ON HILBERT'S TENTH PROBLEM

MARCUS COLLINS

1. Introduction

At the turn of the twentieth century, David Hilbert presented a list of 23 unsolved problems at the International Congress of Mathematicians. Hilbert's list still holds today, with questions like the Riemann Hypothesis still being unsolved. The resolved questions are all significant achievements in mathematics. We will examine the negative solution to Hilbert's tenth problem.

The Greek mathematician Diophantus was concerned only with integer solutions to equations, and Diophantine equations (polynomials with integer coefficients looking for integer solutions) are named in his honor. Hilbert's Tenth Problem asks for a general algorithm to tell if a Diophantine equation has solutions (we will later discuss more formally what we mean by "algorithm"). In 1970, Yuri Matiyasevich finished the proof of a stunning result.

Theorem 1.1. *It is impossible to devise an algorithm to determine whether Diophantine equations have solutions.*

In this article, we will prove the progress that was made before Matiyasevich's final contribution, that Hilbert's tenth question is unsolvable if the exponential function can be represented with a Diophantine equation. Along the way, we will show that all the work that can be done by a computer can also be done by one of these Diophantine equations.

2. Basic Computability Theory

The question of computation will be used in the context of set membership: that is, given a set and a potential element, can you determine if it is a member of the set? This will allow us to talk about sets by the power of algorithms that recognize them - intuitively, "nicer" sets should have simpler algorithms (indeed, any finite set can solve the question of membership by just going through all its elements). We will introduce the following important types of sets.

Definition 2.1. A set is called *decidable* if there exists an algorithm that can determine whether any candidate element is a member of the set or not a member of the set in a finite (although not necessarily bounded) amount of time. In other literature, these sets may be referred to as *recursive* or *computable*.

Definition 2.2. A set is called *semi-decidable* if there exists an algorithm that can recognize if a candidate element is a member of the set in a finite amount of time, but when given a non-member may run forever. Thus, if the algorithm has not given an output after some amount of time, no information can be gleaned: it could be that the input is not a member and the algorithm will run forever, or that the input is a member and the algorithm has just

not stopped yet. In other literature, these sets may be referred to as *recursively enumerable* or *Turing-recognizable.*

Every decidable set is semi-decidable as it a strictly stronger condition to have an algorithm that must terminate instead of an algorithm that is allowed to run forever. Most the common sets we see will be decidable: we describe them as the set of all elements satisfying some property, and there typically exists some finite algorithm to check whether that property holds. Do there exist sets that are semi-decidable but not decidable? This is equivalent to asking whether these algorithms that may run forever while giving no information ever truly appear. If we can recognize when an algorithm is going to run forever, then every semi-decidable set will be decidable. This is known as the halting problem, and is our first example of a problem that has no solution! Let's see why.

We will forego a formal treatment of what constitutes a reasonable algorithm, and instead make some quick assumptions. Firstly, we assume there exists some simple model of computing that models all possible algorithms (for instance, Turing machines). The fact that any reasonable computation can be done by a Turing machine is known as the Church-Turing thesis. Secondly, we assume there are only a countable number of algorithms. For a quick justification of this, we expect every algorithm to be describable in some combination of letters and symbols drawn from a finite alphabet; there are only a countable number of these strings. Thus, every algorithm can be represented as just an integer. Finally, we assume that the inputs will be just integers as well. This allows us to run algorithms that take algorithms as inputs. With these preliminaries, we can prove the unsolvability of the halting problem.

Theorem 2.3. *There does not exist an algorithm that, when given another algorithm and a hypothetical input on that algorithm, can determine if it will run forever or halt.*

Proof. Assume for contradiction such an algorithm exists. That is, there is some algorithm "DoesHalt(Algorithm, Input)" that takes in another algorithm and an input on that algorithm and returns whether or not it will halt. We will now construct a second algorithm as follows. It will take an algorithm as input, and simulate that algorithm running on itself (as both inputs and algorithms are integers, we can pass any algorithm as an input into itself). It then uses DoesHalt to see the behavior of this algorithm running on itself. If it recognizes that the algorithm will halt, it runs forever, and if it recognizes the algorithm will run forever, it halts. What happens when we run this algorithm on itself? It will check what happens when it runs on itself and then do the opposite. If it halts, it should run forever, and if it runs forever, it should halt. This is impossible, and we have reached a contradiction. We conclude no such algorithm DoesHalt can exist. ■

With this, we can now construct a set that is semi-decidable but not decidable: the set of all inputs on which a sufficiently complicated algorithm halts. A member can be recognized in a finite amount of time by simply running the algorithm and waiting for it to halt, but a non-member may have the algorithm running forever by our previous theorem.

3. Diophantine Equations

We now give a formal treatment of Diophantine Equations and sets.

Definition 3.1. A Diophantine equation is of the form $P(x_1, \dots x_m) = 0$, where P is a polynomial with integer coefficients and $x_1, \dots x_m \in \mathbb{Z}$.

Definition 3.2. A set S is called Diophantine if there exists some Diophantine equation $P(x, y_1, \dots y_m)$ such that

$$a \in S \iff \exists y_1 \dots y_m P(a, y_1, \dots y_m = 0.)$$

Definition 3.3. Similarly, a function $f(x_1, \dots x_m)$ is said to be Diophantine if the set of all $m+1$-tuples $(x_1, \dots x_m, f(x_1, \dots x_m))$ is Diophantine.

If Hilbert's Tenth Problem has a solution, every Diophantine set must be decidable. We will show that every semi-decidable set is Diophantine, and thus there exist Diophantine sets that are not decidable. There are a couple helpful simplifications we can make with these sets.

Lemma 3.4. *The system* $P_1(x_1, \dots x_m) = P_2(x_1, \dots x_m) = \dots P_n(x_1, \dots x_m) = 0$ *has a solution if and only if* $P_1^2(x_1, \dots x_m) + P_2^2(x_1 \dots x_m) + \dots + P_n^2(x_1 \dots x_m) = 0$ *has a solution.*

With this, we know that solving a system of equations is the same as solving just one. We can additionally note the following:

Lemma 3.5. *An algorithm that checks for integer solutions to Diophantine equations could check for positive solutions to* $P(x_1 \dots x_m) = 0$ *by adding on the constraints* $x_1 = p_1^2 + q_1^2 + r_1^2 + s_1^2 + 1, \dots x_m = p_m^2 + q_m^2 + r_m^2 + s_m^2 + 1$

Proof. We know from a theorem of Lagrange that every nonnegative integer can be writen as the sum of 4 squares. ■

From this, we know if we can prove there does not exist an algorithm for finding positive solutions, then there cannot exist an algorithm for finding general solutions by contraposition. Thus, we will restrict ourselves to looking for positive solutions for the rest of this article. This allows us to express constraints such as $x > 5$ as $x - 5 = a$, which will eliminate positive solutions if $x < 5$. Let's see some examples of Diophantine sets.

(1) The set of all composite numbers is Diophantine. x is composite $\iff \exists p, q(p+1)(q+1) = x$.
(2) The set $S = \{(p, q) : p|q\}$ is Diophantine. $x \in S \iff \exists y(py = q)$.
(3) The set of powers of even numbers is Diophantine, $a \in S \iff \exists x(2x = a)$.

4. Some Helpful Diophantine Functions

Theorem 4.1. *The function $f(a,b) = a^b$ is Diophantine.*

This theorem was the most difficult part of the proof of Hilbert's tenth problem and the final contribution of Matiyasevich. The proof is here omitted for the time being due to brevity and complexity.

Lemma 4.2. *The sets $S = \{(a,b)|a < b\}$ and $S' = \{(a,b)|a \leq b\}$ are Diophantine.*

Proof. $a < b \iff \exists x(a + x = b), a \leq b \iff \exists x(a + x - 1 = b)$. ■

Lemma 4.3. *The function $f(a,b) = \lfloor \frac{a}{b} \rfloor$ is Diophantine*

Proof. $x = \lfloor \frac{a}{b} \rfloor \iff x < \frac{a}{b} < x + 1 \iff bx < a < bx + b \iff bx < a \wedge a < bx + b$. ■

Lemma 4.4. $\binom{n}{k} = \lfloor 2^{nk}(1 + 2^{-n})^n \rfloor - 2^n \lfloor 2^{n(k-1)}(1 + 2^{-n})^n \rfloor$

Proof. By the binomial theorem,

$$(1 + 2^{-n})^n = \sum_{v=0}^{n} \binom{n}{v} 2^{-nv}.$$

Multiplying both sides by 2^{nk} yields

$$2^{nk}(1 + 2^{-n})^n = 2^{nk} \sum_{v=0}^{n} \binom{n}{v} 2^{-nv} = \sum_{v=0}^{n} \binom{n}{v} 2^{n(k-v)}.$$

Split the sum into $v \leq k$ and $v > k$:

$$\sum_{v=0}^{k} \binom{n}{v} 2^{n(k-v)} + \sum_{v=k+1}^{n} \binom{n}{v} 2^{n(k-v)}$$

For the sum on the left, $n(k - v) > 0$ since $(k - v) > 0$, thus every term in the sum is an integer and the whole sum is an integer. For the sum on the right, $(k - v) \leq -1$, so $2^{n(k-v)} \leq 2^{-n}$. Thus every remaining term is bounded by $\binom{n}{v} 2^{-n}$. We have

$$\sum_{v=k+1}^{n} \binom{n}{v} 2^{n(k-v)} < 2^{-n} \sum_{v=k+1}^{n} \binom{n}{k}$$

Because

$$\sum_{v=0}^{n} \binom{n}{k} = 2^n,$$

we know

$$\sum_{v=k+1}^{n} \binom{n}{k} < 2^n.$$

We conclude

$$2^{-n} \sum_{v=k+1}^{n} \binom{n}{k} < 2^{-n} 2^n = 1.$$

Because the right hand sum is less than 1, it will be eliminated by the floor function. Hence

$$\lfloor 2^{nk}(1 + 2^{-n})^n \rfloor = \sum_{v=0}^{k} \binom{n}{v} 2^{n(k-v)}.$$

The same argument works when replacing k with $k-1$, and we get

$$\lfloor 2^{n(k-1)}(1+2^{-n})^n \rfloor = \sum_{v=0}^{k-1} \binom{n}{v} 2^{n(k-1-v)}.$$

Multiplying both sides by 2^n, we have

$$2^n \lfloor 2^{n(k-1)}(1+2^{-n})^n \rfloor = \sum_{v=0}^{k-1} \binom{n}{v} 2^{n(k-v)}.$$

This matches the powers from our first identity, and subtracting them yields the lemma. ■

Theorem 4.5. *The function $f(n,k) = \binom{n}{k}$ is Diophantine.*

Proof.

$$\binom{n}{k} = y \iff (n = 0 \wedge y = 0) \vee (k = 1 \wedge r = 1)$$

$$\vee (\exists u, v (y = u - 2^n v) \wedge 2^{n^2} u \leq 2^{nk}(1+2^n)^n < 2^{n^2}(1+u) \wedge 2^{n^2} v \leq 2^{nk}(1+2^n)^n < 2^{n^2}(1+v)$$

■

Lemma 4.6. *$x! = \lfloor r^x / \binom{r}{x} \rfloor$ for any $r > (2x)^{x+1}$*

Proof.

$$r^x / \binom{r}{x} = \frac{r^x x!}{r(r-1)\dots(r-x+1)},$$

factoring out $x!$ and canceling r^x yields

$$x!\big(\frac{1}{(1-\frac{1}{r})\dots(1-\frac{x-1}{r})}\big).$$

From this we can conclude $r^x / \binom{r}{x} > x!$. Every term in the denominator will be larger than $1 - \frac{x}{r}$, so

$$x!\big(\frac{1}{(1-\frac{1}{r})\dots(1-\frac{x-1}{r})}\big) < x!\big(\frac{1}{(1-\frac{x}{r})^x}\big).$$

Recall the geometric series $\frac{1}{1-t} = 1 + t + t^2 + \dots$ for $t < 1$. Plugging in $t = \frac{x}{r}$ gives

$$\frac{1}{1-\frac{x}{r}} = 1 + \frac{x}{r} + (\frac{x}{r})^2 + \dots.$$

Because $r > (2x)^{x+1}$ we must have $r > 2x$ and thus $\frac{x}{r} < \frac{1}{2}$, so

$$1 + \frac{x}{r} + (\frac{x}{r})^2 + \dots < 1 + \frac{1}{2} + \frac{1}{4} + \dots = 2.$$

Therefore

$$\frac{1}{1-\frac{x}{r}} = 1 + \frac{x}{r} + (\frac{x}{r})^2 + \dots. = 1 + \frac{x}{r}\big(\frac{x}{r} + (\frac{x}{r})^2 + \dots\big) < 1 + \frac{x}{r}(2) = 1 + \frac{2x}{r}.$$

Because

$$\frac{1}{1-\frac{x}{r}} < 1 + \frac{2x}{r},$$

we know

$$\big(\frac{1}{1-\frac{x}{r}}\big)^x < \big(1 + \frac{2x}{r}\big)^x.$$

By the binomial theorem,

$$(1+\frac{2x}{r})^x = \sum_{j=0}^{x} \binom{x}{j} \cdot (\frac{2x}{r})^j = 1 + \sum_{j=1}^{x} \binom{x}{j} \cdot (\frac{2x}{r})^j.$$

Because $\frac{2x}{r} < 1$, $(\frac{2x}{r})^j < \frac{2x}{r}$. So

$$1 + \sum_{j=1}^{x} \binom{x}{j} \cdot (\frac{2x}{r})^j < 1 + (\frac{2x}{r}) \sum_{j=1}^{x} \binom{x}{j},$$

and since $\sum_{j=1}^{x} \binom{x}{j} = 2^x - 1$ we have

$$1 + \frac{2x}{r} \sum_{j=1}^{x} \binom{x}{j} = 1 + \frac{2x}{r} \cdot 2^x.$$

Combining our inequalities, we have

$$r^x / \binom{r}{x} < x! + \frac{2x}{r} \cdot 2^x = x! + x! \frac{x 2^{x+1}}{r}.$$

Since $x! \leq x^x$,

$$x! + x! \frac{x 2^{x+1}}{r} < x! + \frac{x^{x+1} 2^{x+1}}{r}.$$

By our assumption that $r > (2x)^{x+1}$, we know $\frac{x^{x+1} 2^{x+1}}{r} < 1$, and we conclude $r^x / \binom{r}{x} < x! + 1$. Since

$$x! \leq r^x / \binom{r}{x} < x! + 1,$$

we conclude

$$\lfloor r^x / \binom{r}{x} \rfloor = x!$$

■

Theorem 4.7. *The function $f(x) = x!$ is Diophantine.*

Proof. $y = x! \iff \exists r (r > (2x)^{x+1} \wedge \lfloor r^x / \binom{r}{x} \rfloor = y)$. ■

Theorem 4.8. *The function $f(a, b, k) = \prod_{i=1}^{k} a + bi$ is Diophantine*

Proof.

$$\prod_{i=0}^{k-1} a + bi = \binom{a/b + k}{k} b^k k!$$

■

5. Every Semi-Decidable set is Diophantine

We will begin with the following characterization of Semi-decidable sets:

Theorem 5.1. *For every semi-decidable set S,*

$$x \in S \iff \exists y \forall_{k \leq y} \exists z_1 \dots z_m P(x, y, k, z_1, \dots z_m) = 0$$

for some polynomial with integer coefficients P.

What do each of these quantified variables mean? Intuitively, we want an element to be a member of a semi-decidable set if it satisfies the algorithm determining membership after some finite amount of time. y can be thought of as the amount of time this takes: there must exist some finite point after which the algorithm has accepted the element. It remains to verify that every step of the algorithm progresses properly according to its rules: k is the current state, and the z_i are witness variables that verify the listed sequence is a valid transistion of the algorithm between the states. To show that the basic computations of a computer can be checked by a polynomial is beyond the scope of this article, but one should expect this to be true as a theoretical model of computation is incredibly low-level.

Why doesn't this already characterize every semi-decidable set as Diophantine? The issue is in the bounded quantifier $\forall_{k \leq y}$; Diophantine sets only allow existential quantifiers. There is no obvious way to express a statement required to be true of all k up to an arbitrarily high y with polynomials. However, we can ultimately eliminate this bounded quantifier and show every semi-decidable set is Diophantine.

Theorem 5.2.

$$\exists y \forall_{k \leq y} \exists z_1 \dots z_m P(x, y, k, z_1, \dots z_m) = 0 \iff$$

$$\exists c, t, a_1, \dots a_m (t = G(x,y)! \wedge 1+ct = \prod_{i=1}^{y} (1+it) \wedge (1+ct) | f(x, y, c, a_1, \dots a_m) \wedge \forall i \leq m ((1+ct) | \prod_{j \leq y} (a_i - j))$$

for some $G(x, y)$ satisfying $G(x, y) \geq y$ and $\forall_{k \leq y} \forall z_1, \dots z_m (|P(x, y, k, z_1, \dots z_m)| \leq G(x, y)$

Proof. Fix a particular $k \leq y$. Let p_k be a prime divisor of $1 + kt$. Because

$$1 + ct = \prod_{i=1}^{k} (1 + it),$$

we know

$$p_k | 1 + ct$$

and thus $c \equiv k \mod p$. For each i, let $z_i = \text{Rem}(a_i, p_k)$ (the remainder function is easily seen to be Diophantine). Since

$$(1 + ct) | \prod_{j=1}^{y} (a_i - j)$$

and $p_k | 1 + ct$, we know that

$$p_k | \prod_{j=1}^{y} (a_i - j).$$

Because p_k is prime, it thus must divide some factor from the product by Euclid's lemma, i.e. there is some j such that $p_k | (a_i - j)$, or $a_i \equiv j \mod p_k$. Since z_i is the remainder of a_i

mod p_k, we can conclude $z_i = j$ and $1 \leq z_i \leq y$. Because $a_i \equiv z_i \mod p_k$ and $c \equiv k \mod p_k$, we know

$$P(x, y, c, a_1, \ldots a_m) \equiv P(x, y, k, z_1, \ldots z_m) \mod p_k.$$

Since we assume $p_k | P(x, y, c, a_1, \ldots a_m)$ and thus $P(x, y, c, a_1, \ldots a_m) \equiv 0 \mod p_k$ alongside

$$|P(x, y, k, z_1, \ldots z_m)| \leq G(x, y) = t < p_k,$$

we conclude $P(x, y, k, z_1, \ldots z_m) = 0$, which was to be shown. ■

This theorem defines all semi-decidable sets in terms of products of arithmetic sequences and factorials, which were shown to be Diophantine in section 4. We can at last conclude that every semi-decidable set is Diophantine, there exist Diophantine sets that are not decidable, and Hilbert's tenth problem is unsolvable.

6. Applications and an Open Question

One simple idea for an algorithm to solve Hilbert's Tenth Problem is the following: "Bound the size of the smallest possible solution based on the size of the coefficients, then check all possible answers up to that bound." We know that this cannot work because no algorithm exists: this implies that there is no bound on the size of the smallest solutions given the coefficients, and as a funny consequence there are some very simple equations with very large solutions. Hilbert's Tenth problem over the rationals remains a major open question. There

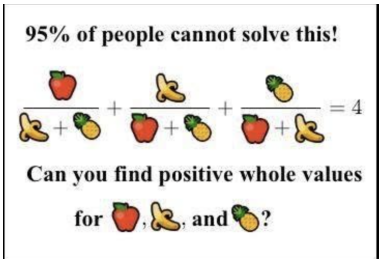

Figure 1. The smallest solution is apple =
1544768021087461664419513150199198374856643256695654317000266348982
53202035277999,
banana =
3687513179412999982719781156522547482549297996897197099628313747163
7224634055579,
pineapple =
4373612677928697257861252602371390152816537558161613618621437993378
423467772036

has been recent progress over other rings.

7. Acknowledgments

This article was completed in the 2026 section of Independent Research and Paper Writing through Euler Circle. The author thanks Simon Rubenstein-Salzado and Arkit Vipul Ray for their guidance.

8. Bibliography

References

[1] Martin Davis. Hilbert's tenth problem is unsolvable. *The American Mathematical Monthly*, 80(3):233–269, 1973.

[2] Martin Davis, Hilary Putnam, and Julia Robinson. The decision problem for exponential diophantine equations. *Annals of Mathematics*, 74(3):425–436, 1961.

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