Hilbert's 10^{th} Problem

PHIL 152 Final Paper

Olivia Lee

June 8, 2023

1 Introducing Hilbert's 10^{th} Problem

Let us set the historical context before Turing's seminal work introducing the concept of Turing Machines. In the 17^{th} century, Gottfried Leibniz successfully created one of the first mechanical calculating machines, which led him to postulate a machine that could determine the truth values of mathematical statements, which would require one to discover a formal language with which to create such a machine. At the 1900 International Congress of Mathematicians, David Hilbert proposed 23 unsolved problems to advance the study of mathematics and determine "what methods, what new facts will the new century reveal in the vast and rich field of mathematical thought?", of which the 10^{th} problem will be the focus of this paper:

> *10. Determination of the solvability of a Diophantine equation. Given a Diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: to devise a process according to which it can be determined by a finite number of operations whether the equation is solvable in rational integers.*

An alternative expression of the problem is as follows:

> *10. Determination of the solvability of a Diophantine equation. Find an algorithm that decides, given a multivariate polynomial equation $f(x_1, \cdots, x_n) = 0$ with coefficients in the ring $\mathbb{Z}$ of integers, whether there is a solution with $x_1, \cdot x_n \in \mathbb{Z}$.*

Hilbert wanted to investigate the potential of automating the decidability of Diophantine solvability. Hilbert (1928) [1] further postulated the creation of "an algorithm to decide whether a given statement is provable from the axioms using the rules of logic". This is known as the *Entscheidungsproblem.* Thus, Hilbert's 10^{th} Problem about Diophantine equations was broadened to a more general question about mathematical statements in general: is there a universally valid algorithm that can tell us if any algorithm will terminate? The answer to his question required a more precise definition of 'algorithm' and 'computation', which did not exist in 1900. Alonzo Church (1935-6) gave one of the earliest definition of effective computability based on λ-calculus, showing that there is no algorithm to decide the equivalence of two given λ-calculus expressions.

2 Turing's Work

After Church's proof using λ-calculus, Turing's seminal paper (1936-7) provides an alternative response to Hilbert's question, by conceptualizing the Turing Machine. A Turing Machine can be described as a triple $\langle n, m, \delta \rangle$ where $n, m \in \mathbb{N}$ represent the number of states and number of symbols respectively, and δ is a partial function from $\{0, \cdots, n-1\} \times \{0, \cdots, m-1\}$ to $\{0, \cdots, m-1\} \times \{0, \cdots, n-1\} \times \{l, r\}$ representing the instruction table mapping the current state and current symbol to the next symbol, next symbol, and direction to move the tape head (left or right). A Turing Machine consists of a finite symbolic alphabet (including a 'blank' symbol), finitely many sates (including a designated 'start' state), a two-way infinite tape with discrete cells (meaning as many as is needed for computation; any halting computation uses a finite subset), and a finite list of instructions with each of the form "if in state i with symbol j, write symbol k, go to state l, and move the tape head left or right".

Turing's proof idea was motivated by Gödel's (1931) [2] invention of numbering to logical formulas in order to reduce logic to arithmetic so as to prove his incompleteness theorem. Using the concept of the Turing Machine, Turing demonstrated that the halting problem is not computable (or decidable), which decides whether a given Turing Machine halts or not. A sketch of the proof is as follows: fix an encoding of programs as natural numbers and identify programs with their associated integers (so that Turing Machines can be enumerated as $\{M_1, M_2, M_3, \cdots\}$). Now, assume for the sake of contradiction the existence of a Turing Machine $H = M_i$ that decides the halting problem, i.e. returns 1 if and only if a program p halts on input n. Using this program, we can build a new Turing Machine M_j with the following property: for any n, M_j halts on input n if and only if program n does not halt on input n. Set n to be the encoding of Turing Machine M_j, and we reach a contradiction: M_j halts on input M_j if and only if M_j does not halt on input M_j. Therefore, the assumption of the existence of the decider for the halting problem, must be false. Therefore, no Turing Machine exists that decides the halting problem, i.e., it is uncomputable.

The uncomputability of the Halting Problem was Turing's negative answer to Hilbert's 10^{th} Problem. Further, Turing showed that Turing Machines and λ-calculus proposed by Church are equivalent models of computation. That is, a function f is Turing-computable if and only if it is representable in λ-calculus. This equivalence led to the Church-Turing thesis, which states that a function is realistically computable if and only if it is computable by a Turing Machine. Intuitively, this asserts that an algorithm is one which can be computed using a Turing Machine, i.e. is a Turing Machine algorithm equivalent to a finite-length computer program, where the computer is assumed to have unlimited memory.

3 Matiyasevich's Solution

3.1 Recursively Enumerable and Recursive Sets

To preface Matiyasevich's Solution to Hilbert's 10^{th} Problem, we define *computably enumerable* and *computable* sets, and an immediate consequence of their definitions, as follows:

Definition 1: A set $\mathsf{Q} \subseteq \mathbb{Z}$ is *computably enumerable* (i.e. *recursively enumerable listable*) if there is an algorithm that prints the elements of Q when left running forever (in any order and with repetitions permitted).

Definition 2: A set $\mathsf{Q} \subseteq \mathbb{Z}$ is *computable* (i.e. *recursive* or *decidable*) if there is an algorithm that decides membership in Q. In other words, there is an algorithm that takes as input an integer n and returns true if $n \in Q$ and false if $a \notin Q$.

Theorem 1: A set S is computable if and only if S and its complement S' are both computably enumerable.

Turing's proof that the Halting Problem is undecidable therefore has an important consequence:

Corollary 1: There exists a recursively enumerable set that is not recursive.

3.2 Davis-Putnam-Robinson-Matiyasevich's Proof

Definition 3: A subset $\mathsf{Q} \subseteq \mathbb{Z}^k$ is *Diophantine* if there exists a polynomial $f(x_1, \cdots, x_k, y_1, \cdots, y_m)$ with integer coefficients such that

$$\mathsf{Q} = \{\vec{x} \in \mathbb{Z}^k : \exists y_1, \cdots, y_m \in \mathbb{Z} : f(\vec{x}, y_1, \cdots, y_m) = 0\}$$

For instance, $\mathbb{N}$ is Diophantine over $\mathbb{Z}$ since

$$x \in \mathbb{N} \Leftrightarrow \exists y_1, \cdots, y_4 \in \mathbb{Z} : y_1^2 + \cdots + y_4^2 - x = 0$$

Davis-Putnam-Robinson-Matiyasevich proved the following:

Theorem 2 (DPRM Theorem): A set $Q \subseteq \mathbb{Z}$ is computably enumerable if and only if it is Diophantine.
Proof: The first direction is simple: if $\mathsf{Q} \subseteq \mathbb{Z}$ is Diophantine, then we can simply write a program that looks through all elements $f(k, y_1, \cdots, y_m) \in \mathbb{Z}^{m+1}$ and prints k if $f(k, y_1, \cdots, y_m) = 0$

Proving the other direction is substantially more complex. Davis made the first attempt by showing the following:

Theorem 2.1 (Davis' Conjecture [3]): For every computably enumerable set S, there exists a polynomial $p(a, k, y, x_1, ..., x_n)$ such that a number a_0 belongs to S if and only if

$$\exists y\ \forall k \leq y\ \exists x_1, \cdots, x_n\ (p(a_0, k, y, x_1, \cdots, x_n) = 0)$$

Such arithmetical representations of computably enumerable sets with a single bounded universal quantifier is known as the Davis normal form, which was an improvement of a previous fundamental result of G odel concerning the existence of arithmetical representations of a general form for all listable sets. This seems fairly close to the desired goal, however, getting rid of the universal quantifier $\forall k \leq y$ to achieve a Diophantine definition for the computably enumerable set S turned out to be challenging.

Robinson attempted a different strategy by showing that exponentiation is Diophantine, i.e., that the set of all triples $\{(a, b, c) \in \mathbb{N}^3 : c = a^b\}$ is a Diophantine set. She ultimately proved the following hypothesis:

Theorem 2.2.1 (Julia Robinson (JR) Hypothesis): There exists a Diophantine set (J) of pairs (a, b) such that

- if (a, b) belongs to J then $b < a^a$
- for all $k \in \mathbb{N}$, there exists a pair $(a, b) \in \mathsf{J}$ for which $b > a^k$.

For instance, the set of pairs (a, b) where $b = 2^a$ satisfies these conditions. Thus, the Diophantineness of exponentiation follows from the existence of a 2-variable diophanetine relation of exponential growth. Therefore,

Theorem 2.2.2 (Robinson, 1952 [4]): Assuming the JR Hypothesis holds, exponentiation if diophatine.

An exponential Diophantine equation is one in which the exponents are variables as well, and an exponential Diophantine set is a set definable by an exponential Diophantine equation. It naturally follows that if exponentiation is Diophantine, then all exponential Diophantine sets are Diophantine.

Davis, Putnam, and Robinson then proved a weaker version of Theorem 2.1 (Davis' Conjecture), and an intermediate version of the DPRM theorem for exponential Diophantine equations:

Theorem 2.3 (Davis-Putnam-Robinson, 1961 [5]): Every computably enumerable set is exponential Diophantine.

Therefore, showing the truth of Theorem 2.2.1 (the JR Hypothesis) was key to finishing the proof of the DPRM theorem, meaning one had to find the two-variable Diophantine relation of exponential growth. Matiyasevich was able to accomplish this using the Fibonacci numbers, and since the Fibonacci numbers grow exponentially, they satisfy the conditions of the JR Hypothesis:

Theorem 2.4 (Matiyasevich, 1970 [6]): Let F_n be the n^{th} Fibonacci number. The relation $m = F_{2n}$ is Diophantine.

Theorem 2.4 completes the proof of the DPRM Theorem. Consequently, it follows immediately from the DPRM Theorem that there is no algorithm that decides Hilbert's Tenth Problem:

Theorem 3 (H10): Hilbert's 10^{th} Problem is undecidable.
Proof: Let $\mathsf{Q} \subseteq \mathbb{Z}$ such that Q is recursively enumerable but not recursive. By the DPRM Theorem, Q is Diophantine with defining polynomial $f(k, y_1, \cdots, y_m)$. If there exists an algorithm that decides Hilbert's 10^{th} Problem, we can simply apply this algorithm to f to decide membership in Q. However, Q is not recursive, so such an algorithm cannot exist. □

The theorem gives an improvement of Gödel's incompleteness theorems by specifying that the unprovable statement can be the assertion that a particular Diophantine equation has no solution. The undecidability of Hilbert's 10^{th} Problem has been a powerful tool for establishing numerous decision problems. For instance, it is fundamental in providing undecidability of a domain R of characteristic zero, via the following theorem:

Theorem 4: If $\mathbb{Z}$ is Diophantine over R, then $H10/R$ is undecidable.
Proof: Assume for the sake of contradiction that $H10/R$ is decidable, meaning there exists an algorithm that decides $H10/R$. From this algorithm for $H10/R$, we can get an algorithm for $H10/\mathbb{Z}$: given a polynomial $f(x_1, \cdots, x_n)$ over $\mathbb{Z}$, we can use the algorithm for R to test whether f has a solution $x_1, \cdots, x_n$ in R, since $\mathbb{Z}$ is Diophantine over R. We can use the Diophantine definition of $\mathbb{Z}$ to add the necessary equations to indicate that the variables x_i take integer values. However, since $H10/\mathbb{Z}$ is undecidable, an algorithm for it does not exist, hence $H10/R$ where $\mathbb{Z}$ is Diophantine over R must also be undecidable. □

4 Open Questions

The generalized version of Hilbert's 10^{th} Problem is as follows:

Generalized H10: Find an algorithm that decides, given a multivariate polynomial equation $f(x_1, \cdots, x_n) = 0$ with coefficients in R, whether it has a solution with $x_1, ..., x_n \in \mathbb{R}$.

Generalization of Definition 3: A subset $\mathsf{Q} \subseteq R^k$ is *Diophantine over R* if there exists a polynomial $f(x_1, \cdots, x_k, y_1, \cdots, y_m)$ with coefficients in R such that

$$\mathsf{Q} = \{\vec{x} \in R^k : \exists y_1, \cdots, y_m \in R : f(\vec{x}, y_1, \cdots, y_m) = 0\}$$

The DPRM Theorem has shown that Hilbert's 10^{th} Problem is uncomputable for $R = \mathbb{Z}$. Shapiro & Shlapentokh (1989) [7] showed Hilbert's 10^{th} Problem uncomputable for any integer ring of an algebraic number field F, with abelian $Gal(F/\mathbb{Q})$. Kim & Roush (1992) [8] also showed that it is uncomputable for finite extensions of $\mathbb{C}(t_1, t_2, \cdots, t_n)$ for $n \geq 2$. Additionally, Hilbert's 10^{th} Problem is uncomputable for function fields of curves over finite fields [9], p-adic function fields [10], large subrings of $\mathbb{Q}$ [11], and large subrings of number fields [12]. Tarski (1930) [13], on the other hand, showed Hilbert's 10^{th} Problem is computable for real closed fields (e.g. $R = \mathbb{R}$) and algebraically closed fields. Additionally, it is computable for finite fields and p-adic fields.

The biggest unsolved question with regards to Hilbert's 10^{th} Problem therefore is whether or not it is computable for $R = \mathbb{Q}$. The solvability of Hilbert's 10^{th} Problem for $\mathbb{C}(t)$ (non-finite extension of $\mathbb{C}$) is also an open question. An even harder problem is determining the computability of Hilbert's 10^{th} Problem for rings of integers in arbitrary number fields K: a recent result of Mazur & Rubin (2010) [14] showed that it is undecidable for arbitrary rings of integers if the Shafarevich-Tate conjecture [15] holds.

PHIL 152 Final Presentation - Olivia Lee
Handout: Hilbert's 10^{th} Problem

David Hilbert proposed 23 unsolved problems to advance the study of mathematics and determine "what methods, what new facts will the new century reveal in the vast and rich field of mathematical thought?". The 10^{th} problem is:

> *10. Determination of the solvability of a Diophantine equation. Given a Diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: to devise a process according to which it can be determined by a finite number of operations whether the equation is solvable in rational integers.*

Alternatively:

> *10. Determination of the solvability of a Diophantine equation. Find an algorithm that decides, given a multivariate polynomial equation $f(x_1, \cdots, x_n) = 0$ with coefficients in the ring $\mathbb{Z}$ of integers, whether there is a solution with $x_1, \cdot x_n \in \mathbb{Z}$.*

In simple terms: can we automate the decidability of Diophantine solvability, or solvability more generally? Is there a universally valid algorithm that can tell us if any algorithm will terminate? (*Entscheidungsproblem*)

Turing's proof of the undecidability of the Halting problem is a negative answer to this question. This presentation focuses on the solution jointly developmed by Davis, Putnam, Robinsom, and Matiyasevich.

Davis-Putnam-Robinson-Matiyasevich's Proof

Definition 1: A set $\mathsf{Q} \subseteq \mathbb{Z}$ is *computably enumerable* (i.e. *recursively enumerable listable*) if there is an algorithm that prints the elements of Q when left running forever (in any order and with repetitions permitted).

Definition 2: A set $\mathsf{Q} \subseteq \mathbb{Z}$ is *computable* (i.e. *recursive* or *decidable*) if there is an algorithm that decides membership in Q. There is an algorithm that takes as input an integer n and returns true if $n \in Q$ and false if $a \notin Q$.

Theorem 1: A set S is computable if and only if S and its complement S' are both computably enumerable.

Corollary 1: There exists a recursively enumerable set that is not recursive. (Consequence of Turing's proof that the Halting Problem is undecidable.)
Definition 3: A subset $\mathsf{Q} \subseteq \mathbb{Z}^k$ is *Diophantine* if there exists a polynomial $f(x_1, \cdots, x_k, y_1, \cdots, y_m)$ with integer coefficients such that

$$\mathsf{Q} = \{\vec{x} \in \mathbb{Z}^k : \exists y_1, \cdots, y_m \in \mathbb{Z} : f(\vec{x}, y_1, \cdots, y_m) = 0\}$$

For instance, $\mathbb{N}$ is Diophantine over $\mathbb{Z}$ since $x \in \mathbb{N} \Leftrightarrow \exists y_1, \cdots, y_4 \in \mathbb{Z} : y_1^2 + \cdots + y_4^2 - x = 0$.

Theorem 2 (DPRM Theorem): A set $Q \subseteq \mathbb{Z}$ is computably enumerable if and only if it is Diophantine.
Proof: The first direction is simple: if $\mathsf{Q} \subseteq \mathbb{Z}$ is Diophantine, then we can simply write a program that looks through all elements $f(k, y_1, \cdots, y_m) \in \mathbb{Z}^{m+1}$ and prints k if $f(k, y_1, \cdots, y_m) = 0$. Proving the other direction is substantially more complex. Davis made the first attempt:

Theorem 2.1 (Davis' Conjecture): For every computably enumerable set S, there exists a polynomial $p(a, k, y, x_1, ..., x_n)$ such that a number a_0 belongs to S if and only if $\exists y\ \forall k \leq y\ \exists x_1, \cdots, x_n\ (p(a_0, k, y, x_1, \cdots, x_n) = 0)$.

Robinson's strategy was to show that exponentiation is Diophantine, i.e., the set of all triples $\{(a, b, c) \in \mathbb{N}^3 : c = a^b\}$ is a Diophantine set.

Theorem 2.2.1 (Julia Robinson (JR) Hypothesis): There exists a Diophantine set (J) of pairs (a, b) such that if (a, b) belongs to J then $b < a^a$, and for all $k \in \mathbb{N}$, there exists a pair $(a, b) \in \mathsf{J}$ for which $b > a^k$. (The the set of pairs (a, b) where $b = 2^a$ satisfies these conditions.)

Theorem 2.2.2 (Robinson, 1952): Assuming the JR Hypothesis holds, exponentiation if diophatine.

Theorem 2.3 (Davis-Putnam-Robinson, 1961): Every computably enumerable set is exponential Diophantine.

Finishing the proof involves finding the two-variable Diophantine relation of exponential growth. Matiyasevich does this using the Fibonacci numbers:

Theorem 2.4 (Matiyasevich, 1970): Let F_n be the n^{th} Fibonacci number. The relation $m = F_{2n}$ is Diophantine.

This completes the proof of the DPRM Theorem. It follows that there is no algorithm that decides Hilbert's Tenth Problem:

Theorem 3 (H10): Hilbert's 10^{th} Problem is undecidable.
Proof: Let $\mathsf{Q} \subseteq \mathbb{Z}$ such that Q is recursively enumerable but not recursive. By the DPRM Theorem, Q is Diophantine with defining polynomial $f(k, y_1, \cdots, y_m)$. If there exists an algorithm that decides Hilbert's 10^{th} Problem, we can simply apply this algorithm to f to decide membership in Q. However, Q is not recursive, so such an algorithm cannot exist. $\square$

References

[1] Hilbert, D. (1928) Die Grundlagen der Mathematik, Abhandlungen aus dem Seminar der Hamburgischen Universität, 6: 65–85.

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

[3] Davis M. (1953) Arithmetical problems and recursively enumerable predicates. Journal of Symbolic Logic, 18(1):33-41.

[4] Robinson J. (1952) Existential definability in arithmetic. Transactions of the American Mathematical Society, 72(3):437–449.

[5] Davis M., Putnam H., and Robinson J. (1961) The decision problem for exponential Diophantine equations. Annals of Mathematics, Second Series, 74(3):425-436.

[6] Matiyasevich Yu. V. (1970) Enumerable sets are Diophantine (in Russian). Doklady Akademii Nauk SSSR, 191(2):279–282 English translation: Soviet Mathematics. Doklady, 11(2):354–358.

[7] Shapiro, H.N. and Shlapentokh, A. (1989) Diophantine relationships between algebraic number fields. Comm. Pure Appl. Math., 42: 1113-1122. https://doi.org/10.1002/cpa.3160420805

[8] Kim K. H. and Roush F. W. (1992) An approach to rational Diophantine undecidability. World Sci. Publishing, 1992, pp. 242–248.

[9] Shlapentokh, A. (2002) Diophantine Undecidability of Function Fields of Characteristic Greater than 2, Finitely Generated over Fields Algebraic over a Finite Field. Compositio Mathematica 132, 99–120. https://doi.org/10.1023/A:1016067603451

[10] Kim K.H. and Roush F.W. (1995) Diophantine Unsolvability over p-Adic Function Fields. Journal of Algebra, 176(1):83-11. https://doi.org/10.1006/jabr.1995.1234.

[11] Poonen, B. (2003) Hilbert's Tenth Problem and Mazur's Conjecture for Large Subrings of Q. Journal of the American Mathematical Society, 16(4), 981–990. http://www.jstor.org/stable/30041462

[12] Poonen, B. and Shlapentokh, A. (2005). Diophantine definability of infinite discrete nonarchimedean sets and Diophantine models over large subrings of number fields. 2005(588), 27-47. https://doi.org/10.1515/crll.2005.2005.588.27

[13] van den Dries, L. (1988) Alfred Tarski's Elimination Theory for Real Closed Fields. The Journal of Symbolic Logic, 53(1), 7–19. https://doi.org/10.2307/2274424

[14] Mazur, B., Rubin, K. (2010) Ranks of twists of elliptic curves and Hilbert's tenth problem. Invent. math. 181, 541–575. https://doi.org/10.1007/s00222-010-0252-0

[15] Shafarevich, I. R. (1959) The group of principal homogeneous algebraic manifolds. Doklady Akademii Nauk SSSR (in Russian), 124: 42–43, ISSN 0002-3264, MR 0106227 English translation