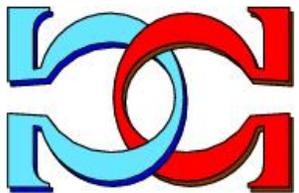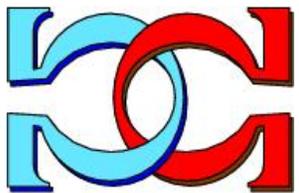# CDMTCS
Research
Report
Series
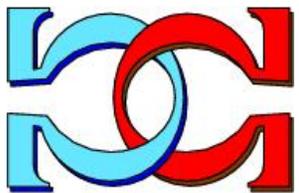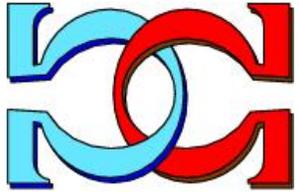
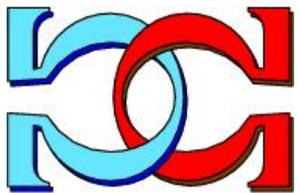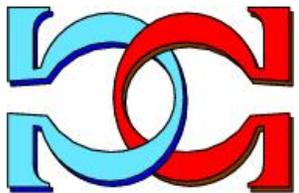# The Riemann Hypothesis in Computer Science

Yuri Matiyasevich
Steklov Institute of Mathematics,
St. Petersburg (POMI), Russia

Centre for Discrete Mathematics and
Theoretical Computer Science

The Riemann Hypothesis in Computer Science

Yu. Matiyasevich

Steklov Institute of Mathematics at St.Petersburg (POMI)
27, Fontanka, St.Petersburg, 191023, Russia
yumat@pdmi.ras.ru

Abstract. The Riemann Hypothesis is reformulated as the statement that particular explicitely presented register machine with 24 registers and 128 instructions never halts.

Keywords: Riemann Hypothesis, register machine

1. Introduction

Alan Turing, one of the founders of modern Computer Science, during the whole his lifetime was fond of Number Theory. His very last paper that he saw published, namely, [36], was in Number Theory. That paper[1] was devoted to computer verification of the celebrated *Riemann Hypothesis*; nowadays it is one of the seven *Millenium problems* [2].

The Hypothesis predicts the positions of complex zeroes of so called *Riemann's zeta function*. This meromorphic function can be defined for $\Re(s) > 1$ by a *Dirichlet series*,

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}. \tag{1}$$

L. Euler gave another definition of this function,

$$\zeta(s) \;=\; \prod_{p \text{ prime}} \frac{1}{1 - p^{-s}}. \tag{2}$$

The equality of the right-hand sides in (1) and (2), known as *Euler identity*, is essentially an analytical form of the *Fundamental Theorem of Arithmetic*, stating that every natural number can be represented in a unique way as a product of powers of prime numbers. This is the reason why Riemann's zeta function plays so important role in the study of prime numbers.

---

[1] More details about Turing's contribution to Number Theory can be found in [8, 7, 17, 9, 4].

In particular, B. Riemann proved in his seminal paper [33] that

$$\pi(x) = \mathrm{Li}(x) - \frac{1}{2}\mathrm{Li}(x^{\frac{1}{2}}) + \sum_{\zeta(\rho)=0} \mathrm{Li}(x^{\rho}) + \text{small terms}. \qquad (3)$$

Here $\pi(x)$ denotes the number of primes below some bound x,

$$\mathrm{Li}(x) = \int_2^x \frac{1}{\ln(t)}\mathrm{d}t. \qquad (4)$$

and the summation is taken over non-real zeroes of Riemann's zeta function. According to the Riemann Hypothesis, all these zeros should have real parts equal to 1/2. In terms of the function $\pi(x)$ the Hypothesis can be reformulated as

$$\pi(x) - \mathrm{Li}(x) = O(x^{\frac{1}{2}}\log(x)). \qquad (5)$$

The Riemann Hypothesis has many other important corollaries in Number Theory and, more surprisingly, in Computer Science as well. In particular, the best known today deterministic test for the primality of a number p has complexity $\Omega(p^6)$; but already in 1976 G. L. Miller [29] proposed an algorithm of complexity $O(p^4)$ *assuming the validity of the (extended) Riemann Hypothesis.*

Table 1 shows results of computational verification of the Riemann Hypothesis for the initial (pairs of conjugate) zeros of the zeta function. It should be emphasized that, while it was done via finite computation with numbers of finite accuracy, the computations present mathematically rigourous proofs that the real parts of these zeros are *exactly* equal to 1/2.

Reported in [36] contribution of Turing, that is, about zeros from 1042nd to 1104th, does not look impressive compared neither with his predecessor no with his follower. However, it was a milestone in numerical verification of the Riemann Hypothesis. On the one hand, it was one of the first usage of an electronic computer for proving non-trivial mathematical statements. But more important is the following: for performing his computation Turing developed an efficient method for testing the Riemann Hypothesis. It is know in Number Theory as *Turing method* and all subsequent computations, up to our days, are based on this method.

In a typical paper about Turing one reads that he has published only two number-theoretical paper, [37, 36]. This is not quite so. Turing dealt with the Riemann Hypothesis also in his Ph.D. thesis which was later published as [35]. There he introduced the notion of *number-theoretical theorems*. Turing wrote:

> By a number-theoretic theorem we shall mean a theorem of the form "$\theta(x)$ vanishes for infinitely many natural numbers x", where $\theta(x)$ is a primitive recursive function. ... An alternative form for number-theoretic theorems is "for each natural number

Year	Number of zeroes	Author
1903	15	J. P. Gram
1914	79	R. J. Backlund
1925	138	J. I. Hutchinson
1936	1041	E. C. Titchmarsh
1953	**1104**	**A. M. Turing**
1956	25000	D. H. Lehmer
1958	35337	N. A. Meller
1966	250000	R. S. Lehman
1968	3500000	J. B. Rosser, J. M. Yohe, L. Schoenfeld
1977	40000000	R. P. Brent
1979	81000001	R. P. Brent
1982	200000001	R. P. Brent, J. van de Lune, H. J. J. te Riele, D. T. Winter
1983	300000001	J. van de Lune, H. J. J. te Riele
1986	1500000001	J. van de Lune, H. J. J. te Riele, D. T. Winter
2004	900000000000	S. Wedeniwski
2004	10000000000000	X. Gourdon

Table 1: Numerical verification of the Riemann Hypothesis

x there exists a natural number y such that $\phi(x,y)$ vanishes", where $\phi(x,y)$ is primitive recursive.

Respectively, a problem is called number-theoretical if its solution can be given in the form of a number-theoretical theorem. It is easy to see that the set of such problems is exactly the class Π_2^0 from the *arithmetical hierarchy*.

As one of the examples of a number-theoretical problems Turing proves that the Riemann Hypothesis can be reformulated as Π_2^0 statement.

It is interesting to note that Turing didn't believe in the validity of the Riemann Hypothesis. He wrote in [36]:

> The calculations were done in an optimistic hope that a zero would be found off the critical line [where $\Re(s) = 1/2$], and the calculations were directed more towards finding such zeros than proving that none existed.

The class of arithmetical statements which can be refuted by a finite calculation is Π_1^0, and in 1958 G. Kreisel improved Turing's result by constructing a Π_1^0 formula equivalent to the Riemann Hypothesis[2].

---

[2] This was just a particular application of a very general technique developed by Kreisel in [25].

Neither Turing's no Kreisel's reformulations of the Riemann Hypothesis immediately attracted attention of specialists in Number Theory. The situation changed in 1970 when the author made the last step in the proof of what is nowadays referred to as *DPRM-theorem*[3]. This theorem establishes that every formula from Π_1^0 with parameters $a_1, \ldots, a_m$ is equivalent to a formula of the special form

$$\forall x_1 \ldots x_n [P(a_1, \ldots, a_m, x_1 \ldots x_n) \neq 0] \qquad (6)$$

where P is a polynomial with integer coefficients.

Together with the above mentioned result of Kreisel, DPRM-theorem has the following corollary: *one can construct a particular polynomial $R(x_1 \ldots x_n)$ with integer coefficients such that the Riemann Hypothesis is equivalent to the statement that Diophantine equation*

$$R(x_1 \ldots x_n) = 0 \qquad (7)$$

has no solutions.

A method for an actual construction of such an equation (7) was described in [19, Section 2]. Later a simplified version was presented in [3, Section 6.4]; more details are supplied in [11]; both methods are discussed in [32].

DPRM-theorem was worked out as a tool to establish the undecidability of Hilbert's 10th problem. It is one of the 23 mathematical problems posed by D. Hilbert in 1900 in [23]. In this problem he asked for an algorithm for recognizing whether given arbitrary Diophantine equation has a solution.

The Riemann Hypothesis is a part of Hilbert's 8th problem. Now an equation (7) shows that this Hypothesis is a very special case of the 10th problem; such a relationship (found via the Computability Theory) between the 8th and 10th Hilbert's problems seems have never been anticipated by specialists in Number Theory.

Hardly we can hope to prove (or refute) the Riemann Hypothesis by examining corresponding Diophantine equation (7). But we can look at such reformulation from a different point of view. Namely, besides the formal proof of the undecidability of Hilbert's 10th problem, we have an informal "evidence" of the difficulty of Diophantine equations – for some of them one cam easily prove that they are equivalent to the tricky Riemann Hypothesis.

But how could one measure the difficulty of a mathematical problem? C. S. Calude, E. Calude, and M. J. Dinneen ([15], for further development see [13, 14, 12]) suggested that the complexity of a statement from Π_1^0 can be defined as the complexity of the simplest machine (or program) that never

---

[3] After M. Davis, H. Putnam, J. Robinson and Yu. Matiyasevich; for detailed proofs see, for example, [1, 3, 27, 18, 24]

```
                    ┌─────────────────────────┐
                    │          NDDM           │
                    ├─────────────────────────┤
   input            │                    ?     │            guess
  ──────────▶       │  P(a, x₁, ..., xₘ) = 0  │   ◀──────   x₁, ..., xₘ
     a              │                         │
                    ├────────────┬────────────┤
                    │    YES     │    NO      │
                    └─────┬──────┴─────┬──────┘
                          │            │
                          ▼            └────────▶
                      accept a            reject
```
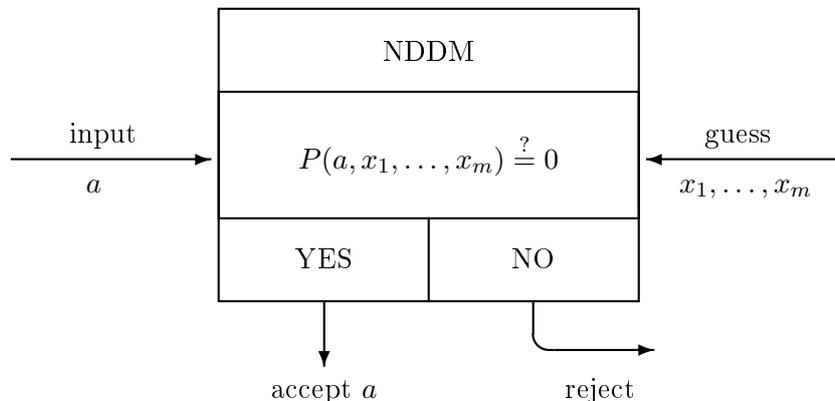
Figure 1: Non-Deterministic Diophantine Machine

halts if and only if the statement is true. Among other famous mathematical problems they estimated (from above) the complexity of the Riemann Hypothesis. Of course, such a bound heavily depends on our current level of knowledge, and it will drastically fell down if someday someone proves or refutes the Riemann Hypothesis.

The numerical value of such complexity measure depends also on the formalism used for describing computations. L. Adleman and K. Manders [6] introduced the notion of *Non-Deterministic Diophantine Machine*, NDDM for short. Each such machine is specified by a parametric Diophantine equation

$$P(a, x_1, \ldots, x_n) = 0 \tag{8}$$

and works as follows: on input a it guesses the numbers $x_1, \ldots, x_n$ and then checks (8); if the equality holds, then a is accepted.

The DPRM-theorem is exactly the statement that NDDMs are as powerful as, say, Turing machines[4]. Thus the difficulty of any problem from Π_1^0 can be measured by any complexity measure (such as the number of the unknowns and the degree) of the Diophantine equation from corresponding NDDM. In particular, equations equivalent to the Riemann Hypothesis (described, for example, in [19, 3, 11]) can perform this role with respect to the Hypothesis.

In [15, 14, 16] a version of *register machines* was used for estimating the complexity of mathematical problems. Such models of computational

_____

[4]The crucial question is whether NDDMs are as *efficient* as Turing machines. If they are, then NDDMs could be used for the study of $P \overset{?}{=} NP$ problem. Partial progress in this direction was archived in [6] but this intriguing question still remains open.

devices were proposed in 1961 by J. Lambek [26], by Z. A. Melzak [28], and by M. L. Minsky [30] (see also [31]). A register machine has a finite number of *registers* capable to contain arbitrary large non-negative integers. Types of possible instructions can varies. In [15, 14, 16] they are rather powerful: assigning an arbitrary value to a register, adding the values of two registers, conditional branches and calling subroutines. The Riemann Hypothesis is presented in [16] by a program with 178 such instructions (this is an improvement over machine with 290 instructions provided in [15]).

More recently, A. Yedidia and S. Aaronson [38] constructed a classical Turing machine with two-letter tape alphabet which, having started with the empty tape, will never halt if and only if the Riemann Hypothesis is true. Their machine has 5372 state. Its construction was based on the reformulation of the Riemann Hypothesis used in [19] for constructing corresponding Diophantine equation (7). The usage of another reformulation presented in [3, Section 6.4] reduced the number of states to 744 (see [5]).

Register machines constructed in [15, 16] are also based on the same reformulation of the Riemann Hypothesis from [3]. In the next Section we introduce a different method of refuting the Riemann Hypothesis (if it false) by a finite computation. It is presented in two forms: as the Python program from Figure 2 and as the register machine from Figure 3. This machine has only 128 instructions, and they are of two very simple types in style of [26] and [30]: to increment or decrement a register by 1. As it was remarked in [30], register machines with such primitive instructions can be viewed as Turing machines with several always empty semi-infinite tapes (only the ends of the tapes are marked with a special symbol).

2. New construction

We start with the well-known reformulation of the Riemann Hypothesis via *Chebyshev psi function*,

$$\psi(n) = \ln(\mathrm{q}(n)), \tag{9}$$

where $\mathrm{q}(n)$ is the least common multiple of numbers $1, \dots, n$. Functions $\pi(n)$ and $\psi(n)$ are closely related but the latter allows simpler (without integral (4)) reformulation of the Riemann hypothesis. For our goal it better to state the necessary and the sufficient conditions separately. Namely,

- the Riemann Hypothesis implies that for all $n > 1$

$$\psi(n) - n < \frac{1}{25} n^{\frac{1}{2}} \ln(n)^2; \tag{10}$$

- *if for some constant C for all sufficiently large n*

$$\psi(n) - n < C n^{\frac{1}{2}} \ln(n)^2 \tag{11}$$

then the Riemann Hypothesis holds.

6

The bound $1/(8\pi)$ on the absolute value of the left-hand side in (10) for $n \geq 74$ was given in [34] (see also [9, Theorem 4.6]); it is easy to verify that one-sided inequality (10) holds for $n = 2, \ldots, 73$ as well. The sufficiency of the condition (11) follows from the well-known $\Omega_{\pm}$-result about the psi function (see, for example, [9, Theorem 4.8]). We shall use the fact that whenever $C > 1/25$, the sufficient condition (11) is weaker than the necessary condition (10).

The main difficulty in the implementation of these conditions via computational devices is caused by the necessity to calculate the real-valued natural logarithms, in (9) and in (10)–(11). Methods for overcoming similar difficulty when constructing Diophantine equation (7) were proposed in [19] and [3] and then adopted in [15, 16] and in [38] for machines. The methods used in this paper are quite different. It is more oriented on computations because main calculated quantities are defined by recursion.

First, instead of computing the *natural* logarithm in (9), we shall work with (the integer part of) the *binary* logarithm,

$$l(n) = \lfloor \log_2(q(n)) \rfloor. \tag{12}$$

Clearly,

$$0 \leq \log_2(q(n)) - l(n) < 1. \tag{13}$$

However, we need to transform the binary logarithms into natural ones; to this end we shall calculate (approximate value of) the natural logarithm, but of single number only, namely, of 2. Let

$$b(n) = \sum_{k=1}^{n-1} (-1)^{k+1} k^{-1}, \tag{14}$$

so $\ln(2) = b(\infty)$. For $n \geq 30$ we have the elementary inequality

$$\left| \ln(2) - b(n) \right| < \frac{3}{5} n^{-1}. \tag{15}$$

Together with (13) this implies that for $n \geq 30$

$$\left| \psi(n) - b(n)l(n) \right| < \frac{1}{25} n^{\frac{1}{2}} \ln(n)^2. \tag{16}$$

Actually, in order to deal with integers only, we shall calculate not $b(n)$ but its multiple

$$d(n) = \frac{(2n-2)!!}{2} b(n). \tag{17}$$

This can be done via the recurrent relations

$$d(1) = 0, \quad d(n+1) = 2nd(n) - (-1)^n (2n-2)!!. \tag{18}$$

7

Our elimination of the logarithms in (10)–(11) is based on the *Prime Number Theorem*[5] stating that function $\pi(n)$ growth approximately as $n/\ln(n)$. We do not need to know the *exact* asymptotic behavior of this function but we need *explicit* lower and upper bounds valid for sufficiently large n. We shall use the *Chebyshev type*[6] *inequalities*: for $n \geq 30$

$$1 < \frac{\pi(n)}{n/\ln(n)} < \frac{13}{10}. \tag{19}$$

The same double factorial, $(2n-2)!!$ from (17), will be used for calculation of an approximation to $n^{\frac{1}{2}}$. Namely, according to the *Stirling formula*, for $n \geq 30$

$$\frac{9}{2}n^{5/2} < \frac{(2n+3)!!}{(2n-2)!!} < 5n^{5/2}. \tag{20}$$

Now (10), (16), (19), and (20) imply that for $n \geq 30$

$$\pi(n)^2 \left(\mathrm{d}(n)\mathrm{l}(n) - \mathrm{f}_0(n)\right) < \mathrm{f}_3(n) \tag{21}$$

where

$$\mathrm{f}_0(n) = \frac{(2n)!!}{2!!}, \qquad \mathrm{f}_3(n) = \frac{(2n+3)!!}{5!!}. \tag{22}$$

In fact, the inequality (21) holds for $n = 1, \ldots, 29$ as well, which can be verified by a direct numerical calculation.

In its turn, the inequality (21), together with (16), (19), and (20), implies (11) with $C = 2/5$. Thus the fulfilment of the inequality (21) for all n is a necessary and sufficient condition for the validity of the Riemann Hypothesis.

The least common multiple, $\mathrm{q}(n)$ from (21), can be calculated via the recurrent equations:

$$\mathrm{q}(1) = 1, \quad \mathrm{q}(n+1) = (n+1)\mathrm{q}(n)/\mathrm{g}(n+1) \tag{23}$$

where

$$\mathrm{g}(m) = \mathrm{GCD}(m, \mathrm{q}(m-1)). \tag{24}$$

Calculated by *Euclidean algorithm* greatest common divisor (24) can be used also for calculating $\pi(n)$ via the recurrent relations

$$\pi(1) = 0, \quad \pi(n+1) = \begin{cases} \pi(n) + 1, & \text{if } \mathrm{g}(n) = 1, \\ \pi(n), & \text{otherwise.} \end{cases} \tag{25}$$

---

[5] The theorem was proved independently by J. Hadamard [22] and by Ch.-J. de la Vallée Poussin [20].

[6] Other inequalities, $0.921 \cdots < \pi(n)/(n/\ln(n)) < 1.105\ldots$, are attributed in many papers and books to P. L. Chebyshev as valid for all $n \geq 30$. It was indicated in [10] that in fact these inequalities fail, for example, for $n = 100$. In reality, Chebyshev had proved inequalities with such bounds but for the ratio $\psi(n)/n$. The validity of (19) can be deduced from sharper bounds known for sufficiently large n (see, for example, [21]) and numerical verification for remaining smaller values of n.

```
from math import  gcd
d=m=p=0
f0=f1=f3=n=q=1
while p**2*(m-f0)<f3:
  d=2*n*d-(-1)**n*f1
  n=n+1
  g=gcd(n,q)
  q=n*q/g
  if g==1: p=p+1
  m=0; q2=q
  while q2>1:
    q2=q2//2; m=m+d
  f1=2*f0
  f0=2*n*f0
  f3=(2*n+3)*f3
```

Figure 2: Python program that never halts if and only the Riemann Hypothesis is true

At last, functions (22) are calculated via the natural recurrent relations:

$$f_0(1) = 1, \quad f_0(n+1) = (2n+2)f_0(n), \tag{26}$$

$$f_3(1) = 1, \quad f_3(n+1) = (2n+5)f_3(n). \tag{27}$$

Combining the recursions (18), (23), and (25)–(27), we come to the following result: *the Riemann Hypothesis is valid if and only if the program on Figure 2 never halts.*

Figure 3 presents a register machine that never halts if and only if the Riemann Hypothesis is true. The machine has 24 registers D, D1, D2, F0–F7, G, G1, M, N, N1, N2, P, P1, Q, Q1–Q3, S which are empty when the machine starts. Instructions have two form: either

$$\langle\texttt{label}\rangle\langle\texttt{register}\rangle\texttt{++}[\langle\texttt{next}\rangle] \tag{28}$$

or

$$\langle\texttt{label}\rangle\langle\texttt{register}\rangle\texttt{--}\langle\texttt{jump}\rangle[\texttt{: }\langle\texttt{next}\rangle] \tag{29}$$

In the former case performing instruction label, the machine increases register by 1 and goes to performing instruction next. In the latter case, the machine tries to decrease register by 1 and then to go to instruction next; however if register was empty, its content does not change and the machine proceeds to performing instruction jump. In the case when next=label+1, the instruction can be abridged by omitting next. The machine starts from instruction 1; there is no instruction 0 so the machine halts whenever it should go to instruction next=0 or jump=0.

```
1F0++           33G++31     65N1++      97Q2++96
2F1++           34Q2--41    66Q3--69    98F2--101
3F3++           35Q2++      67Q++       99F0++
4N++            36G--31     68Q1++66    100M--98:98
5Q++            37N2++36    69Q--71     101P--108
6N--14          38N2++      70Q3++69    102P1++
7N1++           39G--31     71F4--74    103M--106
8D--11          40Q2++39    72F0++      104F7++
9D1++           41G--46     73F2++71    105D2++103
10D2++8         42P++       74F0--77    106F7--101
11D1--6         43G--46     75F4++      107M++106
12D++           44P--45     76F2++74    108P1--115
13D2++11        45G++       77F5--80    109P++
14S--17         46G++       78F3++      110D2--113
15F1--20        47F0--51    79F6++77    111F7++
16D2--15:15     48F1++      80F3--64    112N2++110
17S++           49F1++      81F5++      113F7--108
18F1--20        50F4++47    82F6++80    114D2++113
19D2++18        51G--54     83Q1--86    115F6--0
20D--21:20      52Q3--57    84Q++       116F3++
21D2--24        53G1++51    85Q2++83    117N2--118:115
22D++           54Q++       86Q2--89    118F6--120
23D1++21        55G1--51    87Q2--89    119F3++118
24N1++          56G++55     88Q1++86    120D1--121:120
25Q--28         57Q--59     89Q1--98    121D2--122:121
26Q2++          58Q3++57    90Q2++      122F4--123:122
27Q3++25        59F3--64    91D1--94    123F5--124:123
28N1--31        60F5++      92D2++      124M--125:124
29N++           61F6++      93M++91     125N1--127
30N2++28        62F6++      94D2--96    126N++125
31N2--34        63F6++59    95D1++94    127N2--128:127
32Q2--38        64N--83     96Q1--86    128Q3--6:128
```

Figure 3: Register machine that never halts if and only the Riemann Hypothesis is true

Translation of the program from Figure 2 into a register machine may be done quite straightforwardly. However, a small economy can be achieved. Namely, calculations of `q=n*q/g`, `f0=2*n*f0` and `f3=(2*n+3)*f3` can be performed within the same cycle of length `n`.

Acknowledgement

References

[1] Ю. И. Манин and А. А. Панчишкин. *Введение в теорию чисел*, volume 49 of *Итоги науки и техн. Сер. Соврем. пробл. мат. Фундам. направления*. ВИНИТИ, 1990.

[2] Clay Mathematics Institute Millenium Problems. `http://www.claymath.org/millennium`.

[3] Ю. В. Матиясевич. *Десятая проблема Гильберта*. Физматлит, 1993. English translation: Hilbert's Tenth Problem. MIT Press, Cambridge (Massachusetts) London (1993). French translation: Le dixième Problème de Hilbert, Masson, Paris Milan Barselone (1995). URL: `http://logic.pdmi.ras.ru/~yumat/H10Pbook`,.

[4] Ю. В. Матиясевич. Алан Тьюринг и теория чисел. *Математика в высшем образовании*, 10:111–134, 2012. `http://www.unn.ru/math/no/10/_nom10_012_matiyasevich.pdf`.

[5] Scott Aaronson. The blog. `https://www.scottaaronson.com/blog/?p=2741`.

[6] Leonard Adleman and Kenneth L. Manders. Diophantine complexity. In *17th Annual Symposium on Foundations of Computer Science*, pages 81–88. IEEE, 1976.

[7] Andrew R. Booker. Artin's conjecture, Turing's method, and the Riemann hypothesis. *Exp. Math.*, 15(4):385–407, 2006. `http://doi.org/10.1080/10586458.2006.10128976`.

[8] Andrew R. Booker. Turing and the Riemann hypothesis. *Notices Am. Math. Soc.*, 53(10):1208–1211, 2006.

[9] Kevin Broughan. *Equivalents of the Riemann hypothesis. Volume 1: Arithmetic equivalents.* Cambridge: Cambridge University Press, 2017.

[10] Dietrich Burde. A remark on an inequality for the prime counting function. *Math. Inequal. Appl.*, 10(1):9–13, 2007. `http://doi.org/10.7153/mia-10-02`.

[11] J. M. Hernandez Caceres. The Riemann hypothesis and Diophantine equations, 2018. Master's Thesis Mathematics, Mathematical Institute, University of Bonn.

[12] C. S. Calude and E. Calude. The complexity of mathematical problems: An overview of results and open problems. *International Journal of Unconventional Computing*, 9(3–4):327–343, 2013.

[13] Cristian S. Calude and Elena Calude. Evaluating the complexity of mathematical problems. I. *Complex Syst.*, 18(3):267–285, 2009.

[14] Cristian S. Calude and Elena Calude. Evaluating the complexity of mathematical problems. II. *Complex Syst.*, 18(4):387–401, 2010.

[15] Cristian S. Calude, Elena Calude, and Michael J. Dinneen. A new measure of the difficulty of problems. *J. Mult.-Val. Log. Soft Comput.*, 12(3-4):285–307, 2006.

[16] Elena Calude. The complexity of Riemann's hypothesis. *J. Mult.-Val. Log. Soft Comput.*, 18(3-4):257–265, 2012.

[17] S. B. Cooper and J. van Leeuwen, editors. *Alan Turing – His Work and Impact*. Elsevier Science, 2013.

[18] Martin Davis. Hilbert's tenth problem is unsolvable. *Am. Math. Mon.*, 80:233–269, 1973. `http://doi.org/10.2307/2318447`.

[19] Martin Davis, Yuri Matijasevic, and Julia Robinson. Hilbert's tenth problem: Diophantine equations: Positive aspects of a negative solution. *Proc. Symp. Pure Math.*, 28:323–378, 1976.

[20] Charles-Joseph de la Vallée Poussin. Recherches analytiques de la théorie des nombres premiers. *Annales de la Société Scientifique de Bruxelles*, 20B:183–256, 1896.

[21] Pierre Dusart. Estimates of ψ, θ for large values of x without the Riemann hypothesis. *Math. Comput.*, 85(298):875–888, 2016. `http://doi.org/10.1090/mcom/3005`.

[22] J. Hadamard. Sur la distribution des zéros de la fonction $\zeta(s)$ et ses conséquences arithmétiques. *Bulletin de la Société Mathématique de France*, 24:199–220, 1896. `http://doi.org/10.24033/bsmf.545`.

[23] D. Hilbert. Mathematische Probleme. Vortrag, gehalten auf dem internationalen Mathematiker Kongress zu Paris 1900. *Nachr. K. Ges. Wiss., Göttingen, Math.-Phys.Kl.*, pages 253–297, 1900. Reprinted in Gesammelte Abhandlungen, Springer, Berlin 3 (1935); Chelsea, New York (1965). English translation: Bull. Amer. Math. Soc., 8 437–479 (1901-1902); reprinted in: Browder (ed.) Mathematical Developments arising from Hilbert Problems, Proceedings of Symposia in Pure Mathematics 28, American Mathematical Society, 1–34 (1976).

[24] J. P. Jones and Yu. V. Matiyasevich. Register machine proof of the theorem on exponential diophantine representation of enumerable sets. *J. Symb. Log.*, 49:818–829, 1984. `http://doi.org/10.2307/2274135`.

[25] G. Kreisel. Mathematical significance of consistency proofs. *Journal of Symbolic Logic*, 23(2):155–182, 1958.

[26] J. Lambek. How to program an infinite abacus. *Can. Math. Bull.*, 4:295–302, 1961. `http://doi.org/10.4153/CMB-1961-032-6`.

[27] Yuri Matiyasevich. Hilbert's Tenth Problem: What was done and what is to be done. In *Hilbert's tenth problem: relations with arithmetic and algebraic geometry. Proceedings of the workshop, Ghent University, Belgium, November 2–5, 1999*, pages 1–47. Providence, RI: American Mathematical Society, 2000.

[28] Z. A. Melzak. An informal arithmetical approach to computability and computation. *Can. Math. Bull.*, 4:279–293, 1961. `http://doi.org/10.4153/CMB-1961-031-9`.

[29] Gary L. Miller. Riemann's hypothesis and tests for primality. *J. Comput. Syst. Sci.*, 13:300–317, 1976. `http://doi.org/10.1016/S0022-0000(76)80043-8`.

[30] Marvin L. Minsky. Recursive unsolvability of Post's problem of "Tag" and other topics in theory of Turing machines. *Ann. Math. (2)*, 74:437–455, 1961. `http://doi.org/10.2307/1970290`.

[31] Marvin L. Minsky. Computation: finite and infinite machines. Prentice-Hall Series in Automatic Computation. Englewood Cliffs, NJ: Prentice-Hall, Inc. VII, 317 p. (1967)., 1967.

[32] Aran Nayebi. On the Riemann hypothesis and Hilbert's tenth problem, February 2012. Unpublished Manuscript. `http://web.stanford.edu/~anayebi/projects/RH_Diophantine.pdf`.

[33] B. Riemann. Über die Anzhal der Primzahlen unter einer gegebenen Grösse. *Monatsberichter der Berliner Akademie*, 1859. Included into: Riemann, B. *Gesammelte Werke*. Teubner, Leipzig,

1892; reprinted by Dover Books, New York, 1953. `http://www.claymath.org/publications/riemanns-1859-manuscript`, English translation: `http://www.maths.tcd.ie/pub/HistMath/People/Riemann/Zeta/EZeta.pdf`.

[34] Lowell Schoenfeld. Sharper bounds for the Chebyshev functions $\theta(x)$ and $\psi(x)$. II. *Math. Comput.*, 30:337–360, 1976. `http://doi.org/10.2307/2005976`.

[35] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc*, 42(2):230–265, 1936–37. Correction, ibid. 43 (1937), 544–546.

[36] A. M. Turing. Some calculations of the Riemann zeta-function. *Proceedings of the London Mathematical Society, ser. 3*, 3:99–117, 1953. Reprinted in: *Collected Works of A. M. Turing: Pure Mathematics*(J. L. Britton, ed.), North-Holland, Amsterdam, (1992); *Alan Turing – His Work and Impact*, S. B. Cooper, J. van Leeuwen, eds., Elsevier Science, 2013. ISBN: 978-0-12-386980-7.

[37] A.M. Turing. A method for the calculation of the zeta-function. *Proc. Lond. Math. Soc. (2)*, 48:180–197, 1943. `http://doi.org/10.1112/plms/s2-48.1.180`.

[38] Adam Yedidia and Scott Aaronson. A relatively small Turing machine whose behavior is independent of set theory. *Complex Systems*, 25:297–327, 2016. `http://doi.org/10.25088/ComplexSystems.25.4.297`.