Incompleteness and undecidability were not proven by Gödel and Turing

Sebastian Wingbermühle, independent scientist, Stockholm, Sweden
Mail: independentscientiststockholm@gmail.com

1 Abstract

With incompleteness and undecidability, Gödel and Turing, respectively, introduced two major limitations of mathematics as a logical system. Both their proofs rely on a logical pattern combining self-reference with negation, requiring a statement or machine to have and not have a certain property at the same time. Both proofs correctly arrive at a logical contradiction from this inital premise. But instead of concluding that the combination of self-reference and negation, which was introduced in Russell's paradox before, led to the construction of an ill-defined statement or machine, Gödel used the fact that the logical system of mathematics cannot derive his ill-defined statement to claim incompleteness. Similarly, Turing used the fact that his ill-defined machine cannot exist to conclude that a machine included in the ill-defined machine cannot exist to claim undecidability. In both cases, the proofs thus draw conclusions about a property of a statement or a machine without knowing the relevant pieces of information about the respective statement or machine. However, if the combination of self-reference and negation is removed and self-reference is combined with an identity operation in the statement or machine constructed in the proof, the proofs suggested by Gödel and Turing cannot establish incompleteness and undecidability but rather allow no conclusion about the relevant statement or the machine. Therefore, the works by Gödel and Turing do not prove incompleteness and undecidability, respectively.

2 Introduction

As Russell's paradox, Gödel's proof of incompleteness and Turing's proof of undecidability all rely on the same logical figure of combining self-reference with negation, this logical figure and the way that the above authors apply it in their work will first be illustrated with the help of a less complex example.

The numbers $\{1,2,3,4,5\}$ form a set **S** that is a complete solution to the task to find all natural numbers in $[1,5]$. Let us, however, assume that this set lacks a natural number x. This number x is defined as 2 unless its value is 2. In this definition, we combine self-reference (the value of x depends on the value of x) with negation (the value of x must NOT be the same in the first and second half of the definition). Using this definition of x, we can easily arrive at a contradiction because x cannot be a number and have two different values at the same time (2 and not 2) since a number is defined as a mathematical quantity having exactly one value. Instead of being a number, x is an ill-defined superposition of two numbers that cannot exist. Following Russell's, Gödel's and Turing's argument, it can now be concluded that it is impossible to provide a complete solution to the task to find all natural numbers in $[1,5]$ because x is missing in $\mathbf{S}$ and x cannot be found. However, it was an unproven assumption all the time that x is missing in $\mathbf{S}$, and the above contradiction just showed that x is not a number and must not be an element of $\mathbf{S}$. With the help of x, we cannot draw any conclusion on whether $\mathbf{S}$ is a complete solution to the task to find all natural numbers in $[1,5]$.

In the following, I will show that the set identified by Russell, the statement proposed by Gödel and the machine analysed by Turing are no set, statement or machine but ill-defined superpositions of two sets, statements or machines like the number x discussed above. Therefore, despite arriving at the correct logical contradiction, their theorems and proofs do not establish any limitation of the existing mathematical theory or of the logical framework of mathematics itself. Instead, they just prove that combining self-reference with non-identity operators results in ill-defined quantities and objects that must not exist and that the logical framework of mathematics, therefore, must not be able to form, prove or take decisions on.

3 Russell's paradox

The key idea of Russell's paradox consists in applying a combination of self-reference and negation to set theory. To this end, Russell proposed a set $\mathbf{R}$ defined as the set of all sets that do not contain themselves: [1]

$$\mathbf{R} = \{x : x \notin x\} \tag{1}$$

This definition of $\mathbf{R}$ eventually results in a contradiction because $\mathbf{R}$ has to be a member of $\mathbf{R}$ if and only if it is not a member of $\mathbf{R}$. Consequently, the set $\mathbf{R}$ cannot be formed.

However, the above contradiction is only caused by the fact that $\mathbf{R}$ cannot be a set containing $\mathbf{R}$ and a set not containing $\mathbf{R}$ at the same time because that is an ill-defined superposition of two distinct sets: the set $\mathbf{R}_1$ with the same remaining elements as $\mathbf{R}$ but without $\mathbf{R}$ as element and the set $\mathbf{R}_2$ with the same remaining elements as $\mathbf{R}$ and including $\mathbf{R}$ as element.

If the quantity formed by all possible sets is denoted as $\mathbf{1}$, $\mathbf{R}_1$ and $\mathbf{R}_2$ can be rigorously defined as:

$$\mathbf{R}_1 = \mathbf{1} - \{x : x \in x\} \tag{2}$$

and

$$\mathbf{R}_2 = \mathbf{R}_1 \cup \mathbf{R}_1. \tag{3}$$

The logical contradiction observed for $\mathbf{R}$ does not arise with the set of all sets that contain themselves used in the definition of $\mathbf{R}_1$ because self-reference is no longer combined with negation.

Thus, there is no proof of a shortcoming of set theory, but Russell's paradox just illustrates that combining self-reference with negation leads to ill-defined superpositions of sets that set theory must refuse to form.

4 Gödel's proof of incompleteness

Gödel's incompleteness theorem is centred around the following definition of a class K of natural numbers: [2]

$$n \in K \ \ if\ and\ only\ if \ \ \neg \text{Proof}[R(n);n] \tag{4}$$

or equivalently

$$n \notin K \ \ if\ and\ only\ if \ \ \text{Proof}[R(n);n]. \tag{5}$$

Here, n denotes a natural number, and the statement $[R(n);n]$ consists of a formula with one free variable $R(n)$ in which the variable is replaced with the sign assigned to n. For his theorem, Gödel developed a mapping that assigns a natural number, the above n, to all basic values, logical operations and axioms as well as to all possible formulae with one variable of type natural number in the mathematical, logical system of *Principia Mathematica*, for which he aimed to prove incompleteness. In terms of content, $[R(n);n]$ states that n belongs to K, which allows us to rewrite eq. 4 as:

$$n \in K \ \ if\ and\ only\ if \ \ \neg \text{Proof}[n \in K]. \tag{6}$$

Thus, the natural number n is a member of the class K if and only if there is no proof of it. This definition, like Russell's paradox, combines self-reference with a negation operator ($\neg$Proof) and, as a definition, is not derived but just set by Gödel.

Gödel now argues that the statement $[R(n);n]$ cannot be proven with the help of the following logic:

1. Assume the statement $[R(n);n]$ can be proven. Consequently, $n \in K$ such that $[R(n);n]$ cannot be proven (eq. 4), which leads to a contradiction.

2. Assume the negation of $[R(n);n]$ can be proven. Consequently, $n \notin K$ such that $[R(n);n]$ can be proven (eq. 5). As a result, $[R(n);n]$ and its negation can be proven at the same time, which again leads to a contradiction.

In the above logical argument, Gödel relies on the fact that a statement that can be proven has to be true due to the consistency of the logical system, whereas a true statement does not need to be provable. However, Gödel himself makes the point that $[R(n);n]$ can be proven if and only if it is not true because $n \notin K$ (eq. 5). Therefore, $[R(n);n]$ forces a proof to deliver a false result, which is against the requirement of consistency that Gödel himself stated and used such that $[R(n);n]$ cannot be a valid statement in the context of the class definition provided in eq. 4. To be a valid, a mathematical statement has to be either true or false, i.e. two scenarios have to be considered for $[R(n);n]$:

1. Assume $[R(n);n]$ is false. Consequently, its negation is true such that $[R(n);n]$ can be proven (eq. 5) and has to be true.

2. Assuming that $[R(n);n]$ is true immediately leads to a contradiction because it violates the consistency of the logical system that it is intended to be a part of and has thus to be false.

Consequently, $[R(n);n]$ is always true and false at the same time such that it is not a statement but an ill-defined superposition of two statements, and Gödel's class K is ill-defined, like the set proposed by Russell in his paradox. Therefore, the logical system of *Principia Mathematica* can be complete without proving the Gödel sentence.

However, all logical problems with Gödel's proof are immediately gone if self-reference and negation are no longer combined but self-reference is used together with a unity operator, e.g. if the class K is defined as

$$n \in K \;\; if\ and\ only\ if \;\; \mathrm{Proof}[R(n);n] \tag{7}$$

or equivalently

$$n \notin K \;\; if\ and\ only\ if \;\; \neg\mathrm{Proof}[R(n);n] \tag{8}$$

and if the relation Q in his detailed formal proof is defined as

$$Q(x,y) = xB_x[Sb(y^{19}_{Z(y)})]. \tag{9}$$

As a result, there are no more contradictions in Gödel's logic:

1. Assume the statement $[R(n);n]$ can be proven. Consequently, $n \in K$ such that $[R(n);n]$ can be proven (eq. 7).

2. Assume the negation of $[R(n);n]$ can be proven. Consequently, $n \notin K$ such that $[R(n);n]$ cannot be proven (eq. 8).

However, without the negation operator, Gödel's logic has been reduced to circular statements that do not provide any proof of whether $[R(n);n]$ can be proven. Instead, it can be seen that being provable is a property of a statement, otherwise depending only on the operations and axioms available in the logical system, not on the definition of a class of numbers.

5 Turing's proof of undecidability

To address the question of decidability, Turing represented mathematical processes with the help of computing machines that compute numbers and print them on tape in binary representation, i.e., as a sequence of 0 and 1. [3] On the tapes, F-squares containing information about the computed number alternate with E-squares that can be filled with instructions for the computing machine. In this setup, the state of a machine can be fully defined by its machine configuration, the square that it is scanning and the sequence of symbols that have been printed on the tape.

The work performed by such a computing machine is encoded in tables listing the initial machine configuration, the symbol scanned on the tape, the operations performed by the machine and the final machine configuration. A compact representation of this table as a string of letters is called the standard description (SD) of the machine, the integer number onto which the SD is matched is called the description number (DN) of the machine.

When actually computing a number, a computing machine may come to a halt after writing a finite number of symbols encoding numbers and be unable to write further symbols. In this case, it is circular and the DN encoding its operations is unsatisfactory. In contrast, a computing machine that can complete the computation without coming to a halt is circle-free, and its DN is satisfactory. Knowing whether a machine is circular/circle-free is equivalent to knowing whether a mathematical process converges to a result.

While most computing machines are intended to compute a particular number, a universal computing machine can read in the SD or DN of a computing machine from the E-squares of a tape and behave exactly like the computing machine provided as input. However, a universal computing machine has no intrinsic ability to compute a number such that it cannot work if it is not provided the SD or DN of another computing machine or if it is provided the SD or DN of another universal computing machine.

In this setup, answering the question of whether mathematics is decidable is equivalent to answering the question of whether there exists a machine that is able to compute the sequence of all possible satisfactory DN. To this end, Turing proposes a machine H consisting of a universal machine U followed by a machine D. H takes an integer number n as input, interprets it as the DN of another computing machine and D decides whether it is satisfactory or not. If n is satisfactory, H calculates all $R(n-1)$ satisfactory DN found before n, extends the sequence with n and writes n to tape. If n is not satisfactory, which D can determine in a finite number of steps, H just continues with the next integer number. Thus, H is circle-free because it can either extend the sequence of satisfactory DN within a finite number of steps if provided the DN of a circle-free machine as input, or it can identify the DN as unsatisfactory in a finite number of steps and move on to the next input integer number.

However, if H is given its own DN k as input, a contradiction occurs. D cannot label k as unsatisfactory because H has just been proven to be circle-free. On the other hand, U cannot work if provided k as input such that the $R(k)$-th member of the sequence of satisfactory numbers cannot be computed by H and k cannot be considered satisfactory. From this, Turing concluded that D cannot exist.

The moment that H is given k as input, Turing uses the same pattern of self-reference combined with negation as Russell and Gödel. The self-reference is introduced by the fact that k is the DN of H and H has to examine itself. The negation is introduced by D's ability to identify an unsatisfactory DN in a finite number of steps and then allow H to move on, thus rendering H a machine that, by definition, is circular and circle-free at the same time if it is provided its own DN as input.

1. Assume that H is circular. Then, the universal machine U that is a part of H is circular, too. However, D can classify k as unsatisfactory in a finite number of steps, allowing H to move on to the next input integer number and rendering it circle-free.

2. Assuming that H is circle-free immediately leads to a contradiction because U is circular

if provided the DN of another universal machine as input.

Consequently, by definition, H provided with k as input is circular and circle-free at the same time. As a computing machine has to be either circular or circle-free, the combination of self-reference and negation has again led to an ill-defined superposition of two computing machines. Therefore, the contradiction at which Turing arrived proves that H cannot exist.

But it does not allow to draw any conclusions on whether D can exist. This becomes obvious if the negation is removed from the definition of H, e.g. H is constructed in such a way that D can still classify a DN as satisfactory or unsatisfactory in a finite time but it can no longer render H circle-free if an unsatisfactory DN or the DN of a universal machine is provided as input and U becomes circular. This way, H is properly defined and exhibits the behaviour of a universal machine, but the contradiction that Turing used to conclude that D cannot exist fades away: H, when given k as input, is now just a circular machine. For this reason, Turing's proof does not allow any conclusion on whether mathematics is decidable or not.

6 Conclusions

In the above sections, it has been shown that the works by Gödel and Turing cannot establish incompleteness and undecidability. However, incompleteness and undecidability may indeed be actual limitations of the logical system of mathematics. Moreover, it is beyond the scope of this work to analyse all follow-up publications providing alternative proofs of incompleteness and undecidability or demonstrating these limitations for example systems, e.g. [4] or [5]. Gödel's and Turing's theorems are the basis of several active scientific fields like the physical understanding of life and chaos [6] and have been used to predict the limitations of mathematical tools in other scientific disciplines [7]. The findings of this work suggest to carefully revisit such conclusions derived from Gödel's and Turing's statements and not to take them for granted before replacement proofs for Gödel's and Turing's theorems avoiding the pattern of self-reference in combination with negation have been found.

References

[1] Irvine, A. D.; Deutsch; H. ”Russell’s paradox”. *The Standford Encyclopedia of Philosophy*. Edward N. Zalta (ed.), **2021**.

[2] Gödel, K. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatsh. f. Mathematik und Physik* **1931**, *38*, 173–198. doi: 10.1007/BF01700692.

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

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

[5] Wang, H. Proving Theorems by Pattern Recognition – II. *BSTJ* **1961**, *40*, 1–41.

[6] Prokopenko, M.; Harré, M.; Lizier, J.; Boschetti, F.; Peppas, P.; Kauffman, S. Self-referential basis of undecidable dynamics: From the Liar paradox and the halting problem to the edge of chaos. *Phys. Life Rev.* **2019**, *31*, 134–156. doi: 10.1016/j.plrev.2018.12.003.

[7] Cubitt, T. S.; Perez-Garcia, D.; Wolf, M. M. Undecidability of the spectral gap. *Nature* **2015**, *528*, 207–211. doi: 10.1038/nature16059.