The Complexity of the Twin Prime Conjecture

Frank Vega

May 12, 2020

The Complexity of the Twin Prime Conjecture

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

Abstract

Twin primes become increasingly rare as one examines larger ranges, in keeping with the general tendency of gaps between adjacent primes to become larger as the numbers themselves get larger. The question of whether there exist infinitely many twin primes has been one of the great open questions in number theory for many years. We prove the Twin prime conjecture using the Complexity Theory. An important complexity class is 1NSPACE(S(n)) for some S(n). This mathematical 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 question of whether there exist infinitely many twin primes has been one of the great open questions in number theory for many years. This is the content of the Twin prime conjecture, which states that there are infinitely many primes p such that $p + 2$ is also prime [6]. In addition, the Dubner's conjecture is an as yet unsolved conjecture by American mathematician Harvey Dubner [4]. It states that every even number greater than 4208 is the sum of two t-primes, where a t-prime is a prime which has a twin [4]. We prove there are infinite even numbers that comply the Dubner's conjecture, where this also implies that the Twin prime conjecture is true [4].

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 $2 \times n^2 \neq o(n^2)$ [3]. 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 [8]. 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 [8].

3 Results

3.1 The Complexity of PRIMES

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 [5]. This means that if some language belongs to *1NSPACE*$(S(n))$, then the unary version of that language belongs to *1NSPACE*$(S(\log n))$ [5]. 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*$(o(\log \log n))$ [8], [5]. Since we know that the unary version of $PRIMES$ is non-regular [7], then we obtain that $PRIMES \notin$ *1NSPACE*$(S(n))$ for all $S(n) = o(\log n)$. ◀

3.2 Twin prime conjecture

▶ **Definition 2.** *We define the Dubner's language L_D as follows:*

$$L_D = \{1^{2 \times n} 0^p 0^q : n \in \mathbb{N} \wedge n > 2104 \wedge p \textit{ and } q \textit{ are t-primes } \wedge 2 \times n = p + q\}.$$

▶ **Theorem 3.** *If the Dubner's conjecture is true, then the Dubner's language L_D is non-regular.*

Proof. If the Dubner's conjecture is true, then the Dubner's language L_D is equal to the another language L' defined as follows:

$$L' = \{1^{2 \times n} 0^{2 \times n} : n \in \mathbb{N} \wedge n > 2104\}.$$

L' is a well-known non-regular language using the Pumping lemma for regular languages [10]. ◀

▶ **Definition 4.** *We define the verification Dubner's language L_{VD} as follows:*

$$L_{VD} = \{(2 \times n, p, q) : \textit{ such that } 1^{2 \times n} 0^p 0^q \in L_D\}.$$

▶ **Definition 5.** *We define the Dubner's language with separator L_{SD} as follows:*

$$L_{SD} = \{0^{2 \times n} \# 0^p \# 0^q : \textit{ such that } 1^{2 \times n} 0^p 0^q \in L_D\}$$

where $\#$ is the blank symbol.

▶ **Lemma 6.** *The Dubner's language with separator L_{SD} is the unary representation of the verification Dubner's language L_{VD}.*

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

▶ **Theorem 7.** *There are infinite even numbers that comply the Dubner's conjecture.*

Proof. If the Dubner's conjecture is false, then $L_D \in REG$ or L_D is non-regular and its complement is infinite, since every finite set is regular and REG is also closed under complement [9]. Let's assume the possibility of $L_D \in REG$. Under this assumption, we have that L_{SD} could be reduced to L_D in a nondeterministic constant space, where L_{SD} is the unary version of L_{VD} due to Lemma 6. Certainly, we can reduce in a nondeterministic one-way using constant space the language L_{SD} to L_D just removing the blank symbol $\#$ between the 0's on the input and generating the final output to L_D. But firstly, this nondeterministic one-way reduction replaces the 0's by 1's, but only those 0's which are exactly at the beginning of the original input of L_{SD} (before the first blank symbol). Indeed, we could have that $L_{SD} \in REG$ as result of this nondeterministic one-way reduction in constant space to the language L_D that would be in $\mathit{1NSPACE}(o(\log\log n)$, since $REG = \mathit{1NSPACE}(o(\log\log n)$ and $\mathit{1NSPACE}(o(\log\log n)$ is closed under *1NSPACE*-reductions with constant space [8].

However, this implies that the exponentially more succinct version of L_{SD}, that is L_{VD}, should be in $\mathit{1NSPACE}(S(n))$ for some $S(n) = o(\log n)$, because we would have $REG = \mathit{1NSPACE}(o(\log\log n))$ and the same algorithm that decides L_{SD} within the complexity $\mathit{1NSPACE}(o(\log\log n))$ could be easily transformed into a slightly modified algorithm that decides L_{VD} within $\mathit{1NSPACE}(S(n))$ for some $S(n) = o(\log n)$ [8], [5]. As we mentioned before, the standard space translation between the unary and binary languages actually works for nondeterministic machines with small space [5]. This means that if some unary language belongs to $\mathit{1NSPACE}(S(\log n))$, then the binary version of that language belongs to $\mathit{1NSPACE}(S(n))$ [5]. It is not possible that $L_{VD} \in \mathit{1NSPACE}(S(n))$ for some $S(n) = o(\log n)$, because of $PRIMES \notin \mathit{1NSPACE}(S(n))$ for all $S(n) = o(\log n)$. Certainly, the verification of whether p and q are t-primes needs to be done in order to accept the elements of this language. Consequently, we obtain that $L_{SD} \notin REG$, since it is not possible that $L_{SD} \in \mathit{1NSPACE}(o(\log\log n))$ under the result of $L_{VD} \notin \mathit{1NSPACE}(S(n))$ for all $S(n) = o(\log n)$. In this way, we obtain a contradiction just assuming that the Dubner's conjecture is false and $L_D \in REG$. In contraposition, we have there are infinite even numbers that comply with the Dubner's conjecture, since in the case of L_D would be finite, then we obtain that the Dubner's conjecture is false and $L_D \in REG$, where we just already proved that is not possible. ◀

▶ **Lemma 8.** *The Twin prime conjecture is true.*

Proof. The Theorem 7 implies that there exists an infinite number of t-primes, and thus there will be an infinite number of twin prime pairs as well [4]. ◀

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

4 Harvey Dubner. Twin prime conjectures. *Journal of Recreational Mathematics*, 30(3):199–205, 2000.

5 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`.

6 Godfrey Harold Hardy and Edward Maitland Wright. *An Introduction to the Theory of Numbers*. Oxford University Press, 1979.

7 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 11 May 2020.

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

9 Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.

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