Explication, Open-Texture, and Church's Thesis

Andre Curtis-Trudel

July 19, 2016

Contents

1 Introduction

Mathematicians since at least Euclid have been interested in finding uniform, simple, reliable procedures for solving certain mathematical problems. Hilbert, first in his 1900 address at Paris, and later in 1928 (with Ackermann) in *Principles of Mathematical Logic*, made this interest explicit and focussed on the problem of determining the values of certain functions. Matters culminated in 1936 with Church and Turing independently discovering functions whose values cannot be determined by any uniform, reliable procedure. These results depend crucially on Church's (respectively, Turing's) thesis, which states that the functions whose values can be determined with simple, reliable procedures are just the recursive (respectively, Turing computable) functions. The central questions surrounding the thesis — is it true? and if so, can it be proved? — involve us in a bewildering variety of philosophical issues. Questions about the nature of proof, justification, conceptual analysis, and vagueness are common, and answers to the central questions typically involve secondary answers to these secondary issues. Thus the central questions confront us with challenging philosophical puzzles.

However, the central questions are not *merely* puzzles. Answers to them have deep implications for the foundations of computer science, cognitive science, and mathematics. Computer science is chiefly concerned with algorithms and asks: what problems can be solved by algorithms, and which cannot? Church's thesis is integral to answering this question for it tells us, at least in part, about the bounds of algorithmic solvability. A related problem in cognitive science asks how, and in what respect, human minds are computers? Answers here too rely on Church's thesis. And, finally, deep results in mathematics, such as Gödel's Incompleteness Theorems, rely on Church's thesis for their scope and importance. Clearly the central questions are not merely of academic interest.

That said, I do not intend to answer them; my goals are more modest. I aim to survey some of the issues involved in certain answers to them, to point out where these answers are misguided, and, when possible, to suggest improvements. My project, in essence, is one of philosophical path-clearing. By better understanding what makes the bad answers bad we can better understand what is required for the good ones to be good.

We begin in Section 2 with a brief survey of some central concepts. We will start by reviewing the concept of an algorithm, and will quickly move on to effective calculability. We end Section 2 by properly introducing Church's thesis and clarifying what it says and what it does not. In Section 3 I introduce the standard view of the thesis, which holds it to be true, but unprovable. In light of its unprovability some philosophers have suggested that the thesis is an explication or rational reconstruction. In the latter half of Section 3 I argue that it is not. Finally, in Section 4 I examine in detail and then argue against the premise of the standard view that effective calculability is vague.

2 Algorithms, Effective Procedures, and Church's Thesis

This section introduces some of the main concepts with which we will be concerned. We begin with a general characterization of algorithms and briefly mention the wide variety of known algorithmic techniques. This leads us to a particular, narrow kind of algorithm called an 'effective procedure'. Following this we discuss the epistemologically interesting properties of effective procedures, and from there we introduce Church's Thesis. At this point I am more concerned with introducing ideas than with philosophical rigour. The latter will come later; for now I want to lay out, in broad strokes, our subject matter.

2.1 Algorithms

The central concept of computability theory is that of an algorithm or procedure: a finite, ordered set of instructions designed to solve some problem, namely the problem of finding the value of some function. Often (although not always) algorithms take something as input — a graph, a number, a string of characters, a database, etc. — and, after a series of operations on that input, produce something as output. When, for some function f, and for each valid input, an algorithm eventually ends and produces the correct output we say that the algorithm computes f. As an example, consider the problem of adding two integers in decimal notation. The grade-school procedure is as follows: first, write one integer above the other, lining up corresponding decimal places in columns; then, add corresponding places and write the result below, carrying overflow to the next column, until no more columns remain. Given valid input — two integers in decimal notation — this procedure tells one how to reliably determine their sum, that is, it computes the value of the function $x+y$. Given invalid input — say, a graph — this procedure does not tell us to do anything at all.

This intuitive characterization of an algorithm is clear enough, but too broad to be particularly useful. An immediate problem is that it does not specify what kinds of instructions or operations may occur in the procedure. Indeed, on this account both recipes and driving directions constitute algorithms. But computability theorists (*qua* computability theorists) are neither chefs nor chauffeurs, so we require further specification of (i) what counts as an instruction (and what does not), and (ii) what counts as acceptable input. Computer scientists in the last sixty or so years have investigated a bewildering variety of different algorithmic techniques. Correspondingly there are wildly divergent conceptions of (i) and (ii) in the literature.

There are, for example, conventional algorithms whose instructions include standard arithmetic operations and whose input is some (representation of an) integer. Other algorithms take more sophisticated objects as input, such as graphs, matrices, or even databases or networks. But it is not essential that we use these instruction sets and these inputs. So, for example, in quantum algorithms we work with instructions that manipulate qubits, with DNA computation we work on strings of DNA, and with membrane computing we have instructions for moving molecules across a cell membrane. And certainly there are many more alternatives than these.

I mention all of these examples only so that we may safely ignore them in what follows. From here on we will be concerned with a rather narrow kind of procedure called an effective procedure. Procedures and techniques like those described above are not our subject. This will help to avoid confusion later when we introduce Church's thesis.

2.2 Effective Procedures

Effective procedures constitute a narrow but epistemologically interesting class of algorithms. First I'll describe them, then I'll say what makes them interesting.

Specifying a class of algorithms involves specifying (i) what counts as an instruction, and (ii) what counts as acceptable input. Starting with (i), an instruction is effective (or mechanical) if:

E1 it does not depend on random or outside factors (i.e. is deterministic);

E2 each instruction can be executed by a human agent with finite resources of time and space;

E3 execution requires no ingenuity or insight on the part of the executor — i.e. each instruction must be well-defined, definite, and explicit.

Furthermore, if we have a procedure each of whose instructions is effective, plus

E4 the entire procedure can be specified fully and completely in advance,

then we say that the procedure is an effective (or mechanical) procedure.[1] More vividly, a procedure is effective if it can be carried out by an (idealized) human being using only pencil and paper, given enough time, paper, and pencil. On this account an effective instruction may require an extraordinary (although still finite) number of resources to execute, but it may not tell one to consult a die toss, to pick a number between 1 and 100, or to plug away at a problem until one has an 'aha' moment.

Now for (ii). In what follows we restrict focus to effective procedures that determine the value of number-theoretic functions. Such functions take as input n-tuples of natural numbers and return natural numbers. A function f is *effectively calculable* (or *effectively computable*) if there is an effective procedure (in the sense of E1 — E4 above) that determines $f(x)$, for any x. Both historical and pragmatic reasons motivate this restriction. Firstly, computability theory developed in the first half of the 20^{th} century out of investigations into the properties of certain

[1]This characterization is standard and is consonant with those found in found in, for instance, Copeland (2015), Boolos, Burgess, and Jeffrey (2007, p. 23), and Mendelson (1990, p. 225). I have taken some liberties talking in terms of specific instructions, but this is does not obscure any important points.

functions of natural numbers. Correspondingly, many of the early results and ideas are stated in terms of functions of natural numbers. This is not essential — Post (1944), for instance, works mainly with sets of natural numbers[2] — but it is congruent with standard practice and there is no immediately compelling reason not to follow that practice. Secondly, it is notationally easier to deal with numerical expressions than with, e.g., graphs or matrices. And besides, there are well-known ways to express graphs, matrices, and so forth, numerically, so nothing is lost by working with numerical expressions alone.

So far we have spoken as if effective procedures operate on numbers. But this is misleading. Numbers are abstract objects,[3] hence acausal and non-spatiotemporal, and so cannot be 'operated' upon in the usual sense, especially if our operator is an idealized human. Such agents manipulate finite symbolic configurations on paper or screen. Correspondingly, when we say that effective procedures operate on numbers we mean that our computing agents operate on finite symbolic configurations representing those numbers, given some suitable interpretation.[4] And when we say that an effective procedure P determines the value of a function f (or, equivalently, that that f is effectively calculable), we mean that there is some sequence of symbolic configurations produced by an agent following P that leads from a symbol denoting x to a symbol denoting $f(x)$.

This last point — that the effectively calculable functions are those whose values are determinable by manipulating finite symbolic configurations — brings out why effectively calculable functions are epistemologically interesting. If a function is effectively calculable then the task of determining its values is particularly 'easy' — so easy, in fact, that one need only to follow some simple instructions to do so.[5] With simple enough instructions, entities with extremely limited cognitive capacities — or, as with modern computers, entities with no cognitive capacities at all — can determine the value of an effectively calculable function. It is this property of effectively calculable functions, and of effective procedures more generally, that Hilbert, for example, tried to

[2]And works with coordinate notions such as effective enumerability, instead of effective calculability.

[3]Nominalists may wish to skip the rest of this paragraph.

[4]Just what counts as a suitable interpretation is the subject of philosophical disagreement — cf., the discussion in Rescorla (2007).

[5]Note that we do not mean 'easy' in the complexity-theorist's sense. On this account we may still require an exponential amount of time to determine some functions' value.

capitalize on in pursuit of his finitist programme.[6] Successfully executing this programme involved solving the 'decision problem': the problem of effectively determining whether (the Gödel number of) an arbitrary sentence of first-order is logic is a theorem. In particular (although extremely roughly), Hilbert's idea was to leverage the effectivity of certain inference rules into a solution to the decision problem. This problem exercised Church in the early-to-mid 1930's.

2.3 Church's Thesis

The general question Church tried to address was 'what are the effectively calculable functions?' And the main obstacle Church faced in answering this question was a technical one. As Church saw it, an answer required a precise, formal characterization of the effectively calculable functions. To see why, consider the decision problem. A *formal* characterization allows one to diagonalize: in order to show that the decision problem is unsolvable, one must show that each effective procedure gives the 'wrong answer' for at least one sentence of FOL. And that can be done only if one has a way to list off the effective procedures. A formal characterization provides such a way. The problem with effective calculability, as understood by Church and others in the 1930's, is that it is informal, and so one cannot make a list of all effective procedures. Thus the informality of effective calculability stands in the way of a solution to the decision problem, hence the requirement for a precise, formal characterization.

In virtue of what is effective calculability informal? Church (1936, p. 90) tells us that effective calculability is a "vague intuitive notion" wanting for a definition. It is an unfortunate feature of the literature from the '30s that the conceptions of vagueness and intuitiveness being invoked are left unexplained. We will spend some time below reconstructing and analyzing these notions in detail. For now, it suffices to note that the underlying idea seems to be that effective calculability is informal *because* vague. Vagueness, for Church, stands in the way of formal characterization. Church's solution attempts to sidestep the issue of formally characterizing effective calculability altogether.

[6] For more on this see Sieg (1991, Section 1).

In 1936 Church proposed to define effectivity in terms of recursivity, where a function is recursive if it can be defined from the initial functions plus composition, primitive recursion, and unbounded minimization (i.e. unbounded search). This proposal was not received favourably. Post (1936, p. 291), for example, criticized Church for "masking" the identification of the effectively calculable functions with the recursive functions behind a definition. Post interprets Church as *stipulating* that the effectively calculable functions are recursive, when that is just the question at issue. If by 'define' Church means mere verbal definition, then Post's criticism hits its mark. The pertinent question is just whether all effectively calculable functions are recursive. To define the problem away misses the point of the question altogether. If instead 'define' here means real definition, then we are left wanting an argument that the two notions are indeed equivalent (in the sense of extensional equivalence). Of course, Church gives us an argument — the so-called (by Gandy (1988, p. 77)) 'set-by-step' argument. But this argument had problems and did not appear to convince anyone.[7]

Despite this, by the time Kleene published his 1952 textbook, Church's proposal enjoyed widespread (although not universal) acceptance, bolstered in large part by Turing's 1936 paper. Even so, the proposal was called a 'thesis' and not a 'theorem' precisely due to lingering doubts about its correctness. Certainly Kleene was comfortable enough with the proposal to use it to derive important technical results. But he was quick to point out that the evidence for the proposal falls short of decisively showing that the thesis was true. Rather, we are left with the following as a 'working hypothesis' (to use Post's phrase):

> **Church's Thesis.** A number theoretic function is effectively calculable if and only if it is recursive.[8]

Put extensionally, Church's thesis tells us that the set of effectively calculable number-theoretic

[7]The problem: the argument relies on the requirement that each 'step' be recursive but fails to motivate this requirement. For more see Sieg (1997).

[8]Some historically-minded philosophers (such as Copeland (2015)) call only the left-to-right direction 'Church's thesis', and the right-to-left direction 'the converse of Church's thesis'. A more recent trend calls the entire biconditional 'Church's thesis'. Since the difference is largely immaterial for our purposes, we will follow the more recent trend.

functions is extensionally equivalent to the set of recursive functions.[9]

There are immediate questions about Church's thesis. What kind of claim does it make? Is it true? If so, why? Can it be proved? If so, in what sense of 'prove'? All of these questions, and more besides, will occupy us below. For now, however, a clarificatory remark.

Church's thesis is a claim that the effectively calculable functions are just the recursive functions, and vice versa. It is not the claim that, according to Akl (2005, p. 26), "if a function f can be computed 'somehow' on 'some model of computation', then it is always possible to compute f on a Turing Machine." This is incorrect because there are models of computation which cannot be simulated on a Turing machine. Probabilistic algorithms, for example, may include the instruction 'toss a die', which requires not a Turing machine but an oracle machine on whose oracle tape is written a sequence of random digits.[10] Call 'Akl's thesis' the claim that any somehow-computable function is Turing computable (as above). Now, if Akl's thesis is true, then Church's thesis is true.[11] But the falsity of Akl's thesis does not imply the falsity of Church's thesis. The existence of a 'somehow on some computational model'-computable yet non-Turing computable function does not tell us whether all effectively calculable functions are recursive (or Turing computable). Church's thesis is *not* a claim about what may be calculated 'somehow' on 'some model of computation' — it is a claim about the comparably narrower notion of effective calculability.

2.4 Summary

An algorithm or procedure is a finite, ordered list of instructions. We generate different kinds of algorithms through different restrictions on what counts as an instruction. If we require that instructions be effective, in the sense of E1 — E4, and restrict attention to procedures on natural numbers, we locate the epistemologically interesting class of effectively calculable functions.

[9]The equivalence of the recursive functions and the Turing computable functions leads some philosophers to state Church's thesis in terms of Turing computability instead of recursive definability. In what follow's we'll stick with the above formulation in terms of recursiveness.

[10]Ignoring concerns that such algorithms do not compute functions but relations (presumably the die toss may result in the same input producing two distinct outputs).

[11]As long as effective procedures count as 'some model of computation', which they do.

Church's thesis, if it is true, tells us that this class is extensionally equivalent to the class of recursive functions. Of course, it is not clear at this point that Church's thesis *is* true. In the next section we will explore some of these questions in detail. In particular, we will encounter the so-called 'standard view' which holds that Church's thesis is at once true, but unprovable.

3 Explication and the Standard View

The last section introduced Church's thesis as a claim about the extensional equivalence of the effectively calculable and the recursive functions. We also identified three main questions worth answering: is Church's thesis true? If so, can it be proved? And what kind of claim is it? The distinctions underlying these questions cut across each other. Taking the first two first, we have the following options:

(1) Church's thesis is both true and provable.

(2) Church's thesis is true but unprovable.

(3) Church's thesis is false and provable.

(4) Church's thesis is false and unprovable.

I take it that (3) is inconsistent, at least as long as provability entails truth. And (4) is only as interesting as the bare question of the truth of Church's thesis, since falsehood entails unprovability. And in any event virtually no-one thinks that Church's thesis is false, so I will not worry about (4) in what follows.[12]

So (1) and (2) are the interesting options. What of the third question? If (1) is correct, then Church's thesis is something like a (mathematical) theorem. If (2), then it is less clear what it is. For instance, is it a contingent empirical claim? Or a definition? Or something else entirely? Outlining these positions gives us some sense of the logical geography surrounding Church's thesis. As

[12] Akl (2005) is a counterexample to this trend, but we have already noted that their understanding of Church's thesis is (putting it gently) idiosyncratic. Historical skeptics include Bowie (1973) and Kalmár (1959). Bowie tries to leverage certain supposed intensional features of effective calculability into a construction of a effectively calculable non-recursive function; Berg and Chihara (1974) respond and show that Bowie's argument fails. We will encounter Kalmár's concerns again in Section 4.

we will see, all three questions are intimately linked, and it is difficult to answer one without committing oneself to some answer to the others as well.

In Section 3.1 I introduce the standard stance towards Church's thesis — the so-called 'standard view'. As we will see, the standard view alone does not identify the kind of claim Church's thesis makes. Many proponents of the standard view go on to say that Church's thesis is an explication, in Carnap's sense. Section 3.2 outlines this view, and Section 3.3 argues against it.

3.1 The Standard View

The standard view tracks option (2). This is an old view, with roots in Church's original 1936 paper. Kleene (1952, Section 62) contains the first comprehensive statement of the view and Folina (1998) is a prominent recent advocate. It is difficult to overemphasize the standard view's dominance. Certainly it was the orthodoxy amongst logicians, mathematicians, and philosophers for much of the 20^{th} century.[13] It is still orthodox amongst computer scientists today. Indeed, a popular introductory textbook (Linz, 2012, p. 327) states without argument, or even any acknowledgement that this statement even *needs* argument, that Church's thesis is "not provable". But the standard view is a substantive position and needs to be argued for, and it is to this argument that we now turn.

Kleene (1952, p. 317) tells us that "[s]ince our original notion of effective calculability of a function ... is a somewhat vague intuitive one, the thesis cannot be proved." Presumably Kleene has in mind in-principle unprovability. The concern is not that we have not yet discovered a proof of Church's thesis, rather, it is that there is no such proof to be found in the first place. To make sense of Kleene's remark we need two things: firstly, an account of proof, or in-principle (un)provability, and secondly an account of 'vague intuitive' concepts according to which vague, intuitive concepts cannot be proved. We'll take each in turn.

Kleene does not explicitly state the conception of proof he has in mind. Folina (1998), however, suggests that a broadly Fregean conception of at proof underlies the standard view, and indeed was

[13] There are exceptions, of course. Bowie (1973) and Kalmár (1959) are two examples.

the dominant notion of proof in the 1930's. On this account a

> proof is a mathematical argument from premises accepted as true, each of whose reasoning steps is a deductive inference, and which could be translated into a formal proof.[14]

Here 'formal proof' amounts to purely syntactic derivation in a suitable formal system. This conception of proof is modal in that it involves what *could* (under suitable circumstances) be translated into a suitable formal proof. This is 'in-principle' possibility, since there are mathematical arguments which could not be so translated in actual fact due to restrictions of time or space.[15] On this view even informal arguments may count as proofs, modulo the translation constraint.

Folina does not dispute that there are informal arguments for Church's thesis and cites Mendelson's (1990) argument for the right-to-left direction of Church's thesis as an example.[16] But she insists that Mendelson's argument is not a proof. For an informal argument with true premises to fail to be a proof on Folina's account it must either (a) be non-deductive, (b) non-mathematical, or (c) untranslatable into a suitable syntactic derivation. So one of (a) - (c) must fail in order for Mendelson's argument to fail to be a proof, for there is no indication that Folina thinks that any of Mendelson's premises are false.

It is not clear whether Folina thinks that Mendelson's argument is deductively sound or not. On a wide enough conception of deduction, e.g., necessary truth-preservation, plausibly the argument is deductively sound. On narrower conceptions perhaps not. Presumably Folina does not want to build syntactic derivability directly into the notion of deductive inference; otherwise why explicitly mention the latter in the definition of proof? So I take it that if Mendelson's argument is not a proof, it is not in virtue of failing to be deductively sound. So either it is non-mathematical or untranslatable. The answer, it turns out, is a little of both.

[14] Folina (1998, p. 312).

[15] Say, a proof of an extraordinarily complex theorem of sentential logic which has one unique atomic sentence for each atom in the universe. I have no idea what such a theorem would look like, but such a theorem surely exists.

[16] Briefly: (1) the initial functions are clearly effectively calculable; (2) composition and recursion preserve effective calculability; so (3) every recursive function is effectively calculable.

Effective calculability is not a wholly mathematical property — it involves, in part, spatio-temporal considerations about what can be written down with pencil and paper and psychological considerations about what can be done without insight or ingenuity. And given that effective calculability is not wholly mathematical, the concern becomes that it cannot be suitably represented in an appropriate formal system. As Shapiro (2013, p. 284) puts it, for Church and Kleene (and Folina) "there just is no hope of rigorously proving things when we are dealing with the messy world of space, time, and human psychology." Of course, that does not prevent one from trying. Gandy (1988) and Sieg (2008) are two examples. Such systems need the resources to not only talk about functions of natural numbers but also (descriptions of) physical processes. Sieg and Gandy achieve this by including in their system an infinite set of 'atoms', meant to represent basic physical parts (of computing machines). But by the time we introduce physical properties and objects, or even representations thereof, it is doubtful whether we're left with a *formal* system, in the sense required by Folina's conception of proof, at all.

A second and related concern (although one less emphasized by Folina) is that effective calculability is vague. How does one go about proving that a precise mathematical notion is extensionally equivalent to a fuzzy and at least partly non-mathematical notion? If effective calculability is vague, then proving the thesis "would be like trying to prove that an american philosopher is rich if and only if she has a net worth of at least $783,299.45."[17] Shapiro (2013) locates as the source of the vagueness the idealizations that go into the notion of an idealized human calculator.[18] Suffice it to say that whatever the source of vagueness turns out to be (if there is any), Church and Kleene certainly thought that effective calculability was vague, and that its vagueness stood in the way of a proof of Church's thesis.

Thus the unprovability of Church's thesis results from two claims. First, on the one hand, is the claim that proof is in-principle formal syntactic derivation. Second, on the other, is the claim that no such syntactic derivation can be had for Church's thesis, because effective calculability involves vague, non-mathematical notions. Once we move from the realm of mathematics to the

[17]Shapiro (2015, p. 284).

[18]More on which in Section 4.

realm of effective procedures we move beyond the ambit of rigorous mathematical demonstration, and hence beyond the ambit of what can be proved.

Despite all of this, the standard view also holds that Church's thesis is true. This amounts to the claim that of the set of effectively calculable functions and the set of recursive functions are extensionally equivalent. Correspondingly it is false if there are non-effectively calculable recursive functions or if there are non-recursive effectively calculable functions, and true otherwise. So the standard view holds that the effectively calculable functions and the recursive functions are extensionally equivalent. It is important to be clear about what this entails. Firstly, it is generally agreed that the set of recursive functions is precise and determinate — every object is either in the set or not, and there are no borderline cases. But to hold that the effectively calculable functions are co-extensive with the recursive functions one must hold that the set of effectively calculable functions is similarly precise and determinate. For if not — that is, if there are borderline cases of effectively calculable functions (but not of recursively definable functions) — the two sets would not be identical. Consequently, proponents of the standard view must hold that despite the vagueness of the notion of effectivity, the set of effectively calculable functions is precise and determinate.[19]

If Church's thesis cannot be proved, what evidence is marshalled in its favour? Advocates cite quasi-empirical evidence that seems to count in favour of Church's thesis. Firstly, despite extensive investigation, no effectively calculable non-recursive functions have been discovered, and moreover many effective procedures (such as abacus calculation) have been shown to be recursive. Secondly, all attempts to formally capture effective calculability produce methods that pick out the same class of functions. For instance, the recursive functions, Turing-computable functions, and λ-definable functions are all equivalent. Finally, proponents point to Turing's analysis of effective computation as capturing the essence of effectivity. However they do not think this that this last point provides one with a proof of Church's thesis.[20]

In holding that Church's thesis is true but unprovable the standard view does not commit itself

[19]Incidentally, this means that any vagueness in the concept of effective calculability, if there be such, must be epistemic.

[20]See Kleene (1952) for a catalog of similar arguments.

to a particular view about the nature of Church's thesis. Kleene (1952, p. 318) notices this and remarks that the thesis "may be considered a hypothesis about the intuitive notion of effective calculability, or a mathematical definition of effective calculability." The general point that Kleene picks up on here is that it is at least *prima facie* consistent with the standard view that Church's thesis be an (empirical) hypothesis, or a definition, or even something else entirely. The point is that the standard view still owes us an account of the nature of Church's thesis. Next we examine such an account.

3.2 Explication and Explicationism

Among those who accept the standard view, some think that Church's thesis is a "rational reconstruction, in the sense of Rudolf Carnap and Carl Hempel."[21] A more recent trend in this vein calls Church's thesis an explication, again in Carnap's sense. Murawski and Woleǹski (2006) are the foremost proponents of this latter view. Smith (2007) and Black (2000) both think that explication is a plausible account of Church's thesis, even though both officially take the stronger stance that Church's thesis is provable. For ease of reference, we call those who adopt the view that Church's thesis is an explication 'explicationists,' and the view itself 'explicationism.' Explicationism is more often suggested than argued for (as in Smith (2007)), and even Murawski and Woleǹski (2006)'s treatment is lamentably brief. Indeed, these philosophers seem to take the suggestion that Church's thesis is an explication as largely unproblematic — so unproblematic, in fact, that they neglect to state clearly what explication consists in. So, before we examine what explicationism has to say about Church's thesis we will first look at Carnap's original statement of the process of explication.

On Carnap's view, explication (or 'conceptual explanation') is the process of transforming

> a given more or less inexact concept into an exact one or, rather, in replacing the first by the second. We call the given concept (or the term used for it) the ***explicandum***, and the exact concept proposed to take the place of the first (or the term proposed for

[21] Mendelson (1990, p. 229). See also Rogers (1967, p. 20).

it) the ***explicatum.***[22]

The explicandum "may belong to everyday language, or to a previous stage in the development of scientific language," whereas the explicatum must be given in exact terms, with "explicit rules for its use," and is often a part of a "system of scientific either logicomathematical or empirical concepts."[23] Importantly, explication is not analysis, and instead is closer to Quine's notion of regimentation.[24] Carnap's concern with informal, prescientific concepts is that they are imprecise. In the best case, an imprecise concept leads to unclear truth-conditions; in the worst case, paradox. In order to avoid these difficulties we require a (more) precise concept: the explicatum. A suitably chosen explicatum avoids paradox and facilitates clear, precise truth-conditions.[25]

As an initial example we can consider an explication of 'warmth' in terms of 'temperature'.[26] Explication is required because statements made in terms of 'warmth', such as 'the kettle is warm', have imprecise truth-conditions. Foremost, in this case, is the problem of subjectivity: what is warm to one person is not necessarily warm to another. Moreover, attempts to circumvent this subjectivity seem to fail. The requirement that in order for 'the kettle is warm' to be true, the kettle must feel warm to all possible observers is clearly too stringent, while the requirement that it feel warm to only one is too lenient. With this in mind, rather than muck about with the application conditions of the concept of warmth,[27] Carnap suggests that we avoid 'warmth' altogether by explicating "warmth" in terms of 'temperature'. There are objective and well-defined procedures to determine an object's temperature (just use a well-calibrated thermometer), and consequently statements like 'the kettle is 45°C' have clear and precise truth-conditions.

In this example we explicated 'warmth' by 'temperature' because the former has imprecise applicability conditions while the latter does not. In doing so we implicitly appealed to a general criterion indicating when a concept does or does not have imprecise applicability conditions, and correspondingly when sentences in which the concept occurs have (or do not have) imprecise

[22] (1950, p. 3)
[23] Carnap (1950, p. 3).
[24] As Quine (1964, p. 258) notes, in explication we do not "expose hidden meanings," but instead "supply lacks."
[25] At least, if claims made with the explicatum are imprecise, it is not by virtue of the explicatum.
[26] The example is originally Carnap's, see (1950, p. 12 — 13).
[27] Or the truth-conditions of sentences which involve the concept of warmth.

truth-conditions. Next we will make the criterion explicit. Note, though, that the criterion must tell us when a concept has imprecise applicability conditions (and so is a candidate for explication), but it need not tell us why. There are many reasons why a concept's applicability conditions are imprecise — due to vagueness or underspecificity, for example — but our criterion is not concerned with analyzing those reasons.[28] Now, unfortunately Carnap does not explicitly state the criterion. But we can cull the criterion out by examining both the above example and the applicability conditions of suitable explicata.

In the case of 'warmth', Carnap's concern is that the concept is (partly) subjective.[29] This means that there is no objective, reliable method to determine when an object is warm or not.[30] 'Temperature' rectifies this problem because it is a concept for which such methods *do* exist. More generally, we already noted that Carnap (1950, p. 3) requires the explicatum to have "explicit rules for its use". I take this to mean that the explicatum must have explicit application conditions. And one plausible way to have explicit application conditions is to have an objective, reliable method to determine whether or not the concept applies. This all suggests the following sufficient condition for imprecise application conditions: a concept has imprecise application conditions if there is no objective, reliable method to determine if that concept applies to a given object.[31] And consequently a sentence has imprecise truth-conditions if there is no objective, reliable method to determine the sentence's truth-value. So on Carnap's account, a concept requires explication whenever there is no objective, reliable method to determine either its application conditions or the truth-value of statements which use that concept.

We also need to make explicit the principle in virtue of which we replace talk of warmth with talk of temperature in the example. By adopting 'temperature' as an explication of 'warmth' we avoided the the notion of warmth by *substituting* claims about temperature for claims about

[28]Put another way, the criterion is not an analysis, but an indicator, of imprecise applicability conditions.

[29]It might be objected that subjective concepts are not imprecise, but context-dependent, and in particular dependent upon the context of a particular subject's thoughts, feelings, beliefs, etc. This is probably correct, but largely immaterial to the point being made here.

[30]At least according to Carnap. Note that for him the instruction 'just touch it' does not constitute an appropriate method for reasons outlined in the last paragraph.

[31]Given Carnap's verificationist predilections it is not surprising that this condition looks like the verification criterion, though the present condition in question is not a condition on a statement's *meaning*, just its preciseness.

warmth. Carnap (1950, p. 3) writes that explication involves “replacing” the explicandum by the explicatum and he appears to mean this quite literally. So, in general, to adopt a concept c_2 as an explicatum of some concept c_1 is to adopt a rule according to which we replace talk of c_1 with talk of c_2.[32] More precisely, let t_1 and t_2 be predicates that express concepts c_1 and c_2, respectively. Then to adopt c_2 as an explication of c_1, is to adopt an instance of the following rule-schema:[33]

> **Replacement Rule (RR).** Replace expressions ϕ which (a) contain the predicate t_1, and (b) whose truth-conditions are imprecise due to the presence of t_1, with the expression ψ in which all occurrences of t_1 are replaced by t_2 (while respecting local grammar).

Instances of RR are pragmatic rules that tells us how to swap the explicandum for the explicatum. In order to generate particular rules, replace ϕ and ψ with the explicandum and explicatum, respectively. For example, in the example we took as the explicandum the predicate ‘... is warm’ and as the explicatum the predicate ‘... is 45°C.’ It was in virtue of an appropriate instance of RR that we transformed the statement ‘the kettle is warm’ into the statement ‘the kettle is 45°C.’[34] Obviously not all sentences containing ‘... is warm’ are this grammatically simple, but the reminder to respect local grammar handles these details.[35]

As I’ve cast it, instances of the replacement rule make up the conceptual core of explication. These instances — being pragmatic rules — are not truth-apt; they cannot be correct or incorrect. In this respect explications are unlike ordinary scientific or logical statements, and instead of (in)correctness, Carnap suggests that we should instead assess the (un)satisfactoriness of a proposed explication. There are four criteria governing the satisfactoriness of a proposed explicatum. Carnap requires that the explicatum be (i) similar to the explicandum; (ii) more precise than the explicandum; (iii) fruitful; and (iv) simple.[36] All of these but (i) are straightforward to understand.

[32] Carnap (1950, 1963), Carus (2007, p. 279-280), Maher (2007), Murawski and Woleǹski (2006, p.322).

[33] The schema below is my own invention, but it seems to capture the core of Carnap’s idea.

[34] Likely the best choice of explicatum here is the two-place predicate ‘... is — °C’. Our example predicate is just like this one, but with the second parameter already filled.

[35] To be sure, certain sentences may require some ‘preprocessing’ in order to preserve grammaticality after replacing explicatum for explicandum. And spelling out just how this should work in detail is an important task. But it is not our task, so I mention it only so that I may not worry about it in what follows.

[36] Carnap (1950, p. 5-8).

(i) requires that the extension of the explicatum intersect with the extension of the explicandum, but the intersection need not be exact — some slippage around the edges is permitted. For instance, the concept *fish*[37] at one point included whales in its extension. Carnap notes that the scientific concept *pisces*[38] is an appropriate explicatum for *fish*, even though some fish (namely whales) are not pisces, because *pisces* is much more fruitful than *fish*.

This last point highlights that all of the criteria are defeasible, in the sense that no criterion outweighs the others in all circumstances. Rather, when faced with multiple possible explicata for a given explicandum, we pick the 'best' explicatum by weighing the relative strengths of each criterion for each explicatum against each other. So, for instance, in some circumstances a simpler explicatum is preferable to a more fruitful one, while in others the converse is true. And in the kettle-case, '... is such-and-such a temperature' is a more fruitful concept than '... is warm', because the former leads to more important generalizations. With the former we can state that water boils at 100°C at sea level, whereas with the latter alone we can only state that water boils when it hurts to touch.[39]

Briefly summing up: we require explication when the presence of a concept leads to imprecise truth-conditions, and a statement's truth-conditions are imprecise when there is no objective, reliable method to determine whether that statement is true. We pick a concept as an explicatum according to its similarity, exactness/preciseness, fruitfulness, and simplicity, and once we have chosen a suitable explicatum we invoke an instance of RR in order to replace the explicandum with the explicatum. Instances of RR are not truth-evaluable claims, but are pragmatic rules governing which concepts one allows into their theory.

The suggestion, then, that Church's thesis is an explication amounts to a proposal to adopt recursiveness as an explication of effective calculability. This involves adopting an instance of RR according to which the predicate '... is recursive' replaces the predicate '... is effectively calculable' because the latter has imprecise application conditions, and hence sentences in which it

[37] In Carnap's time, at least.
[38] roughly closer to our modern phylogenetic characterization
[39] Or something along those lines.

occurs have imprecise truth-conditions. Accordingly the biconditional claim 'f is effectively calculable iff f is recursive' has been replaced by an instance of the pragmatic rule RR. This instance instantiates 't_1' with '... is effectively calculable' and 't_2' with '... is recursive'.[40] Additionally, the satisfactoriness of '... is recursive' as an explicatum is to be judged according to its simplicity, fruitfulness, similarity. And explicationists such as Murawski and Woleǹski (2006, p. 322) think that '... is recursive' straightforwardly satisfies all four of these criteria.

It is clear even at this point that working out the details of explicationism exposes a conflict with the standard view. For one, the standard view holds that Church's thesis is true, whereas explications — as instances of the replacement rule — are not even truth-apt, and hence neither is Church's thesis-as-explication. Of course, an explicationist may be comfortable with this, as they still can characterize '... is recursive' as a satisfactory explicatum, even if it turns out that Church's thesis is truth-valueless. Secondly, on the standard view, Church's thesis is a claim in the realm of reference — it is a claim about the extensional equivalence of two sets. Not so on explicationism. Rather, Church's thesis-as-explication is a pragmatic language rule, governing which terms may or may not occur in one's theory of computability. The point I want to emphasize here is that to adopt Church's thesis as an explication is already to abandon the two core commitments of the standard view. This point apparently escaped the notice of the philosophers I mentioned at the beginning of this section. Of course, this does not show that explicationism is incorrect; merely that it is at odds with the standard view. It may still be that explicationism, all things considered, is the way to go. However, as I will argue in the next section, there are good reasons to think that explicationism is not the way to go, and indeed is headed in the wrong direction entirely.

3.3 A Problem for Explicationism

On the standard view, Church's thesis' primary virtue is that it allows one to derive uncomputability results. Uncomputability results tell us about the general non-existence of effective procedures for

[40]To keep things simple, in what follows I will refer to both the rule-schema and its instances with 'RR'.

determining the values of some function.[41] To show that some function is uncomputable it is sufficient to show that it is not effectively calculable. Consequently, uncomputability results are statements that some function or class of functions is not effectively calculable, which is to say that there is no finite, predetermined set of instructions which can be followed without insight by an idealized human agent to determine $f(x)$, for any x and f.

Deriving uncomputability results on the standard view is straightforward and relies on the left-to-right direction of Church's thesis. Recall that this direction states that if a function is effectively calculable, then it is recursive. Taking the contrapositive, we have that if a function is non-recursive, then it is not effectively calculable. It follows that if one can show that some function f is not recursive, f is not effectively calculable. This use of the thesis is common and important enough that it is has a name (actually, a few): Smith (2007, p. 277) calls this the *interpretive* use of the thesis and Odifreddi (1992, p. 103) revealingly calls this the *essential* use. Note that it is by virtue of treating Church's thesis being truth-apt that such results can be derived. For this means that standard rules of inference can be applied to Church's thesis to derive claims about uncomputability.

The essential/interpretive use of the thesis highlights its theoretical importance. A central goal of computability theory is to discover functions whose values can be determined by purely mechanical means, and those whose values cannot. Church's thesis facilitates this by providing a bridge between claims about non-recursive functions and claims about non-effectively calculable functions. Putting the point more metaphorically, Church's thesis allows us to move from claims about abstract functions to claims about (idealized) physical processes and is the point at which the formal apparatus of computability theory 'hooks into' the world of concrete computation.[42]

Uncomputability results are important enough that an account of the nature and status of Church's thesis that prevents one from deriving them is not an account worth countenancing. That is, we have a general constraint on the feasibility of any account of the nature and fit of Church's

[41] Also relevant here are undecidability results, but we can treat these as a degenerate case of absolute uncomputability for functions whose range is $\{0, 1\}$.

[42] Of course, Church and Turing were not concerned with the capabilities of computers as we know them, but with the abilities of idealized human agents.

thesis:

> **CON.** An account of the nature of Church's thesis is satisfactory *only if* it preserves the ability to derive absolute uncomputability results.

If one abandons absolute uncomputability, one must abandon the negative solution to the Entscheidungsproblem, the Halting Problem, and numerous other important results. Since these results are important and widely accepted, it is a strike against any theory that advocates losing them.

Returning now to explicationism, the main problem facing the explicationist is that their account of the nature and status of the thesis does not satisfy CON. The argument for this claim will occupy us below. The upshot of the discussion will be to make clear that explicationism is not a viable account of the nature of Church's thesis. What is surprising about this is not so much that explicationism is defective, but that proponents of explicationism think that their view is entirely capable of deriving absolute uncomputability results. For instance, Smith (2007, p. 324) remarks that "accepting the Thesis in a modest spirit [i.e. as an explication] is quite enough for our purposes," namely the purpose of deriving absolute uncomputability results. Unfortunately for Smith, this is not so: one cannot both adopt Church's thesis as an explication and derive uncomputability results.

Next I will argue that there are functions believed to be not effectively calculable, but whose non-effective calculability cannot be derived on the explicationist's account. While I will discuss only one particular non-effectively calculable function, the remarks here generalize. If we can show that the explicationist cannot derive the non-effective calculability of an uncontroversially non-effectively calculable function, we can show by a symmetrical argument that the explicationist cannot derive the non-effective calculability of any known non-effectively calculable function. This is sufficient to show that the explicationist cannot satisfy CON, and hence is sufficient evidence that explicationism should be abandoned. In particular, we will show that the explicationist cannot prove that the decision function:

$$d(n) = \begin{cases} 1 & \text{if } n \text{ is a code for a theorem of first-order logic.} \\ 0 & \text{otherwise} \end{cases}$$

is not effectively calculable, even though d is widely thought to be not effectively calculable. For our purposes the only essential property of d is that it is not effectively calculable; I choose d simply because it is one of the better known non-effectively calculable functions. The important point is that what goes for d goes for any old non-effectively calculable function.

What is required for the explicationist to prove that d is not effectively calculable? Minimally, the explicationist must be able to state the claim A: 'd is not effectively calculable'. It is in general a necessary condition on proving that p that one be able to state that p. The problem the explicationist faces is that they cannot even *state* A, and hence do not have a hope of proving A. This follows from the explicationist's replacement of Church's thesis as a biconditional with an instance RR. Recall that according to this rule expressions ϕ which (a) contain the predicate '... is effectively calculable', and (b) whose truth-conditions are imprecise owing to the presence of '... is effectively calculable', are replaced with expression ψ which are just like ϕ, except ' ... is effectively calculable' is replaced with '... is recursive'. What we need to show, then, is that both (a) and (b) hold for A.

Clearly (a) holds, because A is equivalent to 'it is not the case that d is effectively calculable', which plainly contains the predicate '... is effectively calculable'. Can A's truth value be precisely determined? That is, is there an objective and reliable method to determine whether d is effectively calculable? From the explicationist's perspective, there is not. Recall the definition of effective calculability introduced in section 1. According to that definition a function is effectively calculable just in case each value at a given input can be discovered by an agent of limited cognitive capacities carrying out, with no insight or ingenuity, and with finite resources of time and space, a step-by-step procedure. But this definition does not specify just when the agent is or is not being insightful and ingenious, and a reliable and objective method *requires* a precise specification of these features. Indeed, explicationism is largely motivated by the concern that no such precise specification can be had. Hence the definition of effective calculability does not suggest a reliable and objective method to determine the truth value of A. And since no other method is forthcoming, the sufficient condition on imprecise truth-conditions tells us that A has imprecise truth-conditions. Consequently the explicationist is compelled to invoke RR on A to produce A':'d is not recursive'. But clearly A' is not A, and what is required to show that d is absolutely uncomputable is that the explicationist be able to prove that A. So the explicationist cannot show that d is absolutely uncomputable. And since we can run a parallel argument for any given (known) non-effectively calculable function, explicationism fails to satisfy CON.

The core of the explicationists' problem stems from the fact that they treat Church's thesis logically as a pragmatic rule and not as a biconditional. By adopting Church's thesis as a pragmatic rule, they bar themselves from using Church's thesis as biconditional to move from non-recursiveness to non-effective calculability. And by replacing statements about (non-)effective calculability with statements about (non-)recursiveness, they bar themselves from even being able to state the desired results. The key point is simple enough: one cannot prove results about non-effectively calculable functions while at the same time adopting a rule which tells one to avoid talking about non-effective calculability in the first place. In some sense the explicationist presents us with a dilemma. Either we can have absolute uncomputability results, or we can have the explicationist's account of the nature of Church's thesis, but not both. The explicationist opts for the latter horn, while I opt for the former. If the cost of an account of the nature of Church's thesis is the ability to derive claims like A, then the cost is too high.

It might be objected that the above remarks beg the question against the explicationist. Explicationism explicitly repudiates various claims about effective calculability on the grounds that they are imprecise, and *a fortiori* repudiates claims about the existence of non-effectively calculable functions. The claim that there are non-effectively calculable functions assumes just what the explicationist says is too unclear to countenance, and so is not an adequate response to explicationism.

This objection misunderstands the dialectic in which explicationists find themselves. The standard view is nearly universally accepted - it is the 'null hypothesis'. And even in spite of its shortcomings, it is the view to beat. What the explicationist offers is an alternative to the standard view, one which purports to make better sense of the nature of Church's thesis than the standard view alone. In order to make their offer attractive, the explicationist needs to show that the value of having an account of the nature of Church's thesis outweighs the utility of being able to derive absolute uncomputability results. The remarks in this section attempt to show that absolute uncomputability results are more important and more useful to have than an account of the nature of Church's thesis. This much, at least, seems to be the attitude of actual researchers in the field, who get along just fine *with* the absolute uncomputability results, and *without* such an account.

There are two further objections available to the explicationist here. The first is that the explicationist might complain that even on the standard view, so-called uncomputability results are not, and never were, results in the first place. After all, the standard view accepts that effective calculability is vague. So what sense can there be of 'deriving' results about effectively calculable functions in the first place? To reiterate, the explicationist's whole point is just that talk of effective procedures is too imprecise to countenance at all. To assume that such talk is *not* too imprecise to countenance begs the question.

I have two responses here. Firstly, absolute uncomputability results derived on the standard view and with the help of Church's thesis are uncontroversially accepted across most of logic and computer science. Doubting that is a little like doubting that the external world exists. Minimally, the explicationist owes us a systematic account about why that acceptance is misguided. Currently no explicationist has presented such an account. Secondly, even if the explicationist's response is correct and such results cannot be had after all, it does not show that explicationism is correct; the fact that absolute uncomputability results cannot be had on the standard view does not alone entail that we ought to adopt recursivity as an explication of effective calculability.

In any event, those still unconvinced can take my argument as establishing the conditional claim that *if* one wants to derive uncomputability results, then explicationism is not the way to go. That much alone is sufficient to show that positions such as Smith's are untenable.[43] And in fact there are good reasons for wanting to derive uncomputability results. Recall from Section 1 our primary interest in effective procedures: they tell us something about functions whose values can be determined by (idealized) human agents. The claim that a given function's values can or cannot be in-principle determined in the actual world is of central interest to computer scientists. Indeed, I submit that the central conceptual tools of computability theorists — Turing machines, the λ-calculus, and so forth — are only interesting insofar as they are a vehicle to claims about (non-)effectively calculable functions. The goal is to find out what can be calculated in the world of concrete computation. By explicating the explicationist gives that up, and proceeds to miss the point of computability theory altogether.

The second objection is that an appeal to the *utility* of explicationism versus the utility of the standard view is misguided, because what we're after is not what is useful, but what is true. And if it turns out that the standard view cannot give a compelling account of what is true (with respect to the nature of Church's thesis), then so much the worse for the standard view. But this objection misunderstands the role of explication in scientific progress. Recall that explications (more precisely, instances of the replacement rule) are not true or false, but are satisfactory or unsatisfactory, according to whether they are simple, similar, exact, and fruitful. Consequently, explications — including the explication of '... is effectively calculable' by '... is recursive' — are adopted or not according in part to their utility, and in this respect the explicationist's suggestion falls short. Ironically, by facilitating the derivation of absolute uncomputability results the standard view shows itself *more* fruitful than explicationism. In this respect, at least, the explicationist is hoist by their own petard: they are committed to thinking that the standard view, and not explicationism, is the preferred view![44] The takeaway point here is that the explicationist cannot appeal to what is *true* in order to defend their view, because explication does not point at truth (at least directly) but instead points to what is useful. And the argument here shows that explicationism's utility does not outweigh the utility of the standard view, even with its defects.

[43] Cf. the Smith quote near the beginning of this section.

[44] Of course, the explicationist still thinks that her view is preferable in light of its exactness, simplicity, and so on.

3.4 Summary

The standard view holds that Church's thesis is true, but unprovable. As a consequence, the standard view leaves open the question of the kind of claim Church's thesis makes. One answer is that the thesis is an explication, in Carnap's sense. I have argued that this view is not plausible. In particular, adopting 'recursivity' as an explication of 'effective calculability' hamstrings our attempts to derive important uncomputability results. All things being equal, we want to be able to derive such results. The fact that explicationism prevents this is a reason to abandon it.

4 Reexamining the Standard View

In the last section we saw that explicationism is motivated, in part, as a response to the supposed unprovability of Church's thesis: because the thesis is not a theorem, the standard view wants for an account of its nature and explicationism provides one. Unfortunately, as we also saw, explicationism's answer is not as viable as it appears. So we are left with essentially the same question we faced at the end of section 3.1 above, namely, what kind of claim does Church's thesis make?

There are two routes one may take at this point. The first develops an alternate account of the nature and status of Church's thesis as we did in Section 3.2. The prospects for this option are dim. Various alternative accounts have been proposed and most are unsatisfactory.[45] Many of them antecedently accept the standard view. Thus it is not unreasonable to suspect that they have failed *because* they take the standard view as their starting point and that these failures are symptomatic of deeper, unaddressed problems. Thus the second route returns to the standard view and examines more carefully its presuppositions. We will take this latter path.

Above we identified two premises underlying the standard view: first, that proof is in-principle syntactic derivation, and second, that no such derivation can be had for Church's thesis because effective calculability is vague and non-mathematical. Focussing on just one aspect of the second premise, here I will argue that effective calculability is not vague. If effective calculability is

[45] For a detailed catalogue of the failures see Murawski and Woleǹski (2006).

not vague, then the concern that Church's thesis is unprovable because vague are unmotivated. Consequently, the overarching interest of the present section is to clear away at least one obstacle standing in the way of a proof of Church's thesis. More modestly, if Church's thesis is unprovable, then it is not in virtue of vagueness on the part of any of its constituent concepts.

Section 4.1 examines the most plausible account, due to Stewart Shapiro, that effective calculability is vague. In Section 4.2 I outline my disagreement with Shapiro. Section 4.3 presents evidence that Shapiro's account is not wholly correct, and Section 4.4 considers some objections and argues that they are not convincing.

4.1 Is Effective Calculability Vague?

We already noted that both Church and Kleene think that effective calculability is vague. But in what sense of 'vague'? It is unlikely that they had anything like contemporary philosophical vagueness in mind. More plausibly, by 'vague' they meant simply 'unsettled' or 'indefinite'.[46] The underlying notion seems to be that of a borderline case. Roughly, a borderline case of a property is an object for which the property neither clearly applies nor clearly does not apply. In order for effective calculability to have borderline cases, then, there must be objects — namely, functions of natural numbers — which are neither clearly effectively calculable nor clearly not effectively calculable.

In a recent series of articles Stewart Shapiro (2006a, 2013, 2015) suggests that effective calculability, at least at one point, was vague in something like this sense. Taking a cue from Waismann, Shapiro (2006a, p. 441) tells us that that "in the thirties, and probably for some time afterward, this notion [effective calculability] was subject to open-texture."[47] According to Shapiro a predicate P exhibits open-texture

> if there are possible objects p such that nothing in the established use of P, or the

[46]Webster's dictionary from 1907 reports that 'vague' means "unsettled; unfixed; undetermined;" and "indefinite". So our suggestion is at least in the right ballpark.

[47]Here I treat open-texture as a kind of vagueness. In actual fact the relationship between the two properties is more subtle. See Chapter 1 and the Appendix of Shapiro (2006b) for details.

non-linguistic facts, determines that P holds of p or that P fails to hold of p.[48]

Thus for the predicate 'effectively calculable' to exhibit open-texture there must be (possible) functions of natural numbers which are neither effectively calculable nor not effectively calculable. We will examine two purported examples shortly. Briefly, though, note that because there is no fact of the matter whether P holds of p, "competent speaker[s] can, in fact, go either way without offending against the meaning of the terms, the non-linguistic facts, and the like."[49] That is, speakers may reasonably disagree about whether or not P holds of p without thereby disagreeing about the meaning of P or about any other relevant facts.[50]

Shapiro does not think that effective calculability *now* exhibits open-texure. Rather, in the last eighty or so years the concept has been *sharpened* to what it is today. 'Sharpening' in the present sense just involves fixing indeterminate cases so that they are definitively included or excluded in the extension of a concept. For example, at one time it was controversial whether or not complex numbers were numbers. That is, the concept 'number' exhibited open-texture with respect to the complex numbers. But 'number' was eventually sharpened so as to include complex numbers, and now they are well-regarded mathematical entities along with the integers, reals, and so on.[51]

It is useful to briefly contrast sharpening to explication, for both are, at least in part, responses to concerns about vagueness.[52] With explication we take a vague, indeterminate concept and find a suitable replacement for it. This may involve excluding objects from the extension of the explicatum which were previously contained in the extension of the explicandum.[53] With sharpening, on the other hand, previously contained objects are still contained post-sharpening. With explication, differences in meaning between the explicandum and explicatum, perhaps even drastic ones, are tolerated if they are accompanied by increased fruitfulness or exactness in the explicatum. No such tolerance is present, at least for the decided cases, when sharpening — once a property is

[48] Shapiro (2013, p. 161).

[49] Shapiro (2006b, p. 12).

[50] Shapiro (2006b) notes that speakers must be reasonably consistent with their own judgements about whether P holds of p or not. In particular, if one judges that P holds of p, then for p' that differ only minimally from p one must judge that P holds of p' also. But we can put aside these details in what follows.

[51] The example is Shapiro's (2013, p. 164).

[52] 'In part' because explication is also concerned with, e.g., subjectivity.

[53] As with the explication of 'fish' by 'pisces', where whales are excluded by the explicatum.

sharpened so as to include an object, no subsequent sharpenings can expell that object from the property's extension.

Very broadly, Shapiro identifies the "possible consideration[s] concerning tools and limitations" involved in the notion of an (idealized) human computing agent as the source of indeterminacy.[54] The idea is that at the time it was unsettled just what sorts of idealizations of time, space, available actions, and so forth, went into the notion of an idealized human computer. This indeteminacy manifests in two kinds of indeterminate cases for effective calculability. On the one hand are computationally infeasible functions, such as Ackermann functions, and on the other are those such as Kalmár's (1959) 'improper minimization' function.[55] Shapiro's claim is that, at one point, these functions, and others like them, were neither effectively calculable nor not effectively calculable — rather, the concept had to be sharpened to include them (as with Ackermann functions) or exclude them (as with Kalmár's). By sharpening effective calculability to include Ackermann functions we fix effective calculability in one direction — a direction according to which feasibility is irrelevant to effective calculability. By sharpening to exclude Kalmár's we fix in favour of definite, well-defined instructions. These functions constitute the core of Shapiro's evidence that effective calculability exhibits (or exhibited) open-texture, so it is worthwhile to examine them in some detail. We start with Ackermann functions.

Ackermann discovered his function in 1927. Here we give a version due to Rósza Péter (1935):[56]

$$A(0,y) = Sy$$

$$A(Sx,0) = A(x,S0)$$

$$A(Sx,Sy) = A(x,A(Sx,y))$$

A is notable for growing extraordinarily fast even for modest inputs — Shapiro (2013, p. 169)

[54] Shapiro (2006a, p. 441).

[55] Shapiro does discuss a third case concerning partial functions. Unfortunately, due to restrictions of space I will not be able to discuss that case here.

[56] S denotes the successor function.

notes that $A(5,5)$ is "larger than the number of particles in the known universe." A is a paradigm case of a computationally infeasible function. And Shapiro identifies this as being central to A (at one time) being neither effective calculable nor not effectively calculable. For Shapiro it was not until Mendelson (1963) and then later Mendelson (1990) that the matter was settled in favour of A being effectively calculable.

Shapiro's idea seems to be that before Mendelson came along considerations about computational feasibility were not built into the concept of effective calculability. As we know it *now* effective calculability determinately does not imply polynomial-time calculability.[57] But, the suggestion goes, at one point it was unclear whether or not this implication held. And it was unclear because the kinds of idealizations involved in effective calculability were unclear. Shapiro (2006a, p. 442 - 443) tells us that "[t]he intuitive notion [of effective calculability] ... leaves open certain interest-relative parameters" with respect to the idealizations imposed on the human calculator. Presumably two such parameters are the amount of time and space required to calculate the value of a function.

According to Shapiro, Mendelson's argument for the so-called 'easy half' of Church's thesis, that recursiveness entails effective calculability, sharpens effective calculability to include Ackermann's function.[58] In the argument Mendelson simply goes through each part of the definition of a recursive function and argues that it is effectively calculable. So, for instance, he starts by noting that the initial functions are all effectively calculable and then argues that composition and recursion both preserve effective calculability. Now, there is nothing here that looks obviously like extension fixing, or A or of anything else. But Shapiro explains:

> [t]he reasoning [in the argument] allows us to see where the idealization from actual human or machine abilities comes in, and, indeed, to see what the idealization is. The argument takes no account on the length of the sequence of functions used to define a recursive function. By examining the argument, we see that we are to ignore, or reject,

[57] A function is polynomial-time calculable if, given an input of size n, the time required to determine a particular value is polynomial in n.

[58] See, for example, Shapiro (2006a, p. 445).

the possibility of a computation failing because the computist runs out of memory or materials.[59]

The mechanism at work here is what Lakatos (1976) calls a 'hidden lemma'.[60] The lemma itself says something like 'ignore bounds on space and time when calculating the value of a function.' And it is in virtue of the lemma that effective calculability is fixed so as to include A.

The lemma is hidden — and so not explicitly justified — but is presupposed by the argument nonetheless. What, according to Shapiro, justifies the lemma? Apparently, not much at all. Shapiro (2013, p. 175) suggests that "[a] community that opted for one of the other notions, such as Turing-computable-in-polynomial-time ... cannot be accused of a failure to understand what they are talking about," which I take to mean that a community that *rejects* the hidden lemma cannot be held irrational or misguided.

Shapiro's second example means to show that the notion of an effective procedure as "a set of instructions that tells the computor what to do at every stage" is indeterminate.[61] The guiding example is Kalmár's (1959, p. 74) argument against Church's thesis. It goes as follows.[62] First, by theorem XIV of Kleene (1935) if ϕ is recursive then

$$\psi(x) = \begin{cases} \mu y(\phi(x,y)=0) & \text{if } (\exists y)(\phi(x,y)=0) \\ 0 & \textit{otherwise} \end{cases}$$

is not recursive. Kalmár calls ψ the 'improper minimalization' of ϕ. Now, since ϕ is recursive, by Church's thesis it is effectively calculable. Hence for any p if $(\exists y)(\phi(x,y)=0)$, then we can find $\psi(p)$ by calculating $\phi(p,0)$, $\phi(p,1)$, ... until we find the appropriate y. Next, if there is no such y and if one can prove $\neg(\exists y)(\phi(x,y)=0)$ "by arbitrary correct means" then we can calculate $\psi(p)$ here also. Thus Kalmár has given us a (purportedly) effective procedure for calculating ψ, even

[59] Shapiro (2006a, p. 445).

[60] Prior to discussion of effective calculability Shapiro spends some time discussing Lakatos' discussion of Euler's theorem. Shapiro sees important parallels between the case Lakatos discusses and Church's thesis.

[61] Shapiro (2013, p. 175).

[62] I follow Mendelson's (1963) cleaner reconstruction.

though ψ is not recursive.

Now, no one, including Shapiro, thinks that Kalmár's proposed procedure is effective; Mendelson (1963), for instance, goes on to criticize Kalmár for the hidden and unmotivated assumption that the proofs "by arbitrary correct means" are effectively enumerable. But, despite this, Shapiro (2013, p. 176) asks "is there something unambiguous in the *pre-theoretic* notion, of computability, in use in the '30s and a few decades after, that rules out Kalmár's "method" definitively?" Shapiro's answer is 'no'.

Shapiro takes the fact that Kalmár even considers that the above procedure is effective as evidence that effective calculability had open-texture. Indeed, Kalmár was a competent mathematician — a recursion theorist, even. If anyone was well positioned to grasp indeterminacy in the notion of an effective procedure Kalmár was. And yet he reached basically opposite conclusion of Church and Turing. That fact wants for explanation and Shapiro finds it in open-texture.

There is an important disanalogy between the present case and that of the Ackermann function. Reflection on the Ackermann function leads us to widen the notion of effective calculability by including more functions as effectively calculable than before. Contrariwise, we "sharpen the concept of computability [i.e. effective calculability] through the rejection" of Kalmár's argument.[63] That is, consideration of Kalmár's procedure allows us to determine what kind of instructions are or are not genuinely effective. Note, however, that it is not as if we discover what are effective instructions. It is an essential feature of open-texture that competent speakers can reasonably differ about positive and negatives instance of a given (open-textured) predicate; such speakers can, as we noted above, "go either way" without offending against either meaning or fact. In order to sharpen, speakers must decide which way. But given that alternate ways can be equally reasonable it cannot be that we discover which way to go, that is, which direction we sharpen in — rather, we decide.

Overall, then, Shapiro's account is as follows. The last eighty years or so have witnessed the concept of effective calculability undergo a series of sharpenings whereby formerly indeterminate

[63] Shapiro (2013, p. 176)

cases of effectively calculable functions were fixed. Some cases, such as the Ackermann function, are fixed so as to be included in the extension of effective calculability. Others, such as as Kalmár's improper minimalization function, are fixed so as not to be included. This sharpening has left us with a notion of effective calculability whose extension is very similar, if not identical, to the extension of the recursive functions. Indeed, given the concept as it is *now*, Shapiro accepts that Church's thesis is true.[64] But this was not always the case: "the notion of computability in use now is the result of the last seventy years [the article is ten years old - AT] of examining texts like Turing (1936), of reacting to proofs and refutations, and of the overwhelming success of the theory of computability."

4.2 Discussion

At first blush it may appear that there is no disagreement between our current goal — making room for the suggestion that Church's thesis is provable — and Shapiro's account. After all, Shapiro thinks that effective calculability *now* is precise, and perhaps even extensionally equivalent to recursivity, even if it wasn't at one point. He even thinks that Church's thesis is susceptible to "proof", given a suitable understanding of what proof is.[65] But isn't this just what we want? Where's the disagreement?

The problem is that Shapiro gets the right conclusion from the wrong premises. Shapiro's view requires us to say that arguments for Church's thesis, like Mendelson's, and analyses of effective calculability, like Turing's, sharpen effective calculability. And on my view this is a radical departure from what is actually going on. I do not think that Turing's analysis fixes the extension of anything, rather, it uncovers necessary conditions on human calculating capacities. And similarly I do not think that Mendelson's argument sharpens anything; instead it exposes necessary entailments between properties.

We can illustrate the disagreement another way. Consider Gödel on effective calculability:

[64](2013, p. 175).
[65]Shapiro (2015, p. 294). Note that the sense of 'proof' used by Shapiro is weaker than Folina's.

> If we begin with a vague intuitive concept, how can we find a sharp concept to correspond to it faithfully? The answer is that the sharp concept is there all along, only we did not perceive it clearly at first. This is similar to our perception of an animal first far away and then nearby. We had not perceived the sharp concept of mechanical procedures before Turing, who brought us to the right perspective. And then we do perceive clearly the sharp concept.[66]

I don't want to get caught up in Gödel exegesis, but I take it that what Gödel is after here is something like epistemic vagueness. The underlying idea seems to be that the extension of a concept can be precise and determinate without the manner by which we express the concept being similarly sharp.[67] And there are mechanisms, such as conceptual analysis of the kind Turing carried out, by which we can bring the two into alignment. That effective calculability appears vague is due to ignorance on our part, but that ignorance can be overcome.

Shapiro directly opposes this Gödelian perspective. He tells us that

> [i]t is not accurate to think of the historical proofs of Church's thesis ... as establishing something about absolutely sharp pre-theoretic notions. Rather, the analytical and mathematical work served to sharpen the very notions themselves.[68]

And earlier in the same paper, when discussing Mendelson's argument, Shapiro tells us that "one might think that the argument proves something determinate about a previously sharp notion. I submit that this would be a mistake."[69] The core of the disagreement concerns the kind of vagueness underlying effective calculability. On the Gödelian account the vagueness is merely epistemic. Despite surface appearances effective calculability has a precise, determinate extension. It took Turing's reflections to sort that out, but the extension was precise all along. On Shapiro's account there was a time when effective calculability had indeterminate cases. Turing's reflections here

[66] Wang (1996, p. 232).

[67] Smith (2007) gives an example: the expression "those philosophers over there" is vague even though it can pick out a precise class of objects.

[68] (2013, p. 178).

[69] (2013, p. 172).

did not uncover that effective calculability was extensionally equivalent to recursivity, rather, they made it the case (at least in part, and if it be the case) that effective calculability and recursivity are extensionally equivalent.

In what follows I defend the Gödelian perspective.[70] However, my strategy is not to present positive arguments for the view, for we do not have the space; instead I will confine myself to some criticisms of Shapiro, and suggest the Gödelian perspective as an alternative. In what follows I will suggest it is not clear that the historical record supports Shapiro's view. The claim that "in the thirties, and probably for some time afterward, this notion [effective calculability] was subject to open-texture" is an empirical claim. Presumably inspection of how the concept was used in the relevant literature at the time can tell us whether or not this is correct. As I will argue, examination of remarks by central figures such as Kleene, Turing, and Church suggests the opposite interpretation, namely that in the '30s and the following few decades the concept was already quite sharp, at least in the respects that interest Shapiro.

Before proceeding, a few methodological remarks are worth making. Firstly, we cannot assume that the concept of effective calculability, as it occurred in the 30s and a few decades after, was anything like our E1 - E4.[71] After all, if Shapiro is correct then E1 - E4 are just the product of eighty years of sharpening. Correspondingly, when I make remarks about the nature of the concept I will make sure to confine my remarks to what is directly supported by textual evidence. Secondly, we should also begin with a clear idea of what Shapiro takes to be the concept in the 30s that exhibited open-texture. Shapiro (2015, p. 283) tells us that "[t]he founders, Church, Stephen Cole Kleene, Alan Turing, et al., were thinking of computability in terms of what can be done by a suitably idealized human following an algorithm." So our question is whether *that* notion exhibited open-texture.

Finally, I have no illusions that I will be able to decisively settle the question of whether or not effective calculability exhibited open-texture in what follows. That task requires a much more

[70]In doing so I do not aim to take a general stance on vagueness. Perhaps the vagueness underlying paradigms such as 'bald' or 'tall' is epistemic, or perhaps not. Either way, the claim that the vagueness underlying effective calculability does not entail a stance on the vagueness underlying the paradigm cases.

[71]Indeed, perhaps we cannot even assume that there was a single concept of effective calculability at all.

thorough treatment of the historical record than I am able to provide below. My sights are set lower. Minimally, the arguments below aim to show that there are some things Shapiro's account cannot explain and that more is needed to vindicate the claim that effective calculability exhibited open-texture.

4.3 Historical Evidence

We start with the evidence surrounding Kalmár's function. Shapiro tells us that the fact that Kalmár even considers that improper minimalization is effectively calculable is evidence of open-texture. The alternate conclusion one might draw is that Kalmár was simply confused about effective calculability. Shapiro discounts this alternative with no discussion. But was Kalmár confused? Consider Kleene on the matter:

> I was present at the Amsterdam Colloquium of 1957 when my good friend László Kalmár presented his argument against the plausibility of Church's thesis ... I immediately concluded, as fast as I heard it, that he had not given an effective procedure for deciding as to the truth or falsity of $(x)\overline{T}(n,n,x)$. He would not be able to tell me in advance in a finite communication (no matter how long we both should live) what set of atomic rules would completely govern the concrete steps in his search for proof by "arbitrary correct means" of $(x)\overline{T}(n,n,x)$.[72]

This comment suggests that, contra Shapiro, in 1957 at least one person (namely Kleene) held a concept of effective calculability that did *not* have open-texture in this direction. But Shapiro has a

[72]Kleene (1987, p. 494 - 495). A note on why Kleene talks about the T function: Kleene's theorem XIV is that $\neg\exists y[T_1(x,x,y)]$ is not recursive — there is no recursive procedure to determine if the function coded by x is defined after y operations when passed x as input. Kleene apparently takes Kalmár, in the argument, to be talking about T_1. Since T_1 is recursive, we may recast Kalmár's function as

$$\psi(x) = \begin{cases} \mu y(T_1(x,x,y)=0) & \text{if } (\exists y)(T_1(x,x,y)=0) \\ 0 & \textit{otherwise} \end{cases}$$

Now, recalling Kalmár's argument, what Kalmár purports to have shown is that, for any n, there is an effective procedure to determine $\psi(n)$. Given how we've recast ψ, this is equivalent to saying that, for a given n, we can determine whether the function coded by n is defined on input n. If Kalmár is correct, then for any n we can determine if $(\exists x)T_1(n,n,x)$ is true, and consequently if $(\forall x)\neg T_1(n,n,x)$ is true. Incidentally, if Kalmár's function is in fact effective, he has given us an effective procedure for solving the Halting Problem.

reply here. He might say that it was in virtue of Kalmár's argument that Kleene fixed the concept in the way that he did. When confronted with Kalmár's "effective procedure," Kleene decided that it was not effectively calculable, that is, he took a stance on how to sharpen the extension of effective calculability.

Unfortunately, this is not a plausible interpretation of what happened at the Colloquium. On the one hand it involves dubious psychological premises about what happened to Kleene's grasp of the concept during Kalmár's talk. On the other, it does not explain why Kleene, "as fast as he heard," Kalmár's argument was able to dismiss it. It is not as if Kalmár prompted Kleene to sharpen his (Kleene's) concepts on the spot; presumably, if Kleene's concept changed during the talk he would have reported so. The alternate account, which is more plausible given Kleene's remarks, is that Kleene brought a pre-sharpened concept with him. Before Kalmár had even started to speak Kleene was operating with the concept of an effective procedure according to which all rules must be finite, simple, and fully specified in advance — that is, Kleene's concept already ruled out "prove by all correct means" as an effective instruction.

But perhaps this is unfair to Shapiro. After all, he recognizes that others, such as Mendelson (1963), were quick to dismiss Kalmár's argument. The salient question is whether anything in the "pretheoretic" notion of effective calculability, used in the '30s and a little while after, rules out Kalmár's procedure. But plausibly by the time Church presented his thesis in 1936, procedures like Kalmár's were already ruled out. Reflection on Church's argument (1936, p. 100-102) for his thesis tells us why.

Church's argument immediately follows the proposal to define effective calculability by recursivity. In the argument, Church surveys two other candidate definientia — calculability by an algorithm and calculability within a logic — in order to show that "no more general definition of effective calculability than that proposed above [i.e. as recursively definable] can be obtained by either of two methods which naturally suggest themselves." I will focus on the second of these; parallel remarks apply to the first. Briefly, for Church, a function F is calculable in a logic if there is an expression f in the logic such that $f(\mu) = \nu$ is a theorem iff F(m) = n is true, where μ and ν

express *m* and *n*, respectively. Importantly, in order for "the system to serve at all the purposes for which a system of symbolic logic is usually intended" Church requires that the logic's inference relation be recursive and that the axioms be effectively enumerable. This all entails that the functions calculable in a logic are extensionally equivalent to the recursive functions. Now, whether or not the argument succeeds is beside the point; for us what matters is that Church chose *this* as an alternate definition of effective calculability. I submit that the fact that Church focussed on this alternative characterizations, instead of some other, is evidence that the notion of an effective procedure was already sharp enough in 1936 to disqualify instructions like Kalmár's.

If effective calculability exhibited open-texture for Church in the dimension of Kalmár-style instructions, then it is difficult to explain why Church focussed on calculability in a logic, instead of some other notion, as an alternate definition of effective calculability. The fact that calculability in a logic is equivalent to recursivity is key here. For that suggests that the kind of permissible operations Church had in mind were no stronger than those available in the course of calculating the value of a recursively defined function, namely, functional application and swapping equals for equals in an equation. And this point generalizes. The fact that all of the different formal characterizations of effective calculability proposed at the time are equivalent to the recursive functions is evidence that the permissible instructions Church had in mind were no more sophisticated than those available in the course of determining the value of a recursively defined function. And that means that the instruction "prove such-and-such by any correct means" would have been automatically disqualified at the time.

We can put the point another way. If in the '30s it was genuinely indeterminate whether or not "prove such-and-such by correct means" was an effective instruction then why did nobody propose an alternate characterization which permitted such an instruction (or something like it)? If Church, as Shapiro thinks, was indeed sharpening effective calculability in one direction, then why did nobody sharpen it in another, mutually incompatible direction? After all, open-texture implies that competent speakers may reasonably disagree about borderline cases, and hence may reasonably disagree about how to sharpen them. Why did it take nearly thirty years for someone (namely Kalmár) to sharpen in the other direction? If Shapiro is correct, then we would expect that at least a few characterizations from around the same timeframe would disagree with Church's. But in fact we see universal agreement, not disagreement.

So, if effective calculability had open-texture it was not in virtue of the kind of instructions allowed to occur in an effective procedure. Next we examine the evidence that infeasible functions, such as Ackermann's, were neither included nor excluded in the extension of effective calculability in the '30s and after.

Shapiro notes that Church, Kleene, and Turing were interested primarily in human calculability. Our dispute concerns what sorts of idealization were imposed on this notion at the time. And, according to Shapiro, reflection on functions like Ackermann's, and subsequent arguments like Mendelson's, impelled researchers to ignore considerations of time and space so long as only a finite amount of each is required. But is there any evidence that the concept of effective calculability, as used in the '30s and a while after, was indeterminate with respect to, say, Ackermann functions? Arguably not.

If effective calculability was indeterminate with respect to A, then we would expect to see some sort of disagreement in the literature as to whether or not A was effectively calculable. But it is telling that no such disagreement exists, at least in the '30s.[73] To be sure, Ackerman functions were well-discussed. But it was apparently not until Porte (1960) that anyone thought to question whether or not A was in fact effectively calculable.[74] But the problematic point for Shapiro is that there is no (apparent) record that A's effective calculability was ever in question. Of course, the fact that there is no record that A's effective calculability was indeterminate does not necessarily entail that A's effective calculability was, at some point, indeterminate. But at the very least Shapiro owes us some evidence *from the 1930s* that A was so indeterminate.

However, even if effective calculability did not exhibit open-texture with respect to A, a general question remains: were any considerations of feasibility built into the concept of effective calculability as it occurred in the 1930s? Here, too, it seems not. Consider, for example, the following

[73] That I am aware of.
[74] Mendelson (1963) quickly dispatches Porte's concern.

example from Turing's 1936 paper. In Section 10 (p. 142) Turing proves that π is calculable in the sense that there is an effective procedure for finding the n^{th} digit of π, for any n. Turing is well aware that the decimal representation of π has infinitely many digits, and is presumably equally aware that there are values of n for which it is infeasible (due to time, space, or whatever) to calculate the n^{th} digit of π. But if Shapiro is correct and the concept of effective calculability (as understood by Turing) was indeterminate for certain cases of infeasible computations, then we should expect that Turing would *not* say that π is calculable. But he does not.

What this tells us is that, contra Shapiro, by the time the '30s came about, the Ackermann function and π were both already sanctioned as effectively calculable. I suggest that this is because the concept was not indeterminate at the time in this respect, for Turing or anyone else.

Before canvassing some objections I want to return briefly to a point brought up in our discussion of Kalmár's function. There we noted that the fact that the different attempts to characterize effective calculability all turn out to be equivalent to recursivity is evidence that the permissible effective instructions exclude Kalmár's. I think that this idea, or something like it, generalizes.

A central component of open-texture is that reasonable, competent agents may disagree about indeterminate cases. In the context of effective calculability this means that if effective calculability has open-texture, then reasonable, competent agents (such as Church, Turing, Gödel, and Kleene) may disagree about the (purported) indeterminate cases. If people may reasonably disagree about the indeterminate cases, then they may reasonably disagree about how to fix them. This entails that it is possible for different (communities of) speakers to fix a single notion in different, mutually incompatible directions. For instance, in the context of feasibility, Shapiro (2013, p. 175) remarks that those who "opted for one of the other notions, such as that of Turing-computable-in-polynomial-time ... or finite-state-computable, cannot be accused of a failure to understand what they are talking about."

Given this consequence of open-texture it is interesting that we do not see different, mutually incompatible notions of effective calculability emerging from the foundational work of the 1930's. Indeed the opposite is true. Not only was there eventual agreement that the effectively calculable functions were all recursive, but Turing's 1936 analysis of effective calculability as Turing machine computability was, for example, hailed by Gödel as "both correct and unique."[75] But what could it mean for a given sharpening to be correct or unique? *Ex hypothesi* open-texture implies that the notion of a 'correct' sharpening is misguided. A given sharpening may be preferred on grounds of fruitfulness, or simplicity, or whatever.[76] But to say that a sharpening is 'correct' misses the point of open-texture altogether. At the very least, Shapiro owes us an account of why there was convergence on a single sharpening at the time, given that those who (would have) disagreed would not be confused about the meaning of 'effective calculability'.

4.4 Objections

Here I discuss three objections to the arguments from Section 4.2 and argue that they both fail. The first objection is that I have changed the subject — the concept I invoke in the empirical evidence above is not the concept Shapiro has in mind. The second is to my general methodology, and that the kind of claims I make, namely predictions about what one would expect to observe, are misguided. The third is that, if it comes down to matters of interpretation, then at least Shapiro's account tells us a story about what Church and the rest meant when they said that effective calculability is vague. So far I have not produced a counter-story, so my account is incomplete, at best.

How could Shapiro's argument so badly misrepresent the concept of effective calculability as it occurred in the '30s? Perhaps it is because Shapiro's target is not the concept of effective calculability in the first place. The above considerations hit their mark only if the concept invoked by them is the same as the concept Shapiro identifies as supporting open-texture. So, the objection goes, we want some further evidence that the two concepts are in fact the same, else the above considerations are simply irrelevant to Shapiro's point.

But plausibly Shapiro has in mind the same notion of effective calculability as does Church,

[75]Quoted in Wang (1974, p. 84).
[76]Echoing Carnap's criteria for a satisfactory explication.

Turing, and Kleene. Firstly, Shapiro (2015, p. 284) tells us that "[t]he founders, Church, Stephen Cole Kleene, Alan Turing, et al., were thinking of computability in terms of what can be done by a suitably idealized human following an algorithm." So plausibly when Shapiro (2013, p. 169) says that ""computability" exhibited open-texture," he means computability in the sense of a suitably idealized human following an effective procedure. Perhaps there is a question here about whether Shapiro's characterization of effective calculability at the time is correct. For instance, it is not entirely clear that the notion of an idealized human calculator was commonly accepted before Turing. Church, for instance, does not explicitly mention idealized humans anywhere in his 1936 paper. But even if this correct it does not mean that our above concerns miss their mark, for they make no prior assumptions about the contents of the notion of effective calculability in play at the time.

The second objection concerns my methodology. In general, I proceed by making claims about what researchers at the time would have written, or more generally about what we would expect to observe in the literature. The fact that these observations are not confirmed, then, I take to be evidence that effective calculability did not exhibit open-texture. The objection, then, is just that what makes it into the literature is not necessarily indicative of the nature of the concept at the time.

Now, if the objection is just that the inferences from what occurs in the literature to the nature of the concept are not deductively sound, then this is just a problem for empirical inquiry more generally. But the point is well-taken: nothing I say above rules out the possibility that open-texture was present in the notion at some point, and was subsequently sharpened away before making it to print. However, this objection cuts both ways. For Shapiro to establish his point he must present us with some evidence (preferably from the '30s, not the '50s or '60s) that effective calculability had borderline cases. As far as I can tell no such evidence is available, at least for the two dimensions of open-texture I've focussed on here. Perhaps examination of, e.g., Church's personal correspondence would provide such evidence. But I've already noted above that my treatment here is not meant to be exhaustive, so at best this is an argument in favour of further investigation.

The final objection is not so much that our present perspective is wrong, but that it falls short. Shapiro's account gives us a story about why (a) Church and Kleene called effective calculability 'vague', and (b) why the question of proving Church's thesis has been so contentious for the past eighty years. But if Shapiro is incorrect then we still require an explanation of (a) and (b), and so far I have not provided anything like such an explanation.

A full answer to this question is far outside our present scope, so I will only gesture towards a response. With respect to (a) it seems as if Church and Kleene and the rest called effective calculability 'vague' because they lacked for a better term. It is perhaps true that effective calculability is vague, if by 'vague' we mean difficult to discern that a given procedure is effective or not. Gödel apparently had this kind of vagueness in mind when making the above quoted remarks. But being difficult to discern does not necessarily entail the existence of indeterminate cases. Moreover, Kleene and Church also call effective calculability 'intuitive' or 'pre-theoretic' or 'informal' just as often as they call it 'vague', with little apparent consistency. Perhaps it is a mistake to put much stock into these remarks. In any case, it appears that there are other conceptions of vagueness available (namely, epistemic) which can explain the claims of vagueness but which don't posit extensional indeterminacy.

So much for (a). What about (b)? I suggest that the perpetual disagreement about the provability of Church's thesis has less to do with vagueness and more to do with the relevant notion of proof at work. In Section 3 we noted Folina (1998)'s view that the notion of proof at work in the '30s and after was that of derivability in a formal system. But the past eighty years has seen that notion relaxed. Mendelson (1990, p. 233), for instance, calls the argument for the right-to-left direction of Church's thesis "as clear a proof as I have seen in mathematics." What has happened here is not that the extension of effective calculability has been fixed, but rather the extension of 'being a proof' has widened to accommodate informal demonstrations alongside formal derivations. Indeed, it is difficult to make sense of much of the history of mathematics if one restricts oneself to too restricted a notion of proof. The question of proving Church's thesis was (and still is) contentious because it represents a struggle between competing conceptions of proof.

4.5 Summary

Shapiro suggests that Church's and Kleene's remarks that effective calculability is vague are best interpreted as claiming that effective calculability exhibits open-texture. Accordingly, Shapiro identifies two functions — Ackermann's and Kalmár's — which, at one point, were neither effectively calculable nor not effectively calculable. On this account it was not until the effective calculability was appropriately sharpened that the functions were included (as with Ackermann's) or ruled out (as with Kalmár's). I have argued that effective calculability did not exhibit open-texture with respect to these two functions. By the time Church published his 1936 paper the kinds of instructions sanctioned as effective already ruled out those required to make true Kalmár's claims, and similarly there is no evidence to suggest that Ackermann's function was ever seriously considered, in the '30s or after, as a possible counterexample to Church's thesis because it relies on unbounded resources for its calculation.

I should note that there is a third option that I have not discussed, which is the development of the partial recursive functions. Perhaps the move from total to partial functions is an instance of sharpening. This strikes me as plausible, but a full investigation is beyond our current scope. Certainly it is an avenue worth investigating; unfortunately it is not one we will here explore.

5 Conclusion

At the beginning of this paper we characterized our project as one of philosophical path-clearing. Thus the project is successful only if matters are indeed clearer now than they were before. I am not so confident in my conclusions to say that matters are crystalline — but perhaps it is clearer now which paths are open to us and which are not. So I will briefly summarize what has been cleared and what remains to be cleared.

As long as one thinks that Church's thesis is true but unprovable there will be a question about its nature — about the kind of claim it makes and how the evidence brought for or against it bears on its truth. We have seen that one possible answer to this question, that the thesis is an explication in Carnap's sense, is implausible. As long as one wants to derive uncomputability results (and there are good reasons to think that one should want to) then explication is not the route to take. This is because explication prevents us from deriving uncomputability results. The purpose of explication is to replace vague, imprecise concepts with determinate, precise ones. But uncomputability results, at least in their most interesting form, tell us something about the vague, imprecise notion of effective calculability, namely, that a given function is not effectively calculable. These results are desirable. And as long as explicationism denies them to us we should reject it.

Explicationism and the standard view are motivated by the thought that effective calculability is vague. But is it? I have argued that it is not, or, at least, that any vagueness is merely epistemic. And such vagueness, importantly, does not stand in the way of a proof of Church's thesis, for it may be ameliorated by careful conceptual analyses, such as Turing's. Stewart Shapiro agrees that vagueness does not stand in the way of a proof, but disagrees about why. According to Shapiro effective calculability exhibited open-texture at one point, but that open-texture has been sharpened away over the last eighty or so years. However, these are an empirical claims and it is not clear that the historical record supports them. If effective calculability exhibited open-texture in the '30s, then we would expect more disagreement between different characterizations than we in fact observe. Moreover, Shapiro cannot explain why Church chose to argue for his thesis the way that he did, or why there is no apparent disagreement over the effective calculability of Ackermann functions when they were first presented. All of this suggests that, at least in the '30s, effective calculability did not exhibit open-texture in the way Shapiro thinks. I suggest that this is because effective calculability never exhibited open-texture in the first place.

There is much that we have not covered. There are questions about the proper conception of proof to consider when thinking about Church's thesis. There are questions about whether there are other kinds of vagueness, not mentioned above, that might be lurking in the concept of effective calculability. And there is the question of whether or not the move to partial recursive functions exhibits a sharpening of effective calculability. This paper has largely been critical in that it points out what doesn't seem to work. But there is still much to be done to figure out what does.

References

Akl, S. G. (2005). The Myth of Universal Computation. *Parallel Numerics*, *2*, 211-236.

Berg, J., & Chihara, C. (1974). Church's Thesis Misconstrued. *Philosophical Studies*, *28*(5), 357–362.

Black, R. (2000). Proving Church's Thesis. *Philosophia Mathematica*, *8*(3), 244–258.

Boolos, G. S., Burgess, J. P., & Jeffrey, R. C. (2007). *Computability and Logic* (5th ed.). New York, NY: Cambridge University Press.

Bowie, G. L. (1973). An Argument Againt's Church's Thesis. *The Journal of Philosophy*, *70*(3), 66 - 76.

Carnap, R. (1950). *Logical Foundations of Probability*. London: Routledge and Kegan Paul.

Carnap, R. (1963). P. F. Strawson on Linguistic Naturalism. In P. A. Schilpp (Ed.), *The Philosophy of Rudolf Carnap* (Vol. 11). La Salle, Ill.: Open Court.

Carus, A. W. (2007). *Carnap and Twentieth Century Thought*. Cambridge, UK: Cambridge University Press.

Church, A. (1936). An Unsolvable Problem of Elementary Number Theory. In M. Davis (Ed.), *The Undecidable* (p. 89 - 107). Hewlett, N.Y.: Raven Press Books.

Copeland, B. J. (2015). The Church-Turing Thesis. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Summer 2015 ed.).

Folina, J. (1998). Church's Thesis: Prelude to a Proof. *Philosophia Mathematica*, *6*(3), 302–323.

Gandy, R. (1988). The Confluence of Ideas in 1936. In R. Herken (Ed.), *The Universal Turing Machine: A Half-Century Survey* (p. 55 - 111). Hamburg: Kammerer & Unverzagt.

Kalmár, L. (1959). An Argument Against the Plausibility of Church's Thesis. In A. Heyting (Ed.), *Constructivity in Mathematics: Proceedings of the Colloquium Held at Amsterdam, 1957.* Amsterdam: North-Holland.

Kleene, S. C. (1935). General Recursive Functions of Natural Numbers. In M. Davis (Ed.), *The Undecidable* (p. 237 - 254). Raven Press Books.

Kleene, S. C. (1952). *Introduction To Meta-Mathematics*. Groningen: Wolters-Noordhoff.

Kleene, S. C. (1987). Reflections on Church's Thesis. *Notre Dame Journal of Formal Logic*, *28*(4), 490 - 498.

Lakatos, I. (1976). *Proofs and Refutations: the Logic of Mathematical Discovery*. Cambridge: Cambridge University Press.

Linz, P. (2012). *An Introduction to Formal Languages and Automata* (5th ed.). Sudbury, MA: Jones and Bartlett Learning.

Maher, P. (2007). Explication Defended. *Studia Logica*, *86*(2), 331–341.

Mendelson, E. (1963). On Some Recent Criticisms of Church's Thesis. *Notre Dame Journal of Formal Logic*, *4*, 201 - 205.

Mendelson, E. (1990). Second Thoughts about Church's Thesis and Mathematical Proofs. *The Journal of Philosophy*, *87*(5), 225–233.

Murawski, R., & Woleǹski, J. (2006). The Status of Church's Thesis. In A. Olszewski, J. Woleǹski, & R. Janusz (Eds.), *Church's Thesis After 70 Years* (p. 310 - 330). Frankfurt: Ontos Verlag.

Odifreddi, P. (1992). *Classical Recursion Theory: The Theory of Functions and Sets of Natural Numbers*. Amsterdam: North-Holland.

Péter, R. (1935). Konstruktion Nichtrekursiver Funktionen. *Mathematische Annalen*, *111*, 42 - 60.

Porte, J. (1960). Quelques Pseudo-paradoxes de la 'Calculabilité Effective'. In *Actes du 2me congrés international de cybernetique* (p. 332 - 334). Namur, Belgium: Association Internationale de Cybernetique.

Post, E. (1936). Finite Combinatory Processes-Formulation I. In M. Davis (Ed.), *The undecidable* (p. 289 - 291). Hewlett, N.Y.: Raven Press Books.

Post, E. (1944). Recursively Enumerable Sets of Positive Integers and Their Decision Problems. In *The Undecidable* (p. 305 - 337). Hewlett, N.Y.: Raven Press Books.

Quine, W. V. (1964). *Word and Object*. Cambridge, MA: MIT Press Press.

Rescorla, M. (2007). Church's Thesis and the Conceptual Analysis of Computability. *Notre Dame*

Journal of Formal Logic, *48*(2), 253–280.

Rogers, H. (1967). *Theory of Recursive Functions and Effective Computability*. Cambridge, MA: MIT Press.

Shapiro, S. (2006a). Computability, Proof, and Open-texture. In A. Olszewski, J. Wolenski, & R. Janusz (Eds.), *Church's thesis after 70 years* (pp. 420–455). Ontos Verlag.

Shapiro, S. (2006b). *Vagueness in Context*. Claredon: Oxford: Oxford University Press.

Shapiro, S. (2013). The Open Texture of Computability. In B. J. Copeland, Carl J. Posy, & O. Shagrir (Eds.), *Computability: Turing, Gödel, Church, and Beyond.* Cambridge, MA: MIT Press.

Shapiro, S. (2015). Proving Things About the Informal. In G. Sommaruga & T. Strahm (Eds.), *Turing's Revolution: The Impact of His Ideas about Computability* (p. 283 - 295). Springer International.

Sieg, W. (1991). Mechanical Procedures and Mathematical Experience. In *Proceedings of the Conference on Mathematics and Mind.*

Sieg, W. (1997). Step by Recursive Step: Church's Analysis of Effective Calculability. *Bulletin of Symbolic Logic*, *3*(02), 154–180.

Sieg, W. (2008). Church Without Dogma: Axioms for Computability. In S. B. Cooper, B. Lowe, & A. Sorbi (Eds.), *New Computational Paradigms* (p. 139-152). New York: Springer.

Smith, P. (2007). *An Introduction To Godel's Theorems*. Cambridge, UK: Cambridge University Press.

Turing, A. M. (1936). On Computable Numbers, With An Application To The Entscheidungsproblem. In M. Davis (Ed.), *The Undecidable* (p. 116 - 151). Hewlett, N.Y.: Raven Press Books.

Wang, H. (1974). *From Mathematics to Philosophy*. London: Routledge and Kegan Paul.

Wang, H. (1996). *A Logical Journey: From Gödel to Philosophy*. Cambridge, MA: MIT Press.

Webster, N., Harris, W., & Porter, N. (Eds.). (1907). *Webster's International Dictionary of the English Language*. Springfield, Mass.: G. & C. Merriam company.