THE HALTING PROBLEM AND OTHER UNDECIDABLE PROBLEMS

Ivan Winters

1 Introduction

The halting problem stands as one of the most foundational results in the theory of computation, dating back to Alan Turing's groundbreaking 1936 paper. Before Turing's formal treatment, various attempts to understand the nature and limits of mechanical computation were underway, notably through the work of Church and Gödel. These efforts were aimed at addressing key questions about what can and cannot be solved by mechanical means—problems that engaged logicians, mathematicians, and philosophers during the early 20th century.

The problem gained its canonical form when Turing introduced the abstract model of a computing machine—now known as the Turing machine—to provide a mathematically precise framework for discussing the notion of computability. By characterizing algorithms as specific sets of instructions executable by a Turing machine, Turing was able to show that certain questions could not be answered by any such mechanical procedure. The halting problem, which asks whether a given Turing machine halts on a particular input, was among the earliest and clearest examples of an insoluble decision problem. Its undecidability demonstrated that there are inherent limitations to what can be effectively computed.

The significance of this finding is underscored by its reverberations across mathematics, logic, and computer science. It revealed fundamental constraints on mechanical reasoning: no universal algorithmic process can exist to determine, in every case, if a computation will eventually terminate. This insight not only closed the door on Hilbert's dream of a complete and decidable system of mathematics but also laid the groundwork for understanding complexity, computability, and the very nature of algorithmic processes.

In modern treatments, including N. J. Cutland's *Computability: An Introduction to Recursive Function Theory*, the halting problem is introduced as a central example of an undecidable problem. Cutland's exposition provides a rigorous foundation for understanding partial recursive functions, their indices, and the sets W_x that represent their halting behavior. By carefully constructing the halting set and employing diagonalization arguments, Cutland presents the result in a structured manner, guiding readers through the intricate interplay of enumerability and decidability.

2 Preliminary Definitions and Concepts

In order to understand the halting problem and other undecidable problems, we first need some foundational concepts. We will consider partial recursive functions (partial computable functions), their indices, and the sets W_x that represent their halting behaviors. Recall that for a standard enumeration $\{\varphi_x\}$ of all partial recursive functions, we define

$$W_x = \{n \mid \varphi_x(n) \text{ is defined (halts)}\}.$$

A key technique in demonstrating undecidability results is the use of a *diagonal argument*. This technique allows us to construct functions or sets that differ from every element in a certain enumerated family at a critical point. The following theorem exemplifies this method, showing how we can derive an undecidable problem through diagonalization.

Theorem (A Fundamental Undecidability Result) '$x \in W_x$' (or, equivalently, '$\varphi_x(x)$ is defined', or '$P_x(x)\downarrow$', or '$\psi(U(x,x))\downarrow$') is undecidable.

Proof. The characteristic function of this problem is given by

$$f(x) = \begin{cases} 1 & \text{if } x \in W_x \\ 0 & \text{if } x \notin W_x. \end{cases}$$

Suppose that f is computable; we shall obtain a contradiction. Specifically, we make a diagonal construction of a computable function g such that $\text{Dom}(g) \neq W_x (= \text{Dom}(\varphi_x))$ for every x; this is obviously contradictory.

The diagonal motto tells us to ensure that $\text{Dom}(g)$ differs from W_x at x; so we aim to make

$$x \in \text{Dom}(g) \iff x \notin W_x.$$

Let us define g, then, by

$$g(x) = \begin{cases} 0 & \text{if } x \notin W_x \text{ (i.e. if } f(x) = 0) \\ \text{undefined} & \text{if } x \in W_x \text{ (i.e. if } f(x) = 1). \end{cases}$$

Since f is computable, so is g (by Church's thesis). But then, if g is computable, let m be such that $g = \varphi_m$. We have

$$m \in \text{Dom}(g) \iff m \notin W_m,$$

a contradiction.

Thus f is not computable, and so the problem '$x \in W_x$' is undecidable.

□

This fundamental undecidability result serves as a cornerstone. It not only provides a direct example of an undecidable set but also is used in proving many other classic undecidability results. Notice that it does not claim we cannot determine whether $\varphi_a(a)$ is defined for some specific number a. Sometimes this can be easy (e.g., if we know the function is total). The theorem asserts that there is no single general method—no algorithm—that works for every x.

3 The Halting Problem

With the diagonal technique and the concept of W_x in hand, we now turn to the halting problem itself. The halting problem asks, given an index x and an input y, whether $\varphi_x(y)$ eventually halts.

Theorem (The Halting Problem) The problem '$\varphi_x(y)$ is defined' (or, equivalently, '$P_x(y) \downarrow$' or '$y \in W_x$') is undecidable.

Proof. Arguing informally, if we had a decider for '$\varphi_x(y)$ is defined' for all x, y, then by setting $y = x$, we would solve '$x \in W_x$', which we have just shown to be undecidable.

Formally, let g be the characteristic function for '$\varphi_x(y)$ is defined'; i.e.

$$g(x, y) = \begin{cases} 1 & \text{if } \varphi_x(y) \text{ is defined} \\ 0 & \text{if } \varphi_x(y) \text{ is not defined.} \end{cases}$$

If g is computable, then so is

$$f(x) = g(x, x),$$

but f is the characteristic function of '$x \in W_x$', which we know is not computable. Thus no such g can exist, and the halting problem is undecidable. $\square$

This result, often called the unsolvability of the halting problem, was one of the earliest and most famous demonstrations of the inherent limitations of mechanical computation. It shows that no algorithmic method exists to universally determine whether an arbitrary program on an arbitrary input will halt.

With the halting problem established as undecidable, we now understand a fundamental boundary: there are limits to what machines can determine about their own behavior and the behavior of other machines. This sets the stage for many other undecidable problems.

4 Other Undecidable Problems

Having established the centrality and intractability of the halting problem, we turn now to several other classical examples of undecidable problems. These results underscore that the halting problem is not unique in its intractability; rather, it is emblematic of a broad class of decision problems that defy algorithmic solution.

4.1 Rice's Theorem

Rice's theorem encapsulates a broad range of undecidability results concerning semantic properties of partial computable functions. Informally, it states that any non-trivial property of the function computed by a program is undecidable. The textbook excerpt (adapted from Cutland) presents a form of Rice's theorem as follows:

Theorem (Rice's Theorem) Suppose that $\mathcal{B} \in \mathcal{C}_1$ and $\mathcal{B} \neq \emptyset, \mathcal{C}_1$. Then the problem "$\varphi_x \in \mathcal{B}$" is undecidable.

Proof. From the algebra of decidability (Theorem 2-4.7) we know that "$\varphi_x \in \mathcal{C}_1 \setminus \mathcal{B}$" is decidable if and only if "$\varphi_x \in \mathcal{C}_1 \setminus \mathcal{B}^\perp$" is decidable. We may assume without loss of generality that the function $f_\emptyset$ that is nowhere defined does not belong to $\mathcal{B}$. (If not, one can prove the result for $\mathcal{C}_1 \setminus \mathcal{B}$.)

Consider a function $g \in \mathcal{B}$. Define

$$f(x,y) = \begin{cases} g(y) & \text{if } x \in W_x \\ \text{undefined} & \text{if } x \notin W_x. \end{cases}$$

By the s–m–n theorem, there is a total computable function $k(x)$ such that $f(x,y) \Rightarrow \varphi_{k(x)}(y)$.

Thus:

$$x \in W_x \implies \varphi_{k(x)} = g \in \mathcal{B},$$
$$x \notin W_x \implies \varphi_{k(x)} = f_\emptyset \notin \mathcal{B}.$$

We have reduced the problem "$x \in W_x$" to the problem "$\varphi_x \in \mathcal{B}$" using the computable function k. Since "$x \in W_x$" is undecidable, it follows that "$\varphi_x \in \mathcal{B}$" is also undecidable. □

4.2 The Word Problem for Groups

The word problem for groups is another fundamental example of an undecidable problem, particularly in group theory. Given a group G generated by a set of elements $S = \{g_1, g_2, g_3, \ldots\}$, every element of G can be represented as a "word" formed from these generators and their inverses. The word problem asks: given a word w on S, does w represent the identity element of G?

The Word Problem for Groups Suppose that G is a group with identity element 1, and that G is generated by a set of elements $S = \{g_1, g_2, g_3, \ldots\} \subseteq G$. A word on S is any expression such as

$$g_2^{-1} g_3^6 g_1^8 g_2^8 g_3^5$$

involving the elements of S and the group operations. Each word represents an element of G, and G being generated by S means that every element of G is represented by some word on S.

The *word problem for G (relative to S)* is the decision problem of determining whether an arbitrary word w on S represents the identity element 1. While some groups (notably finite ones) have decidable word problems, it was a major mathematical achievement when Novikov (1955) and Boone (1957) showed that there exist finitely presented groups with an undecidable word problem. This result, known as the Novikov–Boone Theorem, was a landmark in illustrating that undecidability extends well beyond number-theoretic and logical questions into algebraic structures.

Group theory and modern algebra, in general, are replete with problems that are either known to be undecidable or suspected of being so, with the word problem serving as a central example.

4.3 Diophantine Equations

Diophantine equations—polynomial equations with integer coefficients—form another rich source of undecidable problems. Consider a polynomial

$$p(x_1, x_2, \ldots, x_n)$$

with integer coefficients. The equation

$$p(x_1, x_2, \ldots, x_n) = 0$$

is called a *diophantine equation* if we seek integer solutions. Not every diophantine equation has a solution; for example, the equation

$$x^2 - 2 = 0$$

has no integer solutions.

Hilbert's tenth problem, posed in 1900, asked whether there exists an effective procedure to determine, for any given diophantine equation, whether it has an integer solution. In 1970, Y. Matiyasevich proved that no such procedure exists. His work built on earlier contributions by M. Davis, J. Robinson, and H. Putnam, thereby demonstrating that the solvability of general diophantine equations is undecidable.

Matiyasevich's theorem is often considered one of the crowning achievements in the theory of decidability and number theory. Further details and the full depth of Matiyasevich's result are discussed in references such as Davis [1973], Manin [1977], and Bell & Machover [1977].

5 Conclusion

The halting problem stands as a paradigmatic example of an undecidable problem, illustrating that there are fundamental limits to what can be determined algorithmically. This result, originally proven by Turing, directly undermined the notion that every mathematical question could be settled by a systematic, mechanical procedure. Through the lens of partial recursive functions and the sets W_x, we have seen how diagonalization arguments yield intrinsic barriers to decidability. Once the halting problem is understood, it serves as a stepping stone to uncovering a whole landscape of undecidable problems.

Rice's theorem demonstrates the breadth of undecidability, showing that any nontrivial property of the function computed by a Turing machine is not decidable. This underscores the idea that the halting problem is not a mere curiosity; rather, it is a representative of a deep and pervasive phenomenon in computability. Similarly, the word problem in group theory and the solvability of Diophantine equations are cornerstones in other fields—algebra and number theory—further illustrating that undecidability is not restricted to a narrow technical domain. Instead, it is a fundamental aspect of formal systems across mathematics and theoretical computer science.

In essence, these undecidable problems reveal a tapestry of limitations on algorithmic reasoning. By studying them, we gain insight into the nature of computation, the boundaries of algorithmic methods, and the interplay between logic, number theory, algebra, and other branches of mathematics.

References

- Cutland, N.J., *Computability: An Introduction to Recursive Function Theory.* Cambridge University Press, 1980.
- Davis, M., *Hilbert's Tenth Problem Is Unsolvable*, American Mathematical Monthly, 80(3):233–269, 1973.