Faculty of Science

On Hilbert's tenth problem

Bachelor Thesis, 7.5 ECTS

Jonne Visser (6154174)

Department of Mathematics

Supervisor:

Dr. Jaap van Oosten

June 17, 2021

Contents

1 Introduction

In the third century A.D., the Greek mathematician Diophantus studied the solutions to polynomial equations. While other Greek mathematicians of his time used geometry to solve those, Diophantus began solving polynomial equations in terms of rational integers. This kind of equations turned out to find much use (e.g. in the Chinese Remainder Theorem and Fermat's Last Theorem) and solving them became an important problem. In honor of the advances Diophantus has made on equations of the form

$$D(a_1, ..., a_m, x_1, ..., x_n) = 0, \tag{1.1}$$

where D is a polynomial with integer coefficients, these equations have been called *Diophantine equations*. Note the use of two different kinds of variables; one using a as a base letter and one using x. We will discuss the reason for this in chapter 2.

Example: The equation $3x_1^3x_2 + a_1x_3x_4^5 = 0$ is a Diophantine equation if we demand that $x_1, ..., x_4, a_1 \in \mathbb{Z}$.

Throughout the centuries, many solutions to Diophantine equations have been found. Mathematicians have also proved the insolvability of many others. But while much work has been done, mathematicians realized that the existing solutions and proofs of insolvability only worked on specific classes of equations, and that there was not yet a universal method to recognize the solvability of a Diophantine equation.

During the Second International Congress of Mathematicians in 1900, the German mathematician David Hilbert listed 23 open problems in mathematics that he deemed the most important. These problems became known as Hilbert's problems, and are bundled in his paper, *Mathematische Probleme* (Hilbert 1900). The tenth of these problems is concerned with finding this universal method of determining whether a Diophantine equation is solvable;

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; devise a process according to which it can be determined by a finite number of operations whether the equation is solvable in rational integers. (Matiyasevich 2000)

Over the course of the twentieth century mathematicians worked to prove that such an algorithm did not exist until finally, in 1970, the last piece was provided by Yury Matiyasevich and it was proven that, in fact, no universal algorithm exists to determine whether a Diophantine equation has a solution. This thesis will be concerned with providing a proof to the negative answer to Hilbert's tenth problem. In many places throughout the paper we will follow the theory and structure of (Matiyasevich 2000), Matiyasevich's lecture notes on Hilbert's tenth problem.
To get acquainted with the notation and ideas on the subject of Hilbert's tenth problem, it will be worthwhile to take a look at an overview of the final proof.

2 Overview of the proof

In order to simplify the calculations involved in this proof, it is beneficial to consider Diophantine equations with solutions and coefficients in $\mathbb{N}$, instead of all of $\mathbb{Z}$. This is possible, because the solvability of Diophantine equations in positive integers is equivalent to the solvability of Diophantine equations in $\mathbb{Z}$; a Diophantine equation in $\mathbb{Z}$ reduces to one in $\mathbb{N}$ using the fact that $\{x-y|x,y\in\mathbb{N}\}=\mathbb{Z}$, and a Diophantine equation in $\mathbb{N}$ reduces to one in $\mathbb{Z}$ using the fact that $\{x^2+y^2+z^2+w^2|x,y,z,w\in\mathbb{Z}\}=\mathbb{N}$ (Lagrange's theorem).

Example: Consider the Diophantine equation $3x-6=0$ in $\mathbb{Z}$. This equation has solutions in $\mathbb{Z}$ precisely when the equation $3(a-b)-6=0$ has solutions in $\mathbb{N}$; suppose that $3x-6=0$ has a solution y in $\mathbb{Z}$. Then take $a=y, b=0$ if $y\geqslant 0$, and take $a=0, b=-y$ if $y<0$. Then $3(a-b)-6=0$ holds. Conversely, suppose that there are $p,q\in\mathbb{N}$ such that $3(p-q)-6=0$. Then let $x=p-q$, so that $3x-6=0$ holds.

We will now define a *Diophantine set*:

Definition 2.0.1. A **Diophantine set** Φ is a set that has the representation:

$$(a_1,...,a_n)\in\Phi \iff \exists x_1,...,x_n,\{D(a_1,...,a_n,x_1,...,x_m)=0\}, \tag{2.1}$$

for some Diophantine equation $D=0$. This is where the two different kinds of variables come into play; we fix variables $a_1,...,a_n$, and see whether we can choose the remaining variables $x_1,...,x_m$ such that the equation still holds. If this is the case, precisely then does $(a_1,...,a_n)$ belong to Φ.

Example: The set of all squares S is Diophantine, because it has a representation

$$y\in S \iff \exists x\{y=x^2\} \tag{2.2}$$

Example: The set of all Pythagorean triples T_P is also Diophantine:

$$(a,b,c)\in T_P \iff \{a^2+b^2=c^2\} \tag{2.3}$$
$$\tag{2.4}$$

To arrive at the main idea of the proof, we will need one more definition:

Definition 2.0.2. A **listable** or **effectively enumerable set** $\mathcal{M}$ is a set for which there exists an algorithm that prints in some order, possibly with repetitions, all the elements of the set $\mathcal{M}$.

In 1953, Martin Davis conjectured that the notions of listable and Diophantine sets coincide; a set would be Diophantine if and only if it is listable (Davis 1953). The proof of this conjecture would prove the negative answer to Hilbert's tenth problem, because there are known listable sets for which there exists no algorithm that, given an element that is not in the set, can determine in finite time that this element is not part of the set. Such sets are called undecidable listable sets. Davis's conjecture would imply that there is a Diophantine set $\Phi=\{(a_1,...,a_m)|\exists x_1,...,x_n, D(a_1,...,a_m,x_1,...,x_n)=0\}$ and some tuple $\vec{b}=(b_1,...,b_m)\in\mathbb{N}^m$ such that there is no algorithm that determines whether $\vec{b}\in\Phi$, or equivalently that there is no algorithm that determines whether $D(b_1,...,b_m,x_1,...,x_n)=0$ has any solutions.

It is clear that Diophantine sets must be listable; using the representation (2.1), one could simply try every $(n+m)$-tuple $(a_1,...,a_n,x_1,...,x_m)$ in some order and see if it is a solution to D. If it is, put $(a_1,...,a_n)$ on the list. This way every element of the Diophantine set would eventually end up on the list, albeit possibly with many repetitions. We will devote the third chapter to the construction of a listable set that is undecidable. Then, it remains to be shown that every listable set is also Diophantine. Martin Davis, Hilary Putnam and Julia Robinson published a paper in 1961 (Davis M. 1961) in which they showed that there exists an

exponential Diophantine representation for every listable set. Exponential Diophantine sets are similar to Diophantine sets, except that their representations also allow the use of exponentiation as an operation. In chapter four we will discuss exponential Diophantine sets, and we will prove the equivalence between listable sets and exponential Diophantine sets in chapter five. The final step of proving that exponential Diophantine sets are also Diophantine sets and vice versa was provided by Yuri Matiyasevich in 1970 (Matiyasevich 1970). For this he used the following result published in 1952 by Julia Robinson (Robinson 1952):

Theorem 2.0.3. *There is a polynomial* $A(a, b, c, z_1, ..., z_m)$ *such that*

$$a^b = c \leftrightarrow \exists z_1, ..., z_m \{A(a, b, c, z_1, ..., z_m) = 0\},$$

provided that there is a Diophantine equation $J(u, v, y_1, ..., y_w)$ *such that*

1. *in every solution of the equation we have* $u < v^v$*;*
2. *for every* k *there is a solution such that* $u > v^k$*.*

The sixth chapter will be devoted to the proof that exponential Diophantine sets are equivalent to Diophantine sets.

3 Undecidable listable sets

In chapter 2 it was mentioned that there are known undecidable listable sets. This chapter will be concerned with the construction of such a set. To do this we will first look at a formal definition of an algorithm using the register machine. The description of this register machine is taken from (Matiyasevich 2000, ch. 4.2). The idea of the following construction of an undecidable listable set is borrowed from Poonen 2008, details by me.

3.1 The register machine

A register machine is a computer that has n *registers* $(R_1, R_2, ..., R_n)$ containing positive integers of any size, and that can perform a *program* using m *instructions* $(S_1, S_2, ..., S_m)$, for which S_k can be any of three instructions:

I. $S_k : R_l + +; S_i$

II. $S_k : R_l - -; S_i; S_j$

III. S_k : STOP

An instruction of type I will increment R_l by one and switch to instruction S_i.

An instruction of type II will decrease R_l by one; if this is done successfully we will switch to instruction S_i. If this is not possible because $R_l = 0$, we will switch to instruction S_j instead.

An instruction of type III will terminate the program. Because any two instructions of type III are the same and because the program must terminate, we can assume that there is exactly one instruction of type III, and without loss of generality that this is instruction S_m.

3.2 Algorithms

Now to get to the point, in definition (2.0.2) we said that listable sets are those sets that can be printed by an algorithm. But what exactly do we mean by that? Funnily enough, in 1900 at the time that Hilbert stated his tenth problem, there was no formal definition of algorithms; the process that he requested was just an intuitive notion at the time. Today, we understand that Hilbert asked for an algorithm. Appendix A provides a brief history of the search for what exactly is computable, and gives a motivation for our definition of an algorithm:

Definition 3.2.1. An **algorithm** is a process that can be carried out by a Turing machine.

In this paper we will represent algorithms using the register machine. Although register machines and Turing machines might look quite different, they can be shown to be equivalent in the sense that the operation of one machine can be simulated on the other (Minsky 1967).

3.3 The halting problem

In order to see that there is a listable set that is undecidable, we will first look at the *halting problem*. The halting problem asks whether there is a register machine that, given any register machine program as input, determines whether the program will halt or not. First of all, this raises the question how one would actually use a program as input. For this we use the following insight.

Consider a register machine. It consists of n registers, and m instructions of either of three types. Let $\pi : \mathbb{N}^2 \to \mathbb{N}$ be Cantor's pairing function. We can encode instructions of type $S_k : R_l + +; S_i$ as $\pi(l, i)$, and we can encode instructions of type $S_k : R_l - -; S_i, S_j$ as $\pi(l, \pi(i, j))$. Then, using those encodings together with the integer n we have a finite amount of natural numbers describing the program, and with repeated

use of Cantor's pairing function we have a unique natural number for every possible register machine program.

Now we are ready to address the halting problem. Assume that there is some register machine R that accepts the encoding of any program as input, and halts when this program would run forever. If the program would halt, R will run forever. Now suppose that we use R's program encoding as input for R; Then, by definition, R would halt precisely when R never halts. This is a contradiction, so there can be no such register machine R.

3.4 The construction

Now, consider the set

$$A = \{2^p \cdot 3^a | a \in \mathbb{N}, p \text{ is the encoding of a program that halts on input } x\}. \tag{3.1}$$

This set is listable; simply loop over all tuples (p, a, N) in some order, and if program p halts on input a within N instructions then add $2^p \cdot 3^a$ to the list. However, the set must also be undecidable; suppose that there is some algorithm that determines whether an input c belongs to A. Then this algorithm would also be able to determine for any program p whether it halts on input a or not. By the previous proof of the negative answer to the halting problem this is impossible, so A must be an undecidable listable set.

4 Exponential Diophantine sets

In this section we will take a look at exponential Diophantine equations. The theory and structure is taken from (Matiyasevich 2000, ch. 2).
As discussed in the introduction, exponential Diophantine sets play an essential role in the proof. Exponential Diophantine sets are represented by exponential Diophantine equations:

Definition 4.0.1. An **exponential Diophantine equation** is an equation of the form

$$E_L(a_1, ..., a_n, x_1, x_2, ..., x_m) = E_R(a_1, ..., a_n, x_1, x_2, ..., x_m), \tag{4.1}$$

where E_L and E_R are **exponential polynomials**; equations that combine terms using addition, multiplication and exponentiation.

Note that we do not allow subtraction; doing so would quickly lead us out of the realm of natural numbers, and into the world of complex arithmetics; for example, the expression $(x-y)^{2^{x-y}}$ does not have any meaningful value in the natural numbers (or even in the real numbers for that matter) as soon as $y > x$. We do permit the use of 0 though, despite some ambiguity regarding the value of 0^0; for this proof it will be convenient define $0^0 = 1$.

Similar to (regular) Diophantine equations, we can consider exponential Diophantine sets:

Definition 4.0.2. An **exponential Diophantine set** is a set Φ_E that has a representation:

$$(a_1, ..., a_n) \in \Phi_E \iff \exists x_1, ..., x_n \{E_L(a_1, ..., a_n, x_1, ..., x_m) = E_R(a_1, ..., a_n, x_1, ..., x_m)\}, \tag{4.2}$$

where $E_L(a_1, ..., a_n, x_1, ..., x_m) = E_R(a_1, ..., a_n, x_1, ..., x_m)$ is an exponential Diophantine equation.

4.1 Generalized exponential Diophantine equations

At first glance we are limited to a very select number of operations to use in exponential Diophantine equations, i.e. addition and multiplication. However, we can show that we can use more exotic operations, properties, functions and relations in exponential Diophantine equations as well; we call those exponential Diophantine operations, properties, functions and relations, respectively. We will call expressions using those generalized exponential Diophantine relations. In the next sections we will take a look at a few of those.

4.2 Exponential Diophantine relations

4.2.1 Conjunction

Consider two exponential Diophantine equations:

$$\begin{aligned} E_L(a_1, ..., a_n, x_1, ..., x_m) &= E_R(a_1, ..., a_n, x_1, ..., x_m) \\ F_L(a_1, ..., a_n, x_1, ..., x_m) &= F_R(a_1, ..., a_n, x_1, ..., x_m). \end{aligned} \tag{4.3}$$

We see that the conjunction of these exponential Diophantine equations is an exponential Diophantine equation as well:

$$
\begin{gathered}
E_L(a_1,...,a_n,x_1,...,x_m) = E_R(a_1,...,a_n,x_1,...,x_m)\& \\
F_L(a_1,...,a_n,x_1,...,x_m) = F_R(a_1,...,a_n,x_1,...,x_m) \\
\iff \\
\{E_L(a_1,...,a_n,x_1,...,x_m) - E_R(a_1,...,a_n,x_1,...,x_m)\}^2 + \{F_L(a_1,...,a_n,x_1,...,x_m) - F_R(a_1,...,a_n,x_1,...,x_m)\}^2 = 0 \\
\iff \\
E_L(a_1,...,a_n,x_1,...,x_m)^2 + E_R(a_1,...,a_n,x_1,...,x_m)^2 + F_L(a_1,...,a_n,x_1,...,x_m)^2 + F_R(a_1,...,a_n,x_1,...,x_m)^2 = \\
2E_L(a_1,...,a_n,x_1,...,x_m)E_R(a_1,...,a_n,x_1,...,x_m) + 2F_L(a_1,...,a_n,x_1,...,x_m)F_R(a_1,...,a_n,x_1,...,x_m)
\end{gathered} \tag{4.4}
$$

Here "&" denotes the conjunction of two exponential Diophantine equations.

4.2.2 Disjunction

Instead of conjunction, we can also consider the disjunction ($\vee$) of two exponential Diophantine equations:

$$
\begin{gathered}
E_L(a_1,...,a_n,x_1,...,x_m) = E_R(a_1,...,a_n,x_1,...,x_m)\vee \\
F_L(a_1,...,a_n,x_1,...,x_m) = F_R(a_1,...,a_n,x_1,...,x_m) \\
\iff \\
\{E_L(a_1,...,a_n,x_1,...,x_m) - E_R(a_1,...,a_n,x_1,...,x_m)\} \cdot \{F_L(a_1,...,a_n,x_1,...,x_m) - F_R(a_1,...,a_n,x_1,...,x_m)\} = 0 \\
\iff \\
E_L(a_1,...,a_n,x_1,...,x_m)F_L(a_1,...,a_n,x_1,...,x_m) + E_R(a_1,...,a_n,x_1,...,x_m)F_R(a_1,...,a_n,x_1,...,x_m) = \\
E_R(a_1,...,a_n,x_1,...,x_m)F_L(a_1,...,a_n,x_1,...,x_m) + E_L(a_1,...,a_n,x_1,...,x_m)F_R(a_1,...,a_n,x_1,...,x_m)
\end{gathered} \tag{4.5}
$$

4.2.3 Negation

We can use negation ($\neg$) as well if we introduce an extra variable:

$$
\begin{gathered}
\neg\{E_L(a_1,...,a_n,x_1,...,x_m) = E_R(a_1,...,a_n,x_1,...,x_m)\} \\
\iff \\
\exists y\{[E_L(a_1,...,a_n,x_1,...,x_m) - E_L(a_1,...,a_n,x_1,...,x_m)]^2 = y+1\},
\end{gathered} \tag{4.6}
$$

and we can again expand to yield a valid exponential Diophantine equation.

It is important to note that the set $\{a_1,...,a_m|\exists x_1,...,x_n, E_L(a_1,...,a_n,x_1,...,x_m) \neq E_R(a_1,...,a_n,x_1,...,x_m)\}$ is not the complement of the set $\{a_1,...,a_m|\exists x_1,...,x_n, E_L(a_1,...,a_n,x_1,...,x_m) = E_R(a_1,...,a_n,x_1,...,x_m)\}$. In fact, in general the complement of a Diophantine or an exponential Diophantine set is not itself Diophantine or exponential Diophantine, respectively. This has been proven for Diophantine sets by by Davis Davis 1953, and this proof can be extended to show that there are exponential Diophantine sets whose complements are not exponential Diophantine.

4.2.4 Inequality

One really important result is that inequality is an exponential Diophantine relation:

$$
a < b \iff \exists x\{a+1+x=b\}. \tag{4.7}
$$

Therefore, we can use "$<$" as well as "$>$", "$\leqslant$" and "$\geqslant$" in exponential Diophantine equations. Note that the representation is Diophantine as well, so that "$<$" can be used in Diophantine equations.

4.2.5 Divisibility and congruence

The last two exponential Diophantine relations that we will discuss are those of divisibility and congruence:

$$a|b \iff \exists x\{ax = b\}, \tag{4.8}$$

$$a \equiv b \text{ (mod) } c \iff \exists x\{(a-b)^2 = c^2 x\}. \tag{4.9}$$

For this last equivalence, note that $a \equiv b$ (mod) c precisely when $a-b$ is a multiple of c. Because $a-b$ might be negative while x must be nonnegative, we square both $a-b$ and c to obtain an equivalent result. Note that both representations are also Diophantine, so that we may use both relations in Diophantine equations.

4.3 Exponential Diophantine set operations

Using the fact that disjunction is an exponential Diophantine operation, it is easy to see that the union of two exponential Diophantine sets is exponential Diophantine itself. To see that the intersection of two exponential Diophantine sets is exponential Diophantine as well, we can use an analogue of the conjunction operation in equation 4.4; if we would use equation 4.4 literally, we would require that, for tuples $(a_1, ..., a_n)$ that are in both exponential Diophantine sets, there are numbers $x_1, ..., x_m$ such that both $E_L(a_1, ..., a_n, x_1, ..., x_m) = E_R(a_1, ..., a_n, x_1, ..., x_m)$ and $F_L(a_1, ..., a_n, x_1, ..., x_m) = F_R(a_1, ..., a_n, x_1, ..., x_m)$. However, the variables $x_1, ..., x_m$ that yield a solution to the first equation with tuple $(a_1, ..., a_n)$ need not yield a solution to the second equation with tuple $(a_1, ..., a_n)$. Therefore, we need to introduce m new variables $y_1, ..., y_m$, and demand that:

$$\begin{aligned} E_L(a_1, ..., a_n, x_1, ..., x_m) = E_R(a_1, ..., a_n, x_1, ..., x_m)\& \\ F_L(a_1, ..., a_n, y_1, ..., y_m) = F_R(a_1, ..., a_n, y_1, ..., y_m). \end{aligned} \tag{4.10}$$

This is a generalized exponential Diophantine representation of the intersection between two exponential Diophantine sets.

4.4 Exponential Diophantine properties

A property is called exponential Diophantine if the set of all numbers satisfying this property is an exponential Diophantine set. We can show that the property "odd" is in fact exponential Diophantine:

$$\text{Odd}(a) \iff \exists x\{a = 2x + 1\}. \tag{4.11}$$

We can use exponential Diophantine properties in general exponential Diophantine representations as well.

4.5 Positional notation

For the proof that all listable sets are Exponential Diophantine, we will need to take a look at positional notation; we can write, for any positive base b, every natural number a as

$$a = \Sigma_{k=0}^{\infty} a_k b^k, \tag{4.12}$$

where $a_k < b$ for $a_k, k \in \mathbb{N}$. This means that, if we consider the number a in *base* b, we can write it as a string of digits $...a_3a_2a_1a_0$.

4.5.1 Digit function

Reversely, we can extrapolate the k'th digit a_k of any number a in base b using

$$a_k = \text{Digit}(a, b, k). \tag{4.13}$$

Digit is a generalized exponential Diophantine function:

$$d = \text{Digit}(a, b, k) \iff \exists x, y\{a = xb^{k+1} + db^k + y \quad \& \quad d < b \quad \& \quad y < b^k\}. \tag{4.14}$$

4.5.2 Binomial coefficients

The well-known binomial coefficients can be shown to form an exponential Diophantine set. The binomial coefficients $C_{a,b}$ are defined as the unique numbers that satisfy

$$(u+1)^a = C_{a,a}u^a + C_{a,a-1}u^{a-1} + ... + C_{a,0}, \tag{4.15}$$

for all positive $u \in \mathbb{N}$. This is inconvenient, since we can not use universal quantifiers in exponential Diophantine equations. However, it turns out that we can use (4.15) as a single equation if u is large enough, and this is enough to define the binomial coefficients.

In fact, precisely like in (4.12), we see that the binomial coefficients $C_{a,b}$ are nothing more than the digits of $(u+1)^a$ in base u. So, if we choose u to be larger than any of the $C_{a,b}$, then we can extract the coefficients using Digit. Note that the sum of all binomial coefficients in (4.15) is 2^a (take $u = 1$). Therefore, if we choose $u = 2^a + 1$, then u is indeed greater than any of the coefficients. We can now give a general exponential Diophantine representation of the binomial coefficients:

$$c = \binom{a}{b} \iff \exists u\{u = 2^a + 1 \quad \& \quad c = \text{Digit}((u+1)^a, u, b)\}. \tag{4.16}$$

4.5.3 Kummer's theorem

In 1852, the German mathematician Ernst Kummer found a way to calculate the prime factorization of binomial coefficients of the form

$$\binom{a+b}{b} = 2^{\alpha_2(a,b)}3^{\alpha_3(a,b)}5^{\alpha_5(a,b)}... \tag{4.17}$$

He stated that $\alpha_p(a,b)$ for any prime p can be obtained by writing a and b in base-p notation, and to add them; the number of digit to digit caries you need to perform during this addition is exactly the number $\alpha_p(a,b)$.

To prove Kummer's theorem, note that

$$\binom{a+b}{b} = \frac{(a+b)!}{a!b!}, \tag{4.18}$$

so that $\alpha_p(a,b) = \beta_p(a+b) - \beta_p(a) - \beta_p(b)$, where $\beta_p(k)$ is the exponent of p in the prime factorization of $k!$. This means that

$$\beta_p(k) = \left\lfloor \frac{k}{p} \right\rfloor + \left\lfloor \frac{k}{p^2} \right\rfloor + \left\lfloor \frac{k}{p^3} \right\rfloor + ... \tag{4.19}$$

We can easily see this that there are exactly $\left\lfloor \frac{k}{p} \right\rfloor$ numbers not greater than k that have p as a factor, and there are $\left\lfloor \frac{k}{p^2} \right\rfloor$ numbers not greater than k that have p^2 as a factor, etc. So we have that

$$\alpha_p(a,b) = \Sigma_{l \geqslant 1}\left(\left\lfloor \frac{a+b}{p^l} \right\rfloor - \left\lfloor \frac{a}{p^l} \right\rfloor - \left\lfloor \frac{b}{p^l} \right\rfloor \right) \tag{4.20}$$

The fundamental thing to note is that the l'th summand is either one or zero, depending on whether there is a carry in the $(l-1)$'th digit when adding a and b in base p.

4.5.4 Binary orthogonality

Let a and b be two natural numbers, and consider their binary notation

$$a = \Sigma_{k=0}^{\infty} a_k 2^k, \qquad\qquad b = \Sigma_{k=0}^{\infty} b_k 2^k, \tag{4.21}$$

where $0 \leqslant a_k, b_k \leqslant 1$. We say that a and b are orthogonal if $a_k b_k = 0$, for all $k \in \mathbb{N}$. This is equivalent to saying that there may be no carry in any digit when adding a and b (because there is a carry precisely when the l'th digit is one for both a and b for some l), so Kummer's theorem gives us an exponential Diophantine representation of orthogonality:

$$a \perp b \iff \text{Odd}\left(\binom{a+b}{b}\right). \tag{4.22}$$

4.5.5 Binary masking

Let a and b be natural numbers, and consider their binary notation

$$a = \Sigma_{k=0}^{\infty} a_k 2^k, \qquad b = \Sigma_{k=0}^{\infty} b_k 2^k, \tag{4.23}$$

where $0 \leqslant a_k, b_k \leqslant 1$. The number b is said to *mask* the number a if $b_k \leqslant c_k$ for every $k \in \mathbb{N}$. The masking relation is denoted by "$\preccurlyeq$". Consider the general exponential Diophantine representation

$$a \preccurlyeq b \iff \text{Odd}\left(\binom{b}{a}\right). \tag{4.24}$$

To see why this is an exponential Diophantine representation of the masking relation, suppose that $a > b$. In this case $\binom{b}{a} = 0$, which is not odd. The left hand side is also false in this case. On the other hand, suppose that $a \leqslant b$. Then we can find some natural number c such that $c = b - a$. Then, by Kummer's theorem, $c \perp a$ and thus no carry occurs when adding c and a, or equivalently: $a \preccurlyeq b$.

4.5.6 Digit by digit multiplication (logical and)

Let a, b and c be natural numbers, and consider their binary notation

$$a = \Sigma_{k=0}^{\infty} a_k 2^k, \qquad b = \Sigma_{k=0}^{\infty} b_k 2^k, \qquad c = \Sigma_{k=0}^{\infty} c_k 2^k \tag{4.25}$$

where $0 \leqslant a_k, b_k, c_k \leqslant 1$. We say that c is the result of digit by digit multiplication of a and b if $c_k = a_k b_k$, for all $k \in \mathbb{N}$. Digit by digit multiplication, or logical and, is denoted by "$\wedge$". If $c = a \wedge b$, then

$$c \preccurlyeq a, \qquad c \preccurlyeq b \tag{4.26}$$

$$a - c \perp b - c. \tag{4.27}$$

This is easy to see. But the converse is also true; if $c \preccurlyeq a$ and $c \preccurlyeq b$, then we must have that $c_k \leqslant a_k b_k$ for every $k \in \mathbb{N}$. Suppose there are k for which $c_k < a_k b_k$, and let l be the smallest of those Then we must have that $a_l = b_l = 1$, while $c_l = 0$. This means that the $l-1$'th digits of $a - c$ and $b - c$ are both one (there can be no carry because l is the smallest bit for which $c_l < a_l b_l$), so that $a - c$ and $b - c$ can not be orthogonal, contradiction. This means that (4.26) and (4.27) are also sufficient for c to be the result of digit by digit multiplication of a and b. A generalized exponential representation of digit by digit multiplication is therefore given by

$$c = a \wedge b \iff c \preccurlyeq a \quad \& \quad c \preccurlyeq b \quad \& \quad a - c \perp b - c. \tag{4.28}$$

4.5.7 Diophantine functions

In the proof that exponential Diophantine sets are Diophantine (chapter 6), we will need the notion of Diophantine *functions*; a Diophantine function f is a function whose graph (i.e. the set of all pairs $(\vec{x}, f(\vec{x}))$) is a Diophantine set. We can see that the composition of two Diophantine functions is again Diophantine; let f and g be two Diophantine functions, i.e.

$$a = f(b) \iff \exists x_1, ..., x_m \{D_f(a, b, x_1, ..., x_m) = 0\}, \tag{4.29}$$

$$b = g(c) \iff \exists x_1, ..., x_n \{D_g(b, c, x_1, ..., x_n) = 0\}, \tag{4.30}$$

where $D_f = 0$ and $D_g = 0$ are Diophantine equations. Then the composite function $f \circ g$ is also Diophantine:

$$a = f(g(c)) \iff \exists x_1, ..., x_m, y_1, ..., y_n, b \{D_f(a, b, x_1, ..., x_m) = 0 \quad \& \quad D_g(b, c, y_1, ..., y_n) = 0\} \tag{4.31}$$

$$\iff \exists x_1, ..., x_m, y_1, ..., y_n, b \{(D_f(a, b, x_1, ..., x_m))^2 + (D_g(b, c, y_1, ..., y_n))^2 = 0\} \tag{4.32}$$

5 Listable sets are Exponential Diophantine sets

In this chapter we will prove that all listable sets are exponential Diophantine. For this we will use the theory and structure of (Matiyasevich 2000, ch. 4).
With the definition of an algorithm as a process that can be carried out by a register machine (3.2.1), it is clear that there must also exist some relation between listable sets and register machines. This becomes apparent in the following lemma:

Lemma 5.0.1. *For every listable set $\mathcal{M}$, there is a register machine R that halts on input a precisely when $a \in \mathcal{M}$. Conversely, if a register machine R halts precisely on elements of some set H, then H is a listable set.*

This will become readily apparent; let $\mathcal{M}$ be a listable set. Then, by definition, there is some register machine R that prints out all the elements in $\mathcal{M}$. We now slightly modify this machine to accept an input a, and then let it produce the list $\mathcal{M}$. For every element it produces we check whether it is equal to a; if it is, halt. If it is not, continue producing the list. If we have produced all elements of $\mathcal{M}$, simply repeat the same process indefinitely. We see that this machine halts precisely when $a \in \mathcal{M}$.
Reversely, let R be a register machine, and suppose that it halts precisely on elements in the set H. Then we modify its program slightly to loop over all tuples of the form $(a, N), a, N \in \mathbb{N}$ in some order, for example using Cantor's pairing function; if the machine would not halt after N instructions on input a, go to the next tuple. If the machine would halt, print a. This way, we would end up printing all elements of H, meaning that H must be listable.

Lemma 5.0.1 gives us an excellent way to prove that listable sets are exponential Diophantine sets. The following theorem, together with lemma 5.0.1 would prove that all listable sets are exponential Diophantine.

Theorem 5.0.2. *Given any register machine R, there is an exponential Diophantine equation*

$$E_L(a_1, ..., a_m, x_1, ..., x_n) = E_R(a_1, ..., a_m, x_1, ..., x_n) \tag{5.1}$$

such that the equation has a solution for $(a_1, ..., a_m) = (b_1, ..., b_m)$ precisely when R halts on input $(b_1, ..., b_m)$.

This chapter will be devoted to a proof of this theorem. One way to prove this is to describe the operation of the register machine in terms of exponential Diophantine equations.

The operation of a register machine is a complicated process, where the state of the machine is continuously changing. In order to describe the operation, one may use a *protocol*; a table where each column represents a point in the runtime of the program, such that the variables in a column fully describe the state of the register machine at that time. A protocol looks like this,

	q	...	$t+1$	t	...	0
S_1	$s_{1,q}$	...	$s_{1,t+1}$	$s_{1,t}$	...	$s_{1,0}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
S_m	$s_{m,q}$	...	$S_{m,t+1}$	$s_{m,t}$	...	$s_{m,0}$
R_1	$r_{1,q}$	...	$r_{1,t+1}$	$r_{1,t}$	...	$r_{1,0}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
R_n	$r_{n,q}$	...	$r_{n,t+1}$	$r_{n,t}$	...	$r_{n,0}$
Z_1	$z_{1,q}$	...	$z_{1,t+1}$	$z_{1,t}$	...	$z_{1,0}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Z_n	$z_{n,q}$	...	$z_{n,t+1}$	$z_{n,t}$	...	$z_{n,0}$

where the machine terminates after q steps.

The variables $s_{i,t}$ describe the state of the machine; if the instruction that the machine carries out at time t is instruction S_i, then $s_{i,t} = 1$ and $s_{j,t} = 0$ for $0 < j \leqslant m, j \neq i$. The variables $r_{l,t}$ are the contents of register

R_l at time t. The parameters $z_{l,t}$ are not necessary to describe the state of the machine, but they will help in our calculations; they indicate whether register R_l is empty at time t, so:

$$z_{l,t} = \begin{cases} 0 & \text{if } r_{l,t} = 0 \\ 1 & \text{if } r_{l,t} > 0 \end{cases} \tag{5.2}$$

5.1 Register machine conditions

We now want to find relations between these variables such that these relations have a solution if and only if the register machine terminates on an input a.

5.1.1 Initial conditions

Without loss of generality we can say that the first instruction is going to be S_1, and that all registers are empty except for the first one, which contains the input a. This means:

$$s_{1,0} = 1, \qquad s_{2,0} = ... = s_{m,0} = 0, \tag{5.3}$$

$$r_{1,0} = a, \qquad r_{2,0} = ... = r_{n,0} = 0. \tag{5.4}$$

5.1.2 Step relations

Using one column, we can calculate the next one. For the instruction variables:

$$s_{d,t+1} = \Sigma^0 s_{k,t} + \Sigma^+ z_{l,t} s_{k,t} + \Sigma^- (1 - z_{l,t}) s_{k,t}, \tag{5.5}$$

where:

- Σ^0 is a summation over all instructions of the form $S_k : r_l^+ \implies S_d$
- Σ^+ is a summation over all instructions of the form $S_k : r_l^- \implies S_d, S_j$
- Σ^- is a summation over all instructions of the form $S_k : r_l^- \implies S_j, S_d$

In order to understand this relation it is important to note that, of the variables $s_{1,t}, ..., s_{m,t}$, precisely one is nonzero. This is true for $t = 0$ due to equation 5.3, and this property is propagated using 5.5. Then all 5.5 does is to sum over all possible instructions that route to S_d, so that, if the nonzero instruction from the previous time step routes to S_d, then $s_{d,t+1} = 1$ (and necessarily, because the previous instruction can only route to one instruction, $s_{i,t+1} = 0$ for $i \neq d$).

Similarly for the registers:

$$r_{l,t+1} = r_{l,t} + \Sigma^+ s_{k,t} - \Sigma^- z_{l,t} s_{k,t}, \tag{5.6}$$

where:

- Σ^+ is a summation over all instructions of the form $S_k : r_l^+ \implies S_i$
- Σ^- is a summation over all instructions of the form $S_k : r_l^- \implies S_i, S_j$

5.1.3 Final values

Without loss of generality we can assume that there is only one STOP instruction, since all STOP instructions are identical. We may also assume that this STOP instruction is S_m. We can also assume without loss of generality that the machine is programmed to empty its registers before termination:

$$s_{m,q} = 1, \qquad s_{1,1} = ... = s_{m-1,1} = 0, \tag{5.7}$$

$$r_{1,q} = ... = r_{n,q} = 0. \tag{5.8}$$

5.2 Positional coding

By construction, if the register machine terminates after q steps on input a, we can find values for the variables $s_{i,t}, r_{l,t}, z_{l,t}$ such that they obey conditions (5.1)-(5.7). Conversely, given $a \in \mathbb{N}$, if we can find values for $s_{i,t}, r_{l,t}, z_{l,t}, q$ such that they obey conditions (5.1)-(5.7), then the register machine must halt after precisely q steps on input a. Although it may seem like we are done now, there is one caveat; we have now constructed an indeterminate number of variables, that depends on the run time q of the program. Since the run time of a program may be different for different inputs, we can not define a single Diophantine equation that describes the operation of a register machine on an arbitrary input using this strategy. We must be very careful to avoid this problem.

What we can do to avoid having to deal with this indeterminate number of variables is to 'translate' them into a fixed number of variables. More precisely, we want to find injections $f : \mathbb{N}^m \to \mathbb{N}$ and $g, h : \mathbb{N}^n \to \mathbb{N}$ such that:

$$s_i = f(s_{i,1}, ..., s_{i,m}), \tag{5.9a}$$
$$r_l = g(r_{l,1}, ..., r_{l,n}), \tag{5.9b}$$
$$z_l = h(z_{l,1}, ..., z_{l,n}). \tag{5.9c}$$

One way to do this is to pick a number b, such that b is bigger than all variables $s_{i,t}, r_{l,t} and z_{l,t}$. This allows us to define:

$$s_i = \Sigma_{t=0}^{q} s_{i,t} b^t, \tag{5.10a}$$
$$r_l = \Sigma_{t=0}^{q} r_{l,t} b^t, \tag{5.10b}$$
$$z_l = \Sigma_{t=0}^{q} z_{l,t} b^t, \tag{5.10c}$$

which are easily shown to be injections. Without loss of generality, we can say that $b = 2^{c+1}$ for some $c \in \mathbb{N}$; this allows us to express s_i, r_l and z_l in binary, so that we can use the binary operations and relations defined in chapter 4. Our goal now is to rewrite conditions (5.1)-(5.7) as using the new variables s_i, r_l and z_l.

5.2.1 Zero indicator

We will start with rewriting equation 5.2. In order to do this, we will choose b to be even larger, so that $b/2 = 2^c$ is bigger than all variables $s_{i,t}, r_{l,t} and z_{l,t}$.

Note that $2^c - 1$ is a series of c ones in binary notation, so that $x < 2^c$ is equivalent to the masking relation $x \preccurlyeq 2^c - 1$. Similarly, $x < 2$ is equivalent to the masking relation $x \preccurlyeq 1$. Therefore we have that:

$$r_{l,t} \preccurlyeq 2^c - 1 \tag{5.11}$$
$$z_{l,t} \preccurlyeq 1, \tag{5.12}$$

so that

$$r_l \preccurlyeq d \tag{5.13}$$
$$z_l \preccurlyeq e, \tag{5.14}$$

with

$$d = \Sigma_{t=0}^{q} (2^c - 1) b^t, \tag{5.15}$$
$$e = \Sigma_{t=0}^{q} (1) b^t, \tag{5.16}$$

which follows from repeated application of 5.11 and 5.12 on the terms of 5.10b and 5.10c, respectively. Now consider the number $2^c - 1 + r_{l,t}$. Because $2^c - 1$ can be written as c successive ones in binary, and because $r_{l,t} \leqslant 2^c - 1$ for all l, t, we have that:

$$2^c - 1 + r_{l,t} = \begin{cases} 011...111 \quad \text{(with c ones), if } r_{l,t} = 0 \\ 1 * *... * * * \quad \text{(with c *'s), if } r_{l,t} > 0, \end{cases} \tag{5.17}$$

where $*$ is an unknown bit. The key insight here is that the $c+1$'th bit of this number is precisely our desired value for $z_{l,t}$. This means that

$$2^c z_{l,t} = (2^c - 1 + r_{l,t}) \wedge 2^c, \tag{5.18}$$

and after repeated application to the terms of 5.10c:

$$2^c z_l = (d + r_l) \wedge f, \tag{5.19}$$

where

$$f = \Sigma_{t=0}^{q} 2^c b^t. \tag{5.20}$$

5.2.2 Step relations

In order to write the step relations 5.5 and 5.6, we first note that, if two natural numbers x, y obey the relations $x \leqslant 1$, $y \leqslant 1$, we have that

$$\begin{aligned} 1 \cdot 1 = 1 \wedge 1 = 1, \\ 1 \cdot 0 = 1 \wedge 0 = 0, \\ 0 \cdot 0 = 0 \wedge 0 = 0, \text{ so} \\ x \cdot y = x \wedge y. \end{aligned} \tag{5.21}$$

This is useful, because $s_{i,t} \leqslant 1, z_{l,t} \leqslant 1$ and $(1 - z_{l,t}) \leqslant 1$ for all i, l, t. This means that we can replace multiplication in 5.5 and 5.6 with bitwise multiplication, what makes it easier to rewrite.

For the instruction step relation 5.5, we multiply both sides with b^{t+1} and sum over t from 0 to $q-1$ to obtain:

$$s_i - s_{i,0} = (\Sigma^0 b s_k - b^q s_{k,q}) + (\Sigma^+ b(z_l \wedge s_k) - b^q (z_{l,q} \wedge s_{k,q})) + (\Sigma^- b((e - z_l) \wedge s_k) - b^q (1 - z_{l,q}) \wedge s_{k,q}), \tag{5.22}$$

Using equations 5.7, $s_{k,q}$ must be either zero or represent the STOP instruction, which routes to no other instruction, so that it appears in none of the sums. Therefore, we can simplify:

$$s_i - s_{i,0} = b(\Sigma^0 s_k) + b(\Sigma^+ z_l \wedge s_k) + b(\Sigma^-(e - z_l) \wedge s_k). \tag{5.23}$$

We can use equations 5.3 to obtain:

$$s_i = \begin{cases} b(\Sigma^0 s_k) + b(\Sigma^+ z_l \wedge s_k) + b(\Sigma^-(e - z_l) \wedge s_k) & \text{if } i > 1 \\ 1 + b(\Sigma^0 s_k) + b(\Sigma^+ z_l \wedge s_k) + b(\Sigma^-(e - z_l) \wedge s_k) & \text{if } i = 1 \end{cases} \tag{5.24}$$

Completely similar to the case of s_i, we use 5.4 and 5.8 obtain that

$$r_l = \begin{cases} b r_l + b(\Sigma^+ s_k) - b(\Sigma^- z_l \wedge s_k) & \text{if } l > 1 \\ a + b r_l + b(\Sigma^+ s_k) - b(\Sigma^- z_l \wedge s_k) & \text{if } l = 1 \end{cases} \tag{5.25}$$

5.2.3 Initial conditions

Because the right hand sides of 5.24 and 5.25 are divisible by b in the cases $i > 1, l > 1$ respectively, equations 5.10a and 5.10b guarantee that $s_{2,0} = ... = s_{m,0} = 0$ and $r_{2,0} = ... = r_{n,0} = 0$. In the case where $i = 0$, 5.24 shows that $s_1 = 1 + b \cdot x$, for some $x \geqslant 1$. Therefore it must also be that $s_{1,0} = 1$, and so 5.3 follows from 5.10a and 5.24. In the case that $l = 1$, if we impose that

$$b = 2^{c+1} > 2a, \tag{5.26}$$

then also $r_{1,0} = a$, and initial conditions 5.4 follow from 5.10b and 5.25.

5.2.4 Final values

We can guarantee that $s_{m,q} = 1$ (in condition 5.7) if we impose that

$$s_m = b^q. \tag{5.27}$$

5.3 Listable sets are exponential Diophantine

We have already seen that the machine stopping after q steps on input a is equivalent to there being values to variables $s_{i,t}, r_{l,t}$ and $z_{l,t}$ such that they obey conditions (5.2)-(5.7). This means that we need to show that conditions (5.2)-(5.7) imply conditions (5.12), (5.14)-(5.16), (5.18), (5.19) and (5.23)-(5.26), and vice versa, in order to prove that listable sets are exponential Diophantine; after all, all those conditions are generalized exponential Diophantine equations (or can be easily shown to be in the case of d, e, f), and since the conjunction of two exponential Diophantine equations is an exponential Diophantine equation itself, we can construct an exponential Diophantine equation such as in theorem 5.0.2.
The first implication that, given $a \in \mathbb{N}$, if one can find values for variables $s_{i,t}, r_{l,t}, z_{l,t}$ and q such that they obey conditions (5.2)-(5.7), then one can find values for variables $s_1, ..., s_m, r_1, ..., r_n, z_1, ..., z_n, b, c, d, e, f$ such that they obey conditions (5.12), (5.14)-(5.16), (5.18), (5.19) and (5.23)-(5.26) is by construction.
The converse is also true. Given $a \in \mathbb{N}$, assume we find values for $s_1, ..., s_m, r_1, ..., r_n, z_1, ..., z_n, b, c, d, e, f, q$ such that they obey conditions (5.12), (5.14)-(5.16), (5.18), (5.19) and (5.23)-(5.26). We need a way to obtain values for $s_{i,t}, r_{l,t}$ and $z_{l,t}$. We define:

$$s_{i,t} = \text{Digit}(s_i, b, t), \tag{5.28}$$
$$r_{l,t} = \text{Digit}(r_l, b, t), \tag{5.29}$$
$$z_{l,t} = \text{Digit}(z_l, b, t). \tag{5.30}$$

Note that with this definition, equations (5.10a), (5.10b), and (5.10c) are still valid. Condition (5.2) then follows immediately from (5.19) and (5.20), and conditions (5.3) and (5.4) follow from (5.10a) and (5.24), and (5.10b) and (5.25) respectively, as laid out in section 5.2.3.

In order to prove that (5.5) and (5.6) hold, it is important to make sure that precisely one of the variables $s_{1,t}, ..., s_{m,t}$ is one, and the others are zero. We will show this using induction.
For $t = 0$, we have that $s_{1,t} = 1$, and $s_{2,t}, ..., s_{m,t} = 0$ (follows from (5.3), which we have already shown to hold). Now suppose that precisely one of $s_{1,t}, ..., s_{m,t}$ is equal to one, and the rest is zero. Then, precisely one of the $t+1$'th digits (base b) of the summands s_k in equations (5.24) is equal to one, and the rest is zero. Therefore, no carry across the digits occurs, and we must have that precisely one of $s_{1,t+1}, ..., s_{m,t+1}$ is one, and the rest is zero. We can combine this result together with (5.24) to obtain that (5.5) holds. Together with equation (5.27) we can also guarantee (5.7).
Similarly, precisely one of the $t+1$'th digits (base b) of the summands in equations (5.25) is one, and the rest is zero, except for the first summand (r_l). Due to (5.13), we must have that the value of the $t+1$'th digit of first summand is no greater than $2^c - 1$, and because the value of the other non-zero summand digit is at most one, no carry can occur. This together with (5.25) implies (5.6).

We are now nearly finished; as mentioned previously, we now only need to show that (5.12), (5.14)-(5.16), (5.18), (5.19) and (5.23)-(5.26) are generalized exponential Diophantine equations. In fact, only (5.15), (5.16) and (5.20) are not already generalized exponential Diophantine equations. We can see that they are in fact geometric series, and we obtain the equivalent exponential Diophantine equations:

$$(b-1)d = (2^c - 1)(b^{q+1} - 1), \tag{5.31}$$
$$(b-1)e = b^{q+1} - 1, \tag{5.32}$$
$$(b-1)f = 2^c(b^{q+1} - 1), \tag{5.33}$$

and by conjunction of (5.12), (5.16), (5.18), (5.23)-(5.26) and (5.29)-(5.31) we have obtained the exponential Diophantine equation from theorem 5.0.2. We can conclude that all listable sets are exponential Diophantine sets.

6 Exponential Diophantine sets are Diophantine sets

The final step that we have to prove is that exponential Diophantine sets are Diophantine. As discussed in chapter 2, this step was provided by Matiyasevich 1970, using a theorem by Robinson 1952. In this chapter we will look at Matiyasevich's proof. For this we will use the theory and structure of (Matiyasevich 2000, ch. 3); this chapter in Matiyasevich's notes contains a few minor mistakes that I have implicitly corrected.

Matiyasevich constructed a Diophantine representation of the set

$$U = \{(a, b, c) | c = a^b, a, b, c \in \mathbb{N}\}, \tag{6.1}$$

using a second-order recurrence sequence:

$$\alpha_b(0) = 0, \qquad \alpha_b(1) = 1, \qquad \alpha_b(n+2) = b\alpha_b(n+1) - \alpha_b(n), \tag{6.2}$$

where $b \geqslant 2$. The next few sections will be devoted to the proof of some essential properties of this sequence.

6.1 Properties of α_b

We will start with some basic properties of α_b. The following lemma holds for all $b \geqslant 2$:

Lemma 6.1.1. *The sequence is monotonically increasing, i.e.* $0 = \alpha_b(0) < \alpha_b(1) < ... < \alpha_b(n) < ....$

Proof: We can see this using induction; it is true that $0 = \alpha_b(0) < \alpha_b(1) = 1$. Now suppose that we know that $\alpha_b(k) < \alpha_b(k+1)$. Then, because $b \geqslant 2$, we have that

$$\begin{aligned} \alpha_b(k+2) &= b\alpha_b(k+1) - \alpha_b(k) && (6.3)\\ &> b\alpha_b(k+1) - \alpha_b(k+1) && (6.4)\\ &\geqslant \alpha_b(k+1). && (6.5) \end{aligned}$$

□

The sequence also has an interesting property when $b = 2$:

Lemma 6.1.2. *For all* $n \in \mathbb{N}$*, we have that* $\alpha_2(n) = n$.

Proof: We will prove this using induction. The statement is true by definition for $n = 1$ and $n = 0$. Now suppose that $\alpha_2(k-1) = k-1$ and $\alpha_2(k-2) = k-2$. Then

$$\begin{aligned} \alpha_2(k) &= 2\alpha_2(k-1) - \alpha_2(k-2) && (6.6)\\ &= 2k - 2 - k + 2 && (6.7)\\ &= k. && (6.8) \end{aligned}$$

Therefore, by induction, $\alpha_2(n) = n$ for all n. □

6.1.1 Matrix form

Using matrices, we can rewrite the second order recurrence sequence into a first order relation:

$$A_b(n) = \begin{pmatrix} \alpha_b(n+1) & -\alpha_b(n) \\ \alpha_b(n) & -\alpha_b(n-1) \end{pmatrix}, \tag{6.9}$$

if we take $\alpha_b(-1) = -1$. In this case, we have that

$$A_b(0) = E, \qquad A_b(n+1) = A_b(n)B_b, \tag{6.10}$$

where

$$E = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad B_b = \begin{pmatrix} b & -1 \\ 1 & 0 \end{pmatrix}, \tag{6.11}$$

so that

$$A_b(n) = B_b^n. \tag{6.12}$$

6.1.2 Determinant

The next lemma shows a relation between consecutive elements in the sequence α_b and a Diophantine equation:

Lemma 6.1.3. *Suppose that $x = \alpha_b(n+1), y = \alpha_b(n)$ for some $b, n \in \mathbb{N}, b \geqslant 2$. Then $x^2 - bxy + y^2 = 1$. Conversely, suppose that we find $x, y, b \in \mathbb{N}, b \geqslant 2$ such that $x^2 - bxy + y^2 = 1$. Then $x = \alpha_b(m+1), y = \alpha_b(m)$ or $y = \alpha_b(m+1), x = \alpha_b(m)$ for some $m \in \mathbb{N}$.*

Proof: Let $x = \alpha_b(n+1), y = \alpha_b(n)$, for some $b, n \in \mathbb{N}, b \geqslant 2$. Because both E and B have a unit determinant, we can use the identity $\det(AB) = \det(A) \cdot \det(B)$ to obtain that $\det(A_b(n)) = \det(B_b^n) = \alpha_b^2(n) - \alpha_b(n+1)\alpha_b(n-1) = 1$ for all $b, n \in \mathbb{N}, b \geqslant 2$. This means that

$$\begin{aligned} \alpha_b^2(n) - \alpha_b(n+1)\alpha_b(n-1) &= \alpha_b^2(n+1) - b\alpha_b(n+1)\alpha_b(n) + \alpha_b^2(n) && (6.13)\\ &= x^2 - bxy + y^2 && (6.14)\\ &= 1, && (6.15) \end{aligned}$$

where we used (6.2) to obtain the first equality. We now want to show the converse; suppose we find x, y such that

$$x^2 - bxy + y^2 = 1. \tag{6.16}$$

We want to prove that either

$$x = \alpha_b(m+1), \qquad y = \alpha_b(m), \tag{6.17}$$

or

$$x = \alpha_b(m), \qquad y = \alpha_b(m+1), \tag{6.18}$$

for some $m \in \mathbb{N}$. Note that, because x and y are interchangable in (6.16), we can simply say without loss of generality that $y < x$, and prove that (6.17) holds. We will prove this using induction on y.

Let $x, y \in \mathbb{N}$, and suppose that $x^2 - bxy + y^2 = 1$ and $y < x$. If $y = 0$, then we must have that $x = 1$, and x and y obey (6.17) with $m = 0$.
Now assume $y > 0$, and assume that (6.17) holds for all $k < y$ Then rewriting of (6.16) yields that

$$\begin{aligned} by - x &= \frac{y^2 - 1}{x} \geqslant 0, && (6.19)\\ by - x &= \frac{y^2 - 1}{x} < \frac{y^2}{x} < y. && (6.20) \end{aligned}$$

Now define $x_p = y, y_p = by - x$. Then we have that

$$\begin{aligned} x_p^2 - bx_p y_p + y_p^2 &= y^2 - by(by - x) + (by - x)^2 && (6.21)\\ &= x^2 - bxy + y^2 && (6.22)\\ &= 1. && (6.23) \end{aligned}$$

Using (6.20), we see that $y_p < x_p$, and because also $y_p < y$, by the induction hypothesis there must be some m_p such that

$$x_p = \alpha_b(m_p + 1), \qquad y_p = \alpha_b(m_p). \tag{6.24}$$

Therefore, using $m = m_p + 1$, we have that

$$x = bx_p - y_p = \alpha_b(m+1), \qquad y = x_p = \alpha_b(m). \tag{6.25}$$

□

6.1.3 Divisibility properties I

The following lemma states an interesting divisibility property of the sequence:

Lemma 6.1.4. *Let k and m be natural numbers, with $k > 0$. Then $\alpha_b(k)|\alpha_b(m)$ if and only if $k|m$.*

Proof: Define n, l such that $m = n + kl, 0 \leqslant n < k$. We can write:

$$\begin{aligned}
A_b(m) &= B_b^m \\
&= B_b^{n+kl} \\
&= B_b^n (B_b^k)^l \\
&= A_b(n) A_b^l(k) \\
&= \begin{pmatrix} \alpha_b(n+1) & -\alpha_b(n) \\ \alpha_b(n) & -\alpha_b(n-1) \end{pmatrix} \begin{pmatrix} \alpha_b(k+1) & -\alpha_b(k) \\ \alpha_b(k) & -\alpha_b(k-1) \end{pmatrix}^l \\
&\equiv \begin{pmatrix} \alpha_b(n+1) & -\alpha_b(n) \\ \alpha_b(n) & -\alpha_b(n-1) \end{pmatrix} \begin{pmatrix} \alpha_b(k+1) & 0 \\ 0 & -\alpha_b(k-1) \end{pmatrix}^l (\mathrm{mod} \alpha_b(k)). \qquad (6.26)
\end{aligned}$$

Using this, we can see that

$$\alpha_b(m) \equiv \alpha_b(n) \alpha_b^l(k+1) \bmod(\alpha_b(k)). \qquad (6.27)$$

Now, if we assume that $k|m$, then $n = 0$ so that $\alpha_b(n) = 0$, and the right hand side of (6.27) vanishes. This means that $\alpha_b(k)|\alpha_b(m)$. Conversely, assume that $\alpha_b(k)|\alpha_b(m)$. Then $\alpha_b(m) \equiv 0 \bmod(\alpha_b(k))$, so that $\alpha_b(k)|\alpha_b(n)\alpha_b^l(k+1)$. But by (6.13), $\alpha_b(k)$ and $\alpha_b(k+1)$ must be coprime so that we have that

$$\alpha_b(k)|\alpha_b(n). \qquad (6.28)$$

Because we assumed that $n < k$ and because the sequence $\sigma_b(n)$ is monotonically increasing, we must have that $\alpha_b(n) < \alpha_b(k)$, so that (6.28) is only possible if $\alpha_b(n) = n = 0$, so that $m = kl$ for some l. Hence, $k|m$. □

6.1.4 Divisibility properties II

The next lemma states a second important divisibility property:

Lemma 6.1.5. *Let k and m be natural numbers, with $k > 0$. Then $\alpha_b^2(k)|\alpha_b(m)$ if and only if $k\alpha_b(k)|m$.*

Proof: From (6.1.4) it follows that both sides of (6.1.5) can only be true if $k|m$. So for $k \nmid m$, both sides are false, and (6.1.5) holds. So now suppose that $m = kl$ for some $l \in \mathbb{N}$. In this case:

$$\begin{aligned}
A_b(m) &= B_b^{kl} && (6.29) \\
&= A_b^l(k) && (6.30) \\
&= [\alpha_b(k) B_b - \alpha_b(k-1) E] && (6.31) \\
&= \Sigma_{i=0}^l (-1)^{l-i} \binom{l}{i} \alpha_b^i(k) \alpha_b^{l-i}(k-1) B_b^i, && (6.32)
\end{aligned}$$

where this last step follows from Newton's binomium. If we now evaluate congruence modulo $\alpha_b^2(k)$, all summands except the first two can be ignored:

$$A_b(m) \equiv (-1)^l \alpha_b^i(k-1) E + (-1)^{l-1} l \alpha_b(k) \alpha_b^{l-1}(k-1) B_b \bmod (\alpha_b^2(k)), \qquad (6.33)$$

which implies that

$$\alpha_b(m) \equiv (-1)^{l-1} l \alpha_b(k) \alpha_b^{l-1}(k-1) \bmod (\alpha_b^2(k)). \qquad (6.34)$$

Now assume that $\alpha_b^2(k)|\alpha_b(m)$. Then $\alpha_b(m) \equiv 0 \bmod (\alpha_b^2(k))$, so that we must have that $\alpha_b(k)|l\alpha_b^{l-1}(k-1)$. But, just like before, $\alpha_b(k)$ and $\alpha_b(k+1)$ must be coprime, so

$$\alpha_b(k)|l. \tag{6.35}$$

This together with the fact that $m = kl$ implies that

$$k\alpha_b(k)|m. \tag{6.36}$$

Conversely, suppose that $k\alpha_b(k)|m$. Then $m = kl$ implies that $\alpha_b(k)|l$, which implies that $\alpha_b(m) \equiv 0 \bmod (\alpha_b^2(k))$, so that $\alpha_b^2(k)|\alpha_b(m)$. □

6.1.5 Congruence properties

We need to show a couple more properties of the sequence;

Lemma 6.1.6. *Let b_1, b_2, n be natural numbers with $b_1, b_2 \geqslant 2$, and suppose that $b_1 \equiv b_2 \bmod (q)$ for some positive integer q. Then*

$$\alpha_{b_1}(n) \equiv \alpha_{b_2}(n) \bmod (q). \tag{6.37}$$

Proof: We will use induction; it is clear that this is the case for $n = 0$ and $n = 1$. Now suppose that we know that $\alpha_{b_1}(m) \equiv \alpha_{b_2}(m) \bmod (q)$ for all $m < k$. Then

$$\alpha_{b_1}(k) = b_1\alpha_{b_1}(k-1) - \alpha_{b_1}(k-2) \equiv b_2\alpha_{b_2}(k-1) - \alpha_{b_2}(k-2) \equiv \alpha_{b_2}(k) \bmod (q), \tag{6.38}$$

so we have shown that $\alpha_{b_1}(k) \equiv \alpha_{b_2}(k) \bmod (q)$ as well. □

This result together with the fact that $\alpha_2(n) = n$ for all n proves the following lemma:

Lemma 6.1.7. *Let n and b be natural numbers, with $b \geqslant 2$. Then*

$$\alpha_b(n) \equiv \alpha_2(n) = n \bmod (b-2). \tag{6.39}$$

The final lemma in this section gives a congruence relation between different elements of the sequence:

Lemma 6.1.8. *Let l, m, j be natural numbers. If*

$$n = 2lm \pm j, \tag{6.40}$$

then we have that

$$\alpha_b(n) \equiv \pm\alpha_b(j) \bmod (v), \tag{6.41}$$

where

$$v = \pm\alpha_b(m+1) - \alpha_b(m-1). \tag{6.42}$$

Proof: We see that

$$\begin{aligned} A_b(n) &= B_b^n && (6.43)\\ &= B_b^{2lm\pm j} && (6.44)\\ &= [[B_b^m]^2]^l[A_b(j)]^{\pm 1}, && (6.45)\\ A_b(m) &= \begin{pmatrix} \alpha_b(m+1) & -\alpha_b(m) \\ \alpha_b(m) & \alpha_b(m-1) \end{pmatrix} && (6.46)\\ &\equiv -\begin{pmatrix} -\alpha_b(m+1) & \alpha_b(m) \\ -\alpha_b(m) & -\alpha_b(m+1) \end{pmatrix} \bmod (v). && (6.47) \end{aligned}$$

We note that this latter matrix is $A_b^{-1}(m) \bmod (v)$, so that

$$A_b^2(m) \equiv -E \bmod (v), \tag{6.48}$$

$$A_b(n) \equiv \pm[A_b(j)]^{\pm 1} \bmod (v). \tag{6.49}$$

Note that, because the "$\pm$" at the beginning of the right hand side comes from the exponentiation of $-E$, the two "$\pm$"'s are not correlated, and thus any combination of $+$'s and $-$'s is possible. Looking at the upper right entry of the last matrix (eqn. (6.49)), we see that $\alpha_b(n) \equiv \pm\alpha_b(j) \bmod (v)$, exactly as we required. □

6.2 Diophantine definition

The goal of this section is to show that the expression

$$3 < b \quad \& \quad a = \alpha_b(c) \tag{6.50}$$

is Diophantine.
We will see that (6.50) holds precisely when the following relations hold:

$$3 < b, \tag{6.51}$$

$$u^2 - but + t^2 = 1, \tag{6.52}$$

$$s^2 - bsr + r^2 = 1, \tag{6.53}$$

$$r < s, \tag{6.54}$$

$$u^2 | s, \tag{6.55}$$

$$v = bs - 2r, \tag{6.56}$$

$$w \equiv b \bmod (v), \tag{6.57}$$

$$w \equiv 2 \bmod (u), \tag{6.58}$$

$$2 < w, \tag{6.59}$$

$$x^2 - wxy + y^2 = 1, \tag{6.60}$$

$$2a < u, \tag{6.61}$$

$$2a < v, \tag{6.62}$$

$$a \equiv x \bmod (v), \tag{6.63}$$

$$2c < u, \tag{6.64}$$

$$c \equiv x \bmod (u). \tag{6.65}$$

6.2.1 Sufficiency

The conditions (6.51), (6.52) together with lemma (6.1.3) prove that there is some k such that

$$u = \alpha_b(k). \tag{6.66}$$

Similarly, (6.51), (6.53) and (6.54) together with lemma (6.1.3) imply that there is some positive m such that

$$s = \alpha_b(m), r = \alpha_b(m-1), \tag{6.67}$$

and (6.59) and (6.60) imply that there is some n such that

$$x = \alpha_w(n). \tag{6.68}$$

Now define j, l such that $n = 2lm + j$ or $n = 2lm - j$, and

$$j \leqslant m. \tag{6.69}$$

Now, using lemma (6.1.5) together with (6.55), (6.66) and (6.67) we can see that

$$u | m, \tag{6.70}$$

and it follows from (6.56) and (6.67) that

$$v = \alpha_b(m+1) - \alpha_b(m-1). \tag{6.71}$$

Using lemma (6.1.6), lemma (6.1.8), (6.57), (6.63) and (6.68) we have that

$$a \equiv x \equiv \alpha_w(n) \equiv \alpha_b(n) \equiv \pm\alpha_b(j) \bmod (v), \tag{6.72}$$

and from (6.69), the monotonically increasing property of α_b for $b > 2$, the fact that $b > 2$, and (6.71):

$$2\alpha_b(j) \leqslant 2\alpha_b(m) \leqslant (b-2)\alpha_b(m) < b\alpha_b(m) - 2\alpha_b(m-1) = v, \tag{6.73}$$

which together with (6.62) and (6.72) lets

$$a = \alpha_b(j). \tag{6.74}$$

Now, using (6.65), lemma (6.1.6), lemma (6.1.2) and (6.58) it must be that

$$c \equiv x \equiv \alpha_w(n) \equiv n \bmod (u), \tag{6.75}$$

and from (6.74) and (6.61) and the fact that $n \leqslant \alpha_b(n)$:

$$2j \leqslant 2\alpha_b(j) = 2a < u, \tag{6.76}$$

which is only possible together with (6.75) and (6.64), the definition of $n = 2lm \pm j$ and (6.70) if

$$c = j. \tag{6.77}$$

Finally, (6.74) and (6.77) together imply the second half of (6.50), while the first half follows trivially from (6.51).

6.2.2 Necessity

This proof is going to use the ideas from the previous section; suppose that there are a, b, c that satisfy (6.50). We want to find s, r, u, t, v, w satisfying (6.51)-(6.65).
(6.51) trivially holds. We choose $u = \alpha_b(k)$, selecting some positive k such that (6.61) and (6.64) hold, and such that u is odd. This is possible, since (6.13) guarantees that two successive terms in the sequence $\alpha_b(n)$ are coprime, so that precisely one of them is odd.
Now take $t = \alpha_b(k+1)$,so that by (6.17), (6.52) holds. Now let r and s be defined by (6.67) with $m = uk$. Then using (6.17) we can show that (6.43) and (6.44) hold. Using lemma (6.1.5), (6.55) is also true. We can find v such that it obeys (6.56) and (6.62), because we see that

$$\begin{aligned} bs - 2r &= \alpha_b(m+1) - \alpha_b(m-1) && (6.78)\\ &= b\alpha_b(m) - 2\alpha_b(m-1) && (6.79)\\ &\geqslant 4\alpha_b(m) - 2\alpha_b(m-1) && (6.80)\\ &> 2\alpha_b(m) && (6.81)\\ &\geqslant 2m && (6.82)\\ &\geqslant 2u && (6.83)\\ &> 2a && (6.84) \end{aligned}$$

according to (6.67), the fact that $b > 3$, the monotonically increasing property of the sequence, the fact that $\alpha_b(n) \geqslant n$ for all n, and (6.61).
We see that u and v must be coprime; suppose to the contrary that there is some $d > 1$ such that $d|u$ and $d|v$. Then certainly (6.55) $d|s$, and (6.56) $d|2r$. We chose u to be odd however, so we must have that $d|r$, and because s and r are coprime (6.53), $d = 1$, contradiction. Therefore, u and v are coprime. This means that we can find w satisfying (6.57), (6.58) and (6.59) using the Chinese Remainder Theorem.
Let $x = \alpha_w(c), y = \alpha_w(c+1)$. Then (6.60) holds. We know from lemma (6.1.6) and (6.57) that

$$x = \alpha_w(c) \equiv \alpha_b(c) = c \bmod(v), \tag{6.85}$$

so we have that (6.63) is obeyed. Now from (6.1.7) it follows that

$$x \equiv c \bmod(w-2), \tag{6.86}$$

which together with (6.58) implies (6.65).

6.2.3 Exponentiation is Diophantine

Now having defined a Diophantine representation for the sequence α, we will construct a generalized Diophantine representation for exponentiation.
The eigenvalues λ of the matrix B_b obey the relation

$$\lambda^2 - b\lambda - 1 = 0. \tag{6.87}$$

Now let m, q be such that $\lambda \equiv q \bmod(m)$. This means that

$$q^2 - bq + 1 \equiv 0 \bmod (m), \tag{6.88}$$

so we can for example choose $m = bq - q^2 - q$. We find the eigenvectors of B_b modulo m:

$$B_b \begin{pmatrix} q \\ 1 \end{pmatrix} = \begin{pmatrix} b & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} q \\ 1 \end{pmatrix} \equiv q \begin{pmatrix} q \\ 1 \end{pmatrix} \bmod(m). \tag{6.89}$$

Therefore, we get that

$$A_b(r) \begin{pmatrix} q \\ 1 \end{pmatrix} = B_b^r \begin{pmatrix} q \\ 1 \end{pmatrix} \tag{6.90}$$

$$\equiv q^r \begin{pmatrix} q \\ 1 \end{pmatrix} \bmod (m), \tag{6.91}$$

so that, for example,

$$q\alpha_b(r) - \alpha_b(r-1) \equiv q^r \bmod (m). \tag{6.92}$$

This means that, if we have that $q^r < m$:

$$p = q^r \iff m \quad \& \quad q\alpha_b(r) - \alpha_b(r-1) \equiv p \bmod (m). \tag{6.93}$$

The condition that $q^r < m$ is guaranteed by the choice of b:

$$b = \alpha q + 4(r+1) + q^2 + 2, \tag{6.94}$$

if $q > 0$. If it is the case that $q = 0$ then we can treat that separately, so that we obtain the generalized Diophantine equation for exponentiation:

$$p = q^r \iff (q = 0 \& r = 0 \& p = 1) \wedge \tag{6.95}$$

$$(q = 0 \& 0 < r \& p = 0) \wedge \tag{6.96}$$

$$(\exists b, m\{b = \alpha_{q+4}(r+1) + q^2 + 2 \quad \& \tag{6.97}$$

$$m = bq - q^2 - 1 \& \tag{6.98}$$

$$p < m \& \tag{6.99}$$

$$p \equiv q\alpha_b(r) - (b\alpha_b(r) - \alpha_b(r+1)) \bmod (m)\}) \tag{6.100}$$

Using exponentiation, it is easy to show that all exponential Diophantine equations are Diophantine; if we call the Diophantine equation corresponding to the above Diophantine representation of exponentiation $A(a, b, c, x_1, ..., x_n)$, i.e.

$$a^b = c \iff \exists x_1, ..., x_n, A(a, b, c, x_1, ..., x_n) = 0, \tag{6.101}$$

then we can use that to transform any exponential Diophantine equation into a Diophantine equation. For example, take the exponential Diophantine equation $(x+y)^{(y+z)^{(2x^2)}} + 3^x = 7x^{(z+3x)}$, then we can transform it into the following equivalent Diophantine equation:

$$[A(x+y, y+z, s, x_1, ..., x_n)]^2 + [A(s, 2x^2, s', x_1', ..., x_n')]^2 \tag{6.102}$$

$$+ [A(3, x, s'', x_1'', ..., x_n'')]^2 \tag{6.103}$$

$$+ [A(7x, z+3x, s''', x_1''', ..., x_n''')]^2 \tag{6.104}$$

$$+ [s' + s'' - s''']^2 = 0. \tag{6.105}$$

Using this transformation, we can transform the representation of any exponential Diophantine set into a Diophantine equation, showing that all exponential Diophantine sets are Diophantine, and thereby proving that, given a Diophantine equation, it is not always possible to devise a process that determines whether the equation has a solution or not.

7 Appendix A - Definition of an algorithm and computability

This appendix provides a brief history of the notion of calculability, and a motivation for our definition of an algorithm. It is the result of a literature study, and the reader is invited to read more about the subject in (Hilbert D. 1928), (Gödel 1931), (Church 1936) and (Turing 1937).

Although any mathematician will have a basic understanding of what an algorithm is, it is important from a formal point of view to define precisely what we mean by an algorithm. The definition we will use is the following:

Definition 7.0.1. An **algorithm** is a process that can be carried out by a Turing machine.

What will follow is a motivation for this definition.

In the twentieth century, mathematicians struggled with the *Entscheidungsproblem* Hilbert D. 1928, which asks whether there is an algorithm that, given a set of axioms and a statement as input, outputs 'yes' or 'no', dependent on whether the statement follows from the axioms or not. But before even beginning to find an answer to this problem, mathematicians had to agree on the definition of an algorithm.

Intuitively, an algorithm for a function f is any simple process that accurately returns the function value $f(x)$ given an input x in finite time. This notion can be made more precise using effective methods:

Definition 7.0.2. An **effective method** M on a class of problems P is a procedure satisfying the following rules:

1. There is a finite number of finite allowed instructions.
2. Given a problem $p \in P$, M always returns a correct answer after a finite number of instructions.

An algorithm for a function f then is an instance of an effective method, where P consists of problems of the form: *"Given an input $x \in \mathbb{D}(f)$, find $f(x)$"*.

Ideally there would be such an algorithm for every function. This is not at all clear however, and in fact it turns out that this is not the case. A function for which there is an algorithm is called *computable* or *algorithmically calculable*. Now all we need to do to arrive at a formal definition of an algorithm is to find an effective method on the class of all computable functions.

A-1: Computable functions

Several attempts have been made to describe the class of computable functions. These include:

1. The class of general recursive functions, suggested by Herbrand and described by Gödel (1931) Gödel 1931
2. The class of λ-definable functions using λ-calculus, created by Church (1936) Church 1936
3. The class of Turing computable functions, devised by Turing (1936) Turing 1937

Although these classes of functions seem different, they turn out to describe the same class. For example, the equivalence between Turing computable functions and λ-definable functions is proven by Turing in the same paper in which he defined Turing computable functions (Turing 1937).
Although the concept of computability remains vague, the fact that three different approaches to finding the class of computable functions yield the same result has led to the belief that the class of functions as obtained by any of the methods I, II or III coincides with the class of computable functions. This belief has led to the Church-Turing Thesis:

Thesis 7.1. ***Church-Turing thesis*** *- A function on the positive integers is algorithmically calculable if and only if it is calculable by a Turing machine (or equivalently, λ-definable or recursive).*

For the sake of being able to define an algorithm, we will gladly accept the Church-Turing thesis. The thesis states that any process on a Turing machine can be considered an algorithmically calculable function, and that any algorithmically calculable function can be computed on a Turing machine; this means that our definition of an algorithm as any process that can be carried out by a Turing machine is justified.

References

Church, A. (1936). "An unsolvable problem of elementary number theory". In: *American J. of Math.* 58.2, pp. 345–363.

Davis M. Putnam H., Robinson J. (1961). "The decision problem for exponential Diophantine equations". In: *Ann. Math.* 74.2, pp. 425–436.

Davis, M. (1953). "Arithmetical problems and recursively enumerable predicates". In: *J. of Symbolic Logic* 80.1, pp. 33–41.

Gödel, K. (1931). "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme". In: *I, Monatsh Math. Phys.* 38. Translation: B. Meltzer, Dover publications, inc. (1992), pp. 173–198.

Hilbert D., Ackermann W. (1928). *Grundzüge der theoretischen Logik*. Springer.

Hilbert, D. (1900). "Mathematische Probleme. Vortrag, gehalten auf dem internationalen Mathematiker Kongress zu Paris 1900". In: *Nachrichten von der Königl. Gesellschaft der Wissenschaften*. English translation: Dr. M.W. Newson, Bull. Amer. Math. Soc., 8 (1902), pp. 437-479, Reprinted: Bull. Amer. Math. Soc. 37(4) (2000) pp. 407-436.

Matiyasevich, Yu.V. (1970). "Diofantovost' perechislimykh mnozhestv". In: *Dokl. AN SSSR* 191.2. English translation: Soviet Math. Doklady, **11**(2), 354-358 (1970), pp. 278–282.

– (2000). *On Hilbert's tenth problem*. Ed. by Michael Lamoureux. Pacific Institute for the Mathematical Sciences.

Minsky, M.L. (1967). *Computation: Finite and infinite machines*. 10th. Prentice-Hall, inc.

Poonen, Bjorn (2008). "Undecidability in number theory". In: *Notices Amer. Math. Soc.* 55.3, pp. 344–350.

Robinson, J. (1952). "Existential definability in arithmetic". In: *Trans. Amer. Math. Soc.* 72, pp. 437–449.

Turing, A.M. (1937). "On computable numbers, with an application to the Entscheidungsproblem". In: *Proc. of the London Math. Soc.* 42.2. Correction: **43** (1937), 544-546, pp. 230–265.