The Riemann Hypothesis in computer science

Yu. Matiyasevich

Steklov Institute of Mathematics at St. Petersburg (POMI), 27, Fontanka, St. Petersburg, 191023, Russia

A R T I C L E   I N F O

A B S T R A C T

The Riemann Hypothesis is reformulated as the statement that particular explicitly presented register machine with 29 registers and 130 instructions never halts.

© 2019 Elsevier B.V. All rights reserved.

1. The Riemann Hypothesis and Alan Turing

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, [38], 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* [4].

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.

In particular, B. Riemann proved in his seminal paper [34] that

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

E-mail address: yumat@pdmi.ras.ru.

[1] More details about Turing's contribution to Number Theory can be found in [8,7,17,9,3].

Table 1
Numerical verification of the Riemann Hypothesis.

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

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

$$\mathrm{Li}(x) = \int_2^x \frac{1}{\ln(t)}\, dt, \tag{4}$$

and the summation is taken over the 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

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

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 [30] 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 rigorous proofs that the real parts of these zeros are *exactly* equal to 1/2.

Reported in [38] contribution of Turing, that is, about zeros from 1042nd to 1104th, does not look impressive compared neither with his predecessor nor 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 known 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, [39,38]. This is not quite so. Turing dealt with the Riemann Hypothesis also in his Ph.D. thesis which was later published as [37]. 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 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 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 [38]:

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.
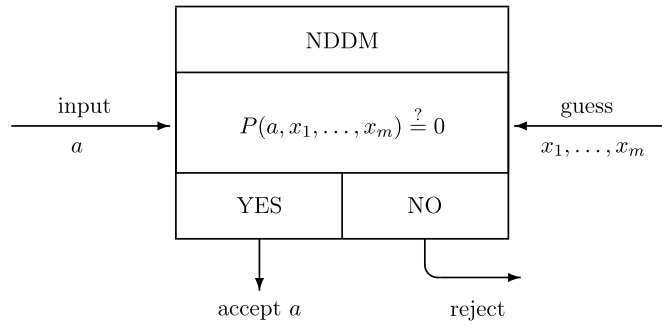
Fig. 1. Non-Deterministic Diophantine Machine.

2. The Riemann Hypothesis as Π_1^0 problem

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

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

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

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*

$$R(x_1 \ldots x_n) = 0 \tag{7}$$

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 [2, Section 6.4]; more details are supplied in [11]; both methods are discussed in [33].

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 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 can 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 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 (Fig. 1). Each such machine is specified by a parametric Diophantine equation

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

---

[2] This was just a particular application of a very general technique developed by Kreisel in [26].

[3] After M. Davis, H. Putnam, J. Robinson and Yu. Matiyasevich; for detailed proofs see, for example, [1,2,28,18,25].

The parameter a is the input to computations. The non-determinism is due to the other variables, $x_1, \ldots, x_n$, the values of which are not specified. For certain choices of these values the equality (8) may hold, but for other choices it may fail. According to the traditional definition of a non-deterministic computation, it is considered successful if (8) holds for at least one choice of values of $x_1, \ldots, x_n$.

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 [19,2,11]) 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 devices were proposed in 1961 by J. Lambek [27], by Z.A. Melzak [29], and by M.L. Minsky [31] (see also [32]). A register machine has a finite number of *registers* capable to contain arbitrary large non-negative integers. Types of possible instructions can vary. 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 the machine with 290 instructions provided in [15]).

More recently, A. Yedidia and S. Aaronson [40] 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 [2, 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 [2]. In Section 3 we present a new Π_1^0 formulation of the Riemann Hypothesis. In Section 4 this criterium is restated in the form of a Python program (see Fig. 2), and in Section 5 as a register machine (see Fig. 4). This machine has only 130 instructions, and they are of two very simple types in style of [27] and [31]: to increment or decrement a register by 1. As it was remarked in [31], 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).

3. New Π_1^0 reformulation of the Riemann Hypothesis

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

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

where $q(n)$ is the least common multiple of numbers $1, \ldots, n$.

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.

The bound $1/(8\pi)$ on the absolute value of the left-hand side in (10) for $n \geq 74$ was given in [36] (see also [9, Theorem 4.6]); it is easy to verify that one-sided inequality (10) holds for $n = 1, \ldots, 73$ as well. The sufficiency of the condition (11) follows from the $\Omega_{\pm}$-result for the psi function established in [35] (see also [24, Theorem 32] and [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 [2] and then adopted in [15,16] and in [40] 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,

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

---

[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.

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 a 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

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

For $n \geq 30$ number $n!/30!q(30)$ is a multiple of each of the numbers $1, \ldots, n$, hence it is not less than $q(n)$ (which is the least common multiple of these numbers). Using Stirling formula we conclude that

$$\psi(n) = \ln\big(q(n)\big) \leq \ln\big(n!/30!q(30)\big) \tag{16}$$

$$< n \ln(n) - n + \ln(n)/2 + 1 + \ln\big(q(30)/30!\big) \tag{17}$$

$$< n \ln(n) - n + \ln(n)/2 - 40, \tag{18}$$

and respectively

$$l(n) = \lfloor \log_2(q(n)) \rfloor < (\ln(n) - n + \ln(n)/2 - 40)/\ln(2). \tag{19}$$

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

$$\left| \psi(n) - b(n)l(n) \right| < \frac{1}{25} n^{\frac{1}{2}} \ln(n)^2. \tag{20}$$

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

$$d(n) = (2n-2)!!b(n). \tag{21}$$

This can be done via the recurrent relations

$$d(1) = 0, \quad d(n+1) = 2nd(n) - 2(-1)^n(2n-2)!!. \tag{22}$$

Our elimination of the logarithms in (10)–(11) is based on the *Prime Number Theorem*[5] stating that $\pi(n)$ 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 $n \geq 30$

$$1 < \frac{\pi(n)}{n/\ln(n)} < \frac{13}{10}. \tag{23}$$

The same double factorial, $(2n-2)!!$ from (21), will be used for calculation of an approximation to $n^{\frac{1}{2}}$. Namely, according to the Stirling formula, for $n \geq 30$

$$\frac{9}{2} n^{5/2} < \frac{(2n+3)!!}{(2n-2)!!} < 5n^{5/2}. \tag{24}$$

Now (10), (20), (23), and (24) imply that for $n \geq 30$

$$\pi(n)^2 \big(d(n)l(n) - f_0(n)\big) < f_3(n) \tag{25}$$

where

$$f_0(n) = \frac{(2n)!!}{2!!}, \qquad f_3(n) = \frac{(2n+3)!!}{5!!}. \tag{26}$$

---

[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 (23) 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
 1   h=m=p=0
 2   d=f0=f3=n=q=1
 3   while p**2*(m-f0)<f3:
 4       d=2*n*d-4*(-1)**n*h
 5       n=n+1
 6       g=gcd(n,q)
 7       q=n*q/g
 8       if g==1: p=p+1
 9       m=0; g=q
10       while g>1:
11           g=g//2; m=m+d
12       h=f0
13       f0=2*n*h
14       f3=(2*n+3)*f3
```

Fig. 2. Program 1 which never halts if and only if the Riemann Hypothesis is true.

In fact, the inequality (25) holds for $n = 1, \ldots, 29$ as well, which can be verified by a direct numerical calculation. Thus the Riemann Hypothesis implies the validity of (25) for all n.

In its turn, the inequality (25), together with (20), (23), and (24), implies (11) with $C = 2/5$. Hence the fulfillment of the inequality (25) for all n is both necessary and sufficient condition for the validity of the Riemann Hypothesis.

The least common multiple, q(n) from (25), can be calculated via the recurrent equations:

$$q(1) = 1, \quad q(n) = nq(n-1)/g(n) \tag{27}$$

where

$$g(n) = \text{GCD}(n, q(n-1)). \tag{28}$$

The greatest common divisor (28) can be used also for calculating $\pi(n)$ via the recurrent relations

$$\pi(1) = 0, \quad \pi(n) = \begin{cases} \pi(n-1) + 1, & \text{if } g(n) = 1, \\ \pi(n-1), & \text{otherwise.} \end{cases} \tag{29}$$

At last, functions (26) are calculated via the natural recurrent relations:

$$f_0(1) = 1, \quad f_0(n) = 2nf_0(n-1), \tag{30}$$

$$f_3(1) = 1, \quad f_3(n) = (2n+3)f_3(n-1). \tag{31}$$

4. Python programs

It is easy to implement the recursions (22), (27), and (29)–(31) within any programming language. Fig. 2 gives an example of Python program; line numbers were added for giving comments below.

Variable n is initiated in Line 2 to the value 1; later the value of n is incremented by 1 inside the loop formed by Lines 3–14, more precisely, on Line 5.

At the moment when the condition in Line 3 is checked, the variables p, m, f0, and f3 should have the following values:

$$\texttt{p} = \pi(\texttt{n}), \ \texttt{m} = \texttt{d}(\texttt{n})\,\texttt{l}(\texttt{n}), \ \texttt{f0} = f_0(\texttt{n}), \ \texttt{f3} = f_3(\texttt{n}). \tag{32}$$

Thus if Riemann's hypothesis is true, then inequality (25) holds, and the computation continue. However, if Riemann's hypothesis is wrong, computation would stop as soon as n reaches the least value violating (25).

When computations in Lines 4–14 are done, the variables p, m, f0, and f3 should again satisfy equalities (32) but now for the incremented value of n. Auxiliary variables d, h, and q should have the following values:

$$\texttt{d} = \texttt{d}(\texttt{n}), \ \texttt{h} = f_0(\texttt{n-1}), \ \texttt{q} = q(\texttt{n}). \tag{33}$$

Let us verify (32)–(33). This is evident for f3: this variable is initiated to 1 on Line 2 and is updated on Line 14 in accordance with the recurrence (31). Similar, variable f0 is initiated to 1 on Line 2 and is updated on Line 13 in accordance with the recurrence (30). The previous value of f0 was copied on Line 13 to h in accordance to (33).

For d we need to consider two cases. When this variable is calculated in Line 4 for the first time, it gets (due to the initializations in Lines 1–2) correct value 2, corresponding to the future value $n = 2$. After that, d gets correct value in Line 4 in accordance with (22) and (33).

The temporary variable g gets in Line 6 the value $\text{GCD}(n, q(n-1))$. Respectively, in accordance with (27) and (29) variables q and p get in Lines 7 and 8 their correct values.

The loop in Lines 10–11 repeats $\lfloor \log_2(q(n)) \rfloor = \texttt{l}(n)$ times, respectively, m gets the value indicated in (32).

```
                  from math import  gcd
          1   h0=m=p=p2=s=0
          2   d=f0=f3=h1=n=q=1
          3   while p2*(m-f0)<f3:
          4     na=n; n=n+1
          5     g=gcd(n,q)
          6     qa=q;
          7     f0a=f0=2f0;
          8     f3a=2*f3; f3=5*f3;
          9     da=2*d; d=0;
          10    while na>0:
          11      d=d+da
          12      q=q+qa
          13      f0=f0+f0a
          14      f3=f3+f3a
          15      na=na-1
          16    if s==0:
          17      s=1; d=d+4*h0; h0=f0
          18    else:
          19      s=0; d=d-4*h1; h1=f0
          20    if g==1:
          21      p2=p2+p; p=p+1; p2=p2+p
          22    m=0; g=q/g
          23    while g>1:
          24      g=g//2; m=m+d
```

Fig. 3. Program 2 which never halts if and only if the Riemann Hypothesis is true.

Summing up, we conclude that *the Riemann Hypothesis is valid if and only if the program on Fig. 2 never halts*.

As an intermediate step for constructing a register machine, let us consider Program 2 from Fig. 3. The main distinctions between the two programs are as follows.

Register machines work with non-negative integers only, so they cannot calculate the factor $(-1)^n$ from Line 4 of Program 1. This obstacle is overcome in Program 2 by the introduction of a new variable s. It alternatively assumes values 0 and 1 depending on the parity of n, and respectively Line 4 of Program 1 is imitated by Lines 16–18 of Program 2.

Another new variable, p2, has the values p^2 which is updated on Line 20 in Program 2 together with updating of the value of p.

The single loop in Lines 10–15 implements four multiplications by n performed in Lines 4, 7, 13, and 14 of Program 1.

5. Register machine

Fig. 4 presents a register machine that never halts if and only if the Riemann Hypothesis is true.

Instructions of the machine are of two forms: either

$$\langle\texttt{label}\rangle\langle\texttt{register}\rangle\texttt{++}[\langle\texttt{next}\rangle] \tag{34}$$

or

$$\langle\texttt{label}\rangle\langle\texttt{register}\rangle\texttt{--}\langle\texttt{jump}\rangle[\texttt{:}\langle\texttt{next}\rangle] \tag{35}$$

In the former case performing instruction label, the machine increases register by 1 and goes to perform 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 an instruction with next=0 or jump=0.

The machine has 29 registers. Eleven of them, namely, D, F0, F3, G, H0, H1, M, N, P, P2, and Q, correspond respectively to variables d, f0, f3, g, h0, h1, m, n, p, p2, and q of Program 2. Eighteen subsidiary registers Da, Db, F0a, F0b, F3a, F3b, F3m, Ga, Gb, Gc, Ma, Na, Nb, P2, P2a, Pa, Qa, Qb are used as mirrors of the above eleven main registers for temporally keeping copies of their values.

The machine starts with all registers being empty, so there is no counterpart to the initialization Line 1 of Program 2. The other initializations, performed in Line 2, are implemented by instructions 1–6.

As it was already mentioned, register machines work with non-negative integers only, so it is impossible to calculate the difference m-f0 from Line 4 when m<f0. But in such a case the inequality in this line is fulfilled, and the difference can be replaced by max(0,m-f0); this value will be assigned to register m in the loop formed by instructions 7–10.

Multiplication by p2 and comparison with f3 in Line 4 of Program 1 are implemented in the register machine by instructions 11–27. Namely, at first the value of f3 is copied by instructions 11–18 to register f3m. Similar, the value of p2 is copied by instructions 19–20 to register p2a. After that the machine tries to perform p^2 times the loop consisting of instructions 21–27. At each cycle the value of f3m should be decreased by the value of m. If the Riemann Hypothesis

```
1D++            34Nb++32        67F3b++66       100P2++
2F0++           35Na-40         68N++47         101Pa++99
3F3++           36Qa-38         69F0-71         102Pa++
4H1++           37G++35         70F0a++69       103Pa-107
5N++            38G-36          71F0b-73        104P2++
6Q++            39Qa++38        72F0a++71       105P++103
7F0-11          40Qa-44         73F3a-75        106G++
8F0b++          41Qa++          74F3++73        107G++
9F0b++          42G-35          75F3b-77        108Q-117
10M-7:7         43Na++42        76F3++75        109Qa++
11F3m-12:11     44D-68          77Qb-79         110Gb++
12F3-19         45Da++          78Q++77         111Q++
13F3m++         46Da++44        79S-88          112G-115
14F3a++         47Nb-69         80H1-85         113Q-115
15F3a++         48Da-51         81D-82          114Ga++112
16F3a++         49D++           82D-83          115Ga-108
17F3b++         50Db++48        83D-84          116G++115
18F3b++12       51Db-53         84D-80:80       117Qa-119
19P2-21         52Da++51        85F0a-97        118Q++117
20P2a++19       53Qb-56         86H1++          119Da-120:119
21P2a-28        54Q++           87F0++85        120Gb-7
22P2++          55Qa++53        88H0-93         121Gb-123
23M-26          56Qa-58         89D++           122Gc++121
24Ma++          57Qb++56        90D++           123Gc-126
25F3m-0:23      58F0b-61        91D++           124Gc-126
26Ma-21         59F0a++         92D++88         125Gb++123
27M++26         60F0++58        93F0a-96        126D-129
28Q-31          61F0-63         94H0++          127M++
29Qa++          62F0b++61       95F0++93        128Da++126
30Qb++28        63F3b-66        96S++           129Da-120
31G-32:31       64F3a++         97G-98          130D++129
32N-36          65F3++63        98G-99:106
33Na++          66F3-68         99P-102
```

Fig. 4. Register machine that never halts if and only if the Riemann Hypothesis is true.

is wrong, then for a certain value of n the inequality (25) fails, respectively, the machine will stop trying to perform instruction 25. Otherwise, after p^2 cycles, computations will continue from instruction 28.

Instructions 28–43 implement the classical Euclidean algorithm for calculation of the greatest common divisor from Line 5 of Program 2.

Instructions 44–77 rather faithfully implement calculations from Lines 6–15 of Program 2. Similar, instructions 79–96 correspond to Lines 16–19 of that Program.

Instructions 97–107 update the values of registers p and p2, if this is required according to Lines 20–21 of Program 2.

Instructions 108–118 implement the division algorithm required for Line 22 of Program 2.

At last, instructions 119–130 correspond to Lines 23–24 of Program 2.

Summing up, we conclude that *the Riemann Hypothesis is valid if and only if the register machine on Fig. 4, having started with the empty registers, never halts.*

Declaration of Competing Interest

There is no competing interest.

Acknowledgement

References

[1] Ю.И. Манин, А.А. Панчишкин, Введение в теорию чисел, Итоги науки и техн. Сер. Соврем. пробл. мат. Фундам. направления, vol. 49, ВИНИТИ, 1990; Translated as: in: Introduction to Modern Number Theory. Fundamental Problems, Ideas and Theories, second edition, Encyclopaedia of Mathematical Sciences, vol. 49, Springer-Verlag, Berlin, ISBN 978-3-540-20364-3, 2005, 3-540-20364-8, xvi+514 pp.

[2] 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, http://logic.pdmi.ras.ru/~yumat/H10Pbook.

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

[4] Clay Mathematics Institute Millenium Problems, http://www.claymath.org/millennium.

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

[6] Leonard Adleman, Kenneth L. Manders, Diophantine complexity, in: 17th Annual Symposium on Foundations of Computer Science, IEEE, 1976, pp. 81–88.

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

[8] Andrew R. Booker, Turing and the Riemann hypothesis, Not. Am. Math. Soc. 53 (10) (2006) 1208–1211.

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

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

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

[12] C.S. Calude, E. Calude, The complexity of mathematical problems: an overview of results and open problems, Int. J. Unconv. Comput. 9 (3–4) (2013) 327–343.

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

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

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

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

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

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

[19] Martin Davis, Yuri Matijasevic, Julia Robinson, Hilbert's tenth problem: Diophantine equations: positive aspects of a negative solution, Proc. Symp. Pure Math. 28 (1976) 323–378, http://doi.org/10.1090/pspum/028.2.

[20] Charles-Joseph de la Vallée Poussin, Recherches analytiques de la théorie des nombres premiers, Ann. Soc. Sci. Brux. 20B (1896) 183–256.

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

[22] J. Hadamard, Sur la distribution des zéros de la fonction $\zeta(s)$ et ses conséquences arithmétiques, Bull. Soc. Math. Fr. 24 (1896) 199–220, http://doi.org/10.24033/bsmf.545.

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

[24] A.E. Ingham, The Distribution of Prime Numbers, Cambridge Tracts in Mathematics and Mathematical Physics, vol. 30, Cambridge University Press, 1932, reprinted in 1990.

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

[26] G. Kreisel, Mathematical significance of consistency proofs, J. Symb. Log. 23 (2) (1958) 155–182.

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

[28] 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, American Mathematical Society, Providence, RI, 2000, pp. 1–47.

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

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

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

[32] Marvin L. Minsky, Computation: Finite and Infinite Machines, Prentice-Hall Series in Automatic Computation, vol. VII, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1967, 317 p.

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

[34] B. Riemann, Über die Anzahl der Primzahlen unter einer gegebenen Grösse, Monatsberichter der Berliner Akademie (1859); Included into: 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.

[35] E. Schmidt, Über die Anzahl der Primzahlen unter gegebener Grenze, Math. Ann. 57 (1903) 195–203.

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

[37] A.M. Turing, On computable numbers, with an application to the Entscheidungsproblem, Proc. Lond. Math. Soc. 42 (2) (1936–37) 230–265, https://doi.org/10.1112/plms/s2-42.1.230, Correction: Proc. Lond. Math. Soc. 43 (1937) 544–546, https://doi.org/10.1112/plms/s2-43.6.544.

[38] A.M. Turing, Some calculations of the Riemann zeta-function, Proc. Lond. Math. Soc., Ser. 3 3 (1953) 99–117, Reprinted in: J.L. Britton (Ed.), Collected Works of A.M. Turing: Pure Mathematics, North-Holland, Amsterdam, 1992, in: S.B. Cooper, J. van Leeuwen (Eds.), Alan Turing – His Work and Impact, Elsevier Science, ISBN 978-0-12-386980-7, 2013.

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

[40] Adam Yedidia, Scott Aaronson, A relatively small Turing machine whose behavior is independent of set theory, Complex Syst. 25 (2016) 297–327, http://doi.org/10.25088/ComplexSystems.25.4.297.