

The Equivalence Is a Theorem and the Thesis Is Not

— That Three Separate Starting Points Arrive at One Class Has Been Proved — But Agreement with
“Computable” Cannot Be the Object of a Proof — [Paper 260]

Yuuki YAMAGISHI (山岸 勇輝)

Yamagishi Complexity Torus Lab, Japan

August 2026

— *From thinking to syncing.* —

Abstract

Two claims of different standing hang from the phrase “can be computed”. This paper asks which of them has been proved—the answer is one of them only. That three formal definitions pick out the same class is a theorem; that this class is everything computable is a thesis and not a theorem. No new mathematical theorem and no new law is claimed. First, the starting points are counted. Between 1934 and 1936, three unrelated starting points were laid down: a machine that moves, the rewriting of expressions, and recursion—the pairwise equivalences among three number **3** (Section 2). Second, this is the core of the paper. Those **3** have all been proved—being comparisons between formal definitions, they can be the object of a proof (Section 3). Third, the thesis cannot be. With an informal “can be computed” on one side, it cannot be written as an equation—it is not that the proof is missing, but that it lies outside the procedure called proof (Section 4). Fourth, it is nonetheless refutable. Exhibit one function outside the class, computed by a procedure everybody grants is computation, and it falls—none has appeared in 90 years, but there is no guarantee that none will (Section 5). Fifth, there are levels inside the class as well. The Ackermann function is computable while not primitive recursive, and $A(4, 2) = 2^{65536} - 3$ has **19729** digits (Section 6). Sixth, those levels are visible in numbers. From $A(4, 0) = 13$ to $A(4, 1) = 65533$ is a factor of **5041**, and from there to $A(4, 2)$ the digit count leaps from **5** to **19729** (Section 6).

Scope of this paper (scope note): No new mathematical theorem and no new law is claimed—Turing machines, the λ -calculus, general recursive functions, their equivalence, the Church-Turing thesis, and the fact that the Ackermann function is not primitive recursive are all standard. No computability theory is built—what is used is one recursive definition, a count of pairs, and a count of digits. The equivalences are not proved—the proofs of the three equivalences are not carried out here. Only the fact that they have been proved, and the reason they can be objects of proof, is written down. The truth of the thesis is not judged—it is called neither correct nor incorrect. What is said is only that its standing is not that of a theorem. The halting problem is not treated—Paper 179 touches on it. This paper treats not an uncomputable object but the standing of the phrase “can be computed”. Complexity is not entered—Paper 04 treats **P** against **NP**. Complexity and computability are different levels, and only the latter is examined here. The Ackermann function is not fully evaluated—from $A(4, 2)$ onwards only the digit count is given, not the value, and $A(4, 3)$ and beyond are outside the tools of this paper. The list of definitions is not exhaustive—Section 2 gives a selection, and other formulations reach the same class. The years are those of first appearance and are not the years the equivalences were proved. Relation to earlier papers: Paper 163 showed that “a good code exists” and “here is a good code” are different sentences—this paper likewise treats a difference in the standing of two claims, but there it was existence against construction and here it is theorem against thesis. Paper 189 showed that unprovability is not a property of a sentence but a relation to an axiom system—Section 4 here treats a claim that is not even a sentence in that sense, lying outside the relation. Paper 179 showed that “cannot be constructed” has distinct roots—this paper counts the three ways of exhibiting “can be computed”. Paper 232 counted four different claims written with the same equals sign—this paper writes that the equals sign of the thesis is not an equals sign at all. Paper 190 measured rare on a logarithmic scale—Section 6 here likewise reports in digits. What is added is counting and setting out the three definitions and the three equivalences, placing the division between theorem and thesis at the single point of whether both sides are formal, writing that the thesis is unprovable and yet refutable, and evaluating the Ackermann function and obtaining the 19729 digits of $A(4, 2)$.

1. Notation and Placement

Setting: split the sentence “this function can be computed” into two.

(A) The formal claim: three formal definitions (machine, rewriting, recursion) pick out the same class of functions.

(B) The informal claim: that class is everything that people call computable.

Question: do (A) and (B) have the same standing?

The answer is no. (A) is a theorem and has been proved. (B) is a thesis and cannot be the object of a proof.

A caution: this paper does not say that (B) is wrong. It says only that its standing is not that of a theorem—being correct and being provable are different questions.

2. Counting the Starting Points

Check 2.1: set out the three starting points. The years are those of first appearance, not those in which the equivalences were shown.

Formulation	What moves	Proposed by	Year
General recursive functions	rules of recursion	Herbrand, Goedel	1934
The λ -calculus	rewriting of expressions	Church	1936
Turing machines	a machine moving on paper	Turing	1936

Observation 2.2

the three do not resemble one another. One is recursion over the numbers, one the rewriting of strings of symbols, one an idealisation of a physical device—nothing in the definitions shows why the same thing should come out.

Check 2.3: count the pairwise equivalences among the three.

Pair	Equivalent?
Recursive functions $\leftrightarrow$ λ -calculus	a theorem
λ -calculus $\leftrightarrow$ Turing machines	a theorem
Turing machines $\leftrightarrow$ recursive functions	a theorem
Total	3

Observation 2.4

three definitions admit $\binom{3}{2} = 3$ pairs, and every one of them has been proved. Register machines and type-0 grammars were later added to this, and all of them fell into the same class—what Section 2 lists is a selection.

3. The Core: the Equivalence Is a Theorem

Observation 3.1

why can it be proved?Because both sides are formal.Both “the set of functions computable by a Turing machine” and “the set of functions expressible by a λ -term” have their definitions written out—so it suffices to show the inclusion in both directions.

Observation 3.2

the shape of the proof is concrete too.One builds a procedure by which one side simulates the other—encoding λ -terms on a tape and having the machine carry out the reductions, and so on.

Observation 3.3

these 3 are the theorem side of this paper.Up to here, matters are settled in the ordinary mathematical sense.

4. The Thesis Is Not a Theorem

Observation 4.1

the Church-Turing thesis says this—“this class is all of the intuitively computable functions”.The right-hand side, “intuitively computable”, has no written-out definition.

Check 4.2: set the two claims side by side by their standing.

Claim	Left side	Right side	Standing
The equivalence	formal	formal	a theorem
The thesis	formal	informal	not a theorem

Observation 4.3

the separator—what has to be changed for these to become different things. Whether the right-hand side is formal, and that alone. The left-hand sides are the same object in both rows.

Observation 4.4

this is not “not yet proved”. Agreement with an informal notion cannot be an input to the procedure called proof—the same matter as Paper 232 counting four standings of one equals sign, and the equals sign of the thesis is not an equals sign at all.

Observation 4.5

what follows from conflating them? One writes that computability has been proved—what has been proved is the 3 equivalences, and the thesis has not been.

5. And Yet It Is Refutable

Observation 5.1

being unprovable and being irrefutable are different. To bring the thesis down, exhibit one function outside the class computed by a procedure everybody grants is computation.

Check 5.2: set out what would bring it down.

What appears	The thesis
A function outside the class, computed by anyone in finite steps	falls
A function outside the class, computed with infinite resources	stands (that is no procedure)
A function inside the class, computed by a new method	stands (the class is the same)
A proof that functions outside the class exist	stands (that was already known)

Observation 5.3

of the four rows, only the first brings it down. Functions outside the class, such as the halting problem, are already known, but that is evidence on the uncomputable side and not a refutation of the thesis.

Observation 5.4

for 90 years the first row has not appeared. But that is evidence and not a proof—a refutable claim that goes long unrefuted enjoys strong empirical support, and is not a theorem.

6. There Are Levels Inside the Class as Well

Setting: the inside of “computable” is not uniform. The Ackermann function is computable while not primitive recursive—a concrete instance of the levels being real.

Check 6.1: evaluate it straight from the definition.

	$n = 0$	$n = 1$	$n = 2$	$n = 3$	$n = 4$
$A(0, n)$	1	2	3	4	5
$A(1, n)$	2	3	4	5	6
$A(2, n)$	3	5	7	9	11
$A(3, n)$	5	13	29	61	125

Observation 6.2

$A(3, n) = 2^{n+3} - 3$ —giving 5, 13, 29, 61, 125, 253, . . . , and the fourth row of the table sits on that formula exactly.

Check 6.3: move to the row $m = 4$.

	Value	Digits
$A(4, 0)$	13	2
$A(4, 1)$	65533	5
$A(4, 2)$	$2^{65536} - 3$	19729

Observation 6.4

from 13 to 65533 is a factor of **5041**. From there to the next, it is the digit count that leaps, from 5 to 19729, rather than the ratio—writing a ratio stops carrying meaning.

Observation 6.5

this function grows faster than every primitive recursive function, so it is not primitive recursive. And yet, as the table shows, it is actually being computed—a working piece of evidence that the inside of “computable” is not one layer.

Observation 6.6

a point to state honestly—the value of $A(4, 2)$ is not written out in this paper. Only its digit count is given, and $A(4, 3)$ and beyond are outside the tools used here.

7. Roll Call and Closing

1. Three starting points(Check 2.1): recursion in 1934, rewriting in 1936, a machine in 1936.
2. The three do not resemble one another(Observation 2.2): nothing in the definitions shows why they agree.
3. Three equivalences(Check 2.3): $\binom{3}{2} = 3$, every one of them proved.
4. Why they can be proved(Observation 3.1): because both sides are formal.
5. The thesis has an informal side(Check 4.2): hence it is not a theorem.
6. The separator is one point(Observation 4.3): whether the right-hand side is formal.
7. It can be refuted(Check 5.2): of the four rows, only the first brings it down.

8. **Ninety years without one**(Observation 5.4): **evidence and not a proof.**

9. $A(3, n) = 2^{n+3} - 3$ (Observation 6.2): **5, 13, 29, 61, 125.**

10. $A(4, 2)$ has **19729 digits**(Check 6.3): **computable while not primitive recursive.**

Closing: two things happened in 1936.One is a theorem: three unrelated starting points arrived at one class—the 3 pairwise equivalences are settled in the ordinary mathematical sense. The other is a thesis: that this class is everything computable—and this one has an informal right-hand side, so it lies outside the procedure called proof. Conflate them and one writes that computability has been proved. And the thesis, while unprovable, can be refuted—that it has not been in 90 years is strong evidence, and not a theorem. One thing separates them—writing down whether both sides are formal. Write it down, and what is settled separates from what is merely well supported. Do not write it down, and the credit earned by 3 theorems flows silently across to the thesis.

References

1. Turing, A. M. (1936). On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42.
2. Church, A. (1936). An Unsolvable Problem of Elementary Number Theory. *American Journal of Mathematics*, 58(2).
3. Kleene, S. C. (1936). General Recursive Functions of Natural Numbers. *Mathematische Annalen*, 112.
4. Turing, A. M. (1937). Computability and λ -Definability. *The Journal of Symbolic Logic*, 2(4).
5. Ackermann, W. (1928). Zum Hilbertschen Aufbau der reellen Zahlen. *Mathematische Annalen*, 99.
6. Soare, R. I. (2016). *Turing Computability: Theory and Applications*. Berlin: Springer.
7. Odifreddi, P. (1989). *Classical Recursion Theory*. Amsterdam: North-Holland.
8. Yamagishi, Y. (2026). *The Physical Resource Bound on Brute-Force Search* [Paper 04]. Zenodo. 10.5281/zenodo.21404131
9. Yamagishi, Y. (2026). *“A Good Code Exists” and “Here Is a Good Code” Are Different Sentences* [Paper 163]. Zenodo. 10.5281/zenodo.22092677
10. Yamagishi, Y. (2026). *“Cannot Be Constructed” Has Distinct Roots* [Paper 179]. Zenodo. 10.5281/zenodo.22098419
11. Yamagishi, Y. (2026). *“Unprovable” Is Not a Property of a Sentence* [Paper 189]. Zenodo. 10.5281/zenodo.22105830
12. Yamagishi, Y. (2026). *“Rare” Is Measured on a Logarithmic Scale* [Paper 190]. Zenodo. 10.5281/zenodo.22106182
13. Yamagishi, Y. (2026). *Learn from a Book with the Proofs Removed and the Equals Sign Loses Its Type* [Paper 232]. Zenodo. 10.5281/zenodo.22127528

— From thinking to syncing. —

Yuuki YAMAGISHI (山岸 勇輝)

