Leon Horsten*

Formalizing Church's Thesis

This paper investigates to what extent Church's Thesis can be expressed and investigated within a formal framework. In particular, we discuss the formalization of Church's Thesis in intuitionistic arithmetic and formalizations of Church's Thesis in systems of epistemic arithmetic.

1. The Formal Investigation of Church's Thesis

Church's Thesis (CT) states that every function on the natural numbers that is effectively computable, is computable by a Turing machine. Effectively computable functions are functions computable by an idealized but human calculator working in a routine or algorithmic fashion, i.e., on the basis of a rule-governed procedure where no ingenuity is required for its execution.

Church's Thesis is true. Most philosophers and mathematicians think that it is not mathematically provable.[1] The reason that is usually given is that it is not a purely mathematical proposition. It is not even expressible in the language of mathematics, for its antecedent contains a notion ("algorithmic computability") that does not belong to the language of mathematics.

CT may be amenable to formal investigation provided we find a suitable interpreted formal language in which it can be expressed, even if this language contains notions which are not strictly speaking

*L. Horsten, University of Leuven, <Leon.Horsten@hiw.kuleuven.be>. Thanks to Mark van Atten for helpful discussions about the subject matter of this paper, and thanks to Jan Heylen for comments on a version of this paper. I am indebted to Anne Troelstra for answering a proof-theoretical question about the intuitionistic reading of Church's Thesis.

[1]But this is controversial. See the discussion between Folina, Mendelson, Black and Shapiro [Mendelson 1990], [Folina 1998], [Black 2000], [Mendelson this volume], [Folina this volume], [Shapiro this volume].

purely mathematical.[2] In a suitable formal context, questions about CT can be investigated with formal rigor. This paper discusses some attempts that have been made in this direction.

It is not hard to express CT in a formal language. One could just take the language of arithmetic, add to it a new predicate intended to apply to all algorithms, a relation symbol expressing the relation that holds between an algorithm and the function it computes, and perhaps some symbols standing for operations on algorithms. But that does not help much. For it is notoriously hard to formulate an acceptable criterion of identity for algorithms. In other words, the interpretation of the resulting formalism appears to be vague. Perhaps an illuminating set of basic axioms concerning the notion of algorithm can be formulated in this formal language, but no existing proposal has been met with widespread approval.

In order to circumvent this difficulty, we shall in this paper settle for *approximations* of CT in formal contexts. We shall look at formal languages with a relatively clear and not overly complicated intended interpretation in which CT can be approximatively expressed. The advantage of this approach is that precise answers to questions concerning CT can sometimes be obtained. But the answer will always be an answer about the question concerning the approximation of CT that one is investigating.

We shall be concerned with approximations, not with variations. I.e., we shall not be concerned with analogues of CT. We shall be concerned with attempts to come close to CT *itself* in a formal context.

We shall not address the question whether CT can be *proved*. We shall be mainly concerned with the question whether CT can be formally expressed within a rigorous framework. And as such, it will be considered as a hypothesis (not as an *axiom*). Rather than being concerned with whether this hypothesis can be proved, we shall to a limited extent address the question whether facts of mathematical and philosophical interest can be derived *from* it.

2. Church's Thesis in Intuitionistic Mathematics

It has been claimed that CT can be expressed in the language of intuitionistic arithmetic, namely roughly as the following scheme:

[2]Such an approach is be favorably looked upon by [Mendelson this volume].

$$\forall x \exists y A(x,y) \rightarrow \exists e \forall x \exists m \exists n [T(e,x,m) \wedge U(m,n) \wedge A(x,n)] \qquad ICT$$

Here A ranges over all formulas of the language of first-order intuitionistic arithmetic.[3] Note that a (weak) choice principle is presupposed in ICT, for $A(x,y)$ expresses a relation, whereas the Turing machine e by definition computes a function.

The fact that ICT at least goes some length in the direction of expressing CT is due to the semantics of the language of intuitionistic arithmetic. $\forall x \exists y A(x,y)$ intuitionistically means that there is a *method* for finding for all x at least one y which can be shown to stand in the relation A to x. And this seems close to asserting that an algorithm exists for computing A.

There is disagreement in the intuitionistic literature about whether the method witnessing the truth of $\forall x \exists y A(x,y)$ should be accompanied by a *proof* that this method computes A. Some intuitionists deny this, saying that the method bears on its sleeves the task that it carries out. But in any case it has to be evident from the interpretation of $\forall x \exists y A(x,y)$.

ICT is false if lawless sequences are allowed in the constructive universe. For these are assumed to be generated by a method (free choice) but already at the outset it seems unlikely that all manners of freely generating sequences of natural numbers in the infinite limit result in a computable function.

Even if lawless or choice sequences are not allowed in the constructive universe, ICT looks suspect. For it asserts that there is a *uniform method* for transforming a method for finding for all x an y which can be shown to stand in the relation A to x into a Turing machine which does the same thing. It is not clear what this uniform method looks like, unless one thinks of methods as almost by definition something like Turing machines.

So most intuitionists believe that ICT has been shown to be false. But this is at first sight puzzling. On the one hand we have intuitionists claiming that they have refuted CT. On the other hand in virtually all of the contemporary literature outside intuitionism CT is regarded as true.

This puzzle disappears once it is noted that ICT does not really express CT. For one thing, a free choice process is not an algorithm in the sense explained in the first section of this paper. But even if we

[3]We are assuming that the reader is familiar with Kleene's T-predicate and U-function.

confine ourselves to the "lawlike" intuitionistic universe, ICT does not express CT. Firstly, the implication in ICT is constructive: it expresses a uniform transformation procedure. But the implication in CT is classical: it does not imply the existence of such a transformation procedure. And secondly, the consequent of ICT requires that for some e we have a *proof* that e computes A. But CT does not require this: it merely requires that some e in fact computes A.

But still, if we confine ourselves to the lawlike universe, ICT is a fair approximation of CT. It gives us reliable implications of CT, provided we read them carefully. For instance, in the context of intuitionistic arithmetic ICT implies a violation of a variant of the law of excluded third. This does not mean that the universal version of the *classical* law of excluded third is false, but only that an effective version of it is incorrect. If CT is correct, then for some properties $\phi(x)$, there is no uniform method for finding out for an arbitrary number whether it has ϕ or not.

Intuitionism provides a useful setting for thinking about aspects of CT. Shapiro discussed the phenomenon that CT is often used to prove theorems that can be stated in the language of (classical) Peano Arithmetic (PA), i.e., to prove statements in which the notion of algorithm does not occur [Shapiro 1983, p. 215]:

> In recursive function theory, there has developed an 'informal' method of argument, often called 'argument by Church's thesis' that employs the [...] inference from computability to recursiveness. Typically, the object is to demonstrate the existence of a recursive function which has a given property P. Using this method, a mathematician first gives a procedure of calculation 'informally' (that is, outside of any particular formulation of algorithms) and then infers that the function computed by this procedure is recursive *because* it is computable. The mathematician then establishes that the function has the property P.

The question then arises whether such uses of CT are *essential*: can every theorem statable in the language of PA, say, that is proved using CT, also be proved without using CT? Generally it is assumed that the answer to this conservativity question is 'yes' but in the same breath it is added that since CT is an 'informal' principle, this cannot be formally proved.

In the intuitionistic setting, a precise answer can be given to *one* way of making this question precise. Let us restrict the discussion to the lawlike universe. Then a proof of a sentence of the form $\forall x \exists y A(x, y)$ consists (again modulo a choice principle) in a proof that $A(x, y)$ is algorithmically computable. So in this setting, it seems that ICT can be meaningfully used to prove that certain functions are Turing-computable. Now the question arises whether modulo the double negation translation, intuitionistic arithmetic (HA, for Heyting Arithmetic) plus ICT is conservative over PA. The answer is affirmative. If we let δ be the double negation translation from the language of intuitionistic arithmetic to the language of classical arithmetic, then we have:[4]

Theorem 1. For every sentence ϕ of the language of classical arithmetic, if $\delta(\phi)$ is provable in $HA + ICT$, then ϕ is provable in PA.

This theorem is a direct consequence of the fact that (1) realizability is conservative over almost-negative formulas and the double-negation translation translation transforms formulas into almost-negative formulas; (2) realizability makes ICT true.

This conservativity phenomenon can be seen as a *weak* form of evidence for the thesis that CT is conservative over classical mathematics.

3. Church's Thesis in Epistemic Mathematics

The $S4$ laws of modal logic describe the propositional logic of the notion of reflexive provability. The reflexive notion of absolute provability should be carefully distinguished from the notion of provability in a formal system. As is well-known, the propositional logic of provability in a formal system is captured by the Gödel-Löb system GL of modal logic. The reflexivity axiom $\Box A \rightarrow A$ is invalid on this interpretation, whereas it is valid on the interpretation of $\Box$ as reflexive provability.

The formal language of epistemic arithmetic then contains a sentential operator whose interpretation is the reflexive notion of provability. Aside from that, the language of epistemic arithmetic contains a constant $\underline{0}$ referring to the number 0, the function symbols

[4]Thanks to Anne Troelstra for pointing this out to me.

s (successor) and $+$ and $\times$, and names for all primitive recursive functions.

$S4PA$, Peano Arithmetic (with the defining equations for the primitive recursive functions) plus the $S4$ axioms formulated in the language of arithmetic extended with the operator $\Box$, describes the notion of reflexive provability in an arithmetical context. Actually, it seems that reflexive provability should be formalized as a *predicate* rather than as an operator. For we would like to be able to express things such as:

"Some sentences are reflexively provable."

But this would seriously complicate the formalism. For we would have to take care to avoid the Kaplan–Montague paradox concerning absolute knowability [Montague 1963]. However, the effect of quantifying over sentences can also be achieved by adding a truth predicate to the language, governed by suitably restricted Tarski-biconditionals, for example. Anyway, here we keep matters simple and express reflexive provability by a sentential operator.

$S4PA$ is known as a system of intensional or epistemic mathematics [Shapiro 1985b], but strictly speaking this is a misnomer. For the notion of reflexive provability is not a purely mathematical notion, although it is of course related to mathematics. The notion of reflexive provability is just as much a philosophical notion as the notion of truth is. It must be admitted that the notion of reflexive provability is less well understood than the notion of truth is: we shall have to return to this later.

Via Gödel's modal translation, which closely follows Heyting's proof semantics for the intuitionistic logical operations, $S4PA$ is related to intuitionistic arithmetic by a faithfulness theorem [Goodman 1984], [Flagg and Friedman 1986]. But $S4PA$ is nevertheless a classical theory. So aside from being able to express intuitionistic statements via the Gödel translation, it can express classical propositions as well as propositions which are in part constructive and in part nonconstructive. This opens the possibility of improving on ICT as an approximation of CT by removing the effectiveness from the implication and from its consequent.

Several proposals have been made for expressing CT in epistemic arithmetic, but here we shall look at the following:

$$\Box\forall x\exists y\Box A(x,y) \rightarrow \exists e\forall x\exists m\exists n(T(e,x,m) \wedge U(m,n) \wedge A(x,n)) \quad ECT$$

Here A ranges over all formulas of the language of classical first-order arithmetic plus the reflexive proof operator. As in the case of ICT, here too an epistemic choice principle is presupposed.

The implication in ECT is classical. Therefore the first reason why ICT did not quite express CT does not apply here. Secondly, the whole consequent of ECT is nonconstructive. So also the second reason why ICT did not really express CT is not applicable here.

The crucial part of ECT is its antecedent: $\Box\forall x\exists y\Box A(x,y)$. It contains no direct mention of the notion of algorithm. But it is equivalent to expressing that a suitable algorithm exists provided that the following thesis holds:

Thesis 1. A proof witnessing the truth of $\Box\forall x\exists y\Box A(x,y)$ for a given formula A must involve displaying an algorithm for computing A.

This thesis is related to the intuitionist claim that there is a close relation between a method for computing a function and a (constructive) proof that the function exists.

It is difficult to see how ECT could be proved in epistemic mathematics from more fundamental principles. But in addition, ECT should probably not be adopted as an extra axiom. ECT is "necessarily" a hypothesis. For otherwise, the Necessitation rule can be applied to ECT. This would mean that there is a *proof* that for any algorithm computing a function, there is a Turing machine which computes that function. And as in the case of ICT, it would seem that this proof would have to involve a uniform transformation procedure for converting algorithms into Turing machines. And short of coming close to identifying algorithms with Turing machines by definition, it is hard to see what this transformation procedure could look like.

ECT is consistent with epistemic mathematics [Flagg 1985]. In $S4PA$, one can prove that functions are effectively computable by proving sentences of the form $\Box\forall x\exists y\Box A(x,y)$. So again the question can be asked whether ECT allows us to prove purely arithmetical theorems that we could not prove without ECT. When ECT is taken as a hypothesis, the situation is the same as for ICT. When taken as a hypothesis, ECT is arithmetically conservative over PA [Halbach and Horsten 2000]. In other words, in epistemic mathematics, the informal notion of algorithmic computability, when it is governed by

ECT, does not give us new mathematical theorems. This is again a weak sort of evidence that CT is conservative over arithmetic.

What if we *do* adopt ECT as an axiom? The question whether $S4PA + ECT$ is arithmetically conservative over Peano arithmetic is still open, as far as I know, although it may well be the case that methods developed by Timothy Carlson can be used to show that $S4PA + ECT$ is indeed conservative over PA [Carlson 1999], [Carlson 2000]. So the notion of algorithm is, at least insofar as we now know, unlike the notion of truth. For it is well-known that truth is nonconservative.

Other proof-theoretical questions can be thought of that have philosophical significance. It was noted in the beginning of this section that HA is faithfully embedded in first-order epistemic arithmetic via Gödel's modal translation from the language of intuitionistic arithmetic to the language of epistemic arithmetic. If one adds ECT to epistemic arithmetic, ICT (under Gödel's translation) does not directly follow from it. So one may wonder whether *any* new intuitionistic statements become provable from ECT (under Gödel's modal translation). This again appears to be question that is still open.

One marked weakness of the whole program of epistemic arithmetic is that the notion of reflexive provability is not very well understood at all. Some suspect that the notion is inherently vague. In any case, our theory of it is very restricted at the moment.[5]

A long time ago, Myhill encouraged researchers to take an axiomatic approach to absolute provability [Myhill 1960, p. 468]. In that respect, Kreisel was pessimistic [Kreisel 1983, pp. 86–87]:

> *Truth and general provability*; at least so far, a distinction without much difference [...] nothing is formulated about general provability that does not also hold for truth [...] In fact, it seems open whether anything can be said about general provability in the language considered that does not also hold for truth.

But even in the light of our feeble grasp on the notion of absolute provability, the situation is not that grim. One cannot say that we have at present *no* (axiomatic) understanding of the notion of the notion of reflexive provability at all. First, the logical principles

[5] I discuss this problem in some detail in [Horsten 2005b].

concerning absolute provability are probably not as strong those for truth. It would seem implausible that the Tarski-biconditionals hold for the notion of absolute (classical) provability. Secondly, ECT is a candidate for precisely the sort of principle that Kreisel was calling for. For ECT might well be true; but if in ECT the concept of provability is replaced by that of (classical) truth, the resulting principle is provably incorrect.

In sum, taking the notion of reflexive provability as primitive is not quite on a par with taking the notion of algorithm as primitive. For in contrast with the notion of algorithm, we do appear to have an elegant theory which yields the basic logical properties of reflexive provability. And even if the content of the notion of algorithm cannot be fully *reduced* to the reflexive notion of provability, if $\Box\forall x\exists y\Box A(x,y)$ turns out to be true if and only if A is algorithmically computable, then this will have consequences that are interesting in their own right.

4. Intensional Aspects of Church's Thesis

Shapiro believes that CT cannot be directly captured in terms of the absolute provability operator. CT concerns the notion of computability: a function is computable if there is an algorithm that computes it. So computable is objective in the sense that it "does not involve reference to a knowing subject" and extensional [Shapiro 1985b, p. 41]. But closely related to the notion of computability there is a *pragmatic* notion, which Shapiro calls *calculability* (or "effectiveness", in the terminology of [Shapiro 1980]). Calculability is a property of presentations of algorithms: a function presentation F is calculable if there is an algorithm $\mathtt{P}$ such that it can be established that F represents $\mathtt{P}$ [Shapiro 1985b, p. 43]. Shapiro suggests that Epistemic Arithmetic can be used to shed light on this latter notion: the antecedent of CT expresses the calculability of a function presentation.

The claim that the notion of computability does not involve reference to a knowing subject strikes me as untrue. For the notion of computability is explicated in terms of the notion of algorithm. And this latter notion does involve reference to a knowing subject, as is evident from Turing's analysis [Turing 1936].[6] Turing explains

[6]Turing's analysis is carefully described and discussed in [Soare 1996].

the notion of algorithm in terms of the notion of a computor, which is a *human being* who calculates following a routing procedure.[7] It is true that human calculation (which is involved in CT) is not the same as human provability (which is involved in ECT). But Thesis 1 is supposed to connect the two notions.

Computability is indeed extensional: if two function presentations denote the same function, then one is algorithmically computable if the other is. We have no reason to think that the antecedent of ECT is extensional. It is not hard to think of two functional expressions which may denote the same function but for which one satisfies the antecedent of ECT while the other does not.

But calculability is at least epistemically more basic than computability. For to determine that a function is computable, we have to determine that there is a function presentation which is calculable, i.e., an algorithm [Shapiro 1980, p. 219]. In fact, calculability seems also to be ontologically more basic than computability. For the notion of computability is obtained by an act of *abstraction* from the notion of calculability.

There is an ambiguity at the heart of the notion of algorithm. On the one hand it is usually said that an algorithm is just a procedure for transforming numbers into other numbers. On the other hand, an algorithm is usually intended to compute a function given under a specific presentation. As such, an algorithm is more than a transformation procedure. Nicolas Goodman put it thus: "The specification of the algorithm is complete only if it includes a statement of the problem it is intended to solve" [Goodman 1987, p. 483]. This seems not quite right: it is rather a transformation procedure plus a *proof* that this procedure computes a function presented in a specific way. But it is not completely clear that even this is what is meant in the vulgar usage by the term 'algorithm'. For it might also be thought that it should be evident from the specification of the algorithm itself which problem it is intended to solve. The indeterminacy here mirrors the ambiguity of the meaning of implication in intuitionism, which was discussed in section 2.

However this may be, this phenomenon explains the puzzlement that students experience upon first being told that the total function

[7]This is emphasized in [Wittgenstein 1980, p. 1096].

f, defined as

$$f(x) = 1 \text{ if Goldbach's conjecture is true;}$$
$$f(x) = 0 \text{ otherwise}$$

is algorithmically computable (either the constant 0 function or the constant 1 function computes it). One has the feeling that, e.g., the constant 1 function is not really an algorithm for computing the function presented above unless it is accompanied by a proof that the algorithm computes the function in question. After all, a transformation procedure involves also a function presentation. The transformation procedure is an algorithm for computing a function presented in a specific way only if the graph of the transformation procedure is provably the graph of the function in the intensional sense of the word.

What has happened is that the contemporary textbook use of the term 'algorithm' is *abstracted* from the intensional use of the term algorithm just described. Mathematics is, at least officially, extensional: it abstracts from the ways in which mathematical objects are presented. The notion of algorithm is not a fully domesticated notion: it remains "informal". But it has become a half-domesticated notion: it has been reduced to an extensional notion.

But the extensional meaning of the contemporary use of the term 'algorithm' can be expressed in terms of the reflexive notion of provability—or at least it can be thus approximated. The idea is that a function is computable if *it has* a calculable presentation. In other words, we come closer to expressing CT if we say:

> If a function has a calculable presentation, then it is Turing-computable.

In this sense, one might attempt some sort of reduction of the concept of computability to the concept of calculability. One advantage of this formulation is that the converse of it, is obviously valid. And this is as it should be, for it is generally held that the converse of CT is obviously valid [Black 2000], [Mendelson this volume]. It is not in the least obvious that the converse of ECT is valid!

But the above attempt to paraphrase CT quantifies over function presentations. So we have to move to a *second-order* epistemic framework. Let $S4PA2$ be exactly like $S4PA$, except that the language of the theory contains second-order predicate variables, and

the background arithmetical theory is second-order (classical) arithmetic.

In the framework of $S4PA2$, the paraphrase of CT in terms of calculability can be expressed:

$$\forall X : [X \text{ expresses a function} \wedge$$

$$\exists Y : \forall x \forall y (X(x,y) \leftrightarrow Y(x,y)) \wedge \Box \forall x \exists y \Box Y(x,y)] \qquad ECT2$$

$$\rightarrow (X \text{ expresses a recursive function.})$$

This formalization comes closer to capturing the content of CT. In contrast to ECT, the converse of ECT2 is clearly true. And this is as it should be, for as was noted above, the converse of CT is clearly true. Moreover, in contrast to the antecedent of ECT, the antecedent of ECT2 is clearly extensional. Therefore ECT2 respects Shapiro's stricture that was discussed in the the beginning of this section. ECT2 shows how computability as an extensional notion is abstracted from what Shapiro calls the intensional notion of calculability.

5. More on the Epistemic Framework

Quine has famously insisted that any regimented theory worthy of the name should have a clear interpretation. Let us apply this to the epistemic background theories of the previous two sections: first- and second-order epistemic arithmetic.

The epistemic framework is an intensional logic. Quine has always felt that intensional logics do not have a clear interpretation. In section 3 it was conceded that the notion of reflexive provability is not as clear as one would wish it to be. But Quine held that there is a *specific* problem with intensional logic: quantifying into intensional contexts is genuinely problematic [Quine 1955].

Against this, I maintain that in a formal context, quantification into epistemic contexts is unproblematic and uncomplicated as long as every object in the domain of discourse can only be referred to by *transparent* terms. And this is the case for the languages of first- and second-order epistemic arithmetic that we have employed. Here is a

sketch of the intended interpretation of quantification in epistemic arithmetic.[8]

Let us first consider first-order quantification. Terms of the language of $S4PA$ must be built from individual variables, $\underline{0}, s, +, \times$ and primitive recursive function symbols. Given an assignment of numbers to the free variables, identity of denotation between two terms s and t is always decidable. So the Kripkean identity and substitution principles

$$s = t \rightarrow \Box(s = t)$$

$$s = t \rightarrow (\phi(x \setminus s) \leftrightarrow \phi(x \setminus t)) \quad \text{for all formulas } \phi$$

are valid. Therefore we can read quantified statements in a "Gödelian", substitutional manner. A statement $\exists x \Box \phi(x)$ can for all intents and purposes be read as: "there is a natural number such that when its standard Peano-numeral replaces the variable x in $\phi(x)$, a provable sentence results." The reason is that modulo provable equivalence, every natural number is nameable in the language by a *unique* term.

The situation becomes more complicated only when not every object in the domain of discourse has a name (such as in the case of the real numbers) or when we allow different terms that in fact refer to the same number, but not *provably* so.[9] But this was avoided in the epistemic frameworks that we have relied on. The opaqueness was restricted to the predicate expressions and was not allowed to spread to the terms.

Substitution of predicates in intensional contexts is governed by the transitivity axiom of $S4$. So we do *not* have in $S4PA2$ the substitution principle

$$\forall X \forall Y : \forall x(Xx \leftrightarrow Yx) \rightarrow (\Phi(X) \leftrightarrow \Phi(Y)),$$

but we do have the weaker:

$$\forall X \forall Y : \Box \forall x(Xx \leftrightarrow Yx) \rightarrow (\Phi(X) \leftrightarrow \Phi(Y)).$$

[8] I have spelled out the intended semantics of first- and second-order languages of epistemic arithmetic in more detail in [Horsten 1998, section 4.2.] and in [Horsten 2005a, section 2].

[9] This latter situation may arise, for instance, when we include a description operator in the formal language.

Still, we have to be clear what an expression of the form $\exists X \Box \Phi(X)$ is supposed to *mean*. I suggest that we take the intended interpretation to be: "there is a *set* of natural numbers S and a *presentation* P_S of S such that when the second-order variable X is replaced in Φ by P_S, a provable sentence results." It is important to be as liberal as possible with respect to admissible presentations of sets [Horsten 1998, p. 17, and footnote 30, p. 24]. Even a set itself is allowed to count as its own presentation. This ensures that the quantifier $\forall X$ in $\forall X \Phi(X)$ has *all* sets of natural numbers in its range—even those that for all we know have no humanly conceivable presentation. Only certain *kinds* of presentations (notably presentations expressible in human languages) can figure in reflexive proofs. For this reason, a sentence such as $\forall X \Box \forall y(Xy \leftrightarrow Xy)$ is not valid. For *of* sets of numbers that have no humanly expressible and usable presentation it is impossible to prove *anything*.

In this way both the absolute generality of second-order quantification and the impossibility of full-fledged de re knowledge of sets of objects can be respected. In sum, the interpretation of quantifying in epistemic contexts is straightforward and unproblematic for the epistemic theories that we have considered.

References

Black, R. [2000], "Proving Church's Thesis", *Philosophia Mathematica* **8**, 244–258.

Carlson, T. [1999] "Ordinal Arithmetic and $Sigma_1$ Elementarity". *Archive for Mathematical Logic* **38**, 449–460.

Carlson, T. [2000], "Knowledge, Machines, and the Consistency of Reinhardt's Strong Mechanistic Thesis", *Annals of Pure and Applied Logic* **105**, 51–82.

Flagg, R. [1985], "Church's Thesis is Consistent with Epistemic Arithmetic", in [Shapiro 1985a, pp. 121–172].

Flagg, R. and Friedman, H. [1986], "Integrating Classical and Intuitionistic Type Theory", *Annals of Pure and Applied Logic* **32**, 27–51.

Folina, J. [1998], "Church's Thesis: Prelude to a Proof", *Philosophia Mathematica* **6**, 302–323.

Folina, J. [this volume], "Church's Thesis and the Variety of Mathematical Justifications".

Goodman, N. [1984], "Epistemic Arithmetic is a Conservative Extension of Intuitionistic Arithmetic", *Journal of Symbolic Logic* **49**, 192–203.

Goodman, N. [1987], "Intensions, Church's Thesis and the Formalization of Mathematics", *Notre Dame Journal of Formal Logic* **28**, 473–489.

Halbach, V. and Horsten, L. [2000], "Two Proof-Theoretic Remarks on $EA + ECT$", *Mathematical Logic Quarterly* **46**, 461–466.

Horsten, L. [1998], "In Defense of Epistemic Arithmetic", *Synthese* **116**, 1–25.

Horsten, L. [2005a], "Canonical Naming Systems", *Minds and Machines* **15**, 229–259.

Horsten, L. [2005b], "Remarks on the Content and Extension of the Notion of Provability", *Logique et Analyse* **189–192**, 15–32.

Kreisel, G. [1983], "Gödel's Excursions into Intuitionistic Logic", in P. Weingartner and L. Schmetterer, *Gödel Remembered. Gödel-Symposium in Salzburg, 10–12 July 1983*, Bibliopolis, pp. 65–186.

Mendelson, E. [1990], "Second Thoughts about Church's Thesis and Mathematical Proofs", *Journal of Philosophy* **87**, 201–205.

Mendelson, E. [this volume], "On the Impossibility of Proving the 'Hard-Half' of Church's Thesis".

Montague, R. [1963], "Syntactical Treatments of Modality, with Corollaries on Reflexion Principles and Finite Axiomatizability", *Acta Philosophica Fennica* **16**, 153–167.

Myhill, J. [1960], "Some Remarks on the Notion of Proof", *Journal of Philosophy* **57**, 461–471.

Quine, W.V. [1955], "Quantifiers and Propositional Attitudes", in W.V. Quine [1976], *The Ways of Paradox and Other Essays*, 3rd edition, Harvard University Press, pp. 100–112.

Shapiro, S. [1980], "On the Notion of Effectiveness", *History and Philosophy of Logic* **1**, 209–230.

Shapiro, S. [1983], "Remarks on the Development of Computability", *History and Philosophy of Logic* **4**, 203–220.

Shapiro, S. (ed.) [1985a], *Intensional Mathematics*, North-Holland.

Shapiro, S. [1985b], "Epistemic and Intuitionistic Arithmetic", in [Shapiro 1985a, pp. 11–46].

Shapiro, S. [this volume], "Church's Thesis: Computability, Proof, and Open-Texture".

Soare, R. [1996], "Computability and Recursion", *Bulletin of Symbolic Logic* **2**, 284–321.

Turing, A. [1936], "On Computable Numbers, with an Application to the *Entscheidungsproblem*", *Proceedings of the London Mathematical Society* **42**, 230–265.

Wang, H. [1974], *From Mathematics to Philosophy*, Routledge & Kegan Paul.

Wittgenstein, L. [1980], *Remarks on the Philosophy of Psychology*, vol. 1, Blackwell.