Log. Univers. 17 (2023), 161–174
© 2023 The Author(s) under exclusive licence to Springer Nature Switzerland AG
1661-8297/23/020161-14, *published online* April 24, 2023
https://doi.org/10.1007/s11787-023-00325-z

Logica Universalis

The Decision Problem for Effective Procedures

Nathan Salmón

Abstract. The "somewhat vague, intuitive" notion from computability theory of an effective procedure (method) or algorithm can be fairly precisely defined even if it is not sufficiently formal and precise to belong to mathematics proper (in a narrow sense)—and even if (as many have asserted) for that reason the Church–Turing thesis is unprovable. It is proved logically that the class of effective procedures is not decidable, i.e., that there is no effective procedure for ascertaining whether a given procedure is effective. This result is proved directly from the notion itself of an effective procedure, without reliance on any (partly) mathematical lemma, conjecture, or thesis invoking recursiveness or Turing-computability. In fact, there is no reliance on anything very mathematical. The proof does not even appeal to a precise definition of 'effective procedure'. Instead, it relies solely and entirely on a basic grasp of the intuitive notion of an effective procedure. Though the result that effectiveness is undecidable is not surprising, it is also not without significance. It has the consequence, for example, that the solution to a decision problem, if it is to be complete, must be accompanied by a separate argument that the proposed ascertainment procedure invariably terminates with the correct verdict.

Mathematics Subject Classification. Primary 03A05; Secondary 03C57, 03D10, 03D40, 03D80, 68Q01, 68Q05.

Keywords. Algorithm, Church's thesis, Church–Turing thesis, Computability, Decidability, Decision procedure, Effectively calculable, Effective procedure, Turing.

I

In computability theory, a *decision procedure* (or a *test*) *for* property F *applicable to* things of kind K(a "decision procedure for Fness of Ks") is an effective procedure (effective method), or an algorithm (in the sense relevant to computability theory), for ascertaining, concerning any input of K, whether it has F or instead lacks it. That is, it is an effective step-by-step procedure for working out through algorithmic calculation, the correct true/false verdict, or the correct answer to the yes/no question, concerning any input of K whether it has F. Decision procedures are not restricted to mathematical properties of natural numbers. To take an example from music theory, there is a decision procedure for the property of being a chord tone in a given musical key. The method of truth tables not only defines propositional-calculus validity (tautological validity) of logical arguments but also provides a decision procedure. A decision procedure for a property F (*effectively*) *decides* a class G iff G is the class of things having the decision procedure's target property F. The method of truth tables thus decides the class of arguments that are derivable (deducible) in the propositional calculus. We say that a class G is (*effectively*) *decidable* (or that "the Gs are decidable") if there exists a (known or unknown) decision procedure that decides G; otherwise G is *undecidable*. *The decision problem* for a property F and a kind $K$is the problem of producing a decision procedure for Fness of Ks. A decision problem (as the term is used here) is not to be confused with the relevant yes/no questions themselves about each appropriate input; it is a demand for an algorithm for ascertaining the answers to those questions. A decision problem is solved by providing a step-by-step decision procedure, or at least by establishing an effective method for producing a decision procedure. A decision problem is said to be *solvable* iff there exists a (known or unknown) decision procedure that solves it, and is *unsolvable* otherwise.[1]

A function is *general recursive* iff it is an element of the smallest class of functions on the natural numbers that includes all constant functions, projections, and the successor function, and is closed under function composition, primitive recursion, and minimization. These are also the functions on the natural numbers that are computable by a Turing machine (and also the λ-definable functions on the natural numbers, as well the functions that satisfy other precise mathematical definitions). General recursive functions are paradigms of effectively, i.e., mechanically or algorithmically, calculable functions. A decision procedure can be provided by representing it as a general recursive function from Gödel numbers of the inputs to $\{\ 0,\ 1\}$, representing falsity and truth (or 'no' and 'yes'), respectively. The Church–Turing thesis is the thesis that all effectively calculable functions on the natural numbers are general recursive. In effect, it is the proposition that the effectively calculable functions on the natural numbers are exactly the general recursive functions, and hence exactly the Turing-computable functions on the natural numbers.[2]

There are alternative characterizations of the notion of an effective or mechanical procedure, but no characterization is universally accepted by experts.[3]

[1] The terminology of 'decides' and 'effective' is not ideal for the concepts under discussion. I endeavor to respect the use of the term 'decidable' in recursion/computability theory to the greatest extent possible. This objective has guided my choice of terms.

[2] The thesis was first advanced in Alonzo Church's [1]. (Turing-computable functions are not restricted to functions on the natural numbers.)

[3] For one such attempt at a precise definition see Jack Copeland [5]. *Cf* Hartley Rogers Jr., [17], at pp. 2–3.

This situation raises questions. Is there nevertheless a generally intended notion? If so, can it be fully captured by a reasonably precise, sense-preserving definition? There is a pressing further question. Suppose a decision procedure is proposed as a solution to a decision problem, but it is questioned whether the proposed procedure is in fact effective. Is there a general recipe for effectively calculating whether a given candidate is, or is not, an effective or algorithmic calculation procedure? In particular, can a precise, sense-preserving definition for 'effective procedure' (assuming there is such) yield such a general recipe? This is, in effect, *the decision problem for effective procedures.* To be sure, there is something circular about the idea of ascertaining by means of an effective procedure whether a given procedure is effective, but it is not unreasonable merely on that ground to seek such a procedure. No vicious circularity is involved in the search. Given a solution to the decision problem for effective procedures, one need only apply that decision procedure to establish whether a proposed solution to a decision problem is effective. On the other hand, the mere nonexistence of known counter-instances is no proof that a proposed decision procedure is effective. If there is no effective recipe for deciding effectiveness, then (unless the effectiveness of a procedure is known immediately and non-inferentially) if a solution to a decision problem is to be definitive, it should be accompanied by a proof of its effectiveness for the question at hand (even if that proof is trivial). It might be demonstrated, for example, that the procedure parallels the standard calculation of a recursive function. In actual practice, the effectiveness of a decision procedure is typically beyond reasonable doubt, making proof of effectiveness otiose. Still, strictly speaking, if the effectiveness of a procedure is undecidable, then the very endeavor to solve a decision problem seems to require a proof of effectiveness, in order for the solution to be definitive. This requires at least a grasp of a reasonably precise, generally intended notion of effectiveness.

The relation *sequence s is a first-order logical proof of formula φ* is general recursive. Church's theorem is that whereas the proof relation of first-order logic is general recursive, provability in first-order logic is only semi-recursive. A once standard view, enshrined in authoritative encyclopedia articles and the like, is that by contrast to Church's theorem, the Church–Turing thesis, even if correct, is not susceptible to proof, because the central notion of an effective procedure is not sufficiently formal and precise as to belong to mathematics proper, in a narrow sense excluding the theory of effectiveness, or to lend itself to proof of equivalence to a mathematical notion. In 1934 Gödel said of a thesis very close to what would come to be known as 'Church's thesis' that it "cannot be proved, since the notion of finite computation is not defined, but it serves as a heuristic principle".[4]

In advancing his thesis, Church characterized the notion of an effectively calculable function as a "somewhat vague intuitive notion" in contrast to the

[4] Kurt Gödel [9], at p. 44*n*3.

mathematical notion of a recursive function.[5] Richard Jeffrey said, "This thesis is not susceptible of mathematical proof, for there is no limit to the variety of forms that clerical routines [effective procedures] might assume, and thus no general, precise definition of the term 'clerical routine' of the sort we would need in order to prove the Church–Turing thesis".[6] László Kalmár said, "Church's thesis is not a mathematical theorem which can be proved or disproved in the exact mathematical sense, for it states the identity of two notions only one of which is mathematically defined while the other is used by mathematicians without exact definition".[7] Similarly, Stephen Kleene said, "Since our original notion of effective calculability of a function (or of effective decidability of a predicate) is a somewhat vague intuitive one, the thesis cannot be proved".[8] Later he said, "This is a thesis rather than a theorem, in as much as it proposes to identify a somewhat vague intuitive concept with a concept phrased in exact mathematical terms, and thus is not susceptible of proof".[9] Hartley Rogers said that the concept of a recursive function is "one way of making precise the *informal* mathematical notion of function computable 'by algorithm' or 'by effective procedure'. ... The claim that each of the standard formal characterizations provides satisfactory counterparts to the informal notions of *algorithm* and *algorithmic function* cannot be proved".[10] Joseph Shoenfield said, "Unfortunately, no one has given a proof of Church's thesis ..., or even isolated the properties of calculable functions which would be needed in such a proof".[11] Alan Turing wrote, "No attempt has yet been made to show that the 'computable' numbers [i.e., the numbers whose decimal expansion are Turing computable] include all numbers which would naturally be regarded as computable. All arguments which can be given are bound to

[5] In [2], Church explicitly put forward his proposal not as a "thesis," but as a "definition" of 'effectively calculable', "so far as positive justification can ever be obtained for the selection of a formal definition to correspond to an intuitive notion" (§ 7). He also said, "The adequacy of this technical definition to represent the intuitive notion of effective calculability.. is not immediately clear, but is... beyond any real doubt", in his [4]. Church obviously did not regard the proposed "technical definition" as sense-preserving, since he was explicit that the two notions differ (one being "intuitive", the other "formal"). This historical, and historic, fact is sometimes overlooked, for example by Robert Irving Soare [20], at p. 250. *Cf.* Herbert Enderton [7], at p. 199; Elliott Mendelson [16]; and Janet Folina [8]. (I recall Church asserting in lecture about the thesis that bears his name that it is unprovable because the technical notion of a general recursive function is formal whereas the intuitive notion of an effectively calculable function is informal.)

The Church–Turing "thesis" might better be termed a *conjecture*. Indeed it was largely confirmed. *Cf.* Saul Kripke [14]. Kripke's proof may be seen as reducing the Church–Turing thesis to Hilbert's thesis that the steps of any mathematical deduction can be given in the language of first-order logic. The proofs given below do not appeal to Hilbert's thesis. *Cf.* also Dershowitz, N. and Y. Gurevich [6]; W. Sieg [19].

[6] Richard Jeffrey [10], p. 105.

[7] László Kalmár [11], pp. 72–80, at p. 72.

[8] Stephen Kleene [12], at p. 317.

[9] Kleene [13], at p. 232.

[10] Hartley Rogers [17], pp. 1, 20.

[11] Joseph Shoenfield [18], at p. 120.

be, fundamentally, appeals to intuition, and for this reason rather unsatisfactory mathematically". [12] The *Wikipedia* entry on 'effective method' says of the Church–Turing thesis, "As this is not a mathematical statement, it cannot be proven by a mathematical proof".

In keeping with the spirit of the existing widespread agreement that the notion of an effective procedure is excessively vague or intuitive to lend itself to mathematical proof of the Church–Turing thesis, Elliott Mendelson said, "Of course, because of the vagueness of the intuitive notions of effectively computable functions and algorithm, it is impossible to *prove* the validity of Church's Thesis".[13] However, in 1990 Mendelson made a more nuanced observation, arguing that even if the converse is not provable, it is in fact rigorously provable that all general recursive functions are effectively calculable. He wrote: The assumption that a proof connecting intuitive and precise mathematical notions is impossible is patently false. In fact, half of CT (the 'easier' half), the assertion that all partial-recursive functions are effectively computable, is acknowledged to be obvious in all textbooks in recursion theory. A straightforward argument can be given for it.... This simple argument is as clear a proof as I have seen in mathematics, and it is a proof in spite of the fact that it involves the intuitive notion of effective computability. ([16], at pp. 232–233)

Here we shall make a case that a substantive theorem about effective procedures is logically provable directly from that intuitive notion—informal, imprecise, and ill-defined though it may be—without invoking any (partially) mathematical thesis, conjecture, or definition.

Insofar as there is a generally intended notion of an effective procedure or algorithm, it is almost certainly definable to a reasonably high degree of precision. (See footnote 3.) I submit that whereas the notion of effective calculability belongs to computability theory, it is ultimately epistemic in content, and is thereby not a notion of mathematics proper in a narrower sense. The logical notion of a proof is also partly epistemic in content, concerning an *a priori* and distinctly mathematical kind of epistemic justification. Mathematics in the narrower sense consists of what is epistemically justified by means of mathematical proofs, and of the justifications, the proofs, themselves, but not of the very notion of epistemic justification and its cognates. The latter belong to meta-mathematics or to philosophy of mathematics, specifically to the epistemology of mathematics. Likewise, effective calculability concerns a distinctly mechanical, computational method of obtaining information and ascertaining the facts. It thereby invokes the philosophically rich and puzzling notions of *knowing which* or *knowing what.* Effective calculability is thus unlike the notion of a general recursive function, which is not a notion special to the epistemology of arithmetic. It is

[12] A. M. Turing [21], at p. 135.
[13] Elliott Mendelson [15], at p. 239.

noteworthy also that effective procedures are *a priori* methods applied to inputs.[14]

Still, insofar as the Church–Turing thesis, which invokes the notion of effective calculability, is a definite proposition, there are indeed substantive theorems primarily concerning effective calculation in general—including effective non-numeric calculation, such as a music-theoretic calculation of whether a given note is, or is not, a chord tone in the key of A♭m. It is proved below directly from the notion of an effective procedure—without invoking the Church–Turing thesis or anything similar—that the decision problem for effective procedures is unsolvable. As a corollary, the notion of an effective procedure cannot be encoded by means of a general recursive function. The result itself is closely related to the proof (attributable to Turing) that no Turing-computable function solves every case of the halting problem for Turing machines. There are differences, however. The latter result concerns ascertaining whether a given Turing program terminates.[15] The former is concerned more broadly with ascertaining whether a given task-performing procedure is effective, including procedures like the effective procedure for bisecting an angle in Euclidean geometry using only a compass and a straightedge. Whereas all effective procedures terminate, not all terminating procedures are effective. Also, the present result is that no effective procedure—whether a Turing program or not—decides effectiveness of each given procedure—whether a Turing program or not. (It is Turing's *thesis* or *conjecture* that all effectively calculable functions from natural numbers to natural numbers are Turing-computable.)

II

Some preliminaries: Among procedures (methods) for performing tasks (accomplishing some end) are procedures for producing or constructing something, including (but not limited to) procedures for producing correct answers to *wh-questions* of a certain class–answers to definite who-what-when-where-which-whether-why questions of that class. Among task-performing procedures (methods) are thus general procedures for determining the *wh-facts*. Let us call such a procedure an *ascertainment procedure for* that class. For example, placing an object onto a bathroom scale is a "what weight" ascertainment procedure for determining the weight of middle-sized objects. A *valuation procedure*

[14] This point is often phrased in terms of reliance on no instruments other than a writing utensil and paper. Such devices are not actually part of the effective procedure itself, which are in principle executable "in one's head", but merely facilitate real-world implementation by limited creatures. Likewise, a compass and a straightedge are no more part of the actual effective procedure for dissecting an angle than are the pencil and paper to which they are applied. They are devices that facilitate real-world implementation of the sub-procedures of constructing such-and-such a circle or line.

[15] The relevant result is that the class of pairs of automatic valuation procedures, or "programs", executable by a Turing machine and inputs for which the valuation procedure eventually terminates is not computed by any Turing machine—in effect, that the binary relation *Turing-valuation program x computes the value of its singulary function for y as argument* is not Turing-decidable. *Cf.* Kleene [12], p. 382. Rogers [17], pp. 24–25, provides a "proof" that relies on Church's thesis.

for a function is a "what value" ascertainment procedure for determining the value of that function for a given argument. A *judgment procedure* for a class of propositions is a procedure for ascertaining whether a proposition of that class is the case, yes or no. A genuine judgment procedure includes steps (at least one) for concluding truth and steps for concluding falsity, even if some of these steps will not be executed. A judgment procedure is representable by a valuation procedure for a corresponding function to $\{0, 1\}$.[16]

Consider how one might go about ascertaining, given an arbitrary entity of kind K, whether it has a particular property F. Let us call any ascertainment procedure for determining whether a given input of kind K has property F or instead lacks it, a *judgment procedure for* property F *applicable to* (things of) kind K("a judgment procedure for Fness of Ks"). An example of a judgment procedure is the classical "They love me; they love me not" method of determining whether one is loved by a particular person, by establishing the oddness-or-evenness of the number of petals on an associated flower.[17] Let us also say that a judgment procedure for F applicable to K *certifies* all and only those things of kind K that it deems to have property F.

As a special case, a judgment procedure might be applicable to judgment procedures (i.e., might be itself of a kind for which it is a judgment procedure). As a very special case, a judgment procedure applicable to judgment procedures—a judgment meta-procedure—might have its own target property. For example, a judgment meta-procedure for consisting of an odd number of steps could consist of an odd number of steps. More typically, a judgment procedure will either be inapplicable to itself or lack its target property (or both). A judgment procedure for being a so-called perfect number is not itself a number (rather, it is a procedure), let alone a perfect number. Let us say

[16] In personal correspondence Gary Mar proposed for consideration the following automatic procedure:

1. Determine whether 4 is a Goldbach number. 2. When it is determined that a given even number n is a Goldbach number, determine whether $n+$ 2 is a Goldbach number. 3. When it is determined that a given even number is not a Goldbach number, conclude that Goldbach's conjecture is incorrect.

This procedure terminates iff Goldbach's conjecture is incorrect. If the procedure 1-3 is a judgment procedure (a "whether" ascertainment procedure), then it is effective iff Goldbach is incorrect. If effectiveness of judgment procedures were decidable, Goldbach would be thereby decidable (albeit perhaps non-constructively if Goldbach is incorrect). However,procedure 1-3 does not qualify as a genuine judgment procedure, as the term is used here, since no step in the procedure has one conclude that the conjecture is true, assuming it is.

There is of course a related genuine judgment procedure for Goldbach's conjecture:

1′. Determine whether the preceding procedure 1-3 terminates. 2′. If it terminates, then conclude that Goldbach's conjecture is incorrect. 3′. If it does not terminate, then conclude that the conjecture is correct.

By contrast with procedure 1-3, procedure 1′-3′ includes a non-vacuous step for concluding that Goldbach is correct. However, given the result about the halting problem and Turing's thesis, procedure 1′-3′ is not effective.

[17] Some judgment procedures are more reliable than others. (Some advice: If you are tempted to implement the flower-petal routine, the most likely explanation is that they love you not.)

that a judgment procedure is *self-certifying* iff it certifies itself, and that it is *non-self-certifying* iff it is not self-certifying.

Consider the following meta-procedure for determining non-self-certification of judgment procedures:

P^{JP}_{SF}: 1. Given an input judgment procedureP^{JP}, determine whether it is self-applicable. 2. If P^{JP} is not self-applicable, conclude that it is non-self-certifying. 3. If P^{JP} is self-applicable, apply it to itself and determine whether there is a unique, unequivocal yes-or-no result of self-application. 5. If there is no such result, conclude that P^{JP} is non-self-certifying. 5. If the result of self-application is negative, conclude that P^{JP} is non-self-certifying. 6. If, but only if, the result of self-application is unequivocally positive, conclude that P^{JP} is not non-self-certifying.

Is P^{JP}_{SF}self-certifying for non-self-certification? Or is it non-self-certifying? By inspection, P^{JP}_{SF}certifies exactly those judgment procedures that are genuinely non-self-certifying. Since it is self-applicable, if it certifies itself, then it is non-self-certifying. Therefore P^{JP}_{SF} does not certify itself. But again by inspection, the only way that can be (step 6) is that the result of self-application is positive. In that case, P^{JP}_{SF}certifies itself. We have thus deduced a contradiction from P^{JP}_{SF}itself. This is *the paradox of non-self-certifying judgment procedures.*

The resolution to the non-self-certifying judgment-procedures paradox is straightforward. P^{JP}_{SF}does not qualify as a judgment procedure. The applicability class of $P^{JP}_{SF}$is all judgment procedures. If it were itself a judgment procedure, it would be self-applicable. But when applied to itself it yields contradictory results: that $P^{JP}_{SF}$is both non-self-certifying and not. This precludes $P^{JP}_{SF}$from being a judgment procedure.

Although the paradox of non-self-certifying judgment procedures is straightforwardly resolved, parallel considerations yield a significant result about an epistemically special class of task-performing procedures: effective procedures.

III

A valuation procedure for a function may be infallible, in the sense that it never delivers a wrong value for a given argument, and also automatic or deterministic, in that it never calls for (or even permits) initiative or creative choice ("ingenuity"), and furthermore *a priori*, while still failing to be effective. To be effective a valuation procedure must also invariably eventually deliver the right value after a finite number of steps in a finite duration. An (*effective*) *calculation procedure* for a function is an effective valuation procedure for the function, i.e., an algorithm for calculating the function. A function is *effectively calculable* iff there exists a (known or unknown) calculation procedure for it. A decision procedure for Fness of Ks (as defined above) is simply an effective judgment procedure for Fness of Ks. (See footnote 1.) There are judgment procedures for validity in full first-order logic. However, assuming the Church–Turing thesis, by Church's theorem (and Gödel's completeness theorem) there can be no decision procedure for validity in full first-order logic.

A decision procedure$P_F^K$for Fness of Ks immediately yields its mirror image—a decision procedure$P_{\sim F}^K$for the complement property (λx˜Fx) applicable to K—simply by applying P_F^K and reversing the results:

$P_{\sim F}^K$: 1. Given an input x of K, apply P_F^K to it. 2. If the result of applying P_F^K to x is negative, conclude that $x$is non-F. 3. If the result of applying P_F^K to x is positive, conclude that x is not non-F.

A decision procedure$P_F^K$for Fness of Ks certifies all and only those things of K that are elements of the class that P_F^K decides. Let us say that a judgment procedure is *effectively self-certifying* if it is both a decision procedure (i.e., effective) and self-certifying; otherwise it is *non- effectively-self-certifying.* Let us say that a judgment procedure is *effectively non-self-certifying* iff it is both a decision procedure and non-self-certifying. Any decision procedure is either effectively self-certifying or effectively non-self-certifying and not both.

A lemma is obtained by diagonalization on effective certification.

The effective self-certification lemma: *The class of self-certifying decision procedures is undecidable.*

Proof. Any decision procedure for effective self-certification of judgment procedures would immediately yield its mirror image, a decision procedure for non-effective-self-certification of judgment procedures. There is no such thing. For suppose there is. If it certifies itself, then since it is effective, and therefore infallible, it is non-self-certifying. Therefore, if there is such a thing, it does not certify itself. But in that case, since it is effective, it has its own target property: being not both effective and self-certifying. And in that case, if there is such a thing, since it is effective it certifies itself. This is a contradiction.

If one is interested in the question of whether a given judgment procedure applicable to judgment procedures is effectively self-certifying, in at least some cases some creative initiative may be required to determine the answer, since a general algorithm for resolving every case is not possible. This result alone is sufficient to establish Mendelson's observation that the notion of an effective procedure, though intuitive and not properly mathematical, also permits proofs invoking it. □

IV

Consider now the following judgment meta-procedure for effective self-certification:

P_{SC}^{JP}: 1. Given an input judgment procedureP^{JP}that is applicable to judgment procedures, determine whether P^{JP} is a decision procedure (effective for some property or other). 2. If P^{JP} is not a decision procedure, conclude that $P^{JP}$is not effectively self-certifying. 3. If P^{JP} is a decision procedure, apply it to itself. 4. If the result of self-application is positive, conclude that P^{JP} is effectively self-certifying. 5. If the result of self-application is negative, or there is no unique, unequivocal yes-or-no result of self-application, conclude that P^{JP} is not effectively self-certifying.

The result of P^{JP}_{SC} is positive for effective self-certification iff the input judgment procedure is effectively self-certifying. P^{JP}_{SC} is thus an infallible judgment procedure for effective self-certification of judgment procedures if anything is. However, as a consequence of the effective self-certification lemma, P^{JP}_{SC} is not a decision procedure for effective self-certification of judgment procedures, since nothing can be. How does P^{JP}_{SC} fail to be a decision procedure?

The undecidability of effectiveness: *The class of effective procedures is undecidable.*

Proof. The first step of P^{JP}_{SC} is to ascertain whether the input judgment procedure is effective. None of the remaining steps disqualify P^{JP}_{SC} from being effective. If the effective ascertainment procedures were decidable, then the first step of P^{JP}_{SC} would be effectively executable and P^{JP}_{SC} as a whole would then represent a specific decision procedure, one of the very sort precluded by the effective self-certification lemma. □

The effective self-certification lemma is little more than a truth of first-order logic. The undecidability of effectiveness, by contrast, belongs to computability theory, more specifically to the theory of effective procedures. The only non-logical observation employed in the proof is that the entirety of P^{JP}_{SC} represents a specific decision procedure provided its first step can be executed effectively.[18] This conditional is not subject to any real doubt. The proviso that the first step is effectively executable is a very big 'if'.

That effectiveness of procedures is not itself decidable is not surprising. (*Cf.* Rice's theorem.) What is noteworthy is that the result is provable directly from the "somewhat vague intuitive" notion of an effective step-by-step procedure. No appeal is made to any lemma, conjecture, or thesis invoking recursiveness or Turing-computability. In fact, there is no reliance on anything very mathematical. The proof does not even appeal to a precise definition of

[18] The proof may be regarded as invoking a principle stating a conditional relationship regarding the composition of an effective decision procedure out of effective sub-procedures, perhaps something like the following:

If a judgment procedure P^{K}_{F}, for Fness of Ks, consists of a finite sequence of steps such that

(i) each of the steps is individually effectively executable (executable by an effective procedure);

(ii) the first step is applied to an initial input I;

(iii) any input to a successor step is the result of an earlier step, or else is the initial input I; and

(iv) for any initial input $I\prime$ to P^{K}_{F}, if $I\prime$ is of K, then: $I\prime$ has F iff invariably when the sequence of steps is fully implemented P^{K}_{F} culminates in a finite duration in certifying $I\prime$,

then the result of replacing each effectively executable step of P^{K}_{F} by its corresponding effective procedure is a decision procedure for Fness of Ks.

This principle is not merely a conjecture but presumably a theorem concerning effectiveness. Unlike the Church–Turing thesis, it is a trivial and obvious analytic truism concerning effective procedures.

Cf. Alonzo Church [3], at p. $52n118$.

'effective procedure'. Instead, it relies solely and entirely on a basic grasp of the intuitive notion of such a procedure.

Although the notion of an effective procedure is not itself susceptible to a decision procedure, as noted above the generally intended notion is evidently definable with a reasonably high degree of precision. In fact, an appropriate definition provides a checklist of necessary and sufficient conditions, which in turn yields a judgment procedure for effectiveness of judgment procedures. However, the judgment procedure does not qualify as a decision procedure. In particular, the requirement that the procedure invariably eventually culminate in the correct verdict ("halts") after finitely many steps in a finite duration is not always decidable. In fact, it is a corollary of the undecidability of effectiveness that any proposed definition for 'effective procedure' that is itself decidable does not correctly capture the intended notion. There is thus a kernel of truth in the claim that the notion of an effective procedure is not precisely definable. Whereas the intended notion can evidently be defined reasonably precisely, effectiveness is not a notion of mathematics proper (in the relevant sense), and no definition, however precise, can provide a decision procedure for it. A definition that provides a correct checklist without providing a test is the best of all possible worlds. This situation is not unfamiliar. The definition of a theorem of first-order logic, although precise, does not generate a decision procedure for first-order-logical provability. Likewise, the definition of a recursive function, although precise, does not generate a decision procedure for recursiveness. Otherwise, theoretically Church's theorem that the class of first-order-logical validities is not recursive could be proved by an effective calculation.[19]

The situation does pose a theoretical difficulty for the endeavor of seeking solutions to decision problems. The effectiveness of a proposed judgment procedure may be questioned, and there is no certain means by which all such challenges may be effectively either validated or invalidated.[20] But the difficulty is surmountable. It does not follow from the undecidability of effectiveness that it is unknowable whether some judgment procedures are effective. Rather no general algorithm is possible for resolving in every case whether a given ascertainment procedure is effective. If one is interested in the question of whether a proposed procedure is effective, in at least some cases some creative initiative may be required to determine the answer.[21] (Though the question

[19] Given the Church–Turing thesis, the class of effectively calculable functions on the natural numbers is effectively enumerable.

[20] *Cf.* Church's defense, in [3], pp. 52–53, of the demand that the notion of well-formedness be effective.

[21] In deriving the Church–Turing thesis from Hilbert's thesis Kripke [14], p. 80, argues that an effective calculation ("computation") of a function for a given argument is "a special form of mathematical argument ... In particular, the conclusion of the argument follows from the instructions as given and perhaps some well-known and not explicitly stated mathematical premises." By the present result, there is no algorithm for determining whether a given set of instructions (a given procedure) for valuating a function is of the specialized form in question. It does not follow that the specialized form is undefinable, or that it is unknowable whether a given set of instructions is indeed of the right form.

is undecidable, it is known whether P^{JP}_{SC} is a decision procedure, since it is provably not one.) In particular, if a procedure is proposed as a solution to a decision problem, its effectiveness must be established without the aid of an effective recipe for doing so in all cases. (The present result notwithstanding, special subclasses of effective procedures are decidable.)

The undecidability of effectiveness echoes the proof that no Turing machine solves the halting problem for Turing-valuation procedures. (See footnote 15 above.) An argument similar to the present proof demonstrates—without invoking the notion of a Turing-computable function, or the mathematically equivalent notion of a recursive function or that of a λ-definable function—that the relation, *valuation procedure x effectively calculates the value of its singulary function for y as argument*, is also undecidable. The present results, however, do not specifically concern computer programs or recursive functions. The undecidability of effectiveness applies more broadly to judgment procedures for properties of any sort—mathematical or otherwise—including judgment procedures for properties not known to be representable by means of a mathematical function, such as some procedures for ascertaining which course of action among options will best serve one's interests in the long run. Indeed, the undecidability of effectiveness applies to the full range of task-performing procedures.

Acknowledgements

The paper was written as the U.S. Senate was implementing a judgment procedure in the second impeachment trial of Donald Trump. I am grateful to Harry Deutsch, Melvin Fitting, Gary Mar, Oron Shagrir, and Teresa Robertson Ishii for their reactions and discussion. I am deeply indebted to Alonzo Church for his masterful tutelage.

References

[1] Church, A.: Preliminary report of [2]. Bull. Am. Math. Soc. **40**, 332–333 (1935)

[2] Church, A.: An unsolvable problem of elementary number theory. Am. J. Math. **58**, 345–363 (1936)

[3] Church, A.: Introduction to Mathematical Logic I. Princeton University Press, Princeton (1956)

[4] Church, A.: Logistic System. In: Runes, Dogobert D. (ed.) Dictionary of Philosophy. Philosophical Library, New York (1983)

[5] Copeland, J.: The Church-Turing Thesis. Stanford Encyclopedia of Philosophy, Stanford (2004)

[6] Dershowitz, N., Gurevich, Y.: A natural axiomatization of computability and proof of Church's Thesis. Bull. Symb. Logic **14**, 299–350 (2008)

[7] Enderton, H.: A Mathematical Introduction to Logic. Academic Press, New York (1972)

[8] Folina, J.: Church's Thesis: Prelude to a Proof. Philos. Math. **6**, 302–323 (1998)

[9] Gödel, K.: On Undecidable Propositions of Formal Mathematical Systems. In: Davis, M. (ed.) The Undecidable, pp. 41–74. Raven Press, Hewlett (1965)

[10] Jeffrey, R.: Formal Logic: Its Scope and Limits. Hackett, Indianapolis (2006)

[11] Kalmár, L.: An Argument Against The Plausibility of Church's Thesis. In: Heyting, A. (ed.) Proceedings of the Colloquium Held. North-Holland, Amsterdam (1959)

[12] Kleene, S.: Introduction to Metamathematics. North-Holland, Amsterdam (1952)

[13] Kleene, S.: Mathematical Logic. Wiley, New York (1967)

[14] Kripke, S.: The Church–Turing 'Thesis' as a Special Corollary of Gödel's Completeness Theorem. In: J. Copeland, C. Posy, and O. Shagrir, eds, Computability: Turing, Church, and Beyond. pp. 77–104 (2013)

[15] Mendelson, E.: Introduction to Mathematical Logic, 2nd edn. D. Van Nostrand, New York (1979)

[16] Mendelson, E.: Second thoughts about Church's Thesis and mathematical proofs. J. Philos. **87**(5), 225–233 (1990)

[17] Rogers, H., Jr.: Theory of Recursive Functions and Effective Computability. MIT Press, Cambridge (1987)

[18] Shoenfield, J.: Mathematical Logic, Reading. Addison-Wesley, Cambridge (1967)

[19] Sieg, W.: 2008, Church Without Dogma: Axioms for Computability. In: Lowe, B., Sorbi, A., Cooper, B. (eds.) New Computational Paradigms, pp. 139–152. Springer Verlag, New York (2008)

[20] Soare, R.I.: Interactive Computing and Relativized Computability. In: Copeland, J., Posy, C., Shagrir, O. (eds.) Computability: Turing, Church, and Beyond. MIT Press, Cambridge (2013)

[21] Turing, A.M.: On computable numbers, with an application to the Entscheidungsproblem. Proc. London Math. Soc. **42**(2), 230–265 (1936)

Nathan Salmón
Department of Philosophy
University of California
93106 Santa Barbara, CA
USA
e-mail: nsalmon@ucsb.edu

Received: July 9, 2022.
Accepted: March 17, 2023.