

# Gödel's Reception of Turing's Model of Computability: The “Shift of Perception” in 1934

Juliette Kennedy<sup>(✉)</sup>

University of Helsinki, Helsinki, Finland  
juliette.kennedy@helsinki.fi

## 1 Introduction

The emergence of the mathematical concept of computability in the 1930s was marked by an interesting shift of perspective, from viewing the intuitive concept, “human calculability following a fixed routine” in terms of calculability in a logic, to viewing the concept as more adequately expressed by Turing’s model.<sup>1</sup> This happened in spite of, or in parallel with, *confluence*, as Gandy called it in his [1], namely the proven extensional equivalence of the models of computability which had been given prior to Turing’s model.

In this talk we consider this shift—one in which Gödel was a key figure—in relation to Gödel’s philosophical outlook subsequently. On the way we consider a question that Kripke has asked recently [2]: why did Gödel not see that the *Entscheidungsproblem* is an *immediate* consequence of the Completeness and Incompleteness Theorems? Kripke’s analysis depends upon viewing computability in terms of calculability in a logic. We thus suggest that Kripke’s own explanation for Gödel’s purported blindness to the fact of having solved what was arguably viewed as the single most important problem in logic remaining open at the time,<sup>2</sup> be complemented by the story of the difficulties logicians had with viewing computability in this sense.

In particular we will suggest that the inherent circularity which characterises such a conception, may have contributed to a sense of reluctance on the part of the Princeton group of logicians at the time to embrace it: for if computability is understood in terms of calculability in a logic, then it must be the case that not any logic will serve, rather the logic in question must be given effectively. But then, how to understand the concept “given effectively”, as applied to the logic in question?

What follows is an abbreviated history of the development of computability in the 1930s, from the point of view of this shift in perspective.<sup>3</sup>

---

Juliette Kennedy: I thank Liesbeth de Mol for the invitation to address a special session of CiE2017, and to contribute this extended abstract to its proceedings.

<sup>1</sup> Gandy quote. See below.

<sup>2</sup> See, e.g. Sieg [3], p. 387.

<sup>3</sup> Some of the text below is adapted from the author’s [4].

## 2 Isolating the Concept of “Computable Function”

Gödel defined the class of primitive recursive functions in his 1931 [5], while Church at the same time had developed the  $\lambda$ -calculus together with Kleene.

Church's first presentation of the  $\lambda$ -calculus in 1932 [6], which embeds that calculus in a *deductive formalism*, in the Hilbert/Bernays terminology, was found to be inconsistent in 1934. Two years later Church published a, in Gandy's words, logic-free definition of the  $\lambda$ -calculus, based on the primitives “function” and “iteration”:<sup>4</sup>

When it began to appear that the full system is inconsistent, Church spoke out on the significance of  $\lambda$ -definability, *abstracted from any formal system of logic*, as a notion of number theory.<sup>5</sup>

In the period between these two versions of the  $\lambda$ -calculus Church suggested his *thesis*, namely the suggestion to identify the  $\lambda$ -definable functions with those which are effectively computable.<sup>6</sup> The plausibility of the thesis became especially clear when Church, Kleene and Rosser established *confluence*, proving the equivalence of  $\lambda$ -definability with computability in the sense of the Herbrand-Gödel equational calculus, in 1935.<sup>7</sup>

Interestingly, Gödel was not persuaded of its adequacy, when he was told of Church's suggestion to view intuitive or effective computability in terms of  $\lambda$ -definability in early 1934, finding the proposal “thoroughly unsatisfactory.”<sup>8</sup> Church described Gödel's then suggestion to instead take an axiomatic approach to the problem, in a letter to Kleene:

His [Gödel's] only idea at the time was that it might be possible, in terms of effective calculability as an undefined notion, to state a set of axioms which would embody the generally accepted properties of this notion, and to do something on that basis. . . . At that time he did specifically raise the question of the connection between recursiveness in this new sense and effective calculability, but said he did not think that the two ideas could be satisfactorily identified “except heuristically.”<sup>9</sup>

Church may have been influenced by Gödel's negative view of the adequacy of  $\lambda$ -calculus, as in his lecture on his Thesis to the American Mathematical Society in 1935, Church used the Herbrand-Gödel equational calculus as a model of

<sup>4</sup> [1], Sect. 14.8. Following Gandy we use the term “effectively computable,” or just “effective,” to mean “intuitively computable.”

<sup>5</sup> See [7], which relies on Church's [8]. See also [9]. Emphasis the author's.

<sup>6</sup> The phrase “Church's Thesis” was coined by Kleene in 1943. See his [10].

<sup>7</sup> The proof of the equivalence developed in stages. See Davis, [11] and Sieg, [12].

<sup>8</sup> Church, letter to Kleene November 29, 1935. Quoted in Sieg, op. cit., and in Davis [11].

<sup>9</sup> Church, op.cit. In this talk and elsewhere we distinguish the axiomatic from the logical method, viewing the former as an informal notion, and the latter as involving a formalism.

effective computation, i.e. recursiveness in the “new sense,” rather than the  $\lambda$ -calculus.

In fact Church presented two approaches to computability in the AMS lectures and in his subsequent 1936 [13], based on the lectures: Firstly *algorithmic*, based still on what is now known as the untyped  $\lambda$ -calculus, i.e. the evaluation of the value  $fm$  of a function by the step-by-step application of an algorithm—and secondly *logical*, based on the idea of calculability in a logic:

And let us call a function  $F$  of one positive integer *calculable within* the logic if there exists an expression  $f$  in the logic such that  $f(\mu) = \nu$  is a theorem when and only when  $F(m) = n$  is true,  $\mu$  and  $\nu$  being the expressions which stand for the positive integers  $m$  and  $n$ .<sup>10</sup>

In order to view computability in terms of calculability in a logic, one must first restrict the class of formal systems representing those computable functions. As we noted, not just any formalism will serve! For Church this meant that the theorems of the formal system should be recursively enumerable.<sup>11</sup> A number-theoretic function is taken to be effective, then, if its values can be computed in such an effectively given formalism.

The argument appears to be circular.<sup>12</sup> In Hilbert and Bernays’ 1939 *Grundlagen der Mathematik II*, the computable functions are also presented in terms of a logical calculus, but here effectivity is now reduced to primitive recursion. In his [14], Sieg has this to say about Hilbert and Bernays’ improvement of Church’s system:

In this way they provided the mathematical underpinnings for . . . Church’s argument, but only *relative* to the recursiveness conditions: the crucial one requires the proof predicate of deductive formalisms, and thus the steps informal calculations, to be primitive recursive!

We now come to one of the topics of this note, the shift in perspective in 1934 noted by Gandy, writing: “. . . in 1934 the interest of the group shifted from systems of logic to the  $\lambda$ -calculus and certain mild extensions of it: the  $\lambda - \kappa$  and the  $\lambda - \delta$  calculi.”<sup>13</sup> Gandy does not speculate on the deeper reasons for this shift; and indeed it is interesting that there are so few direct references to

<sup>10</sup> [13], p. 357.

<sup>11</sup> In detail, recursive enumerability would be guaranteed here by the so-called step-by-step argument: if each step is recursive then  $f$  will be recursive; and three conditions: (i) each rule must be a recursive operation, (ii) the set of rules and axioms must be recursively enumerable, (iii) the relation between a positive integer and the expression which stands for it must be recursive. Conditions i-iii are Sieg’s formulation of Church’s conditions. See Sieg, [12], p. 165. For Gandy’s formulation of the step-by-step argument, see [1], p. 77.

<sup>12</sup> See also Sieg’s discussion of the “semi-circularity” of the step-by-step argument in his [12].

<sup>13</sup> [1], p. 71. Of course this shift presaged a much more dramatic shift of perspective in 1936, inaugurated by the work of Turing together with Post’s earlier work.

the circularity problem in the writings of logicians working on computability at the time. However it seems plausible that the circularity problem was at least partly responsible for this shift away from logical systems, especially given the initial inconsistency of the  $\lambda$ -calculus.

## 2.1 The “Scope Problem”: How General Are the Incompleteness Theorems?

We turn now to Gödel's role in these developments. Gödel was perhaps the first to suggest isolating the concept of effective computability.<sup>14</sup> His interest was driven, at least in part, by an important piece of unfinished business as far as the Incompleteness Theorems are concerned, in that it was not clear at the time to which formal systems the theorems apply, outside of the fragment of the type theory of *Principia Mathematica* he used.

In short, solving the scope problem depends on articulating a precise and adequate notion of effective computability, because the formal systems at issue in the Incompleteness Theorems, are to be given *effectively*.<sup>15</sup>

To this end, that is, with a wish to “make the incompleteness results less dependent on particular formalisms,”<sup>17</sup> Gödel introduced in 1934 the general recursive, or Herbrand-Gödel recursive functions. It is striking from the point of view of this note that he begins the section introducing those functions thus: “Now we turn to some considerations which for the present have nothing to do with a formal system.”, and defines the notion of “formal system” in the next section as consisting of “symbols and *mechanical rules* relating to them.”<sup>18</sup> Gödel seems here to have all the elements behind the Turing conception of computability in place. What is missing of course, is the model itself.

The Herbrand-Gödel calculus allows for forms of recursions that go beyond primitive recursion, however it was not clear to Gödel at the time, that the schema captured *all* recursions.<sup>19</sup>

Gödel gave, in the same presentation, a precise definition of the conditions a formal system must satisfy so that the incompleteness theorems apply to it. These included the restriction that:

<sup>14</sup> See Gandy's [1], p. 72.

<sup>15</sup> Gödel was careful not to claim complete generality for the Second Incompleteness Theorem in his 1931 paper:

For this [formalist JK] viewpoint presupposes only the existence of a consistency proof in which nothing but finitary means of proof is used, and it is conceivable that there exist finitary proofs that cannot be expressed in the formalism of  $P$  (or of  $M$  and  $A$ ).<sup>16</sup>

<sup>16</sup> [15], p. 195.  $P$  is a variant of *Principia Mathematica*.

<sup>17</sup> Sieg [16], p. 554.

<sup>18</sup> [15], p. 346 and 349, resp. Emphasis added.

<sup>19</sup> As Gödel would later write to Martin Davis, “. . . I was, at the time of these lectures, not at all convinced that my concept of recursion comprises all possible recursions.” Quoted in [15], p. 341.

Supposing the symbols and formulas to be numbered in a manner similar to that used for the particular system considered above, then the class of axioms and the relation of immediate consequence shall be [primitive JK] recursive.<sup>20</sup>

By 1935, Gödel’s reflections on computability in the higher order context began to point towards the possibility of a definitive notion of formal system. Nevertheless, an, in Gödel’s terminology now, *absolute* definition of effective computability was still missing at that point.<sup>21</sup>

### 3 Turing’s Analysis of Computability

In 1936, Turing gave a self-standing analysis of human effective computability and used it to solve the *Entscheidungsproblem*.

Rather than calculability in a logic, Turing analyzed effectivity informally but exactly, via the concept of a Turing Machine—a machine model of computability, consisting of a tape scanned by a reader, together with a set of simple instructions in the form of quadruples.<sup>23</sup>

We alluded to circularity in connection with approaches to computability that are centered on the idea of calculability in a logic.<sup>24</sup> The crucial point here is that Turing’s analysis does not require the specification of a logic *at all*.

The reaction to Turing’s work among the Princeton logicians was immediately positive. As Kleene would write in 1981, “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).”

For Gödel, as he would later explain to Hao Wang, Turing’s model of human effective calculability is, in some sense, perfect:

---

<sup>20</sup> [15], p. 361.

<sup>21</sup> As Gödel wrote to Kreisel in 1965:

That my [incompleteness] results were valid for all possible formal systems began to be plausible for me (that is since 1935) only because of the Remark printed on p. 83 of ‘The Undecidable’ . . . But I was completely convinced only by Turing’s paper.<sup>22</sup>

<sup>22</sup> Quoted in Sieg [17], in turn quoting from an unpublished manuscript of Odifreddi, p. 65.

<sup>23</sup> Or quintuples, in Turing’s original presentation.

<sup>24</sup> As Sieg puts it [14], “The work of Gödel, Church, Kleene, and Hilbert and Bernays had intimate historical connections and is still of deep interest. It explicated calculability of functions by *exactly one core notion*, namely calculability of their values in logical calculi via (a finite number of) elementary steps. But no one gave convincing and non-circular reasons for the proposed rigorous restrictions on the steps that are permitted in calculations.”.

The resulting definition of the concept of mechanical by the sharp concept of “performable by a Turing machine” is both correct and unique . . . *Moreover it is absolutely impossible that anybody who understands the question and knows Turing's definition should decide for a different concept.*<sup>25</sup>

And Turing's analysis led to the complete solution of the scope problem:

In consequence of later advances, in particular of the fact that, due to A. M. Turing's work, a precise and unquestionably adequate definition of the general concept of formal system can now be given, the existence of undecidable arithmetical propositions and the non-demonstrability of the consistency of a system in the same system can now be proved rigorously for every consistent formal system containing a certain amount of finitary number theory.

Turing's analysis thus settled definitively the adequacy question for computability. For Gödel, and for the logicians of the time, the Turing Machine was not just another in the list of acceptable notions of computability—it is the *grounding* of all of them.

We suggested that the issue of circularity was in the background of the shift away from a logical orientation to the problem of isolating and precisifying the notion of intuitive computability. Once again this is the problem of how to spell out the concept of effectivity of a logical system that embeds one's notion of computability, without invoking that very notion. How did the Turing model of computation solve this problem for Gödel? We explain it thus: Turing's model of computation allows for an autonomous logical perspective, because it is itself logic free.

What was the role of confluence? *Church's Thesis* identified effective calculability with  $\lambda$ -calculability, and then with Herbrand-Gödel calculability, after the equivalence between the two was established. Prior to Turing's work, the available confluence was seen as justifying adequacy in a weak sense only. Once one has a grounding example in hand this changes—confluence now plays an epistemologically important evidentiary role.

It is striking that Emil Post, who called for the return to meaning and truth in the opening of his [19], and the downplaying of what he called “postulational thinking,” aligns him both ideologically and mathematically with these developments. As it turns out, Post's recommendation to develop recursion theory mathematically, by stripping off the formalism with which the theory was encumbered, led to the formalism free development of recursion theory just along the lines he advocated. It also gave rise to the development of Post Systems, a model of computability very similar to Turing's.

The project of developing an autonomous logical perspective permeated Gödel's outlook from then on. Gödel alludes to it a number of times in his Princeton Bicentennial Lecture, in connection with finding absolute notions of

<sup>25</sup> Remark to Hao Wang, [18], p. 203. Emphasis added.

decidability and provability. One can also read the perspective into Gödel's overarching goal of attaining decidability in set theory—for how else to achieve decidability in set theory, except by remaining, as we have called it, formalism free?

## 4 Kripke and the *Entscheidungsproblem*

Kripke advocates a, as he calls it, logical orientation to the problem of isolating the notion of effective computability.

My main point is this: a computation is a special form of mathematical argument.<sup>26</sup>

He then asks, given that such an approach would have been very easily within reach of Gödel and other logicians working in the period immediately after the 1931 Incompleteness Theorems, why wasn't it noticed that a negative solution to the *Entscheidungsproblem* follows immediately?

Suppose we had taken derivability by a computation expressible in a first-order language as one's basic definition of computability. Then given the Gödel Completeness Theorem, any conventional formalism for first-order logic will be sufficient to formalise such derivability. . . . It will be a short and direct step to conclude the undecidability in this sense of the *Entscheidungsproblem*.<sup>27</sup>

Kripke gives the argument, which necessarily includes the notion of *validity*. He concludes, very reasonably in our view, that Gödel would have been reluctant to invoke a notion of truth in the proof, had he arrived at that proof. Indeed it is well known that Gödel's initial proof of the Incompleteness Theorem followed from the undefinability of truth together with the observation that first order provability is not only definable, but r.e.<sup>28</sup>

We suggest in this talk that the evidence in the record regarding Gödel's response to the Turing model, together with the developments leading up to it (as recounted here), might also be taken into account in explaining Gödel's so-called blindness. It just might be that for Gödel, *grounding* the notion of computability required an altogether new *mathematical* idea, and not a logical one. We saw how close he came to Turing's conception in 1934, just before he turned his attention to the continuum problem in set theory.

## References

1. Gandy, R.: The confluence of ideas in 1936. In: The universal turing machine: A half-century survey, pp. 55–111. Oxford Science Publications, Oxford University Press, New York (1988)

<sup>26</sup> [2], p. 80.

<sup>27</sup> *ibid*, p. 85.

<sup>28</sup> See Gödel's letter to van Heijenoort of February 22, 1964, in [20].

2. Kripke, S.: The church-turing "Thesis" as a special corollary of gödel's completeness theorem. In: Copeland, B.J., Posy, C.J., Shagrir, O. (eds.) *Computability: Gödel, Church, and Beyond*. MIT Press, Cambridge (2013)
3. Sieg, W.: *Hilbert's Programs and Beyond*. Oxford University Press, Oxford (2013)
4. Kennedy, J.: Turing, Gödel and the "Bright Abyss". In: *Philosophical Explorations of the Legacy of Alan Turing*, vol. 324 of *Boston Studies in Philosophy* (Springer)
5. Gödel, K.: Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatsh. Math. Phys.* **38**, 173–198 (1931)
6. Church, A.: A set of postulates for the foundation of logic I, II. *Ann. Math.* **33**(2), 346–366 (1932)
7. Kleene, S.C., Rosser, J.B.: The inconsistency of certain formal logics. *Ann. Math.* **36**(2), 630–636 (1935)
8. Church, A.: The richard paradox. *Amer. Math. Monthly* **41**, 356–361 (1934)
9. Kleene, S.C.: Origins of recursive function theory. *Ann. Hist. Comput.* **3**, 52–67 (1981)
10. Kleene, S.C.: Recursive predicates and quantifiers. *Trans. Am. Math. Soc.* **53**, 41–73 (1943)
11. Davis, M.: Why Gödel didn't have church's thesis. *Inf. Control* **54**, 3–24 (1982)
12. Sieg, W.: Step by recursive step: Church's analysis of effective calculability. *Bull. Symbolic Log.* **3**, 154–180 (1997)
13. Church, A.: A note on the Entscheidungsproblem. *J. Symbolic Log.* **1**(1), 40–41 (1936). (Correction 1:101–102)
14. Sieg, W.: Church without dogma: Axioms for computability. In: Cooper, S.B., Löwe, B., Sorbi, A. (eds.) *New Computational Paradigms*, pp. 139–152. Springer, New York (2008)
15. Gödel, K.: *Collected works*. Vol. I. The University Press, New York (1986). Publications 1929–1936, Edited and with a preface by Solomon Feferman. Clarendon Press, Oxford
16. Sieg, W.: On computability. In: *Philosophy of Mathematics. Handbook of the Philosophy of Science*, pp. 535–630. Elsevier/North-Holland, Amsterdam (2009)
17. Sieg, W.: Gödel on computability. *Philos. Math.* **14**(3), 189–207 (2006)
18. Wang, H.: *A Logical Journey. Representation and Mind*. MIT Press, Cambridge (1996)
19. Post, E.L.: Recursively enumerable sets of positive integers and their decision problems. *Bull. Am. Math. Soc.* **50**, 284–316 (1944)
20. Gödel, K.: *Collected Works*. V: Correspondence H-Z. Oxford University Press, Oxford (2003). Feferman, S., et al. (eds.)