradiction $\not\vdash_{\mathcal{T}} \text{Converges}[\text{Lp}_{\bullet}[x]]$ follows by consistency of \mathcal{T} .
2. Suppose $\text{Converges}[\text{Lp}_{\bullet}[x]]$. Then $\vdash_{\mathcal{T}} \text{Converges}[\text{Lp}_{\bullet}[x]]$ by Adequacy of \mathcal{T} . It follows that $\text{Inferable}_{\mathcal{T}}[\text{Converges}[\text{Lp}_{\bullet}[x]]] \rightarrow_1 \text{True}$.

But this contradicts $\neg \text{ComputationallyDecidable}[\text{HaltingProblem}]$ because $\text{Halt}[p, x] \Leftrightarrow \text{Inferable}_{\mathcal{T}}[\text{Converges}[\text{Lp}_{\bullet}[x]]]$

Consequently,

$\text{Consistent}_{\mathcal{T}} \Rightarrow \neg \text{ComputationallyDecidable}[\text{InferenceProblem}_{\mathcal{T}}]$