Philosophers' Imprint

volume 16, no. 13
august, 2016

COMPUTATION IN NON-CLASSICAL FOUNDATIONS?

Toby Meadows *Zach Weber*

University of Queensland *University of Otago*

© 2016, Philosophers' Imprint
This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 3.0 License
<www.philosophersimprint.org/016013/>

Comparing the incomparable

Different logics, different computers?

The Church-Turing Thesis is one of the great success stories in twentieth-century analytic philosophy and mathematical logic. It is a harmonious conceptual and formal analysis of what it means for a procedure to be, in principle, computable. But it is now the twenty-first century. There is now a great variety of logics [Priest, 2008], some proposed as serious rivals to classical logic (e.g. [Beall, 2009]; [Field, 2008]; [Routley et al., 1982]); or, even more daringly, as genuinely co-equal notions of logical consequence [Beall and Restall, 2006]. With this plurality of logics in view, it becomes plausible that there could be a corresponding plurality of notions of computation [Sylvan and Copeland, 2000].[1]

Initially, one would want to know the status of the Church-Turing Thesis (CTT) in non-classical settings, to know to what extent the classical theory of computation survives under change of logic. But — and this is the main point of our paper — the question is *itself* sensitive to changes in logic. Before we can answer it, we need some guidelines for how we should even go about asking. Setting up a fair test for viability requires some fixed reference to test against, but fixing a reference (for instance, insisting that any legitimate proof relation must be undecidable) inherently privileges some frameworks over others. How can we test, with minimal bias, which non-classical logics support a robust notion of computation?

On the plurality of logics

The Church-Turing Thesis is an instance of a more general project, of taking informal philosophical notions and making them formally precise. The informal notion of *logical consequence* or *validity* has been a similar center of a such a project. But unlike the consensus story of the CTT, after over a century of research in non-classical logic, there

1. Our discussion is orthogonal to the literature on hypercomputation; see [Copeland and Sylvan, 1999], [Ord and Kieu, 2005].

is a growing sense that this plurality of logic is to be taken seriously.[2] The idea has emerged that logical validity itself may be polysemous, a cluster concept [Shapiro, 2014]. One need not be a radical to recognise that there is earnest disagreement amongst learned people on what is necessary and sufficient to determine validity — that non-classical logics have "been seriously proposed by (a minority of) expert logicians, and rationally debated" [Williamson, 2012]. These divergent theories are "studied by mathematicians whose credentials can hardly be challeneged" [Shapiro, 2014, p.38]. It is not a big leap from here to the (pessimistic?) induction that there is, as Smith puts it [Smith, 2011, p.27],

> no hope for a[n] argument to show that our initial inchoate, shifting, intuitions about validity — such as they are — succeed in pinning down a unique extension.

Smiley [Smiley, 1998] offers a nice image: "[T]hose who blithely appeal to an 'intuitive' or 'pre-theoretic' idea of consequence are likely to have got hold of just one strand in a string of diverse theories."

Logical pluralism is often *contrasted* with the absolutism of computation; as it is often presented, the CTT would be a highly unusual example in the history of science, where it is much more common that multiple non-equivalent models fit the data [Shapiro, 2014, p.47]. But this contrast does not withstand much scrutiny, as we will see. As Shapiro puts it [Shapiro, 2014, p.48],

> It seems counterproductive — to the philosophical and scientific purposes at hand — to insist that [other models of computability] are incorrect, and should give way to the One True Model of computability. It might be better to think of [alternatives] as different ways to sharpen or model the intuitive notion.

If consequence may be plural, then computability may be, too.

2. It's even been in the newspaper [Williamson, 2012]. Cf. [Williamson, 2014].

Frameworks and convergence

Let us call the combination of a logic and foundational theory a *framework*. Different frameworks deliver different results (see 2.2 below), so it is natural to ask about the prospects for comparison. Computability is a particularly useful case study: it serves both as a good example and a baseline requirement for a reasonable foundation. If a foundation is able to capture computability, it can represent the syntax of languages and replicate mechanical manipulations thereof.

It is not yet clear if many non-classical frameworks can do this.[3] While employing logics *weaker* than classical is known to avoid the semantic and set-theoretic paradoxes generated by "naive" principles of truth and set theory, much less is known about how well the revised logics make for an effective foundations.[4]

The most compelling evidence for the Church-Turing Thesis has been *convergence theorems*, which show that all known formalisations of computation are equivalent. On the one hand, we are highly sympathetic to the idea that any logical foundation must deliver on a theory of computation, in the form of convergence theorems; this is a reasonable test for the viability of a proposed framework. On the other hand, we are highly sympathetic to the idea that robust non-classicality about logic can give rise to incomparable but serious notions of computability; in which case, if the CTT turned out to be a peculiarly classical result, its importance becomes more limited.

We will show the limits of pinning down adequacy conditions for a non-classical system *in as neutral a fashion as possible*. How to do this without begging questions is a very difficult problem. Even if one framework were, from its own perspective, properly stronger than another and able to represent the other as a sub-theory, we argue that this does nothing to advance the conversation, since from the perspective

3. Some "proof of concept" results are in [Bacon, 2013].
4. The non-classicist chimes in that, in light of the paradoxes, it is not clear how well *classical* logic facilitates the development of an effective foundation — but we leave this aside; see [Priest, 2006, ch.1–3]. For one of the more developed non-classical foundations in naive set theory, see [Brady, 2001].

of the “sub"-theory, the “stronger" theory is simply *false*. (And there is nothing very impressive about a false theory being able to imitate in parts a true one.) Simply having this problem in detailed view can help the conversation move forward. Our goal is not to finish the story, but rather to lay the dialectical groundwork to tell it.

1. The Church-Turing Thesis revisited

To begin, let us look at the CTT and what is required to establish it.

1.1 *The idea of the CTT and its justification*

As is well-known, the Church-Turing thesis provides a bridge between informal and formal notions of computation. So, to start, we first need a couple of definitions.[5]

Definition 1 *A function is* informally computable *if it takes a finite amount of input and can be calculated by an ideal person in a finite amount of time on a finite amount of paper by following a finite set of instructions.*

This informal definition is given using mathematically imprecise notions like ‘ideal person’, ‘calculated’, ‘paper’ and ‘instructions’ — to say nothing of ‘finite’. In contrast, the formal definition is much less vague (pending a definition of ‘Turing machine’).

Definition 2 *A function is* formally computable *if there is a Turing machine that can compute it.*

The Church-Turing Thesis is then the substantial claim that the mathematically precise definition captures a philosophically nebulous concept:

CTT All informally computable functions are formally computable functions.

5. These are really two definitions of computation, rather than definitions of two different kinds of computation.

There are a few ways to argue the CTT. From the armchair, the conceptual analysis involved in defining a Turing machine is a kind of argument showing that any informally computable function is formally computable. Empirically, many observe that after almost eighty years of robust research, no counter-examples (putatively, a function that is informally computable but not formally computable) have emerged (see [Copeland and Sylvan, 1999], [Smith, 2007, ch.34]).

The most compelling evidence by far, though, is from *convergence*. There are many different ways of formally representing computable functions: Turing machines; μ-recursive functions; register machines; Kleene’s equational characterisation; and λ-calculus. Despite this diversity, they can all be shown to give the same results. Every way that people have come up with to capture the informal notion of computation has turned out to be equivalent.

To illustrate, and to make the discusssion ahead more concrete, let us see that any Turing machine can simulate any register machine, and vice versa. A *Turing machine*:

- *operates upon* an infinite tape of cells which contain finitely many cells marked “1" and the rest blank
- *can* move left, move right, write, and erase

The machine is given instructions in the form of an *algorithm*, which is a set of instruction-quadruples each composed of: a label; the contents of a cell; a basic operation to perform; and a label for the next action to execute.[6] A *register machine*

- *operates upon* a finite number of buckets
- *can* move stones from one bucket to another

Turing machines operate on a piece of tape, while register machines move stones between registers. To compare them, we find a way for

6. An additional requirement is consistency: an algorithm does not lead to more than one output; we will revisit this assumption later. For details, see [Rogers, 1967, p.13–15]; [Smith, 2007, p.290].

Turing machines to *represent* register machines and a way for register machines to represent Turing machines. Suppose a register machine's starting state consists of three registers containing two, five and seven stones respectively.

B1	B2	B3
2	5	7

In order to simulate the register machine on a Turing machine, represent this on the tape:

1	1	-	1	1	1	1	1	-	1	1	1	1	1	1	1

Then for every possible register algorithm, we prove by induction (starting from the small initial number of operations) that there is a Turing algorithm which replicates it. Similarly, in the other direction, we show a way of representing the computation space of Turing machines using register machines,[7] and a way of mimicking any Turing algorithm by a register algorithm. All up, with both directions of representation given, Turing machines and register machines *converge*.

1.2 Convergence in general: FAC-structures

Let's look at this from a more general perspective, to get at a tool for considering formal analyses of computation and their role in establishing convergence theorems. There are three key components involved in a formal analysis of computation:

7. To do this we represent the state of the tape as a number and then place this number of stones in the first register. For example, suppose the initial statement of the tape was as follows:

1	-	1	1	-	1	1

Then, working backwards and representing blanks by 0s, we get 1101101 which we take as a numeral. In base-2, the number is represented as $2^6 + 2^5 + 2^3 + 2^2 + 2^0 = 109$.

1. a *space* for the computation;
2. a list of possible ways of *manipulating* the computation space; and
3. a list of possible *instruction* sets.

The space of computation can be represented as the collection of states which a computation could reach; to use a little set-theoretic vocabulary, the collection of states is represented as a *set*, X. We then represent ways of manipulating the space with a set F of partial functions $f : X \rightharpoonup X$.[8] And finally, we shall have a set I of instructions for manipulating the space, *algorithms*. We then write

$$\mathcal{A} = \langle X, F, I \rangle$$

to represent a particular formal analysis of computation and call this a *FAC-structure*. Returning to the Turing example: the different states of the Turing tape would be X; F is the set of functions taking the Turing tape from one state to another; and I is the set of possible instructions for Turing machines.[9]

A flag: We have started talking about sets. But in any standard set theory, there are no sets which are Turing tapes or register buckets! So already at this level, a non-trivial amount of machinery is being called upon for representation; we return to this point in 2.1 below.

With *FAC*-structures in hand, we may characterise convergence between models of computation as follows:

8. That is, functions that do not return an output in X for every input from X. These functions must be partial, since some sets of instructions are insufficient always to determine an output; though see 2.2.3 below.
9. For any algorithm $a \in I$, there will be exactly one partial function $f \in F$ which is such that for any input state x, $f(x)$ is the correct output state that occurs (if there is one) when a is correctly followed. There may, however, be more than one algorithm representing any particular partial function $f \in F$.

Definition 3 *Let $\mathcal{A} = \langle X_{\mathcal{A}}, F_{\mathcal{A}}, I_{\mathcal{A}} \rangle$ and $\mathcal{B} = \langle X_{\mathcal{B}}, F_{\mathcal{B}}, I_{\mathcal{B}} \rangle$ be two different formal analyses of computation. We say that $\mathcal{A}_1$ and $\mathcal{A}_2$* converge*,*

$$\mathcal{A} \simeq \mathcal{B},$$

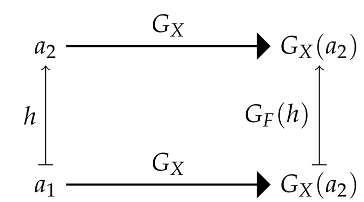
if there exist functions

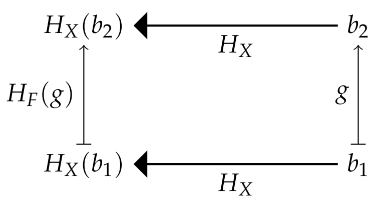
$$G_X : X_{\mathcal{A}} \longrightarrow X_{\mathcal{B}}; \quad G_F : F_{\mathcal{A}} \longrightarrow F_{\mathcal{B}}; \quad H_X : X_{\mathcal{B}} \longrightarrow X_{\mathcal{A}}; \quad H_F : F_{\mathcal{B}} \longrightarrow F_{\mathcal{A}}$$

such that:

1. *for all $h \in F_{\mathcal{A}}$ and for all $a_1, a_2 \in X_{\mathcal{A}}$, $h(x) = y$ iff $G_F(h)G_X(a_1) = G_X(a_2)$; and*
2. *for all $g \in F_{\mathcal{B}}$ and all $b_1, b_2 \in X_{\mathcal{B}}$, $g(b_1) = b_2$ iff $H_F(g)H_X(b_1) = H_X(b_2)$.*

In a diagram:





Informally speaking, G is a formal representation of the way in which we represent one analysis of computation using another.[10] Background details fixed, the official claim is:

Claim 4 *All formal analyses of informal computation converge.*

This claim is not a mathematical theorem. Given any pair of *FAC*-structures, we may prove that they converge. But there is no mathematically precise test for which *FAC*-structures really are a formal analysis of computation.[11] Rather we have an *ad hoc* enumeration of analyses provided by experts working independently on the question. And yet all these analyses turn out to be equivalent. For this reason, the CTT is widely held to be true.

So goes the standard story. Let us now reconsider matters from a non-classical perspective.

2. The challenge: a 'non-classical' Church-Turing Thesis?

2.1 Reasoning non-classically about buckets and tapes

The CTT presents a challenge for a non-classical logician. The stakes are high: CTT looks like a true philosophical *result*, one that has radically advanced our understanding. The stakes are high enough that one could reasonably *dismiss* any purported logico-mathematical framework for which CTT fails, as having strayed beyond the pale and not requiring further serious engagement.

---

10. For example, if $\mathcal{A}$ is the Turing machine representation, then a particular Turing machine can be described by an algorithm $a \in I_{\mathcal{A}}$ giving a partial function $g \in F_{\mathcal{A}}$. Particular states of the tape can then be represented by $x, y \in X_{\mathcal{A}}$. Now suppose that when presented with state x, the Turing machine represented by g outputs y; i.e., $g(x) = y$. Then $G_X(x)$ and $G_X(y)$ are the representation of the Turing tape states as states of the registers, and $G_F(g)$ is Register machine corresponding to g.

11. It is not difficult to satisfy the requirements of being a *FAC*-structure. It is easy to cook up function sets that are not plausibly computable. We are not saying that being a *FAC*-structure in itself gives us a good approximation of computation. The relationship with informal computation is given by conceptual analysis and is not visible from this perspective.

But how could such a thing even happen? After all, a common thought is that convergence theorems are proven *informally*, without foundational background; cf. [Smith, 2007, ch.34–35]. The reasoning is in a pre-theoretic, jargony-but-natural language — and since all that is required is plausibility, not proof, the CTT is established before any logical paraphernalia even kicks in. Aren't the tools required for a convergence theorem so simple that we don't *need* to lay out our foundational assumptions?

Well, sober as it sounds, it only seems unnecessary to formalise the CTT argument given a presupposed background, most likely of classical orthodox set theory, *ZFC*. That is what most philosophers were trained in, and it remains an unreflective fallback. But — insofar as that little bit of pop-psychology is accurate — the very preconditions for the possibility of a (philosophically interesting) non-classical logic is that our informal reasoning practices are up for discussion, and we cannot presume some large settled core like *ZFC*. After all, it may turn out that some logics are too weak to argue convergence. To advance the conversation, we are forced to make our foundations more specific: to lay them bare.

There are, *prima facie*, multiple foundational frameworks with which to approach computation, and so, *prima facie*, multiple notions of computation generated. As a first pass for thinking about this question, imagine three worlds. In one, classical logic was dominant in the twentieth century; but in another, fuzzy logic/set theory won the day, and became the orthodoxy; and in another, intuitionistic logic/set theory was preferred. Then three counterparts of Alan Turing are working in each world, formulating the notions of "informal effectiveness" and "recursively computable" in each. Three very natural and very difficult questions quickly emerge:

1. In which worlds can we, from outside, see that any formalisations of computability will be equivalent?
2. In which worlds can it be proven, from inside that world, that "all formalisations of computability are equivalent"?
3. In which worlds can it be represented what is going on in other worlds?

To foreshadow section 3, a way of understanding what is at stake here is through the *de re*/*de dicto* distinction [Quine, 1976]: roughly, the difference between "outer" and "inner" quantification: "there is something that I can prove to be such-and-so" versus "I can prove there is something that is such-and-so". One might hope that frameworks may be compared by evaluating claims of the form:

> For any two formalisations of computation, framework $\mathcal{F}$ proves that they are equivalent.

For this to work, mathematical objects must be stable across frameworks, allowing the claims to be *de re*. But the frameworks are foundational. They determine (or at least characterise) their respective domains of mathematical objects; different frameworks may have different domains. This makes it look like the claims should be:

> $\mathcal{F}$ proves that any two formalisations of computation are equivalent.

We cannot shift from *de dicto* to *de re* claims here: the second formulation does not imply the first, unless the relevant notions can come out from the scope of a provability operator.

The main burden of this section is to provide evidence that no such shift should be expected.

2.2 *The instability of mathematical objects across frameworks*

How radical could a non-classical foundational framework be? A major difference will be the objects recognised by a theory: the ontology. Orthodox *ZFC* set theory has an ontology, about which other formulations of set theory diverge: about the existence of a universal set, unrestricted complements, sets that are members of themselves, and so forth. In a "naive" set theory, there is a set of all sets not members of themselves — and we all know what happens then; but a non-classical logic makes the Russell set tolerable (e.g. [Priest, 2006, ch.18]). Not only does the ontology of these theories differ, but the range of possible properties that objects can take on is expanded.

The same kind of shift can be expected in what might be called a "naive" theory of computability, for example in the infamous *halting problem*. Consider the Turing machine

H halts on input x iff (the machine with code name) x does not halt on input x

Classically, H cannot be a Turing machine: H will halt on $\ulcorner H \urcorner$ iff H does not halt on $\ulcorner H \urcorner$, a contradiction. Non-classically, though, we might want to maintain that any apparently describable algorithmic process determines a computation, and allow that H exists. Then its apparently impossible behaviour could be reclassified as merely novel non-classical properties, e.g. the machine is not deterministic [Agudelo and Carnielli, 2010]; cf. 2.2.3 below. The point for now is not what to make of such a bizarre object, only to flag that there are frameworks in which it might be allowed to exist.

Let us consider a few examples to this effect. In each case below, some basic differences about logical operators (conjunction, negation, implication, identity) have drastic down-stream effects for mathematical objects, like sets and numbers. The sequence of subsections here shows that what is proved *inside* a framework may be so divergent that it cannot be easily transferred *between* frameworks.

2.2.1 *Sets and functions*

Sets themselves are theory-laden. For instance, consider a framework within the linear logic tradition, in which the logical rule of contraction is not valid: $p \wedge p$ is not equivalent to p; see [Petersen, 2000]. So now, an exercise: For some set a, is it the case that $a = a \cap a$? There's no problem with $a \cap a \subseteq a$, but for the other direction we'd need to show that $x \in a$ implies $x \in a \wedge x \in a$. Without contraction, the identity does not obtain.

This is a disagreement with the classical theory of sets at the level of finite intersection, something that could be drawn with a Venn diagram. The question: Is a person who is proving the CTT using non-contractive set theory to represent buckets and tapes using the same sets?

Similarly, for ordered pairs, one expects

$$\langle a, b \rangle = \langle c, d \rangle \qquad \texttt{iff} \qquad a = c \wedge b = d$$

to obtain. Again, in a non-contractive theory, this may or may not be possible, beginning with the reduction of ordered pairs: the common definition $\{\{a\}, \{a, b\}\}$ is not equivalent to the more advantageous $\{\{\{a\}, \emptyset\} \{b\}\}$, as shown in [Petersen, 2000, p.380]. Since functions — univocal relations — are developed out of ordered pairs, this opens a way for more than one non-equivalent notion of 'function'.[12]

2.2.2 *Numbers and identity*

If objects as basic as sets could vary between frameworks, then perhaps it is not surprising that "the natural numbers" may differ between frameworks. Consider Peano Arithmetic in the relevant logic R, known as R# [Meyer and Mortensen, 1984]. This is given by the standard *PA* axioms, but with all the material implications replaced by relevant implications. A domain of numbers is described. In relevant logic, it is possible that, given numbers $n, m \in \mathbb{N}$, it can be that $n = m$ and $n \neq m$ are both satisfiable, without it following that $0 = 1$. (That would be an irrelevant inference.) Rather stunningly, then, R# can have *finite* models [Priest, 2006, ch.17]. All the expected classical non-identities still hold — for instance, $1 \neq 2$ — but it is nevertheless possible that their negations hold, too. From the classical perspective, it looks like distinct objects are being mis-identified. From a (deeply committed) R#

12. A most familiar example is constructive mathematics, which since Brouwer has been been working with a notion of function such that all functions are continuous [Bridger, 2007, ch.4, appendix 1].

perspective, one non-self-identical number has been mis-interpreted as many numbers.

Again the question: Is a person using R# to count the pebbles in a register machine counting the same way? Do they even mean the same thing by 'counting' as someone using classical PA?

By the same token, one of the conditions on being a Turing machine is consistency: there are no two outputs p, q of an algorithm on the same input such that $p \neq q$. In a dialetheic set theory [Restall, 1992], there are objects (like the Russell set) such that $x \neq x$. The consistency requirement can therefore be *met*, to the letter if not the spirit of the law, by a construction in the logic LP [Priest, 1979] where programs are not (consistently) consistent! Let us skip the details and just get to the question: Is the LP programmer even writing a program?

2.2.3 *Algorithms*

A non-classicist is rethinking some of the basic decisions made in the twentieth century about recursion theory. One such decision is prompted by a standard diagonal argument, saying that some procedures do not determine a function, but only a *partial* recursive function. Following [Rogers, 1967], Sylvan and Copeland [Sylvan and Copeland, 2000] use the same facts to argue instead for a "paradox of all algorithms". Consider a list of all computable functions in one argument $f_0, f_1, \ldots$.. Consider the function

$$g(z) = f_z(x) + 1$$

This is a computable function: to compute it, first compute $f_z(x)$ and then add 1. Therefore it is on the list of all computable functions: $g = f_k$ for some k. At which point

$$f_k(k) = g(k) = f_k(k) + 1$$

Sylvan and Copeland conclude from this that the list of all computable functions is inconsistent — that "there are more algorithmic functions than all algorithmic functions" (p.194), and, for good measure, there may be some natural number such that $n = n + 1$ (which sounds absurd, but would be technically possible in paraconsistent arithmetic [Priest, 2006, ch.17]).

Orthodox recursion theory of course has a reply to this line of thought [Rogers, 1967, p.11]. Our only point at the moment is that, to the extent that what they propose is cogent, Sylvan and Copeland are heading for a very unorthodox notion of 'computable function'. Without rashly ruling out their approach a priori, we can ask: Is someone who takes it to be decidable whether or not a set of instructions constitutes a total recursive function working with anything like classical recursive functions?

2.3 *Taking plurality seriously*

From an orthodox standpoint, it is tempting to answer all the above questions rather swiftly. Someone who takes "functions" (with scare quotes) to be inconsistent is not talking about functions (no scare quotes); "sets" without contraction are multi-sets, not sets; etc. The justification for this swift appraisal is that classical *ZFC* objects just *are* sets, classical *ZFC*'s ω just *is* the natural numbers, and in general, *ZFC* objects are canonical mathematical objects. Any other objects, like those described above, may exist, but are e.g. some sort of multi-sets, or fragment of the numbers, or something else entirely — but certainly not genuine sets or numbers! This attitude is tempting because it organises otherwise alien practices in a familiar way. It might be described as the principle of charity, to say with Quine that deviants do not change the logic, but only the subject [Quine, 1986, p.61–94]. Let us urge against charity here.

2.3.1 *Translations?*

There is nothing wrong with rephrasing unfamiliar notions into a familiar language, to make them accessible. But it would be a mistake — "dogmatic and pointlessly controversial" [Williamson, 2014, p.214], says Williamson — to exclude the original intent as mistaken, or think that this rephrasing can dispense with the original, un-reconstructed idea. Translation theorems can suggest that, e.g., intuitionistic negation is really just the $\Box\neg p$ of the classical modal logic S4 (for which it is obvious that

$$p \vee \Box\neg p$$

fails; not everything is either true or *necessarily* false). But this is hardly a fair explication of Brouwer's intuitionism; if an S4 frame is used to model intuitionistic arithmetic, for example, there is a (platonistic) completed infinity at every node of the frame [Shapiro, 2014, p.34–35]. Be it 'negation', 'set', or other crucial notion, insisting that the non-classical object is really a crypto-classical object is to reduce this entire literature to a merely verbal dispute. Lewis cautions that

> to suppose that [a non-classicist] mistakes mere terminological difference for profound philosophical disagreement is to accuse him of stupidity far beyond belief [Lewis, 1990, p.30].

Just because one can interpret another's language does not mean that interpretation is semantically faithful. There's a world of difference between an interpretation where terms and descriptions are in harmony, and an interpretation done upon some abstract scratch pad in some disused corner of one's foundational framework; cf. [Shapiro, 2014, ch.4].

Cross-framework foundations is not the place for invoking overly charitable principles of charity. If we disambiguate by giving away names (there is classical negation, intuitionistic negation, dialetheic negation, etc.;) then the dispute is not about names, but reference: Which one delivers the correct theory of negation? Of sets? Of computation? These are substantive questions that cannot be translated away.

2.3.2 *Classical default?*

The standing offers for cross-cultural discourse in the literature assume that, by and large, classical logic is perfectly fine and safe to use. This idea is invoked particularly at the level of metametatheory — that is, the plane of reasoning between frameworks. For example, Bacon has developed an algebraic metatheory for non-classical theories that is itself, purportedly, non-classical. But there is a caveat:

> I am assuming that the theory of syntax would be free of the kind of phenomena responsible for nonclassicality, allowing one to assume classical logic about syntax [Bacon, 2013, p.347, fn12].

All the main strategies on offer are some variation on this thought. For example, Field, in the context of giving a non-classical account of validity, writes:

> the non-classical logician needn't doubt that classical logic is "effectively valid" in the part of set theory ... [that] suffices for giving a model-theoretic account of validity for the logic that is at least extensionally correct [Field, 2008, p.109].

If we are confident that the domain in which we are working is consistent, for example, then ex falso quodlibet can be deployed reliably; if we can presume bivalence, then double negation elimination is fine; and metatheoretic reasoning conditions allow for all this.

How tenable is this idea? In [Field, 2008], a paracomplete theory of truth is proposed. In [Beall, 2009], a paraconsistent theory is similarly proposed. Both argue that, due to paradox, classical logic is not correct (in very different ways). Nevertheless, both crucially avail themselves of classical model theory in *ZFC* in order to build models of these theories, in the Kripkean fixed-point style. The problem is simply that now the justification offered *for* the theory is not justified according *to* the theory itself. So the proffered theory of truth — paid for by a logic revision, in the hopes of retaining a universal truth predicate — turns out not to be the whole story about truth. However much one can say in reply to this charge, it certainly *seems* bad. A classical meta-theorist simply does not take non-classical logic as seriously as the rhetoric elsewhere insists one should.[13]

An all-else-fails suggestion is that the non-classicist can view the *ZFC* models as a temporary communication device, a ladder for climbing up over classical accounts that will be eventually kicked away. Let us simply assume (quite an assumption) that with enough work it could be shown that a classical framework can build up and legitimate a non-classical framework. What would it show? Suppose the non-classicist is asked to justify various aspects of their formalism, to reproduce the arguments (from *ZFC* model theory) that underlie their theory; then there are two possibilities. Either the proofs were inherently classical, involving reasoning that cannot be recast, in which case the non-classicist cannot justify their own position. The underlying classical results, if only obtainable classically, will always remain classical-only, and can never be appropriated or used by the non-classicist. Or else the proofs could have been done in a non-classical setting all along, in which case there never was need of classical metatheory except as a *façon de parler*. One can kick a ladder only once one is safely off of it; and when it comes to mathematical foundations, being safely off amounts to having never needed the ladder in the first place.

2.4 The challenge going forward

The discussion of this section leaves us in a difficult position with respect to the Church Turing Thesis. If we wanted to ask our question — in which frameworks can the notion of computation be developed? — in a way that treats non-classical logics as toys, to be manipulated by a classical *ZFC*-type framework, then this would be a straightforward mathematical problem. (To be candid, that was the project we originally intended to take on.) But it would operationally presuppose that the classical framework is the arbiter of truth. While there is no doubt we can make toy interpretations of non-classical frameworks under a classical framework, working that way would divest the question of much philosophical interest.

Shapiro suggests that working in this 'eclectic' spirit is not so difficult, after all: "Usually, it is not hard to keep one's perspective" [Shapiro, 2014, p.209]. We are less sanguine. If one is genuinely entertaining different frameworks, then matters become genuinely confusing. Nevertheless, flying without a safety net is worth the risk. It promises more insight into the nature of computation, its philosophical analysis, logic, and limits. It is in this spirit that we now proceed.

13. Both anticipate the objection, ultimately suggesting that model theory is an instrumental or heuristic device. In a different vein, Beall and Restall consider this problem [Beall and Restall, 2006]. Shapiro devotes two full chapters to the issue in his [Shapiro, 2014]. The common theme throughout is that the worry is somewhat overblown or misplaced. Nevertheless, the worry is prima facie very serious — why else would everyone devote space to explaining why it is not? — and this is sufficient for our purposes.

3. Towards a framework for comparison of frameworks (when there can be comparison at all)

Our goal is to take pluralism in foundations seriously. With this in mind, let us now return to our main problem: the Church-Turing Thesis and convergence theorems. In this section we shall provide a clear way of highlighting the distinction between *de dicto* and *de re* approaches to comparison of *FAC*-structures (section 2.3), and what is required to make a plausible representation of them.

3.1 Convergence revisited

Let's try to make the foundational background behind the convergence argument more explicit. The orthodox way to formulate Claim 4 (all formal analyses of computation are equivalent, therefore CTT) using our apparatus of frameworks and *FAC*-structures is as follows:

Claim 5 *(ZFC-convergence,* de re*) Let Γ be the finite set of acceptable formal analyses of computation that have been produced. Then for all $\mathcal{A}, \mathcal{B} \in \Gamma$,*

$$ZFC \vdash_{CL} \mathcal{A} \simeq \mathcal{B}.$$

Regarding notation, we place ZFC on the left of the $\vdash$ to indicate that this is the underlying theory. We place CL to the bottom right of the $\vdash$ to indicate that the underlying logic is classical. The list Γ will need to include those analyses listed at the beginning of Section 1.1 and on the basis of its sufficiency — a philosophical premise — we go on to argue that CTT holds. We use classical logic and ZFC.

Given our goal to compare alternative frameworks, we might generalise Claim 5 as follows:

Problem 6 *(*$\mathcal{F}$ *convergence,* de re*) Let* $\mathcal{F}$ *be a framework and let* Γ *be the finite collection of acceptable formal analyses of computation that have been produced. Then for all* $\mathcal{A}, \mathcal{B} \in \Gamma$*,*

$$\mathcal{F} \vdash_{\mathcal{F}} \mathcal{A} \simeq \mathcal{B}.$$

This time we have written $\mathcal{F}$ both to the left and to the bottom right of the $\vdash$, to mean the axioms and logic of $\mathcal{F}$ respectively. We might then think that if $\mathcal{F}$ is in a position to solve Problem 6, then it could be in a good position to mount the standard convergence argument and claim CTT; and if not, then computability according to $\mathcal{F}$ is going to look very different. However, this is exactly the point that we should *not* rush off to start comparing and contrasting foundational frameworks. The formulation of Problem 6 is too ambitious. It presupposes that notions such as 'finite collection' are invariant across frameworks. So far, we have given several reasons to doubt that this assumption holds. In the classical ZFC framework, we've explained how to represent formal analyses of computation, but we have not said how to do this for frameworks in general.[14]

Corresponding to our *de re* formulation above, one can address the convergence problem in a *de dicto* fashion. In the context of the classical ZFC framework, we simply forget about the informal notions of a formal analysis of computation and use the *FAC*-structures instead:

Claim 7 *(*ZFC*-convergence,* de dicto*) Let* Γ *be the set of acceptable FAC-structures. Then*

$$ZFC \vdash_{CL} \forall \mathcal{A}, \mathcal{B}(\mathcal{A} \in \Gamma \wedge \mathcal{B} \in \Gamma \Rightarrow \ \mathcal{A} \simeq \mathcal{B}).$$

What is the difference between Claim 5 and Claim 7? On the outside of the ZFC context, in Claim 5, we are talking about formal analyses of computation involving tapes and buckets. On the inside of the context, in Claim 7, we are talking about classical ZFC objects. The move from outside to inside is a philosophical one — the claim that the *FAC*-structures provide a faithful representation of their targets outside the framework. So, generalising again, the *de dicto* version of Problem 6 would then be:

Problem 8 *(*$\mathcal{F}$ *convergence,* de dicto*) Let* Γ *be the set of acceptable representations of formal analyses of computation according to* $\mathcal{F}$*. Then*

$$\mathcal{F} \vdash_{\mathcal{F}} \forall \mathcal{A}, \mathcal{B}(\mathcal{A} \in \Gamma \wedge \mathcal{B} \in \Gamma \Rightarrow \ \mathcal{A} \simeq \mathcal{B})$$

We should understand Γ here as a list given with the language and framework of $\mathcal{F}$. Quantification occurs *inside* the context of the framework.[15]

Would a positive solution to Problem 8 in some framework show that CTT held there and that computability in that framework was, in some sense, comparable to its orthodox understanding? Not obviously.

14. More explicitly: When trying to formalise the standard approach to convergence theorems in Section 1.2, we adopted some set-theoretic tools to provide an algebraic representation of formal analyses of computation, and we noted some possible issues with this move. When we represented, for example, Turing's model of computation using a *FAC*-structure $\mathcal{A} = \langle X, F, I \rangle$, X didn't really contain particular states of a tape, but rather a set-theoretic surrogate which contains objects which are isomorphic to states of a tape: they contain all the information that is required. This move then allows us to take the informal notion of a formal analysis of computation and represent it by a *FAC*-structure.

15. Setting aside the (major) issue of restricted quantification with a non-material conditional.

There is no reason to think that the list of representations $\Gamma^{\mathcal{F}}$ used by framework $\mathcal{F}$ is in any reasonable way comparable to the list of representations Γ^{ZFC} used by ZFC. This is the point where the limitations of the *de dicto* result start to squeeze. An alternative framework may provide a theory of computability which only *sounds* orthodox. A framework $\mathcal{F}_1$ could accuse $\mathcal{F}_2$ of merely proving something that *looked like* $\mathcal{F}_1$'s result, for example (as per section 2.2.2) by using a single inconsistent number to simulate many distinct numbers. At bottom, this is going to be a philosophical point rather than a mathematical one. There needs to be some agreement between $\mathcal{F}_1$ and $\mathcal{F}_2$ with regard to the faithfulness of their respective representation to warrant comparison. For two parties to communicate, they have to agree that they are talking (roughly) about the same thing. Doing *de dicto* comparisons by taking a framework at its word is only as good as that framework's word.[16]

3.2 *The conditions for* de re *comparison: arguing about sets and reals*

One might hope that *FAC*-structures are simple enough to be invariant across frameworks. This hope is unrealistic. Even for two frameworks that agree on *almost everything*, there are significant hurdles for comparison, as we will now illustrate with an example. From this we shall learn that we should choose the objects from which we construct our representations very carefully.

Consider a disagreement between two proponents of classical ZFC, who are at odds over how many ordinals there are. We start from here to show that *even under extremely favourable conditions*, namely, agreement about logic and a vast amount of ontology, the difficulties of comparison still emerge. Let $\mathcal{F}_1$ be classical ZFC plus the assumption that there is an inaccessible cardinal; and let $\mathcal{F}_2$ be classical ZFC plus the assumption that there are no inaccessible cardinals. Moreover, let us assume that the adherents of $\mathcal{F}_1$ and $\mathcal{F}_2$ have roughly the same realist attitude toward set theory. They both take it that their axioms are correctly describing an ontology of sets which is, so to speak, out there and logically prior to its description.[17]

It is well-known that from the perspective of $\mathcal{F}_1$ one can produce a relatively natural interpretation of the $\mathcal{F}_2$ framework. Supposing that κ is the first inaccessible cardinal, $\mathcal{F}_1$ can procure a set V_κ which provides a model for the $\mathcal{F}_1$ framework; i.e., V_κ is a model of the theory ZFC and the statement that there are no inaccessible cardinals [Kanamori, 2003]. This model will allow $\mathcal{F}_1$ to simulate anything that $\mathcal{F}_2$ can do. On the other hand, one cannot provide an interpretation of the $\mathcal{F}_1$ framework from the perspective of $\mathcal{F}_2$.[18] One might be tempted to say that $\mathcal{F}_1$ thinks that *the ordinals extend further* than $\mathcal{F}_2$ thinks.

This is not a good way of understanding the situation. It makes heavy use of $\mathcal{F}_1$'s interpretation of $\mathcal{F}_2$, but without justifying why this interpretation is acceptable for making comparisons between $\mathcal{F}_1$ and $\mathcal{F}_2$'s representation of the ordinals. This is not to say that there is no value in $\mathcal{F}_1$'s "simulation" interpretation of $\mathcal{F}_2$ — it provides a means of predicting what $\mathcal{F}_2$ will say about any given matter — but it is, no more than an instrument for this purpose. It would be more faithful to let the terms of one framework denote naturally in the other — a reasonable condition, given that both $\mathcal{F}_1$ and $\mathcal{F}_2$ are trying to denote the same collection with the term "ordinal". It is not that one framework's ordinals are *longer* than the other's; rather each thinks that the

16. It is at least possible for a framework to incorrectly self-report. A framework couched in a dialetheic arithmetic could prove $\exists \mathcal{A}, \mathcal{B} \in \Gamma\ \neg(\mathcal{A} \simeq \mathcal{B})$ *even though* it also proves $\forall \mathcal{A}, \mathcal{B} \in \Gamma\ \mathcal{A} \simeq \mathcal{B}$. This could be due to the subtleties of how the equivalence relation is defined, or facts about which collapse model was used [Priest, 2006, ch.17]; but the main point is that, paraconsistently, $\forall xp$ and $\exists x \neg p$ do not rule each other out.

17. We could have considered a more logical example, like the representation of intuitionistic negation by $\Box\neg$ in $S4$ from Section 2.3; but by sticking with sets, our target comparison is between things rather than language, which makes the same basic point but without complications about the "referent of negation" or other such hairy questions.

18. This is for the intuitive reason that $\mathcal{F}_1$ is "bigger" than $\mathcal{F}_2$, and the technical reason that $\mathcal{F}_1$ has strictly greater consistency strength than $\mathcal{F}_2$. Note that the ZFC framework doesn't strictly provide an object which is the ordinals, which should give pause to the orthodox [Shapiro and Wright, 2007].

other has made a *mistake* about the existence of a particular ordinal. Moreover, their disagreement is quite blatant. It can be read straight off the frameworks themselves, making use of the fortuitous circumstances in which each framework is formulated in the same language. Most starkly, $\mathcal{F}_2$ entails that $\mathcal{F}_1$ is false. For $\mathcal{F}_2$ there is no value in a model furnished by a framework that is false.[19]

Stepping back, the lesson is that the more economically we can represent a formal analysis of computation, not making use of arbitrary sets, the better our prospects for meaningful comparison across a wide variety of frameworks. The only method of making good *de re* comparison of theories of computation is to represent formal analyses of computation in a region over which there is no disagreement, e.g. in the example just given, below the first inaccessible cardinal.[20] The same descriptions suffice for both parties in the debate. So when it comes to comparing non-classical frameworks, agreement up to an inaccessible cardinal is far too ambitious.

But there is no reason to aim so high. A *FAC*-structure has three components to represent: a computation space, partial computable functions, and possible algorithms. Each could be represented by a set of natural numbers, or, in the case of partial computable functions, a countable collection of real numbers. The entire *FAC*-structure can then be represented by a *single real number*, using well-known coding techniques [Simpson, 1999]. If we can find frameworks that agree about the real numbers, then we'll be in a position to make meaningful comparisons about their respective theories of computation.

And still, representing *FAC*-structures with a real number is not economical enough. Let $\mathcal{F}_1$ be classical *ZFC* plus the assumption that some real number is not constructible; and let $\mathcal{F}_2$ be classical *ZFC* plus the assumption that every real is constructible [Kunen, 1980]. Each of these frameworks clearly rules the other out, so they are distinct. Given that $\mathcal{F}_1$ thinks there is a non-constructible real, we might be tempted to say that $\mathcal{F}_1$ thinks there are more reals than $\mathcal{F}_2$. Indeed $\mathcal{F}_1$ can provide an interpretation known as the constructible hierarchy, which models $\mathcal{F}_2$ in much the same way as the previous example. But as with inaccessibles, so with constructibles. First and foremost, $\mathcal{F}_1$ and $\mathcal{F}_2$ think the other has got something *wrong*: $\mathcal{F}_1$ says there's a non-constructible real and $\mathcal{F}_2$ says there isn't. If $\mathcal{F}_1$ simulates $\mathcal{F}_2$'s talk about the reals, it will not be faithful to $\mathcal{F}_2$'s intended interpretation. Both being in *ZFC*, $\mathcal{F}_1$ and $\mathcal{F}_2$ will still agree on almost any question of ordinary mathematics. Yet they have substantively different theories of the real numbers. If we represent *FAC*-structures using real numbers, we may have problems finding counterparts from one framework in another.

3.3 *Representations on the cheap: From reals to naturals*

In the case of the classical set theorists disagreeing about reals, we can restrict our attention to the constructible reals, whence $\mathcal{F}_1$ and $\mathcal{F}_2$ do not disagree.[21] But such a move is possible only under ideal conditions

19. After all, the *trivial* theory — in which every sentence is held true — contains every other theory as a fragment, but no non-trivialist finds this very impressive.

20. Strictly, we should say the sets whose rank is below the first inaccessible cardinal, however, this technical point isn't particularly relevant here. More formally, for any sentence of set theory φ and any definable ordinal α below the first inaccessible cardinal, we have

$$\mathcal{F}_2 \vdash_{\mathcal{F}_2} \varphi^{V_\alpha} \;\Rightarrow\; \mathcal{F}_1 \vdash_{\mathcal{F}_1} \varphi^{V_\alpha}.$$

V_α denotes the family of sets with rank less than α [Jech, 2003]. Regarding notation, φ^{V_α} denotes the result of binding all the of the quantifiers in φ to V_α [Kunen, 1980]. However, it can be informally read as saying that φ is true in V_α.

21. More formally, for any set-theoretic sentence φ, we have

$$\mathcal{F}_1 \vdash_{\mathcal{F}_1} \varphi^{L} \;\Rightarrow\; \mathcal{F}_2 \vdash_{\mathcal{F}_2} \varphi^{L}.$$

Moreover, for any constructible real number described in $\mathcal{F}_1$ there is an obvious counterpart to it in $\mathcal{F}_2$. We simply use the same description in $\mathcal{F}_2$. As in the for *de re* comparison. What can we hope for common ground between frameworks in general?

Given our goals of comparing representations between different frameworks, the best prospects lie in representing analyses of computation as economically as possible. Now, what is good about *FAC*-structures is that they anchor us: a fixed representational device gives us reason to think that what some framework thinks is a Turing machine is comparable with what another framework thinks is a Turing machine. The problem with *FAC*-structures is that they cost too much, by demanding a great deal of agreement between alternative frameworks. How low can we go?

Despite the fact that the ingredients of a Turing machine (algorithms, partial computable functions, computation space) can be naturally represented by *FAC*-structures, in Section 1.1 we were able to describe them using a finite amount of information. This suggests that (with a little coding) there ought to be a way of capturing this with a *finite representation*, with a *mere natural number* rather than a real. First, rather than giving a complete (and infinite) list of all the algorithms, we might just give the finite vocabulary of the instructions and the finite set of rules which determine whether or not they are well-formed. This could be coded by a natural number. Second, rather than giving (an infinite list of) all the (infinite) partial computable functions, we just say how the algorithms are supposed to be read and how they are supposed to be implemented. This information again could be coded by a natural number. The computation space is naturally represented by a set of natural numbers, but it should be a computable set. Provided a framework $\mathcal{F}$ has some ability to talk about computability, then it will be able represent the computation space with a Gödel number coding instructions for how to build it. So we can represent *FAC*-structures with three natural numbers which are essentially codes for instructions. We thus obtain a significant amount of compression from a real number down to a natural number.

previous example, we are able to use the simulation interpretation to provide a region of common ground between the frameworks.

There are two observations to make here. First, finite representations are cheap. Previously with a *FAC*-structure, we were simply given all the algorithms and all of the functions as objects within the ontology of a particular framework. With this finite representation, however, we are given the instructions for how to *construct* those objects within the framework. We are therefore getting an exterior view of a framework. To use our planetary metaphor: we, looking at the world from outside, can see that it has the resources for computation; but the world itself does not yet have the requisite objects on hand. Second, and more importantly, the kind of constructions required can be executed by computable functions. Thus, finite representation requires a framework capable of capturing a certain amount of computability theory already. It piggy-backs on the host framework's ability to represent computation in order to take the natural number and turn it into a more natural representative of an analysis. Any framework that is able to talk about the natural numbers and represent computation will be able to take such a representation and attempt a solution to Problem 6. In this way we can substantially lower the agreement bar between frameworks and get into position to seriously consider non-classical approaches to computation.

3.4 Barriers to comparison still?

No matter how low we manage to set the bar, there is still a level of agreement that must be met even if we are only interested in solving the *de dicto* Problem 8. A framework $\mathcal{F}$ must be able to talk about the natural numbers (even if it merely interprets them)[22] and it must be

22. For two frameworks, at the least, we'd want to know that for all Σ^0_1 sentences φ

$$\mathcal{F}_1 \vdash_{\mathcal{F}_1} \varphi \;\Leftrightarrow\; \mathcal{F}_2 \vdash_{\mathcal{F}_2} \varphi.$$

able to talk about computation. Without this, a framework does not just fail to get into position to address Problem 6; it fails to be able to do any computation at all. Or, to put it more generously, it will not do computation in a manner recognisable to more classically oriented approaches. If, for example, $\mathcal{F}_1$ is an ultra-finitist framework and $\mathcal{F}_2$ is not, then $\mathcal{F}_2$ will have a very different idea of what a partial computable function is. After all, an easy corollary of ultra-finitism is that all programs halt and first-order logic is decidable, just for example. At this point it would be reasonable to think that the word 'computation' means different things in different frames. Radically different theories of natural numbers should cause us to wonder if there is deep failure of common subject matter at the core.

Nevertheless, our cheap representations might seem to be begging the question. In order to produce a finite representation, we're assuming that a framework can do some computability theory right at the beginning. Isn't that a problem? No. We are trying to establish that $\mathcal{F}$ does computability in *just one* way, rather than several. This is the content of of the target convergence theorems, and it is critical to our everyday mathematical application of computability theory. There is nothing, in principle, stopping a particular framework from having several different ways of handling computability; in which case, any one of them will suffice. The demand that a framework be able to talk about computation is not trivial, but it does not presuppose the CTT.

Whether there could be even more basic levels, lower than arithmetic, to resort to in very radical cases — e.g. two frameworks that only agree up to some finite number n, or a framework that says of its own proof relation that it is undecidable, but which from a *ZFC* perspective is clearly undecidable — may be achievable. And, for all we've said, *which collection* of "the natural numbers", or some fragment thereof, must be agreed upon? We leave such negotiations to another day.

4. Conclusion: Colonialism and courtesy

Computation in different foundational frameworks, when possible at all, may be compared *de dicto* with relative ease: just let each framework say what it has to say about convergence theorems (or lack thereof). This will give a faithful report on the situation that respects the integrity of different systems, but without adding much insight. If only *de dicto* versions of convergence theorems were provable in $\mathcal{F}$, this would only establish that from $\mathcal{F}$'s perspective, its own version of computability theory looks like a theory of computability. Establishing how the frameworks compare *de re* would be much more illuminating. In full generality, though, it does not appear to be possible to answer absolute questions about computation between worlds. As Shapiro sums it up,

> ...Nothing prevents a theorist committed to a substantial logic from studying a range of other logics and learning how they relate to each other and her own. Nor does it prevent another logician from taking on a similar study, using a different logic. Nor does it prevent a third logician from studying how the first two meta-theoretic projects relate to each other. The three of them might not be able to share all of their results with each other, in a straight homophonic manner, but that's life [Shapiro, 2014, p.202].

At heart, the problem is about transworld identity: how do we interpret the names and terms of one framework in terms of another? The canonical approach of rigid designation is not available in these situations, so an attractive option instead would be something like Lewis's counterpart theory [Lewis, 1968]. The idea is to make sense of a plurality of theories the same way we make sense of a plurality of worlds. When moving from a model of one framework to that of an alternative, we seek counterparts in the new model which are most *similar* to that in the original. A seemingly bedrock requirement is agreement about (most of the) natural numbers. So worlds that are closer to each other with respect to agreement about numbers are more comparable. Worlds are organised into clusters, and in these local neighborhoods, some *de re* comparison should be possible; worlds in clusters that are vastly far away from each other are too different to say whether or not computation in one is the same as computation in another. The similarity metric can be based on level of agreement about the natural numbers.[23] Rather than betraying some pro-classical prejudice, the minimal conditions we have isolated are just a way to say that for two people to be able to speak to each other, they need to agree on some basics; and it does not get much more basic than natural numbers.

We have tried to split the difference between two possible methods. On the one hand we have a kind of *colonial charity*, where we absorb our interlocutor's framework into our own but without taking its content seriously. This allows the alternative framework's statements to come out, in some weak sense, true, but it can fail to take that framework seriously as a rival. On the other hand, we have a kind *intellectual courtesy*, where we take our interlocutor at their word and interpret their language as we would our own. On this way of doing things, our interlocutor's theory may come out false according to our own lights, but we have at least paid them the courtesy of respecting their autonomy. Under particularly favourable conditions there is common ground between two frameworks, where charity and courtesy coincide; less so in less favourable conditions.

"When people talk to each other, they never say what they mean," says the character of Turing in *The Imitation Game* (2014). "They say something else and you're expected to just know what they mean." As elsewhere in life, we each speak from our own perspectives, and cannot do much more than hope to be understood.[24]

23. There is a possible revenge problem here — is the similarity relation itself expressed *de dicto* or *de re*? But if this is a problem, it would be a problem for Lewis too; in which case, we safely ignore it for now.

24. Thanks to audiences at the Institute of Computer Science, Academy of Sciences of the Czech Republic, a Pluralism Workshop at Yonsei University, and the Otago Logic Group. This research was supported by a Marie Curie International Incoming Fellowship within the 7th European Community Framework Programme, and by the Marsden Fund, Royal Society of New Zealand.

References

[Agudelo and Carnielli, 2010] Agudelo, J.C. and Carnielli, W. (2010). Paraconsistent machines and their relation to quantum computing. *Journal of Logic and Computation*, 20(2):573–595.

[Bacon, 2013] Bacon, A. (2013). Non-classical metatheory for non-classical logics. *Journal of Philosophical Logic*, 42(2):335–355.

[Beall, 2009] Beall, J. (2009). *Spandrels of Truth*. Oxford University Press.

[Beall and Restall, 2006] Beall, J. and Restall, G. (2006). *Logical Pluralism*. Oxford University Press.

[Brady, 2001] Brady, R. (2001). *Universal Logic*. CLSI Publications.

[Bridger, 2007] Bridger, M. (2007). *Real Analysis: A Constructive Approach*. Wiley.

[Copeland and Sylvan, 1999] Copeland, B.J. and Sylvan, R. (1999). Beyond the universal Turing machine. *Australasian Journal of Philosophy*, 77(1):46–66.

[Field, 2008] Field, H. (2008). *Saving Truth from Paradox*. Oxford University Press.

[Jech, 2003] Jech, T. (2003). *Set Theory*. Springer.

[Kanamori, 2003] Kanamori, A. (2003). *The Higher Infinite: Large Cardinals in Set Theory from Their Beginnings*. Springer.

[Kunen, 1980] Kunen, K. (1980). *Set Theory: An Introduction to Independence Proofs*. Elsevier.

[Lewis, 1968] Lewis, D.K. (1968). Counterpart theory and quantified modal logic. *Journal of Philosophy*, 65(5):113–126.

[Lewis, 1990] Lewis, D. (1990). Noneism or allism? *Mind*, 99(393):24–31.

[Meyer and Mortensen, 1984] Meyer, R. K. and Mortensen, C. (1984). Inconsistent models for relevant arithmetics. *Journal of Symbolic Logic*, 49(03):917–929.

[Ord and Kieu, 2005] Ord, T. and Kieu, T. D. (2005). The diagonal method and hypercomputation. *British Journal for the Philosophy of Science*, 56(1):147–156.

[Petersen, 2000] Petersen, U. (2000). Logic without contraction as based on inclusion and unrestriced abstraction. *Studia Logica*, 64(3):365–403.

[Priest, 1979] Priest, G. (1979). The logic of paradox. *Journal of Philosophical Logic*, 8(1):219–241.

[Priest, 2006] Priest, G. (2006). *In Contradiction: A Study of the Transconsistent*. Oxford University Press.

[Priest, 2008] Priest, G. (2008). *An Introduction to Non-Classical Logic: From If to Is*. Cambridge University Press.

[Quine, 1986] Quine, W. (1986). *Philosophy of Logic*. Harvard University Press.

[Quine, 1976] Quine, W. V. (1976). Quantifiers and propositional attitudes. In *The Ways of the Paradox and Other Essays*. Harvard University Press, Cambridge.

[Restall, 1992] Restall, G. (1992). A note on naïve set theory in *LP*. *Notre Dame Journal of Formal Logic*, 33(3):422–432.

[Rogers, 1967] Rogers, H. (1967). *Theory of Recursive Functions and Effective Computability*. McGraw-Hill.

[Routley et al., 1982] Routley, R., Plumwood, V., Meyer, R. K., and Brady, R. T. (1982). *Relevant Logics and Their Rivals*. Ridgeview.

[Shapiro, 2014] Shapiro, S. (2014). *Varieties of Logic*. Oxford University Press.

[Shapiro and Wright, 2007] Shapiro, S. and Wright, C. (2007). All things indefinitely extensible. In Rayo, A. and Uzquiano, G., editors, *Absolute Generality*. Oxford University Press.

[Simpson, 1999] Simpson, S. G. (1999). *Subsystems of Second Order Arithmetic*. Springer.

[Smiley, 1998] Smiley, T. (1998). Conceptions of consequence. In Craig, E., editor, *Routledge Encyclopedia of Philosophy*. Routledge, London.

[Smith, 2007] Smith, P. (2007). *An Introduction to Gödel's Theorems*. Cambridge University Press.

[Smith, 2011] Smith, P. (2011). Squeezing arguments. *Analysis*, 71(1):22–30.

[Sylvan and Copeland, 2000] Sylvan, R. and Copeland, J. (2000). Computability is logic relative. In Hyde, D. and Priest, G., editors, *Sociative Logics and Their Applications*. Ashgate.

[Williamson, 2014] Williamson, T. (2014). Logic, metalogic and neutrality. *Erkenntnis*, 79(2):211–231.

[Williamson, 2012] Williamson, T. (May 13, 2012). Logic and neutrality. *New York Times*.