Hilbert’s Tenth Problem

Caleb Alexander

Background

Diophantus

- Greek mathematician from the 3rd century AD.
- Introduced Diophantine Equations which, as a problem, were one of the inspirations for the field of algebraic number theory.

Hilbert

- 19th/20th century German mathematician
- Advocated for the adoption of set-theory as a common language in mathematics.
- Introduced his famous **23** problems which would be some of the most heavily studied problems throughout the 20th century.

Formal Construction

A diophantine equation is polynomial equation with integer coefficients where only the integer solutions are considered. For example,

$$x^3 - y^2 = 2$$

$$\text{solutions} = \{(3,5),(3,-5)\}$$

Is there an algorithm which, for any diophantine equation, will decide whether the equation has a valid solution?

Enumerable List Equivalence Pt. 1

- Worked with so-called exponential diophantine equations.
  - Can be of the form $x^z + y + c = 0$ (Notice a variable as an exponent)
- Established a one-to-one correspondence between sets of integers (defined existentially by addition and multiplication) and diophantine polynomials. These sets of integers are also called diophantine because they are the set of integers under which a diophantine equation is solvable.
  - A set of integers defined existentially by addition and multiplication will look like

$$S = \{x \in \mathbb{Z} : \exists y \in \mathbb{Z} \text{ and } x = 2 + 4y\}$$

- This same one-to-one correspondence was extended to exponential diophantine sets.

$$S = \{x \in \mathbb{Z} : \exists y, z \in \mathbb{Z} \text{ and } x = 2 + (4y)^z\}$$

Enumerable List Equivalence Pt. 2

- Established that every diophantine set was recursively enumerable.
  - Since every diophantine set contains some number of finite-tuples of integers it should be true that for a diophantine set D

$$D \subseteq \prod_{i=1}^{k} \mathbb{Z} \quad \text{so} \quad |D| \leq |\prod_{i=1}^{k} \mathbb{Z}| = |\mathbb{Z}|$$

    And clearly the integers are countable.

- Showed that recursively enumerable set was at least exponentially diophantine.
  - A recursively enumerable set, is one in which a turing machine can enumerate (list) the set.
  - Did so by showing that every recursively enumerable set can be represented as a recursively enumerable relation, which is of course a function. Each of these functions can be represented as an exponentially diophantine function; thus an exponentially diophantine set is implied.

A Thought

- Notice that if we could prove that all recursively enumerable sets were not just exponentially diophantine, but strictly diophantine, then there is an equivalence between diophantine sets and recursively enumerable sets.

- Thus there must be some diophantine set and polynomial which are equivalent to the "halting problem"; and thus is undecidable.
    - so $\exists P$ so that $P(x_1, \ldots, x_n) = 0$ is undecidable.

Matiyasevich Shows the equivalence

- Matiyasevich defined a diophantine set in terms of the fibonacci numbers. Recall that fibonacci numbers grow faster than any polynomial and are thus on the order of exponential.
- By defining a diophantine set in terms of the fibonacci numbers he showed that for an exponential function he could find a diophantine polynomial which may approximate it.
    - If you give me some exponential diophantine polynomial P, I will give you some strictly diophantine polynomial P’ such that P’ approximates P at values before some arbitrarily large integer N, and some of, or all of, the coefficients of P’ are fibonacci numbers.
- This allowed him to show that every RE set was diophantine.

The Problem is undecidable and therefore NP-Hard

The problem is equivalent to the “halting problem”, which is undecidable, therefore Hilbert’s 10th problem is undecidable and NP-Hard.

Gödel’s Incompleteness and the significance of undecidable problems

- In his two famed incompleteness theorems Gödel showed that even a formal system which has almost no suppositions can be used to construct a problem which is undecidable.
- This undecidable problem is, in most senses, equivalent to the “halting problem”.
- Thus we can say that, in some sense, all undecidable problems are (or at least reduce to) a single undecidable problem.
- In general, despite the frustrations intrinsic in knowing that some things are beyond computation, it is a very powerful tool for mathematicians and computer scientists to have a problem which is provably unsolvable.

Sources

A good explanation of both the RDP and Matiyasevich's Proofs.

https://www.logicmatters.net/resources/pdfs/MRDP.pdf

David, Putnam, & Robinson's original Proof

https://www.jstor.org/stable/pdf/1970289.pdf?refreqid=fastly-default%3Abf6ac99d35cb2c093a06eaf73dcd53a9&ab_segments=&initiator=&acceptTC=1