CRITIQUE OF THE CHURCH-TURING THEOREM

TIMM LAMPERT

Humboldt University Berlin, Unter den Linden 6, D-10099 Berlin
e-mail address: lampertt@staff.hu-berlin.de

Abstract. This paper first criticizes Church's and Turing's proofs of the undecidability of FOL. It identifies assumptions of Church's and Turing's proofs to be rejected, justifies their rejection and argues for a new discussion of the decision problem.

Contents

1. Introduction

First-order logic without names, functions or identity (FOL) is the simplest first-order language to which the Church-Turing theorem applies. I have implemented a decision procedure, the FOL-Decider, that is designed to refute this theorem. The implemented decision procedure requires the application of nothing but well-known rules of FOL. No assumption is made that extends beyond equivalence transformation within FOL. In this respect, the decision procedure is independent of any controversy regarding foundational matters.

However, this paper explains why it is reasonable to work out such a decision procedure despite the fact that the Church-Turing theorem is widely acknowledged. It provides a critique of modern versions of Church's and Turing's proofs. I identify the assumption of each proof that I question, and I explain why I question it. I also identify many assumptions that I do not question to clarify where exactly my reasoning deviates and to show that my critique is not born from philosophical scepticism. My critique, for example, does not

apply to Gödel's incompleteness proof of Peano Arithmetics (PA) nor does it question the axiomatic method in general. Instead, it is directed against the possibility of expressing a decision function for FOL in a language based on FOL. I argue that any indirect undecidability proof for FOL is underdetermined, because rather than reducing the hypothetical decidability assumption to absurdity one can instead reduce to absurdity the assumption that the translation of such a decision function into the language of FOL indeed expresses the decision function.

2. Critique of Church's Proof

2.1. **Church's Proof.** In the following, I refer to modern versions of Church's proof.[1] Church's undecidability proof of FOL rests on the proof that the property of provability in Robinson's arithmetic (Q) is undecidable. Q is based on seven axioms concerning properties of natural numbers, addition and multiplication; [Smith (2013)], p. 67, lists these axioms. The inference rules of Q are the rules of FOL$_=$ (FOL including identity, names and function symbols). The language of Q, L_A, is based on the language of FOL= and includes the arithmetic expressions 0, S (successor), + (addition), and $\times$ (multiplication) in addition to the logical constants of FOL=. In contrast to FOL=, the language of Q does not contain any vocabulary that has no fixed interpretation (such as names or predicates). The only predicate is equality (=), and objects are all denoted by 0 and the successor function S. In the following, I take it for granted that L_A is interpreted using its standard, fixed arithmetic interpretation $\Im_A$ and that Q is sound in relation to this interpretation.

Church argues that if FOL is decidable, then Q is decidable. The proof of this implication is based on (i) the translation of L_A-formulas into FOL formulas while preserving satisfiability and (ii) the deduction theorem applied to the seven axioms of Q.[2] I do not doubt this proof, and thus, I do not doubt that decidability of FOL implies decidability of Q. The translation rules that eliminate function symbols as well as identity are defined by [Boolos et al. (2003)], chapter 19.4. These authors prove the correctness of the translation procedure by proving their propositions 19.12 and 19.13 (cf. [Boolos et al. (2003)], pp. 256 and 257). I have implemented this procedure as an additional tool in the FOL-Decider. Since I claim the decidability of FOL, I therefore also claim that it is decidable whether a given L_A-formula is provable in Q.

The claim that provability in Q (Q-provability for short) is decidable, of course, does not imply that Q is negation-complete. Indeed, it is well known that Q is Π_1-incomplete; $\forall x(0 + x = 0)$ is a trivial example that is true according to $\Im_A$ but not provable in Q (cf. [Smith (2013)], Theorem 10.8, p. 69.). A decision procedure for Q makes it possible to prove the incompleteness of Q without any semantic or meta-mathematical reasoning. All

[1] For convenience, I will restrict the presented references to [Smith (2013)]. Other versions of Church's proof do not significantly differ with respect to my critique. [Smith (2013)] proves Church's theorem on p. 303. One favourable aspect of his book is that he nicely identifies and proves the underlying assumptions of Church's theorem.

[2] The deduction theorem applied to the seven axioms states the following equivalence:

$$\{\text{Axiom } 1, \ldots, \text{Axiom } 7\} \vdash A \text{ iff } \vdash \text{Axiom } 1 \wedge \ldots \wedge \text{Axiom } 7 \rightarrow A$$

.

that must be done is to prove for some formula A, e.g., the Π_1-formula $\forall x(0 + x = 0)$, that neither A nor $\neg A$ is provable from the axioms of Q by applying the decision procedure.[3]

Before considering the proof of the undecidability theorem for Q, let me first introduce some useful terminology.

According to *Church's thesis*, every decision procedure can be defined in terms of a total μ-recursive function. In the following, I will speak of "recursive functions" for brevity. As I will argue in section 2.3, the use of the term "function" is not without presuppositions. Since recursive functions are computable, one can think of the values of recursive function as outputs of and the arguments as inputs to a computation. The peculiarity of recursive functions is the way in which they can be defined, namely, by a chain of definitions by means of composition, primitive recursion and minimization from a set of initial functions (the zero, successor and identity functions) (cf. [Smith (2013)], chapter 38, p. 287). The details of defining recursive functions, however, are not relevant to my critique.

In the following, I use Smith's terminology for the important distinction between *expressing* and *capturing* a property, relation or function (cf. chapter 5, p. 41 to 44). I apply his general definitions to L_A, $\Im_A$ and Q:

Definition 2.1. A property P is *expressed* by the open well-formed formula (wff) $\varphi(x)$ with one free variable in L_A iff, for every n,

(i) if n has the property P, then $\varphi(\overline{n})$ is true according to $\Im_A$, and
(ii) if n does not have the property P, then $\neg\varphi(\overline{n})$ is true according to $\Im_A$.

Definition 2.2. Q *captures* the property P by means of the open wff $\varphi(x)$ iff, for any n,

(i) if n has the property P, then $Q \vdash \varphi(\overline{n})$, and
(ii) if n does not have the property P, then $Q \vdash \neg\varphi(\overline{n})$.

In place of "open formula", I will also use the term "propositional function" to emphasize that open formulas are correlated with an intended interpretation. $\overline{n}$ is the representation of n in L_A. The definitions are applied to the properties of *expressions* (instead of properties of numbers) by means of referring to expressions (terms, formulas or sequence of formulas) by their Gödel numbers. The definitions of expressing *relations* and *functions* are similar to that of expressing properties. The same applies to capturing relations and weakly capturing functions. Weakly capturing functions does not require to capture that the value of a given argument is unique (= capturing a function as a function or, in short: capturing a function). To fulfil this requirement, the following stronger definition is needed (cf. [Smith (2013)], p. 119):

Definition 2.3. Q *captures* the one-place numerical function f by means of the open wff $\varphi(x, y)$ iff, for any m, n,

(i) if $f(m) = n$, then $Q \vdash \varphi(\overline{m}, \overline{n})$, and
(ii) $Q \vdash \exists!y\varphi(\overline{m}, y)$.

[3]The translation of Goldbach's conjecture, also a Π_1-formula GC (cf. [Smith (2013)], p. 78), may serve as a less trivial example assuming that the conjecture is true according to $\Im_A$ and not provable in Q. The decision of the Q-unprovability of the negation of Goldbach's conjecture, which is a Σ_1-formula, would prove its truth due to the Σ_1-completeness of Q. Unfortunately, the FOL-translation of the formula $Q \rightarrow \neg GC$, NUN090+1 of the Thousands of Problems for Theorem Provers (TPTP) library, is too complex to apply the FOL-Decider (cf. figure 1, p. 10 for a formula of similar complexity and section **??** for the status of the FOL-Decider). We can hope that future improvements to the FOL-Decider will make it possible to apply the FOL-Decider to formulas of such complexity.

Note, that (i) and (ii) imply if $f(m) \neq n$, then $Q \vdash \neg\varphi(\overline{m}, \overline{n})$. (ii) satisfies the uniqueness condition. The definition can be trivially extended to n-place functions.

For brevity, I often refer merely to expressing and capturing properties without mentioning relations or functions. I will also tacitly identify the talk of deciding, expressing and capturing properties (relations) with deciding, expressing and capturing their *characteristic functions* (cf. [Smith (2013)], chapter 2.2, p. 9f., p. 18, Theorem 5.1, p. 43, p. 107).

The *Diagonalization Lemma* states that for *any* open formula $\phi(x)$, some formula γ exists such that the following formula can be proven in Q: $\gamma \leftrightarrow \phi(\lceil\gamma\rceil)$, where $\lceil\gamma\rceil$ refers to the Gödel number of γ. I call the diagonalization of a propositional function $\neg\phi(x)$ the "*diagonal case*". In particular, I call $\neg P(\lceil\gamma\rceil)$ the diagonal case of $P(x)$, where $P(x)$ abbreviates the propositional function that would express and capture a presumed decision procedure for FOL- and consequently, Q-provability.

The proof that Q is undecidable is based on the following premises:

(1) Q is consistent;
(2) any decision procedure can be defined in terms of a total recursive function (Church thesis);
(3) if Q-provability is decidable by a recursively defined characteristic function, then this function can be *expressed* in the language L_A and, moreover, *captured* in Q;
(4) the Diagonalization Lemma (cf. [Smith (2013)], Theorem 24.4, p. 181); and following from this Lemma
(5) the theorem that Q-provability cannot be captured in Q (cf. [Smith (2013)], Theorem 24.8, p. 183).

Since the Diagonalization Lemma applies to any propositional function, it also applies to $P(x)$ (cf. [Smith (2013)], Theorem 24.3, p. 180). The assumption that Q-provability can be captured in Q results in deriving a contradiction in Q in the diagonal case $\neg P(\lceil\gamma\rceil)$ (cf. the proof of Theorem 24.8 in [Smith (2013)], p. 183). Since Q is assumed to be consistent one must conclude that Q-provability cannot be captured in Q. I doubt neither this theorem nor the Diagonalization Lemma. Thus, I neither doubt (4) nor (5). However, I will question that Q-provability can be captured in Q by questioning (3).[4]

(1) is not only presumed in the reasoning for (5) but also for (3): If Q were not consistent, the notion of capturing would be reduced to absurdity. Moreover, Church's proof intends to prove that a contradiction is derivable in Q if FOL-provability (and thus, Q-provability) were decidable and could consequently be captured in Q. Therefore, one can only reduce the assumption that FOL-/Q-provability is decidable to absurdity if Q is assumed to be consistent. (2), the Church thesis, makes it possible to refer in (3) to a procedure for translating decision functions defined in terms of recursive functions into propositional L_A-functions. Taken together, (1) to (3) imply that decidability of Q implies the possibility of capturing the presumed decision function within Q. This implication contradicts (5), which presumes (4). Therefore, the hypothetical assumption that Q is decidable can be reduced to absurdity on the basis of (1) to (5).

[4]The Diagonalization Lemma is based on the "*capturing theorem*", which states that any primitive recursive function can be captured in Q (Theorem 17.1 in [Smith (2013)], p. 125). I doubt this theorem in the case of Q- or FOL-provability. However, in the proof of the Diagonalization Lemma, Theorem 17.1 is applied only to diagonalization as a primitive recursive function. I do not doubt that this primitive recursive function can be captured in Q.

Since my critique is related to (3), let me spell out the premises of (3) in more detail to precisely identify the assumption (or "theorem"[5]) to which my critique relates. Then, in section 2.4, I will explain the reasons for my critique.

(3) is based on the general theorem that *every* recursive function can be captured in Q (cf. [Smith (2013)], Theorem 17.1, p. 125, for primitive recursive functions and [Smith (2013)], Theorem 39.2, p. 298, for recursive functions in general). I call this theorem the "*capturing theorem*". The proof of this theorem rests on the following premises:[6]

- (α) the theorem that *every* recursive function can be *expressed* in L_A (cf. [Smith (2013)], Theorem 15.1, p. 113 and [Smith (2007)], p. 277f. or [Smith (2013)], p. 297 for minimization),
- (β) the theorem that the specified translation procedure for translating recursive functions results in propositional Σ_1-functions in L_A (cf. [Smith (2013)], Theorem 15.2, p. 118, p. 297 and [Smith (2007)], p. 277f.),
- (γ) the theorem that Q is Σ_1-complete (cf. [Smith (2013)], Theorem 11.5, p. 79), and
- (δ) the theorem that Q is strong enough to ensure that if a function f is captured in a weak sense by a propositional function φ, then there is a function $\tilde{\varphi}$ that captures f as a function (cf. [Smith (2013)], Theorem 16.2, p. 120).

This proof applies to properties P by representing properties by their characteristic functions c_P (cf. [Smith (2013)], Theorem 15.3, p. 118 and Theorem 16.1, p. 119). So, in fact, in the case P does not hold, the Π_1-formula $\neg\varphi(\overline{n})$ is equivalent to the formula expressing $\neg c_P(\overline{n}) = 0$, which is equivalent with and derivable from the Σ_1-formula expressing $c_P(\overline{n}) = 1$. Expressing and capturing a property, thus, is reducible to Σ_1-formulas not only in the positive case, in which the property holds but also in the negative case, when it does not hold. In general, expressing and capturing total recursive functions can be reduced to Σ_1-formulas.

δ is needed to ensure that the requirement to capture a function as a function is fulfilled, which is needed to infer that a property is captured by Q if its corresponding function is captured by Q (cf. the proof of Theorem 16.1 in [Smith (2013)], p. 119f.).

I call (α) the "*expressing theorem*". If it were not assumed, then the Σ_1-formulas resulting from the translation procedure for a property P in question were not correlated with P. I question neither (β) to (δ) nor that Theorem 17.1 can be inferred from (α) to (δ). I do not question (β) because I take this theorem to merely state that the specification of the general translation procedure for recursive functions results in L_A-propositional functions of the rather simple Σ_1-type (i.e. propositional functions preceded by a sequence of unbounded existential quantifiers followed by an expression with no further unbounded quantifiers). This procedure is presumed for (α). However, (α) additionally states that the resulting propositional Σ_1-function indeed expresses the recursive function in question. This is what I question in the case of a recursive characteristic function for FOL-provability and, consequently, for Q-provability. The rejection of premise (3) above follows from this rejection of the possibility to express FOL-provability in L_A.

[5]Henceforth, I will no longer use quotation marks to qualify acknowledged theorems that I am questioning.

[6]I follow the proof strategy of the first edition [Smith (2007)] (as well as [Boolos et al. (2003)], chapter 16.2), which bases the proof of the *capturing theorem* on the Σ_1-completeness. This makes clear that my critique applies already to the weaker *expressing theorem* (given Q's soundness in respect to $\Im_A$, which I do not question). It should be noted that the proof of the *capturing theorem* in [Smith (2013)] also makes use of the Σ_1-completeness of Q.

Let me further specify what is in question. The proof of the *expressing theorem* is based on the specification of a translation procedure that translates recursive functions into propositional Σ_1-functions in L_A (cf. (β) above and section 2.4 below). I do not doubt that this procedure is well defined and results in propositional Σ_1-functions in any case. Such a translation procedure is itself recursively definable (and, thus, computable). However, the question is whether the translation procedure is *correct* in the case of a recursively defined decision procedure for FOL. More specifically, the question can be formulated as follows: given that the propositional Σ_1-function $P(x)$ translates the decision procedure for FOL-/ Q-provability, is it indeed the case that for *all* n, if n is the Gödel number of a provable FOL formula, then $P(\overline{n})$ is true according to the standard *arithmetic* interpretation $\Im_A$ of L_A, and if n is the Gödel number of an unprovable FOL formula, then $\neg P(\overline{n})$ is true according to $\Im_A$? The positive answer to this question claims a universal proposition that even applies to the hypothetical diagonal case $\neg P(\ulcorner \gamma \urcorner)$ that expresses its own unprovability according to its meta-mathematical interpretation.

In contrast to the syntactic notion of capturing, the notion of expressing is a semantic notion. Thus, the question arises of how one can prove the correctness of a procedure for translating a recursively defined decision procedure for FOL into a propositional function $P(x)$ in L_A. I will argue that the proof of the *expressing theorem* does not provide a satisfiable answer in the case of FOL-provability.

For now, it is sufficient to state what is at issue; the details of the critique will be presented later in section 2.4. The decision problem is the problem whether it is possible to define an algorithm that applies the rules of a correct and complete FOL calculus such that the proof search can determine not only provability but also unprovability for FOL formulas. A positive answer does not deal with expressing such an algorithm within FOL. Thus, it is another and *prima facie* completely different kind of question whether one can express and, moreover, capture such an algorithm by means of a propositional L_A-function $P(x)$ such that for any Gödel number n of a provable FOL formula, the L_A-formula $P(\overline{n})$ is true according to $\Im_A$ and provable in Q, whereas for any Gödel number n of an unprovable FOL formula, the L_A-formula $\neg P(\overline{n})$ is true according to $\Im_A$ and provable in Q. An answer to this question applies meta-mathematical proof methods not needed in the case of positive answer to the decision problem. Church's proof presumes that the answers to these two questions must be coextensive. I will argue that this is not proven, in contrast to what is purported.

Let me list the specific assumptions of Church's proof that one can most reasonably doubt:

Assumption 1: the consistency of Q,
Assumption 2: Church's thesis, and
Assumption 3: the assumption that decidability of FOL implies the existence of a propositional function $P(x)$ that *expresses* FOL-provability in the language L_A (even in the diagonal case).

In the following sections 2.2 and 2.3, I explain why I do not question Assumption 1 or 2. Section 2.4 then explains why I do question Assumption 3.

2.2. **Consistency of Q.** Certain types of finitism question Assumption 1 insofar as they neither accept semantic consistency proofs of Q nor relative consistency proofs, such as Gentzen's or Gödel's consistency proof of the stronger theory PA. To date, no absolute or

finitistic consistency proof of Q has been presented. Instead, since (i) Q presumes quantification over an infinite domain (which may be understood as implying an illegitimate reference to the extensional infinite or to infinity at all) and (ii) the Diagonalization Lemma holds in Q (which may be understood as causing antinomies), a certain finitistic attitude may question the consistency of Q, which gives rise to attendant reservations against Church's proof.[7]

I do not question Assumption 1. First of all, I take it for granted that Q is sound in relation to $\Im_A$. Since consistency follows from soundness, this is an even stronger assumption than consistency. Of course, one can object that the soundness, like the consistency, of Q is not rigorously proven in accordance with the strict standards of a certain finitist point of view that eschews any semantic notions extending beyond computing symbols. Furthermore, there are also good reasons to argue for alternative semantics of arithmetic propositions. This, in turn, calls for alternative calculi, e.g. intuitionistic logic or even pure mathematical calculi that do not make use of logical constants. However, my critique is independent of any specific conceptualization of arithmetic. It takes L_A and Q as well as FOL for granted and questions that a decision procedure for FOL can be expressed in a language based on FOL. It is rather the conceptualization of a decision procedure for FOL in FOL that is questioned in the first place than that of arithmetic.

Instead of questioning Q's consistency, I conjecture that it is possible to prove the consistency of Q by applying the implemented decision procedure for FOL. Deciding the conjunction of the FOL translations of Q's axioms, NUM009+0.ax in the TPTP library, to be non-refutable (satisfiable) proves their consistency. Such a proof results from a purely syntactic decision procedure that relates to nothing but FOL (plus the standard elimination of functions and identity) and thus is an absolute consistency proof that may serve as a finitistic consistency proof. At present, merely the complexity of the formula in question, the lack of exploitation of optimization strategies and the limits of the hardware make it impossible to provide such a proof by running the FOL-Decider program. However, one may hope that future improvements of the FOL-Decider will make such proofs possible. For now, my claim may serve as a conjecture for future verification. Currently, powerful existing semi-deciders such as Vampire can already verify that significant minimal changes to the axioms of Q can be proven to result in either inconsistent axioms ("strengthening") or axioms with finite models ("weakening"). In particular, such verification is possible for the case of varying the FOL translation of axiom 3 that replaces the inductive axiom schema of PA. Such behaviour is typical of satisfiable axioms that have complicated (and, in particular, infinite) models. Nevertheless, for the following, it is sufficient to distinguish my critique of undecidability proofs from doubts concerning Assumption 1.

2.3. **Church's Thesis.** Church's thesis (i.e., Assumption 2) is, as is often emphasized, admittedly a thesis and not a theorem. If it is taken to be a formal explication of the informal concept of mechanical computation, it cannot be proven rigorously. Originally, Church and

[7][Rodych (1999)], for example, explains Wittgenstein's rejection of undecidability proofs in relation to (i); my paper [Lampert (2018a)] offers an alternative explanation. [Gumaňski (1988)] blames proofs in which the diagonal method is applied. Therefore, he also rejects Church's theorem. To my knowledge, Gumaňski was the first to also present a detailed argument for the decidability of FOL by defining upper limits for proofs in Beth's tableaux calculus (cf. [Gumaňski (2000)] and [Gumaňski (2008)]). My reasoning for questioning the Church-Turing theorem and my decision procedure are inspired by Wittgenstein and are independent of Gumaňski.

Turing thought of computation carried out by humans with paper and pencil according to some given routine. However, the relevant question today is whether a decision procedure for FOL can be written in some prevalent computer language. The significance of the Church-Turing theorem is that nobody seeks for such a program because it is believed that the decision problem has no positive solution. The FOL-Decider, written in *Mathematica*, intends to refute the Church-Turing theorem; the relevant question is whether this program somehow goes beyond what could be defined by recursive functions or Turing machines. So, I take the Church-Turing thesis to state that any program that decides a computable problem can be defined in terms of recursive functions (Church's thesis) or, alternatively, in terms of Turing machines (Turing's thesis). I do not question that. Given that the Church-Turing thesis merely states the possibility that typical program code (such as the *Mathematica* code of the FOL-Decider) can be reduced to a certain kind of simplest program code, this statement can be verified in the case of the FOL-Decider. I do not presume any idiosyncratic understanding of decidability, and I have no doubt that my decision procedure can be redefined in the language of either recursive functions or Turing machines. This reduction is itself computable.

However, Church's or Turing's thesis is sometimes said to state that any computable function can be spelled out in terms of a specific kind of *number-theoretic function*. This conception implies a general, extensional understanding of "functions", which are most often defined in set-theoretic terms (and not as computations of output from initial input, which are merely understood as functions of a specific kind), and it further interprets program code in terms of functions that generate mappings from natural numbers to natural numbers. Such syntax and semantics are presumed in the course of proving the *expressing theorem* since this proof relates recursive functions to propositional functions that express the former in accordance with the standard set-theoretic arithmetic interpretation of L_A (cf. the following section 2.4). The interpretation of recursive functions as extensional number-theoretic functions, however, is neither trivial nor insignificant since all that, in fact, occurs in computing is the manipulation of certain symbols (e.g., sequences of digits such as 0, $S0$, etc., or of strokes on a tape) in accordance with rules of a specific form (i.e., rules that can be defined by recursion or by the instructions of Turing machines). When this process is understood in terms of a specific type of number-theoretic function, doing so requires interpretation involving theoretical conceptualization.

The all-important point underlying my critique is that such a conceptualization begins to reformulate the language of computation (i.e., program code) as a language based on a general, extensional understanding of functions that paves the way towards the application of logical syntax by means of its set-theoretic or, more generally, extensional semantics. However, I do not consider the interpretation of Turing-computable "functions" or recursive "functions" as a specific type of number-theoretic functions to be an essential part of the Church-Turing thesis. Instead, I interpret this thesis merely as a thesis concerning the possibility of reducing typical program code to simplified program code of a certain kind (namely, in terms of recursive definitions including minimization or in terms of Turing machines). According to this understanding, the Church-Turing thesis is a thesis concerning the reduction of computer languages to most simplified computer languages. The syntax of a language of computation is simply the syntax of program code or, more specifically, the syntax of the corresponding recursive definitions or Turing machines, and the semantics of such a language is given by the paraphrases of these definitions in terms of instructions for generating and manipulating symbols. The relation to the syntax and semantics of a

language based on FOL is a separate question that should not be confused with the Church-Turing thesis. It is this relation that is at issue in Assumption 3 – the assumption that I intend to refute with my work.

2.4. **The Problem of Translation.** Assumption 3 is not considered on its own in Church's proof. Instead, it is inferred from the *expressing theorem*. This theorem states that *any* recursive function is expressible by a propositional function in L_A. The question is whether the proof of the *expressing theorem*, in fact, applies to a presumed recursive function for FOL-provability. I argue that this is not the case as follows:

Step 1: The proof of the *expressing theorem* presumes a number-theoretic extensional interpretation of recursive functions and relates this interpretation to the arithmetic interpretation $\Im_A$ of propositional L_A-functions.

Step 2: A recursive function for FOL-provability first and foremost presumes a logical (or meta-mathematical) interpretation of a recursively defined decision procedure for FOL: it refers to logical formulas and their properties. Thus, it must be assumed that the arithmetic and meta-mathematical interpretations, $\Im_A$ and $\Im_M$, are correlated in the interpretation of recursive functions as well as their translation into propositional functions.

Step 3: That this correlation holds in the case of diagonalizing the translation $P(x)$ of a presumed decision function for FOL-provability is questionable and does not follow from the proof of the *expressing theorem*.

My critique does not concern the expressibility of known recursive functions computing arithmetic, logical or meta-mathematical properties. Instead, it questions only the expressibility of the presumed decision procedure for FOL-provability (and, consequently, properties that are reducible to FOL-provability, such as Q-provability). Thus, it does not concern step 1 and it does not question in general the assumption of step 2 that $\Im_A$ and $\Im_M$ are correlated. It merely questions the validity of this assumption in the special case of expressing a decision function for FOL-provability in the diagonal case.

2.4.1. *Step 1 – Comparing Arithmetic Interpretations.* The proof of the *expressing theorem* for recursive functions proves the theorem for the initial functions, for the composition of recursive functions, for primitive recursion and for minimization. The main part of the proof consists of specifying the procedure for translating recursive functions into propositional Σ_1-functions. The key is to define such a translation for recursion by encoding recursion in terms of the β-function. Such a translation procedure is not trivial. As an example, one may consider the translation of the primitive recursive factorial function into a two-place propositional Σ_1-function that expresses $x! = y$ in L_A and captures it in Q (cf. Table 1)[8].

One can ensure that the factorial function is also captured by Q by (i) instantiating values for x and y in the L_A formalization and (ii) deriving the corresponding formulas in the case that these values satisfy the factorial function and deriving their negation in the case that these values do not satisfy the factorial function.

The FOL formula shown in Figure 1 is equivalent to substituting $S[0]$ for x and y, translating the result into a pure FOL formula ϕ without identity, translating the seven axioms of Q into a pure FOL formula ψ without identity and generating the formula $\neg(\psi \rightarrow \phi)$. This

[8]This translation was generated by implementing the definitions given in [Smith (2013)], p. 72, p. 116 and p. 117.

Recursive Def.	L_A **Formalization**
$0! = 1$ $(Sx)! = x! \times Sx$	$\exists_c\, \exists_d\, (\exists_u\, (c = u \times S(d \times S(0)) + S(0) \wedge$ $\exists_m\, m + u = c \wedge \exists_m\, m + S(0) = d \times S(0)) \wedge$ $\exists_u\, (c = u \times S(d \times S(x)) + y \wedge \exists_m\, m + u = c \wedge$ $\exists_m\, m + y = d \times S(x)) \wedge$ $\forall_z\, (\exists_m\, m + z = x \Rightarrow (z \neq x \Rightarrow \exists_v\, \exists_w\, (w = v \times S(z) \wedge$ $\exists_u\, (c = u \times S(d \times S(z)) + v \wedge \exists_m\, m + v = d \times S(z) \wedge$ $\exists_m\, m + u = c) \wedge \exists_u\, (c = u \times S(d \times S(S(z))) + w \wedge$ $\exists_m\, m + w = d \times S(S(z)) \wedge \exists_m\, m + u = c)))))$

Table 1: Translation of the Factorial Function into a Propositional L_A Function

formula should be refutable if the whole translation procedure is correct. Unfortunately, no modern logic engine from the TPTP site (including the FOL-Decider) is able to refute this formula due to its complexity.[9] This example illustrates the difficulty of controlling the relevant translation procedure, even with the best available logic engines.

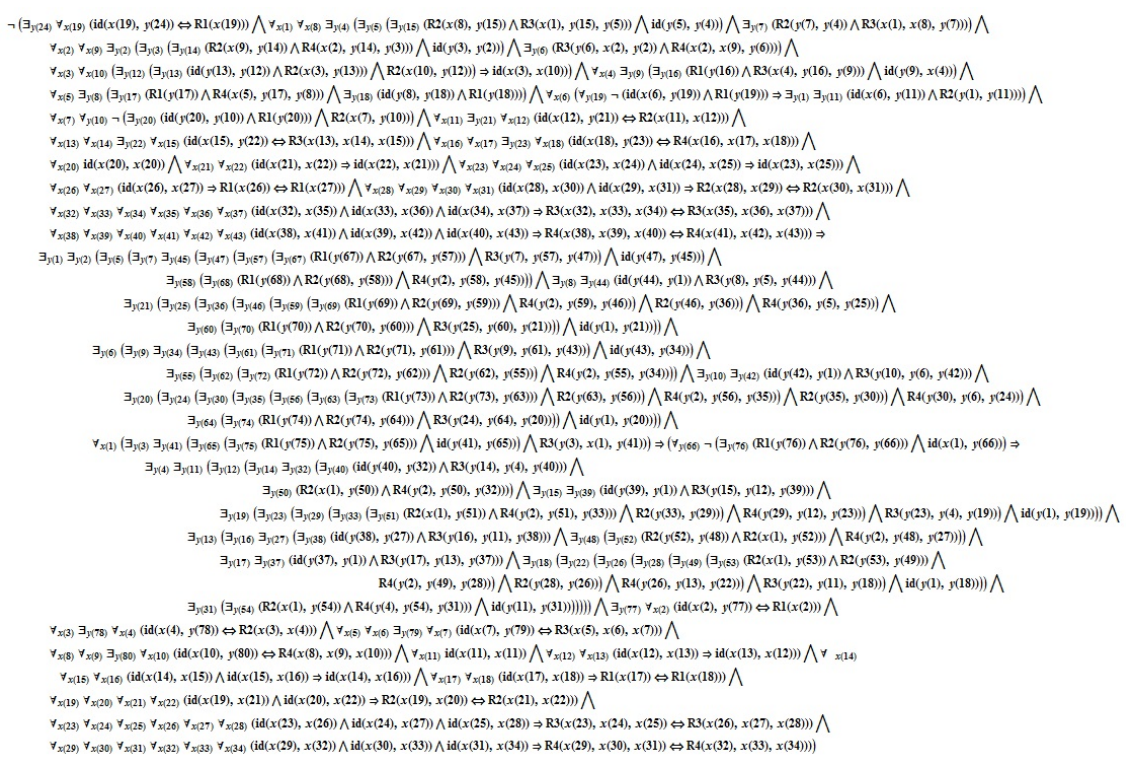

Figure 1: $\neg$(FOL-translation(Q) $\Rightarrow$ FOL-translation(L_A-translation($S0! = S0$)))

However, my point of critique does not concern the complexity of the translation procedure. I have no doubt that recursively defined arithmetic functions (such as the factorial function) can be expressed and captured by the translation procedure (including the FOL

[9]Thanks to Geoff Suttcliffe for testing this with all engines from the TPTP-site with extended time limits.

reduction). Rather, I question how it can be justified that such a procedure is *correct in any arbitrary case*, including the hypothetical case of a recursive definition of FOL-provability.

Proofs of the *expressing theorem* do not explicitly address this question. Let me state in advance that I do not advocate any sceptical or philosophically motivated critique of any answer to this question in the following. For the sake of the argument, I take the underlying semantics on which the translation procedure is based for granted. However, I argue that the semantic reasoning cannot be generalized to the diagonalization of a propositional function $P(x)$ that is intended to express FOL-provability (i.e., the diagonal case of $P(x)$).

The question of the criterion for the correctness of the translation procedure already arises in the case of the simplest translations of the initial functions. Smith's proof of the *expressing theorem* starts with proving the theorem for these functions. I quote the complete proof for the successor and zero functions from [Smith (2013)], p. 110:

> *Proof for (1)* There are three easy cases to consider:
>
> i. The successor function $Sx = y$ is expressed by the open wff $\mathtt{Sx} = \mathtt{y}$.
>
> ii. The zero function $Z(x) = 0$ is expressed by the wff $\mathtt{Z(x, y)} =_{\mathtt{def}} (\mathtt{x} = \mathtt{x} \wedge \mathtt{y} = \mathtt{0})$.

In contrast to the tricky case of expressing recursion, the translation procedure is trivial in the case of the initial functions. In this case, its correctness seems to be plain and unquestionable. However, it is interesting to note that the proof, in fact, consists of prescribing *how* the functions are to be translated without explicitly stating *why* the translation is *correct*. The answer to this question must relate the recursively defined functions to the *arithmetic* interpretation $\Im_A$ of the propositional functions in L_A. $\Im_A$ assigns truth values to propositional functions for given instances of their arguments. The translation of a recursively defined function $f(x) = y$ into a propositional function $\phi(x, y)$ is correct iff for all m and n, $\phi(\overline{m}, \overline{n})$ is true according to $\Im_A$ iff $f(m) = n$ (cf. [Smith (2013)], p. 43). How can one know that this is the case?

To answer this question, one must answer the following questions: (i) How can one gain knowledge of the truth values of $\phi(x, y)$ for specific values of m and n? (ii) How can one know that the truth values of $\Im_A(\phi(m, n))$ correspond to the computation of $f(m) = n$? (iii) How can one know the answers for *all* of the infinite number of possible values?

With regard to (i), one cannot refer to provability within Q to justify the truth value of $\Im_A(\phi(m, n))$ since Q is justified by its soundness and the *capturing theorem* is based on the *expressing theorem*: the syntax is measured by the presumed semantics, not the other way round. So, one must simply accept truth values of L_A-formulas according to $\Im_A$ as primitive (as emphasized by [Smith (2013)], section 5.3). Let us accept this in the case of $\Im_A$ for the sake of argument, although it is unsatisfying, at least from a philosophical point of view. Furthermore, it should be noted that we do not know truth values of the arithmetical interpretation $\Im_A$ of L_A-formulas in non-trivial cases. Thus, we cannot justify or control the translation procedure in such cases by independently comparing truth values.

The same applies regarding the second question. We might think of running a computation and comparing the result of what the computation *does* with the truth value of what the corresponding L_A proposition *says* according $\Im_A$. However, let us ignore any specific problems with this method since it cannot work because of (iii): it is obvious that we cannot justify any general answer by controlling specific values.

What we, in fact, do in justifying the correctness of the translation procedure is to relate *paraphrases* of the relation between x and y in the recursively defined function to *paraphrases* of the L_A expression $\phi(x, y)$. In doing so, strikingly, one also relates paraphrases of the recursively defined functions in terms of *number-theoretic (or arithmetic)*

extensional functions to the extensional understanding of the arithmetic interpretation $\Im_A$ of the corresponding propositional L_A-functions.[10] In the "Proof for (1)" quoted above, for example, Smith interprets $Sx = y$ as the successor function and $Z(x) = 0$ as the zero function: this is a number-theoretic interpretation since the values of the variables are numbers, not symbols, and it is extensional since it does not matter how the values of the functions are assigned to their arguments. Such an interpretation is, indeed, best suited to be compared with the $\Im_A$ of the corresponding propositional functions. In the case of $Sx = y$ and $\mathtt{Sx} = \mathtt{y}$, one must judge that "y is assigned to the successor to x iff y is equal to the successor of x" is true. In the case of the zero function, one must compare "0 is assigned to x" and "x is identical to itself and y equals 0". The translation is correct iff the two paraphrases are coextensive. Note that the paraphrases depend on the syntax of the paraphrased expressions and differ the more complex the expression to paraphrase is. Image, e.g., paraphrases of the expression in Table 1. The similarity of the paraphrases in the case of the successor function is an exception in this respect.

That paraphrases in the case of the initial functions are coextensive is, of course, rather trivial and beyond doubt as soon as one takes the underlying semantics for granted. However, the implications of the method of evaluating the correctness of the translation are important: the recursively defined functions to be translated are interpreted in light of the arithmetic interpretation of L_A propositions, and the "criteria" for correctness are judgements concerning the coextensionality of arithmetic paraphrases. This method is not only the implicit method used in the proof of the initial functions; the same method is applied in the case of composition, recursion and minimization by Smith and in alternative proofs of the *expressing theorem*. These proofs basically consist of defining *how* the primitive recursive functions are translated, relying on the intuitive evidence provided by the arithmetic interpretations of the related expressions. Smith, for example, ends his translation of the factorial function into the formula given in Table 1 with the following emphatic statement: "For this *evidently* expresses the factorial function" (my emphasis). Likewise,he ultimately proves that the described procedure works in any arbitrary case by presenting the general translation schema φ and asserting that "it is then *evident* that φ will serve to express the p.r. defined function f" ([Smith (2013)], p. 118, my emphasis). And in the case of minimization, he again simply defines *how* minimization is translated into Σ_1-formulas by simply ensuring that this "*evidently* expresses" ([Smith (2007)], p. 277, my emphasis)[11] the function defined by minimization. Smith's proof is in no way less explicit than others, nor does it differ in method from other proofs of the *expressing theorem*. On the contrary, his proof is admirably clear and understandable.

Of course, a sceptic or a radical finitist may doubt such a method of justifying the correctness of a translation procedure that is based on evidence concerning the coextensionality of real or schematic paraphrases. One might object that this is not an acceptable proof method in pure, rigorous mathematics. However, I do not advance such an argument, nor do I doubt the correctness of translations of recursively defined arithmetic functions (e.g., the factorial function) into L_A expressions. It is evident that the translation is correct from (i) placing all intermediate steps of the translation procedure in accordance with the recursive definitions on one side and the compositional semantics of L_A on the other and (ii)

[10] Cf. the "remarks about extensionality" by [Smith (2007)], pp. 29f., pp. 34f. and 35ff. and [Smith (2013)], chapter 14.4.

[11] Cf. [Smith (2013)], p. 297: "Intuitively, $\mathtt{F}$ expresses f; i.e. $f(m) = n$ iff $F(\overline{m}, \overline{n})$ is true (think about it!)."

considering the comparability of arithmetic interpretations. If one is trained, one may even be able to recognize the recursive definitions of the translated functions in the complicated L_A expressions.

In the following, the question is the extent to which this method can be generalized to cases of recursively defined logical relations between formulas (not numbers).

2.4.2. *Step 2 – Arithmetic and Meta-mathematical Interpretations.* Given a computer program for deciding FOL formulas, the intentions and paraphrases of its functions will most likely be logical; one intends to manipulate logical formulas in accordance with syntactic rules of logic with the purpose of identifying logical properties. Given the Church thesis, these functions can be reduced to recursive functions, and a proper paraphrase of the resulting recursive functions will still be logical or meta-mathematical. Assigning '0' in the final step, for example, might be paraphrased as assigning the value 'provable' to the initial formula, which is given in the form of some binary code. How can one be sure that the translation into a propositional function $P(x)$ is still correct, such that this decision on a logical property of formulas is, in fact, properly correlated with the truth values of the corresponding arithmetic interpretation of $P(x)$?

The answer is that meta-mathematical properties can be expressed within L_A via Gödelization: "We yield an isomorphic image [...] in the domain of arithmetics" ([Gödel (1931)], p. 174, footnote 9). Since we decode our expressions – terms, formulas, or sequences of formulas – by means of signs of numbers, we can read functions between expressions as arithmetic functions and thus correlate arithmetic interpretation with an meta-mathematical interpretation. This statement applies to both the recursively defined functions and the propositional functions that result from the translation procedure. Let us therefore distinguish an arithmetic interpretation $\Im_A$ from a meta-mathematical interpretation $\Im_M$ in the case of recursive functions and their L_A correlates. Now, we can again justify the correctness of the translation procedure by going through it step by step and comparing the corresponding paraphrases in terms of the interpretations $\Im_M$ and $\Im_A$ of the recursive functions and L_A expressions.

Let me illustrate the method of justification again by means of a simplest possible example. Smith stipulates that even numbers are the basic codes for variables in L_A. The Gödel number for an expression is defined by taking the basic code number for each individual symbol i of the expression as the exponent for the i-the prime number and then taking the product of the results ($1 \geq i$). Given this encoding, Smith proves that the primitive recursive function "n is a variable" ($Var(n)$) can be expressed in L_A as follows (cf. [Smith (2013)], p. 147):

> *Proof of (i)* We just note that we can put
>
> $$Var(n) =_{def} (\exists x \leq n)(n = 2^{2(x+1)}).$$
>
> For the basic code for a variable always has the form $2n$, for $n \geq 1$, and the g.n. [= Gödel number, T.L.] for a single expression is 2 to the power of that basic codes.

The propositional L_A-function $(\exists x \leq n)(n = 2^{2(x+1)})$ can be interpreted according to $\Im_A$. However, Gödel numbering makes it possible to additionally read this formula as expressing a meta-mathematical property.[12] The proof presents the propositional function that expresses the recursively defined meta-mathematical property and justifies that this propositional function indeed expresses the recursive function by recalling the Gödel numbering that lies at the heart of both $\Im_M(Var(n))$ and $\Im_M(\exists x \leq n)(n = 2^{2(x+1)}))$.

I do not doubt that the ability to express meta-mathematical properties works well in the case of known recursive functions and their specified L_A translations. This is the case, for example, for "x is a PA-formula" or "y is a PA-proof of x" (abbreviated as $B(y, x)$). One can justify the translations by explaining how, in fact, the conventions of the stipulated Gödelization enforce arithmetic properties and relations that are, in turn, expressed by propositional L_A-functions. This can be illustrated by many examples, that do not give rise to doubt that the paraphrases or intended interpretations are coextensive.

I also do not question that PA-provability can be expressed but not captured by existentially binding the variable y in the propositional function $B(y, x)$ that captures "y is a PA-proof of x" (cf. Definitions 45 and 46 in [Gödel (1931)], p. 186). My reasoning against the undecidability proof of FOL does not question Gödel's proof, which proves that his Π_1-formula G, i.e., the diagonal case $\neg\exists y B(y, \lceil G \rceil)$ of $\exists y B(y, x)$, is undecidable in PA. As Gödel himself remarks, $\exists y B(y, x)$ is generated not by translating a total recursive function but by existentially binding y in the propositional function $B(y, x)$. Therefore, it cannot be inferred that provability can be captured by $\exists y B(y, x)$; it cannot be presumed that the negative case of unprovability can be reduced to a positive, existential claim.

A decision function for FOL is a completely different entity than a procedure for deciding, for a given sequence of formulas (or their Gödel numbers) and another formula (or its Gödel number), whether the former is a proof of the latter (i.e. deciding "y is a FOL-proof of x"). First of all, a decision procedure for provability must also decide unprovability. According to the translation schema underlying the *expressing theorem*, a translation of the decision that a formula is unprovable results not in a Π_1-formula but in a Σ_1-formula (regardless of whether the translation is correct). More importantly, the strategy of a reasonable decision procedure does not involve checking whether given Gödel numbers satisfy the relation "y is a proof of x". Once the decidability of provability is of concern, the question is how to *find* a proof or disproof by means of an intelligent, fully automated decision strategy. It will not be sufficient to check pairs of numbers by trial and error and hope to serendipitously find a positive pair satisfying "y is a proof of x". This is not a reasonable search method in an infinite space, nor is it a reasonable procedure to identify unprovability. Instead, the intent of a decision procedure for provability is to *generate* a proof or disproof given a formula in question as input. What is looked for is a decision criterion for provability that can be applied after a finite number of steps of manipulating symbols; a criterion that allows one to decide on single instances of an infinite number of candidates for proofs of a given formula ϕ does not provide such a criterion since it cannot identify that ϕ is unprovable.

Gödel's incompleteness proof must be clearly distinguished from Church's proof. Gödel does *not hypothetically* assume a recursive definition of "y is a PA-proof of x" but *actually*

[12]The recursive definition of the property of being a variable is not provided as part of the proof. $(\exists x \leq n)(n = 2^{2(x+1)})$ is not the result of a translation of the recursive definition. Being the result of a canonical translation procedure is not a necessary condition for expressing a recursive function; it is only a sufficient one.

provides the recursive definition and its translation to a propositional function, abbreviated by Gödel as $B(y, x)$. On that basis, he specifies a concrete propositional function $\exists y B(y, x)$ that *actually* expresses "x is provable in PA" without capturing it. That $\exists y B(y, x)$ does not capture provability is not surprising. First of all, it depends on the strength of the axiomatic system what can be captured by it. Furthermore, it is not surprising that general, universally quantified propositions are not provable within PA since logical quantification is not based on induction but on quantifying over infinite sets, which involves an extensional understanding of infinity. Therefore, even though induction is implemented in PA in contrast to Q, this does not imply that PA should be expected to be Π_1-complete (which it is not according to Gödel's incompleteness proof). Universal quantification can only be proven logically in non-trivial cases from universal axioms. That an axiomatic proof conception cannot reduce all proofs of universally quantified propositions to a finite number of axioms comes as no surprise; it simply shows the limits of a logical conceptualization of arithmetic.

In contrast, Church's proof *hypothetically* assumes a recursive definition for FOL-provability and its translation into a propositional function $P(x)$ in L_A. On this basis, the assumption is reduced to absurdity that such a recursive function (and consequently, its translation) exists. Not a concrete propositional function expressing what goes beyond the realm of computation is at stake but expressing what is hypothetically assumed to be computable within a language of logic that is not restricted to express what is computable is in question. Similar to interpreting Gödel's incompleteness proof as a proof of the limits of an axiomatic proof conception, one may argue that it comes as no surprise that a decision function for FOL is not expressible in L_A since what is expressible depends on the syntax of a language and, contrary to computer languages, propositional languages that are based on FOL are not designed to express what is computable. However, instead of arguing this way, Church's proof questions the existence of a decision procedure for FOL rather than questioning the expressiveness of FOL.

2.4.3. *Step 3 – Underdetermination of Church's indirect proof.* As soon as one is considering a decision procedure for FOL, or any other system based on the language and inference rules of FOL, the situation becomes different to all positive and known translations of recursive functions of meta-mathematical properties since now the reasoning is hypothetical and it is intended to justify a reduction to absurdity of the assumption of FOL's decidability. What is at issue is not the translation of a specific given decision function; instead, it is argued that it is impossible to define such a function. To do so, it must be presumed that the *expressing theorem* holds not only in the positive and justified cases of recursive definitions of meta-mathematical properties but also in the hypothetical case of FOL-provability.

This is true only if the arithmetic and meta-mathematical interpretations are also isomorphic and coextensive in this case. However, this cannot be judged since in this case, no translations are available to be compared. There is no intuitive evidence that paraphrases may also be coextensive in this hypothetical case as soon as one also considers that diagonalization is involved. There is no evidence whatsoever that generalizing to the translation procedure to the very special case of translation a decision procedure for FOL-provability (or any case of provability reducible to FOL-provability) and diagonalizing it will still preserver the correctness of the translation procedure. This special case is not comparable to known positive cases since by meta-mathematical reasoning, we know that in the special diagonal case, no propositional Σ_1-function $P(x)$ can capture provability. The meta-mathematical interpretation of $\neg P(\lceil\gamma\rceil)$ as "$\lceil\gamma\rceil$ is not provable" is contradictory to what is provable,

i.e., both in the case of the provability of $\neg P(\lceil\gamma\rceil)$ and in the case of the unprovability of $\neg P(\lceil\gamma\rceil)$, which implies the provability of $P(\lceil\gamma\rceil)$ due to capturing. This situation has no equivalent among the usual cases, and a reliable means of interpreting this situation in advance has not been determined.

Since capturing and expressing are equivalent in the case of Σ_1-functions, the situation can be interpreted in two ways: either $P(x)$ *does not exist* since FOL is not decidable (the standard understanding), or FOL-provability *is not expressed* by the translation of the assumed decision function into $P(x)$ (the alternative understanding). The standard understanding rests on the belief that the translation procedure correctly generalizes to the abnormal diagonal case $\neg P(\lceil\gamma\rceil)$; the alternative understanding questions this generalization due to the differences between the cases and the fact that no positive evidence can be provided that $\Im_M$ and $\Im_A$ are also isomorphic in this diagonal case. In fact, the alternative understanding denies that $\Im_M$ and $\Im_A$ are correlated in the abnormal diagonal case of provability: the truth values of "$\lceil\gamma\rceil$ is not provable" and the corresponding arithmetic interpretations of $\neg P(\lceil\gamma\rceil)$ may well be different.

Church's proof reduces the assumption that FOL (and, consequently, Q) is decidable to absurdity. This presumes the standard understanding. However, the proof by contradiction is underdetermined since the alternative understanding cannot be ruled out. According to this understanding, the assumption that $\Im_M(\neg P(\lceil\gamma\rceil))$ is coextensive to $\Im_A(\neg P(\lceil\gamma\rceil))$ is reduced to absurdity. If $\Im_M(\neg P(\lceil\gamma\rceil)) \neq \Im_A(\neg P(\lceil\gamma\rceil))$, then FOL-provability is not expressed because expressing is defined only in relation to $\Im_A$ (cf. Definition 2.1). Given that, for example, $\neg P(\lceil\gamma\rceil)$ is provable in Q, $\Im_A(\neg P(\lceil\gamma\rceil))$ is true due to Q's soundness, while $\Im_M(\neg P(\lceil\gamma\rceil))$ is false: According to the alternative understanding, this may well be the case.

There is no reason to infer that Q is not sound (or even is inconsistent) according to the alternative understanding since soundness is defined in relation to the standard arithmetic interpretation $\Im_A$. One might say that Q is not sound with respect to an intended meta-mathematical interpretation $\Im_M$ in the diagonal case $\neg P(\lceil\gamma\rceil)$. However, it is more appropriate to say that a meta-mathematical interpretation $\Im_M(\neg P(\lceil\gamma\rceil))$ is not admissible in this case, or simply that FOL-provability is not expressible (or definable) within L_A because the intended meta-mathematical interpretation fails in the diagonal case.

In the alternative understanding, undecidability proofs are interpreted as undefinability proofs, such as Tarski's proof of his theorem that L_A-truth cannot be expressed within L_A. In fact, Church's proof is much more similar to Tarski's proof than to Gödel's proof: in Church's proof as well as in Tarski's, it is hypothetically assumed that a certain property (FOL- or Q-provability / truth according to $\Im_A$) can be expressed in L_A and this is reduced to absurdity on the basis of the consistency or soundness of Q, the Diagonalization Lemma and the diagonalization of the propositional function in L_A that is assumed to express the property in question. The key difference of undefinability proofs and Gödel's incompleteness proof is not between the semantic and syntactic properties but between an indirect proof based on the hypothesis of expressing a property (Tarski, Church) and an indirect proof based merely on the presumption of capturing a property while providing a propositional function ($\exists y B(y, x)$) that expresses the property in question (PA-provability) without capturing it (Gödel). The former can be used to reduce expressing a property to absurdity, while the latter can only be used to reduce capturing a property to absurdity. The only reason why Church's proof is taken as an undecidability proof rather than as an undefinability proof is the belief in the *expressing theorem*, which applies to recursive functions.

However, according to the alternative understanding the general validity of this theorem is what is in question.

In the remainder of this section and the next section **??**, I intend to provide further evidence that this is not an hopeless, idiosyncratic alternative understanding conceived of by a wrong-headed mind but rather an obvious interpretation that cannot be ruled out in advance.

2.4.4. *Inadmissible Interpretations.* The proof of the *expressing theorem* relies on arithmetic interpretations of the recursively defined functions and the L_A translations. Thus, it is evident that it does not apply to the question of whether FOL-provability is also expressible in L_A. Instead, this claim is based on the further assumption that $\Im_A$ and $\Im_M$ are coextensive *in any arbitrary case*. This coextensionality is confirmed in many cases, but the question is whether the postulated isomorphism also holds in the abnormal diagonal case of FOL-provability. Nothing within the proof of the *expressing theorem* or in the undecidability proofs justifies this assumption.

Instead, the meta-mathematical reasoning leaves open the option of reducing to absurdity the assumption that $P(x)$, in fact, expresses provability, and consequently, it is questionable whether $\Im_A$ and $\Im_M$ are correlated in the diagonal case of $P(x)$. Moreover, there is evidence that intended interpretations of diagonal cases do not obey the extensional semantics of a language based on FOL and, therefore, are inadmissible.

It is a common phenomenon that effective procedures for translating expressions in one language into FOL formulas with intended interpretations may fail because the translated expressions do not behave in accordance with the semantics of FOL. The theory of logical formalization and philosophical logics are full of examples. One standard example from intensional logic is as follows: One may define a translation procedure for translating sentences of the grammatical form "Person P ϕs a ψ" into FOL formulas of the form "$\exists x(\phi(P,x) \wedge \psi(x))$". If one applies this procedure to "Jack loves a woman", it works fine. However, applying it to "Jack seeks a unicorn" fails since $\exists x\psi(x)$ follows from $\exists x(\phi(P,x) \wedge \psi(x))$ but "Some unicorn exists" does not follow from the truth of the proposition that someone seeks a unicorn.

The common explanation for this fact is that "seeking" is not a predicate that behaves in accordance with the principle of extensionality of FOL semantics since the object one seeks is not an object of reference. Such examples are no reason to question the soundness of FOL since soundness is defined in relation to admissible interpretations that do behave in accordance with the semantics of FOL (and not in relation to *arbitrary* instances/paraphrases/intended interpretations of FOL formulas). Instances/paraphrases or intended interpretations that do not behave in accordance with the semantics of FOL are inadmissible. They cannot be expressed (defined, represented) by propositional FOL functions. FOL would become unsound if its soundness were measured in relation to such interpretations. However, this is not what is done. Instead, the interpretations are withdrawn as admissible interpretations of a language based on FOL. Translations of expressions on the basis of inadmissible interpretations are not correct.

It is important to note the phenomenon of inadmissible interpretations where the question of expressing properties by means of propositional functions is concerned. Any proof based on expressing properties in FOL that are defined in another language must not be threatened by the possibility of inadmissible interpretations. We know, however, from many semantic and logical paradoxes that predicates such as "x is not true", "x is heterological",

"x is a set of normal sets", "x is a Richard number", etc., are not expressible by propositional FOL functions without implementing certain restrictions on or changes to the syntax and/or semantics of FOL (e.g. type theory or the distinction between meta-language and object language). The reason is always the diagonal case, which gives rise to contradictions due the intended interpretations, thus showing that they are inadmissible.

One possible way to explain this inadmissibility is to maintain that the intended interpretations involved do not behave in accordance with the extensional principles of FOL semantics in the diagonal case since this case implies an intended self-reference. Wittgenstein is a prominent advocate of such an analysis (cf., e.g., TLP 3.33-3.334; WVC, p. 121; and PR, p. 207f.). According to him, an expression must share its form with the object to which it refers. In the case of self-reference, however, there is an ambiguity of form; $\lceil\gamma\rceil$ is an argument and, thus, part of a sentence, and it refers to the very sentence of which it is part. This ambiguity may induce the impossibility to refer properly in the diagonal case. This, in turn, may cause contradictions and, hence, inadmissible interpretations. According to my understanding, this analysis motivates Wittgenstein's critique of undecidability proofs (cf. RFM, I, appendix I, §12f, §19). The meta-mathematical interpretation $\Im_M$ of a presumed propositional function $\neg P(x)$ in terms of unprovability fails in the diagonal case since $\lceil\gamma\rceil$ does not refer properly in this context, similar to the intended self-reference in the case of a paradox. One should be aware that this ambiguity concerns only $\Im_M$, not $\Im_A$. The crucial difference between these two intended interpretations is the diagonal case: there is no self-reference involved in the case of $\Im_A$. This is why $\Im_M$ may be inadmissible while $\Im_A$ *cannot* be inadmissible.

Unfortunately, Wittgenstein seemed to believe that his analysis applies not only to Church's and Turing's proofs but also to Gödel's proof (cf. [Lampert (2018a)] for details). In contrast, I neither maintain that a Wittgensteinian analysis applies in general nor that it is the only possible one. I merely claim that it is one possible explanation of why one can reasonably doubt the assumed expressibility of FOL-provability within a language based on FOL despite the fruitfulness and success of expressing meta-mathematical properties within L_A. The analysis of paradoxes is a highly controversial topic. I do not believe that the question in how far undecidability proofs are affected by inadmissible interpretations giving rise to paradoxes can be solved by adhering to a rather philosophical or foundational discussion. I do not intend to argue for any particular view on paradoxes, nor do I presume a Wittgensteinian analysis of paradoxes. The details of the analysis of paradoxes are not important for my critique of the Church-Turing theorem and need not be shared. What is important is that the admissibility of $\Im_M$ in the diagonal case $\neg P(\lceil\gamma\rceil)$ can be reasonably doubted and that Wittgenstein's analysis is one possible approach to doing so. No effective translation procedure, and no reasoning in the case of positive examples of successfully expressing meta-mathematical properties, can rule out the possibility that the translation procedure does not successfully express FOL-provability in the case of a given decision procedure for FOL. To do so, it would be necessary to prove that the intended meta-mathematical interpretation in the diagonal case is admissible and, thus, does not deviate in truth value from its arithmetical counterpart. However, this is not proven on the basis of hypothetical reasoning; the general isomorphism between $\Im_M$ and $\Im_A$, or the extensionality of $\Im_M$ in any arbitrary case (including $\neg P(\lceil\gamma\rceil)$), is stipulated or implicitly assumed, not proven. That is why the undecidability proof of FOL is not compelling.

Since the undecidability proof is based on hypothetical reasoning concerning the expressibility of FOL-provability in the diagonal case it can be reasonably doubted. This is

true even in the case one cannot conceive what exactly goes wrong. Paradoxes (antinomies or anomolies) come as a surprise evidenced by external reasoning, not by making evident some internal mistake that may be discovered by going through each step of the proof. The standard understanding of Church's proof is not incoherent. Instead, it is an admirable paradigm of thorough and ingenious reasoning. Yet, it is underdetermined. One may claim that rigorous proofs should not be underdetermined. However, strictly speaking, any indirect proof based on more than one assumption is underdetermined and leaves the option open to reduce some alternative assumption to absurdity. The alternative understanding argues for such an alternative reading of the indirect proof. However, it is likewise underdetermined. Without, in fact, proving that FOL is decidable, Assumption 3 (cf. p. 6) cannot be refuted. The specification of the FOL-Decider shall serve as external evidence to reconsider what is taken for granted.

However, given the peculiarity of the hypothetical diagonal case $\neg P(\ulcorner\gamma\urcorner)$ and its analogy to paradoxes, one may argue that the admissibility of $\Im_M$ is not only not proven in the diagonal case of $P(x)$ but also unlikely due to the following facts: (i) diagonalization implies Gödelization and, thus, is involved only in $\Im_M$, not $\Im_A$, of the corresponding formulas; (ii) the diagonal case of a presumed characteristic propositional function $P(x)$ expressing provability in FOL (or some axiomatic system based on FOL) is designed to rule out the one-to-one correspondence between the provability of the diagonalization of $\neg P(x)$ in Q and its meta-mathematical interpretation $\Im_M$; and (iii) since the arithmetic interpretation $\Im_A$ is not affected by diagonalization, it might be the case that $\Im_A$ and $\Im_M$ are no longer correlated in the diagonal case. Thus, it might well be that in the case of diagonalization, $\neg P(\ulcorner\gamma\urcorner)$ is true (false) according to $\Im_M$ but is false (true) according to $\Im_A$. However, since expressing is defined in relation to the arithmetic interpretation (cf. Definition 2.1), this means that $P(x)$ does not express provability. Consequently, $P(x)$ also does not capture Q-provability because the *capturing theorem* presumes the *expressing theorem*.

To question the correctness of a presumed translation of a decision procedure for FOL does not, of course, mean that *every* translation of a recursive function into a Σ_1-formula is questionable. Interpretations of diagonal cases do not necessarily induce contradictions. Identifying *one* inadmissible interpretation, however, also does not mean that *no other* translation is inadmissible. Similar to the principal question of distinguishing admissible from inadmissible intended interpretations, one is confronted with the problem of distinguishing correct from incorrect translations. Claiming the decidability of FOL does not mean rejecting undecidability proofs in general. It merely implies that one should not base undecidability proofs on translations of presumed recursive functions into L_A when the meta-mathematical interpretation induces contradictions. The impossibility of expressing and capturing FOL-provability in L_A or Q is not the correct standard against which to measure decidability as long as one might likewise interpret Church's undecidability proof as a reductio of Assumption 3.

3. Critique of Turing's Proof

The critique that the correctness of the procedure for translating a decision procedure into FOL formulas is questionable in the diagonal also applies to Turing's undecidability proof of FOL. Turing's proof relies on *Turing's thesis*, which states that anything that is computable (by a program or by a human being) can be written as a program consisting of instructions for a Turing machine. As in the case of Church's thesis, I take this reduction

of computability for granted. Turing does not refer to Q or to any axiomatic system that is assumed to be consistent. Instead, he directly specifies a procedure for translating Turing machines into FOL formulas with an intended interpretation that relates to the behaviour of the translated machines. His proof is based on a lemma that correlates the provability of the formalizations of a Turing machine M started with an input I with the occurrence of M entering a specific configuration or state if started with I (i.e., *Turing's Lemma*, cf. [Turing (1936)], pp. 261f.): the computational behaviour of Turing machines is described by the intended interpretations of propositional FOL formulas. In modern variants of Turing's original proof, the state in question is the halting state, and *Turing's Lemma* states that the formalization $Un(M, I)$ of M (started with I) is provable iff M, started with I, halts.

The question is whether *Turing's Lemma* also applies in the case of Turing machines involving a (hypothetical) Turing machine that decides FOL (i.e., the diagonal case). Turing bases his proof of *Lemma 2*,[13] which concerns the left-to-right direction of his *Lemma*, on a naive principle that must not be confused with the soundness of FOL. He assumes that the provability of an FOL formula implies the truth of its intended interpretation (i.e., *Turing's Principle*). However, Turing's intended interpretations are paraphrases of FOL formulas in terms of descriptions of configurations and of instructions for Turing machines. It is questionable whether the semantics of FOL that is presumed in any soundness proof of FOL applies to these paraphrases in any arbitrary case (including diagonal cases). There are many counter-examples to *Turing's Principle* involving non-extensional contexts; given that diagonalization gives rise to paradoxes, it cannot be assumed that diagonal cases are extensional and behave in accordance with the semantics of FOL in cases involving intended interpretations of FOL formulas in terms of descriptions of Turing machines.

Similar to Church's proof, Turing's proof assumes that the intended interpretations of the formulas that translate a decision procedure behave in accordance with the semantics of FOL even in the diagonal case of interpreting the FOL translations of Turing machines involving a decision machine for FOL. In analogy to the use of the phrase "expressing a property in accordance with the standard interpretation of L_A", let us use the phrase "expressing a property in accordance with an admissible interpretation of FOL". What is in question is whether it can be justified that the *intended* interpretations of FOL formulas that relate $Un(M, I)$ to the behaviour of the translated machine M started with I behave in accordance with the semantics of FOL. The problem is not to justify this for given cases but rather to generalize the admissibility of the intended interpretations to *any case*, including hypothetical diagonal cases that induce contradictions. These contradictions can be used to reduce to absurdity the assumption that the interpretations involved in the diagonal case are admissible instead of reducing the hypothetical presumed decision procedure to absurdity. On the basis of this indeterminacy, one can use *Turing's Principle* as a criterion for identifying inadmissible intended interpretations rather than referring to it as an assumption of the undecidability proof: if an intended interpretation of a provable formula

[13]Cf. [Turing (1936)], p. 262. The proof of Turing's *Lemma 2* consists of only two sentences. In the first sentence, Turing quotes his general principle: "If we substitute any propositional functions for function variables in a provable formula, we obtain a true propositions." *Function variables* is Turing's term for open formulas, and *propositional functions* his term for instances (or intended interpretations) of the open formulas. The second sentence of his proof then applies this general principle to his intended interpretations of the *function variables* in $Un(M, I)$ in order to infer the truth of the intended interpretation of $Un(M, I)$ in the case of its provability.

is not true (but rather is either false or somehow ambiguous or meaningless), then it is inadmissible.

Turing's Lemma is the analogue to *capturing theorem*, which is based on the *expressing theorem*. Similar to the questionable general validity of the *expressing theorem*, the question is whether the general claim of *Turing's Lemma* also applies to the specific diagonal case of the presumed decision procedure for FOL. The specific assumption that is the analogue to the specific Assumption 3 of Church's proof can be formulated regarding the halting problem as follows:

> **Assumption 3':** The decidability of FOL implies the existence of an FOL formalization $Un(M, I)$ for a combination of Turing machines M involving a decision machine for FOL and started with input I, such that $Un(M, I)$ expresses the halting of M in accordance with an admissible interpretation of $Un(M, I)$ (even in the diagonal case).

Questionable is neither the existence of a well-defined translation procedure resulting in $Un(M, I)$ given M and I nor one's ability to paraphrase $Un(M, I)$ in accordance with an intended interpretation such that the paraphrase states that M halts if M is started with I. The question is whether the provability of $Un(M, I)$, in fact, correlates with the truth of the intended interpretation in the diagonal case.

It is clear that the fact that the assumption of a decision machine for FOL contradicts *Turing's Lemma*. To see this, one needs to consider diagonal cases (similar to Church's proof). It is sufficient to consider a set of machines M that ends in a combination of a decision machine FOL for FOL with the dithering machine D. FOL returns exactly one stroke iff its input formula is provable. The dithering machine halts iff its input is not identical to exactly one stroke (cf. [Boolos et al. (2003)], p. 39). As an example of such a set of machines ending with $FOLD$, one might consider the machine $CTFOLD$, composed of (i) a copy machine C that copies the number of strokes with which it is started (cf. [Boolos et al. (2003)], p. 39) and (ii) a translation machine T that generates a formula $Un(M, I)$ given the description number of M and the input I of M. When started with its own description number i, $CTFOLD$ generates a formula $Un(CTFOLD, i)$, which is provable iff $CTFOLD$ does not halt when started with its own number. This contradicts *Turing's Lemma*. The question, however, is whether one should interpret this result as a reduction to absurdity of FOL or whether one should interpret it as a reduction to absurdity of the admissibility of the intended interpretation in this diagonal case and, therefore, reject Assumption 3' (and, consequently, the general validity of *Turing's Lemma*). Similar to the various versions of Church's proof, the versions of Turing's proof do not provide a forcible reason to answer this question. Hence, these proofs are underdetermined.

By referring to intended interpretations of logical formalizations of Turing machines, Turing's proof reaches beyond what must be conceded if one does not take for granted the admissibility of intended interpretations, particularly if diagonalization is involved. Instead of inferring that FOL is undecidable, one might reject the general validity of *Turing's Lemma* since it does not apply to cases such as $CTFOLD$ started with its own number. The provability of $Un(CTFOLD, i)$ will not be correlated with the truth of the paraphrase of $Un(CTFOLD, i)$ according to the intended interpretation: $Un(CTFOLD, i)$ may be provable, while $CTFOLD$ started with i does not halt. There is no evidence that this may not happen since we cannot generalize known, normal cases of applying *Turing's Lemma* to the abnormal, hypothetical diagonal cases such as $CTFOLD$ started with its own number. According to this critique, it may well be that FOL is sound and decidable and that the

halting problem (or some other unsolvable problem) is not solvable: what one must reject is the claim that the intended interpretations of the logical formalizations of Turing machines are admissible in any arbitrary case.

Modern versions of Turing's undecidability proof refer to the logical validity (truth in all interpretations), instead of the provability, of $Un(M, I)$ (cf., e.g., [Boolos et al. (2003)], chapter 11.1). Here, the proof of the analogue to *Lemma 2* (cf. [Boolos et al. (2003)], p. 130) implies that "truth in all interpretations" specifically applies to the intended interpretations involved. However, this is the case only if the intended interpretations are admissible, since "all interpretations" refers to all interpretations that obey the principles of FOL semantics. Questions concerning the admissibility of interpretations are related to the application of formal logic to meaningful propositions. The literature on the formalization and application of FOL to meaningful propositions is full of examples that show that meaningful propositions may not behave in accordance with the principles of the extensional semantics of FOL. Problems regarding the relation of the syntax and semantics of FOL to meaningful propositions of some other language should not affect rigorous proofs concerning the limits of what can be decided by purely formal means.

References

[Boolos et al. (2003)] Boolos, G.S., Burgess, J.P., Jeffrey, R.C.: *Computability and Logic*, forth edition, Cambridge: Cambridge University Press.

[Gödel (1931)] Gödel, K. 1931: "On formally undecidable propositions of *Principia Mathematica* and related systems I", *Monatshefte für Mathematik und Physik* 38, 173-198.

[Gumaňski (1988)] Gumaňski, L.: "Remarks on Cantor's Diagonal Method and Some Related Topics", in B. Dyankov (ed.): *Types of Logical Systems and The Problem of Truth (Logic and its Application)*, 45-56.

[Gumaňski (2000)] Gumaňski, L.: "The Decidability of the First-Order Functional Calculus", *Ruch Filosoficzny* 57(3/4), 411-438.

[Gumaňski (2008)] Gumaňski, L.: "A New Proof of Decidability of First-Order Functional Calculus", *Ruch Filosoficzny* 65(3), 419-437.

[Lampert (2018a)] Lampert, T. 2018: "Wittgenstein and Gödel: An Attempt to make 'Wittgenstein's Objection' reasonable", *Philosophia Mathematica 25.3*, 324-345.

[Lampert (2019b)] Lampert, T.: "Turing's Fallacy of Substitution", preprint: http://www2.cms.hu-berlin.de/newlogic/webMathematica/Logic/turingsfallacyofsubstitution.pdf.

[Rodych (1999)] Rodych, V.: 'Wittgenstein's Inversion of Gödel's Theorem', *Erkenntnis* 51, 173-206.

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

[Smith (2013)] Smith, P.: *An Introduction to Gödel's Theorems*, second edition, Cambridge University Press, Cambridge.

[Turing (1936)] Turing, A.: 'On Computable Numbers, with an Application to the Entscheidungsproblem', in *Proceedings of the London Mathematical Society* 2 (42), pp. 230-265.