The Complexity of Mathematics

Frank Vega
Joysonic, Uzun Mirkova 5, Belgrade, 11000, Serbia
vega.frank@gmail.com

Abstract

The Goldbach's conjecture is one of the most important and unsolved problems in number theory. Nowadays, it is one of the open problems of Hilbert and Landau. We prove the Goldbach's conjecture is false using the Complexity Theory. An important complexity class is 1NSPACE(S(n)) for some S(n). This proof is based on if some unary language belongs to 1NSPACE(S(log n)), then the binary version of that language belongs to 1NSPACE(S(n)) and vice versa.

2012 ACM Subject Classification Theory of computation → Complexity classes; Theory of computation → Regular languages; Theory of computation → Problems, reductions and completeness

Keywords and phrases complexity classes, regular languages, reduction, number theory, primes, one-way

1 Introduction

The Goldbach's original conjecture, written on 7 June 1742 in a letter to Leonhard Euler, states: "... at least it seems that every number that is greater than 2 is the sum of three primes" [5]. This is known as the ternary Goldbach conjecture. We call a prime as a natural number that is greater than 1 and has exactly two divisors, 1 and the number itself [14]. However, the mathematician Christian Goldbach considered 1 as a prime number. Euler replied in a letter dated 30 June 1742 the following statement: "Every even integer greater than 2 can be written as the sum of two primes" [5]. This is known as the strong Goldbach conjecture.

Using Vinogradov's method, Van der Corput and Estermann showed that almost all even numbers can be written as the sum of two primes (in the sense that the fraction of even numbers which can be so written tends towards 1) [13], [6]. In 1973, Chen showed that every sufficiently large even number can be written as the sum of some prime number and a semi-prime [3]. The strong Goldbach conjecture implies the conjecture that all odd numbers greater than 7 are the sum of three odd primes, which is known today as the weak Goldbach conjecture [5]. In 2012 and 2013, Peruvian mathematician Harald Helfgott published a pair of papers claiming to improve major and minor arc estimates sufficiently to unconditionally prove the weak Goldbach conjecture [8], [9]. In this work, we prove the strong Goldbach's conjecture is false.

2 Theory and Methods

We use o-notation to denote an upper bound that is not asymptotically tight. We formally define $o(g(n))$ as the set

$$o(g(n)) = \{f(n) : \textit{ for any positive constant } c > 0, \textit{ there exists a constant}$$

$$n_0 > 0 \textit{ such that } 0 \leq f(n) < c \times g(n) \textit{ for all } n \geq n_0\}.$$

For example, $2 \times n = o(n^2)$, but $2 \times n^2 \neq o(n^2)$ [4]. In theoretical computer science and formal language theory, a regular language is a formal language that can be expressed using a regular expression [2]. The complexity class that contains all the regular languages is REG. The two-way Turing machines may move their head on the input tape into two-way (left and right directions) while the one-way Turing machines are not allowed to move the head on the input tape to the left [11]. The complexity class *1NSPACE*($f(n)$) is the set of decision problems that can be solved by a nondeterministic one-way Turing machine M, using space $f(n)$, where n is the length of the input [11].

3 Results

The checking whether a number is prime can be decided in polynomial time by a deterministic Turing machine [1]. This problem is known as $PRIMES$ [1].

▶ **Theorem 1.** *$PRIMES \notin$ 1NSPACE($S(n)$) for all $S(n) = o(\log n)$.*

Proof. If we assume that $PRIMES \in$ *1NSPACE*($o(\log n)$), then the unary version should be regular. Certainly, the standard space translation between the unary and binary languages actually works for nondeterministic machines with small space [7]. This means that if some language belongs to *1NSPACE*($S(n)$), then the unary version of that language belongs to *1NSPACE*($S(\log n)$) [7]. In this way, when $PRIMES \in$ *1NSPACE*($o(\log n)$), then the unary version should be in *1NSPACE*($o(\log \log n)$) and we know that $REG =$ *1NSPACE*($o(\log \log n)$) [11], [7]. Since we know that the unary version of $PRIMES$ is nonregular [10], then we obtain that $PRIMES \notin$ *1NSPACE*($S(n)$) for all $S(n) = o(\log n)$. ◀

▶ **Definition 2.** *We define the unary Goldbach's language L_{UG} as follows:*

$$L_{UG} = \{0^{2\times n}0^{p}0^{q} : n \in \mathbb{N} \wedge n > 2 \wedge p \textit{ and } q \textit{ are odd primes } \wedge 2 \times n = p + q\}.$$

We define the language coL_{UG} as

$$coL_{UG} = \{0^{2\times n}0^{2\times n} : n \in \mathbb{N} \wedge n > 2 \wedge \neg \exists \textit{ odd primes } p \textit{ and } q \textit{ such that } 2 \times n = p + q\}$$

where coL_{UG} is the complement language of L_{UG}. On the other hand, the language S_{UG} is equal to $L_{UG} \cup coL_{UG}$.

▶ **Theorem 3.** *The language S_{UG} is regular.*

Proof. The language S_{UG} can be stated as

$$S_{UG} = \{0^{4\times n} : n \in \mathbb{N} \wedge n > 2\}.$$

Certainly, we can easily decide S_{UG} in constant space using a deterministic Turing machine and thus, S_{UG} is a regular language [12]. ◀

▶ **Definition 4.** *We define the binary Goldbach's language L_{BG} as follows:*

$$L_{BG} = \{(2 \times n, p, q) : \textit{ such that } 0^{2\times n}0^{p}0^{q} \in L_{UG}\}.$$

▶ **Lemma 5.** *L_{UG} is the unary representation of language L_{BG}.*

Proof. This is trivially true from the definition of these languages. ◀

▶ **Theorem 6.** *$L_{BG} \notin$ 1NSPACE($S(n)$) for all $S(n) = o(\log n)$.*

Proof. The language L_{BG} cannot be computed in *1NSPACE*$(S(n))$ for some $S(n) = o(\log n)$, because of this would imply that the problem $PRIMES$ belongs to *1NSPACE*$(S(n))$ for some $S(n) = o(\log n)$ as well. Certainly, if this could be true, then we can take any number p and check whether p is prime. This could be nondeterministically done on input p just deterministically generating the numbers $p + 3$ and 3 and nondeterministically choosing an arbitrary number q, but instead of putting in the work tapes, then we will put them to the output tape just using constant space in one-way. After that, we use the space composition reduction just using the previous output of $(p + 3, 3, q)$ as input into a new nondeterministic Turing machine that would decide whether the instance belongs to L_{BG} in *1NSPACE*$(S(n))$ for some $S(n) = o(\log n)$. Indeed, the nondeterministic one-way computation will accept this input if and only if the nondeterministic generated number q is equal to p and p is prime. In this reduction, we assume the initial string p has a binary representation with the least significant bit in the first position within the input tape from left to right. In this way, it will be possible to deterministically generate $p + 3$ in one-way using constant space. Since *1NSPACE*$(S(n))$ for some $S(n) = o(\log n)$ is closed under *1NSPACE*-reductions with constant space, then the whole computation could be done in *1NSPACE*$(S(n))$ for some $S(n) = o(\log n)$. Nevertheless, this would be a contradiction according to Theorem 1, since the language $PRIMES \notin$ *1NSPACE*$(S(n))$ for all $S(n) = o(\log n)$. Consequently, we obtain that $L_{BG} \notin$ *1NSPACE*$(S(n))$ for all $S(n) = o(\log n)$. ◀

▶ **Theorem 7.** *The strong Goldbach's conjecture is false.*

Proof. We may have only two options: $L_{UG} \in REG$ or L_{UG} is non-regular and its complement coL_{UG} is infinite, since for every finite set F, the language $S_{UG} - F$ is always regular [12], because of S_{UG} is regular due to Theorem 3. Certainly, if coL_{UG} is finite, then the language $L_{UG} = S_{UG} - coL_{UG}$ could never be non-regular [12]. Let's assume the possibility of $L_{UG} \in REG$. However, this implies that the exponentially more succinct version of L_{UG}, that is L_{BG}, should be in *1NSPACE*$(S(n))$ for some $S(n) = o(\log n)$, because we would have $REG =$ *1NSPACE*$(o(\log\log n))$ and the same algorithm that decides L_{UG} within the complexity *1NSPACE*$(o(\log\log n))$ could be easily transformed into a slightly modified algorithm that decides L_{BG} within *1NSPACE*$(S(n))$ for some $S(n) = o(\log n)$ [11], [7]. Actually, L_{UG} is the unary version of L_{BG} due to Lemma 5. As we mentioned before, the standard space translation between the unary and binary languages actually works for nondeterministic machines with small space [7]. This means that if some unary language belongs to *1NSPACE*$(S(\log n))$, then the binary version of that language belongs to *1NSPACE*$(S(n))$ [7]. Consequently, we obtain that $L_{UG} \notin REG$, since it is not possible that $L_{UG} \in$ *1NSPACE*$(o(\log\log n))$ under the result of $L_{BG} \notin$ *1NSPACE*$(S(n))$ for all $S(n) = o(\log n)$ as result of Theorem 6. In this way, we obtain a contradiction just assuming that $L_{UG} \in REG$. Therefore, L_{UG} is non-regular and this implies there is only one remaining option: coL_{UG} is infinite. Nevertheless, we have the strong Goldbach's conjecture has an infinite number of counterexamples when coL_{UG} is infinite. Hence, we demonstrate the strong Goldbach's conjecture is false. ◀

References

1 Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Annals of Mathematics*, 160(2):781–793, 2004. `doi:10.4007/annals.2004.160.781`.

2 Alfred V. Aho and John E. Hopcroft. *The Design and Analysis of Computer Algorithms*. Pearson Education India, 1974.

3 Jing-run Chen. On the Representation of a Large Even Integer as the Sum of a Prime and the Product of Two Primes at Most. *Sci. Sinica*, 16:157–176, 1973.

4 Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 3 edition, 2009.

5 Leonard Eugene Dickson. *History of the Theory of Numbers: Divisibility and Primality*, volume 1. New York, Dover, 2005.

6 Theodor Estermann. On Goldbach's problem: Proof that almost all even positive integers are sums of two primes. *Proceedings of the London Mathematical Society*, 2(1):307–314, 1938.

7 Viliam Geffert and Dana Pardubská. Unary Coded NP-Complete Languages in ASPACE (log log n). *International Journal of Foundations of Computer Science*, 24(07):1167–1182, 2013. doi:10.1007/978-3-642-31653-1_16.

8 Harald A. Helfgott. Minor arcs for Goldbach's problem. *arXiv preprint arXiv:1205.5252*, 2012.

9 Harald A. Helfgott. Major arcs for Goldbach's theorem. *arXiv preprint arXiv:1305.2897*, 2013.

10 David Matuszek. Pumping Lemma Example 3, February 1996. In The Pumping Lemma Lecture at https://www.seas.upenn.edu/~cit596/notes/dave/pumping6.html. Retrieved 9 June 2020.

11 Pascal Michel. A survey of space complexity. *Theoretical computer science*, 101(1):99–132, 1992. doi:10.1016/0304-3975(92)90151-5.

12 Michael Sipser. *Introduction to the Theory of Computation*, volume 2. Thomson Course Technology Boston, 2006.

13 Johannes Gualtherus Van der Corput. Sur l'hypothèse de Goldbach pour presque tous les nombres pairs. *Acta Arithmetica*, 2(2):266–290, 1936.

14 David G. Wells. *Prime Numbers, The Most Mysterious Figures in Math*. John Wiley & Sons, Inc., 2005.