

# Turing and the Art of Classical Computability

Robert Irving Soare

---

---

## Contents

<b>1 Mathematics as an Art</b>	<b>1</b>
<b>2 Defining the Effectively Calculable Functions</b>	<b>2</b>
<b>3 Turing's Achievement</b>	<b>3</b>
<b>4 Why Turing and not Church?</b>	<b>4</b>
<b>5 Why Michelangelo and not Donatello?</b>	<b>4</b>
<b>6 Classical Art and Classical Computability</b>	<b>6</b>
<b>7 Computability and Recursion</b>	<b>7</b>
<b>8 The Art of Exposition</b>	<b>8</b>
<b>9 Conclusion</b>	<b>8</b>

## 1. Mathematics as an Art

Mathematics is an art as well as a science. It is an art in the sense of a *skill* as in Donald Knuth's series, *The Art of Computer Programming*, but it is also an art in the sense of an *esthetic endeavor* with inherent *beauty* which is recognized by all mathematicians.

One of the world's leading art treasures is Michelangelo's statue of *David* as a young man displayed in the Accademia Gallery in Florence. There is a long aisle to approach the statue of David. The aisle is flanked by the statues of Michelangelo's unfinished slaves struggling as if to emerge from the blocks of marble. These figures reveal Michelangelo's work process. There are practically no details, and yet they possess a weight and power beyond their physical proportions. Michelangelo thought of himself not as

carving a statue but as seeing clearly the figure within the marble and then chipping away the marble to release it. The unfinished slaves are perhaps a more revealing example of this talent than the finished statue of David.

Similarly, it was Alan Turing [1936] and [1939] who saw the figure of computability in the marble more clearly than anyone else. Finding a formal definition for effectively calculable functions was the first step, but *demonstrating* that it captured effective calculability was as much an artistic achievement as a mathematical one. Gödel himself had expressed doubt that it would be possible to do so.

## 2. Defining the Effectively Calculable Functions

The *Entscheidungsproblem*, the decision problem for first order logic, emerged in the early 1920's in lectures by Hilbert and was described in Hilbert and Ackermann [1928]. To show this problem unsolvable one first had to mathematically define the effectively calculable (informally computable) functions. From 1931 to 1934, Church and his student Kleene developed the  $\lambda$ -definable functions. Church privately proposed to Gödel around March, 1934 that  $\lambda$ -definable functions should be identified with the effectively calculable functions. Gödel rejected this as “thoroughly unsatisfactory.”

Gödel [1934] gave a series of lectures at Princeton in which he defined the *Herbrand-Gödel (HG) recursive functions*, a class of functions as a deductive formal system with initial functions (like axioms) and with two rules of inference to derive new functions. Church [1936] proposed *Church's Thesis* that a function is effectively calculable if and only if it is Herbrand-Gödel recursive. Gödel still did not accept it.

Kleene [1936] then defined the  *$\mu$ -recursive functions* by combining the (Gödel) numbering of syntax in Gödel's Incompleteness Theorem [1931] with the HG recursive functions. Kleene defined a primitive recursive predicate  $T(e, x, y)$  which asserts that  $e$  is the Gödel number of a system  $E$  of HG equations,  $y$  is the Gödel number of an HG computation from  $E$  on input  $x$ , and the primitive recursive function  $U(y) = z$  decodes the output  $z$  from  $y$ . This definition is mathematically correct and prevailed for several decades in research papers from 1935 to 1965, but it is not intuitive, being based on two unintuitive formalisms.

By 1936 Gödel knew these definitions, and their mathematical equivalence. Gödel was responsible for the second and largely for the third, but he did not accept any of them as defining an effectively calculable function. Indeed Gödel suggested that it might not be possible to give a mathematical

definition of calculability, and he wrote in footnote 3 of [1934] “. . . the notion of finite computation is not defined, but serves as a heuristic principle.”

### 3. Turing’s Achievement

At that moment in 1936 a twenty-three year old youth in Cambridge brought a new vision which immediately convinced Gödel. Turing’s unparalleled achievement in [1936] consisted of at least several remarkable parts which we sketch only briefly because they are very well-known. Turing [1936]: (1) defined an automatic machine (Turing machine) based on his model of how a human being might carry out a calculation; (2) defined a universal Turing machine whose inputs included both programs and integers and could simulate any Turing machine on any input; (3) gave a remarkable demonstration that any function calculated by a human being could be computed by an *a*-machine. Turing [1936] then stated what was later known as *Turing’s Thesis* that a function on the integers is computable by a finite procedure if and only if it is computable by a Turing machine.

First, Turing gave a model based on a mechanistic approach to human computing, something the previous models lacked. Perhaps even more impressive was Turing’s careful analysis in component parts of how a human being might calculate and then an argument why his Turing machine could simulate this calculation. By comparison, Church [1936] tried to carry out a similar argument that any calculable function is HG recursive, but Gandy [1988, p. 79] and Sieg [1994, pp. 80, 87] pointed out the flaws in Church’s argument. Gödel never accepted Church’s Thesis, but he accepted Turing’s Thesis at once, and stated:

“That this is really the correct definition of mechanical computability was established beyond any doubt by Turing.” *Gödel Collected Works Volume III [1995, §3.3]*;

“ But I was completely convinced only by Turing’s paper.” *Gödel: letter to Kreisel of May 1, 1968 [Sieg, 1994, p. 88]*.

“The greatest improvement was made possible through the precise definition of the concept of finite procedure, . . . This concept, . . . is equivalent to the concept of a ‘computable function of integers’ *Gödel [1951, pp. 304–305], Gibbs lecture*.

Kleene [1981b, p. 49] wrote, “Turing’s computability is intrinsically persuasive” but “ $\lambda$ -definability is not intrinsically persuasive” and “general recursiveness scarcely so (its author Gödel being at the time not at all persuaded).”

Kleene wrote in his second book [1967, p. 233], “Turing’s machine concept arises from a direct effort to analyze computation procedures as we know them intuitively into elementary operations. Turing argued that repetitions of his elementary operations would suffice for any possible computation. For this reason, Turing computability suggests the thesis more immediately than the other equivalent notions and so we choose it for our exposition.”

Church, in his review [1937] of Turing [1936] wrote, *Computability by a Turing machine*, “has the advantage of making the identification with effectiveness in the ordinary (not explicitly defined) sense evident immediately—*i.e.*, without the necessity of proving preliminary theorems.”

#### 4. Why Turing and not Church?

Why give so much credit to Turing and not to Church? Church was a full professor at Princeton in 1936 when Turing was a mere graduate student. Church was working in the HG recursive functions defined by Gödel, the most eminent logician at the time. It used the concept of recursion (induction) which had appeared in mathematics since Dedekind [1888] while Turing machines were a fanciful new invention without such a concise, mathematical definition in a well-known formalism. By 1934 Church and Kleene had shown that most number theoretic functions were  $\lambda$ -definable and therefore recursive, giving clear evidence for Church’s Thesis. Church was the first to propose Church’s Thesis when even Gödel did not believe it. Church got it right and he got it first. The effectively calculable functions *are* the recursive functions. By any purely quantifiable evaluation Church’s contribution was at least as important as Turing’s. However, characterizing human computability was *not* a purely quantifiable process.

#### 5. Why Michelangelo and not Donatello?

Donatello (1386–1466) was a sculptor in Florence. In 1430 he created the bronze statue of *David*, his most famous work. This was a remarkable work, innovative in many ways, the first free-standing nude statue since ancient times, the first major work of Renaissance sculpture. Now compare this to Michelangelo’s *David*, in 1504, the most famous statue in the world. Michelangelo broke away from the traditional way of representing David, with sword in hand and with the giant’s head at his feet (as with Donatello). Michelangelo has caught David tense with increasing power as he is about

to go to battle. Michelangelo places him in perfect contraposto outdoing the Greek representations of heros.

Michelangelo and Turing both completely transcended conventional approaches. They created something *completely new* from their own visions, something which went far beyond the achievements of their contemporaries. Second, both emphasized the *human form*. Michelangelo brought out the human form in his statues and the Sistine ceiling. Turing invented a system which simulates how a human being computes and then demonstrated that his creation did capture human computing.

Frank Zölner wrote in his book *Michelangelo Life and Work*, [2010, p. 7]

“As innovative as Leonardo da Vinci, who was a generation older, as productive as his slightly younger contemporary Raphael of Urbino, as secretive as Giorgione in Venice and blessed like Titian with a long life and unbridled creativity, Michelangelo Buonarroti embodies, perhaps most completely, the concept of the artist in the modern era.”

Likewise, Alan Turing embodies, perhaps most completely, the concept of human computability in the modern era. Regarding the creative process, Turing himself [1939, §11] wrote,

“Mathematical reasoning may be regarded rather schematically as the exercise of a combination of two faculties, which we may call *intuition* and *ingenuity*. The activity of the intuition consists in making spontaneous judgments which are not the result of conscious trains of reasoning. These judgments are often but by no means invariably correct. . . .

The exercise of ingenuity in mathematics consists in aiding the intuition through suitable arrangements of propositions, and perhaps geometrical figures or drawings.”

Turing’s first great contribution was by intuition. While others were studying deductive systems like HG recursion, Turing’s intuition drew him to a completely new model more clearly reflecting in mechanical terms how a calculation is carried out. His second contribution was his exercise of ingenuity which led him to a demonstration that anything computed by a human being could be computed by a Turing machine. Gandy [1988, p. 82] observed, “Turing’s analysis does much more than provide an argument for” Turing’s Thesis, “*it proves a theorem.*” Furthermore, as Gandy

[1988, pp. 83–84] pointed out, “Turing’s analysis makes no reference whatsoever to calculating machines. Turing machines appear as a result, a codification, of his analysis of calculations by humans.” Sieg [1994], [2006], [2009], gives a full analysis of Turing’s contribution. Wittgenstein remarked about Turing machines, “These machines are *humans* who calculate.”

## 6. Classical Art and Classical Computability

The term “recursive” to mean “computable” began with Church [1936] and was developed by Kleene to prevail for sixty years in the form, “recursive function,” “recursively enumerable set,” and Recursive Function Theory.” In June, 1979, Gerald Sacks and I had each been invited to give a series of four lectures at the Italian Mathematical Summer Center (C.I.M.E.). Gerald was to lecture on *generalized recursion theory (GRT)* and I on recursion theory on the integers  $\omega$ , at that time sometimes called *ordinary recursion theory (ORT)*. I began the trip with a few days in Florence revisiting the Renaissance art treasures of the Uffizi gallery and Michelangelo’s statues in the Accademia. As the train made its way to Bressanone at the very north of Italy in the Dolomite alps, I was still basking in the memories of the art of the Renaissance.

As our courses began in Bressanone, Gerald kept repeating the term “Ordinary Recursion Theory (ORT).” He was doing nothing wrong, simply using the term as it had come to be used in the previous decade to distinguish  $\omega$ -recursion theory from GRT. And yet as the phrase kept cascading down it clashed more and more with my esthetic sense. It seemed far too impoverished to describe the magnificent theory created by Turing [1936], [1939], Post [1944], and the others. My colleagues at the University of Chicago, Alberto Calderone and Antoni Zygmund, worked in singular integrals and classical analysis, but classical analysis was never called “ordinary analysis” to distinguish it from functional analysis. No one ever called the art of the Renaissance “ordinary art” to distinguish it from Baroque art or Impressionism.

By my third lecture it all came together. I coined the term “*Classical Recursion Theory (CRT)*” and developed a whole lecture about the analogies between CRT and the classical art of the Italian High Renaissance. The lectures and art analogies were published in Soare [1981] but it is a rather obscure reference and not widely read. The lectures were expanded at the Cornell AMS meeting in July, 1982, but not published there. Some of the analogous characteristics are these.

*Human Scale.* A Roman arch such as the Arch of Constantin next to the Colosseum in Rome is designed to arouse awe and to dwarf the human figure. In contrast the Loggia della Signoria in Florence is on a human scale and designed to display statues of human size. The art and sculpture of the High Renaissance were designed to display the human form. Analogously, the computability theory of Turing and Post works on the integers which can be represented as in Turing by a finite sequence of ones and blanks. GRT works on infinite ordinals or on functionals of higher type.

*Composition and Balance.* The paintings of the Renaissance were characterized by highly complex compositions which were balanced to keep the eye from leaving the painting. Leonardo's *The Virgin and Saint Anne* has a very complicated and carefully designed composition around the three figures, Mary, her mother, Anne, and her son Jesus. The heads and feet form one large triangle. The arms and child form an inner rectangle. Everything holds the eye and prevents it from leaving the painting as it might in a Baroque painting. In classical computability, theorems such as the Friedberg Muchnik theorem are proved by a delicate balance of opposing requirements, positive requirements wanting to enumerate elements into a set  $A$  and negative requirements wanting to keep elements out. These constructions are often defined with as much intricacy and balance as a Renaissance composition. Other characteristics will be developed in later papers.

## 7. Computability and Recursion

Church [1936] and Kleene began to use the term "recursive" to mean "computable" as well as "inductively defined." Since Kleene was the main figure in the subject after 1940, this term had become standard for sixty years from 1936 to 1996. By the 1990's this usage had become problematic. When one referred to a recursive function did one mean "inductively defined" or "effectively computable?" Also, the term "recursive" was not well understood in the mathematical and scientific community and, if understood at all, it was identified with the elementary methods of iteration and recursion in a first programming course. Neither Turing nor Gödel ever used the word "recursive" to mean "computable" or "recursive function theory" to name the subject. When others did, Gödel reacted sharply negatively stating, "the term in question [recursive] should be used with reference to the kind of work Rosza Peter did."

Soare [1996] and [1999a] analyzed the history and meaning of computability and recursion and suggested that the terms "Computability The-

ory” and “computably enumerable set” be used in place of the recursive version. This was largely adopted within a few years.

## 8. The Art of Exposition

In the art of exposition it is not sufficient to have a correct theorem with a correct proof. It must be the right theorem with the right proof, relating the results which came before and those which will come after in an esthetically pleasing mix. The entire work must be artistically beautiful and must appeal to the imagination.

The initial expositions in Turing [1936] and Post [1944] were clear, intuitive, and very well motivated. In contrast, Kleene [1936] had developed the Kleene  $T$ -predicate as a Gödel coding of the Herbrand-Kleene recursive functions, which had little appeal to the imagination. Kleene’s mathematical results were very difficult but his  $T$ -predicate notation was hard to read. It dominated the proofs in the subject for over thirty years. For example, Friedberg used the Kleene  $T$ -predicate style proofs in his solution [1957a] to Post’s problem and his completeness criterion [1957c] which made the proofs difficult to read. Compare these proofs with the informal style of Rogers’ book [1967] written in a clear, intuitive style, which opened the subject to a generation of students and which was continued in Soare’s book [1987].

## 9. Conclusion

Mathematicians are not assigned projects like building bridges. Like artists, they choose which problems to work on according to taste and beauty. Like artists, what they produce is evaluated on the basis of beauty as well as mathematical results. The greatest results are those arising from a completely new vision and a profound intuition into the area.

Bibliographic references are given in the following.

R. I. Soare, Turing Oracle Machines, Online Computing, and Three Displacements in Computability Theory, *Annals of Pure and Applied Logic*, 160 (2009), 368–399.

R. I. Soare, *Computability Theory and Applications: The Art of Classical Computability*, Computability in Europe Series, Springer-Verlag, Heidelberg, 2012, to appear.

## References

- [1] [Church, 1936]  
A. Church, An unsolvable problem of elementary number theory, *American J. of Math.*, **58** (1936), 345-363.
- [2] [Feferman, 2006]  
S. Feferman, Turing's Thesis, *Notices of the Amer. Math. Soc.* **53** 2007, 1200–1206.
- [3] [Friedberg, 1957a]  
R. M. Friedberg, Two recursively enumerable sets of incomparable degrees of unsolvability, *Proc. Natl. Acad. Sci. USA* **43** (1957), 236–238.
- [4] [Friedberg, 1957c]  
R. M. Friedberg, A criterion for completeness of degrees of unsolvability, *J. Symbolic Logic* **22** (1957), 159–160.
- [5] [Friedberg, 1958a]  
R. M. Friedberg, Three theorems on recursive enumeration: I. Decomposition, II. Maximal Set, III. Enumeration without duplication, *J. Symbolic Logic* **23** (1958), 309—316.
- [6] [Gandy, 1988]  
R. Gandy, The confluence of ideas in 1936, In: Herken, [1988], 55–111.
- [7] [Gödel, 1931]  
K. Gödel, Über formal unentscheidbare sätze der Principia Mathematica und verwandter systeme. I, *Monatsch. Math. Phys.* **38** (1931) 173-178.
- [8] [Gödel, 1934]  
K. Gödel, On undecidable propositions of formal mathematical systems, Notes by S. C. Kleene and J. B. Rosser on lectures at the Institute for Advanced Study, Princeton, New Jersey, 1934, 30 pp.
- [9] [Gödel, 1995]  
K. Gödel, *Collected works Volume III: Unpublished essays and lectures*, S. Feferman et. al., editors, Oxford Univ. Press, Oxford, 1995.
- [10] [Kleene, 1936a]  
S. C. Kleene, General recursive functions of natural numbers, *Math. Ann.* **112** (1936), 727–742.

- [11] [Post, 1944]  
E. L. Post, Recursively enumerable sets of positive integers and their decision problems, *Bull. Amer. Math. Soc.* **50** (1944), 284–316.
- [12] [Rogers, 1967]  
H. Rogers, Jr., Theory of Recursive Functions and Effective Computability, McGraw-Hill, New York, 1967.
- [13] [Sieg, 1994]  
W. Sieg, Mechanical procedures and mathematical experience, In: A. George (ed.), *Mathematics and Mind*, Oxford Univ. Press, 1994, 71–117, .
- [14] [Sieg, 1997]  
W. Sieg, Step by recursive step: Church’s analysis of effective calculability, *Bull. Symb. Logic* **3** ( 1997), 154–180.
- [15] [Sieg, 2006]  
W. Sieg, Gödel on computability, *Philosophia Mathematica* **14** (2006), 189–207.
- [16] [Sieg, 2009]  
W. Sieg, On computability, (Andrew D. Irvine, editor) *Handbbook of the Philosophy of Mathematics* (2009), Elsevier, 535–630.
- [17] [Soare, 1981]  
R. I. Soare, Constructions in the recursively enumerable degrees, In: *Recursion theory and computational complexity*, G. Lolli ed., Proceedings of Centro Internazionale Matematico Estivo (C.I.M.E.), June 14–23, 1979, in Bressanone, Italy, Liguori Editore, Naples, Italy, 1981.
- [18] [Soare, 1987]  
R. I. Soare, *Recursively Enumerable Sets and Degrees: A Study of Computable Functions and Computably Generated Sets*, Springer-Verlag, Heidelberg, 1987.
- [19] [Soare, 1996]  
Computability and Recursion, *Bulletin of Symbolic Logic* **2** (1996), 284–321.
- [20] [Soare, 1999a]  
R. I. Soare, The history and concept of computability, In: *Handbook of Computability Theory*, ed. E. R. Griffor, North-Holland, Amsterdam, 1999, 3–36.

- [21] [Soare, 2007]  
 R. I. Soare, Computability and Incomputability, Computation and Logic in the Real World, in: Proceedings of the Third Conference on Computability in Europe, CiE 2007, Siena, Italy, June 18–23, 2007, Lecture Notes in Computer Science, No. 4497, S.B. Cooper, B. Löwe, Andrea Sorbi (Eds.), Springer-Verlag, Berlin, Heidelberg, 2007.
- [22] [Soare, 2009]  
 R. I. Soare, Turing Oracle Machines, Online Computing, and Three Displacements in Computability Theory, *Annals of Pure and Applied Logic*, 160 (2009), 368–399.
- [23] [Soare, 2012a] = [Soare, CTA]  
 R. I. Soare, *Computability Theory and Applications: The Art of Classical Computability*, Computability in Europe Series, Springer-Verlag, Heidelberg, 2012, to appear.
- [24] [Soare, 2012c]  
 R. I. Soare, Turing-Post Relativized Computability and Interactive Computing, in: Jack Copeland, Carl Posy, and Oron Shagrir (Editors), *Computability: Gödel, Church, Turing, and Beyond*, MIT Press, 2012, to appear.
- [25] [Soare, 2012d]  
 R. I. Soare, The Impact of Turing on Computable Reducibility and Information Content, in: Rodney Downey (Editor) Association for Symbolic Logic Lecture Notes in Logic (CUP), 2012, to appear.
- [26] [Soare, 2012e]  
 R. I. Soare, Turing Computability and Information Content, *Philosophical Transactions of the Royal Society A*, Ed. Samson Abramsky, to appear.
- [27] [Turing, 1936]<sup>1</sup>  
 A. M. Turing, On computable numbers, with an application to the

---

<sup>1</sup>Many papers, Kleene [1943, p. 73], [1987], [1987b], Davis [1965, p. 72], Post [1943, p. 200], and others, mistakenly refer to this paper as “Turing, [1937].” The journal states that Turing’s manuscript was “Received 28 May, 1936–Read 12 November, 1936.” It appeared in two sections, the first section of pages 230–240 in Volume 42, Part 3, issued on November 30, 1936, and the second section of pages 241–265 in Volume 42, Part 4, issued December 23, 1936. No part of Turing’s paper appeared in 1937, but the two page minor correction [1937a] did.

Entscheidungsproblem, *Proc. London Math. Soc.* ser. 2 **42** (Parts 3 and 4) (1936) 230–265; [Turing, 1937a] A. M. Turing, A correction, *ibid.* **43** (1937), 544–546.

- [28] [Turing, 1937b]  
A. M. Turing, Computability and  $\lambda$ -definability, *J. Symbolic Logic*, **2** (1937), 153–163.
- [29] [Turing, 1939]  
A. M. Turing, Systems of logic based on ordinals, *Proc. London Math. Soc.* **45** Part 3 (1939), 161–228; reprinted in Davis [1965], 154–222.
- [30] [Zöllner-Thoenes, 2010]  
Frank Zöllner and Christof Thoenes, Benedikt Taschen, editor Michaelangelo Life and Work, Taschen GmbH, Köln, 2010. ISBN: 978-3-8365-2117-8.