CDMTCS Research Report Series

Long and Short Proofs

Cristian S. Calude[1], **Ludwig Staiger**[2]

[1]School of Computer Science, University of Auckland, New Zealand
[2]Institut für Informatik, Martin-Luther-Universität Halle-Wittenberg, Halle, Germany

CDMTCS-561
March 2022

Centre for Discrete Mathematics and
Theoretical Computer Science

Dedicated to the memory of our
friend and collaborator D. Ştefănescu

Long and Short Proofs

Cristian S. Calude† and Ludwig Staiger*

†School of Computer Science, The University of Auckland,
Private Bag 92019, Auckland, New Zealand
*Institut für Informatik, Martin-Luther-Universität
Halle-Wittenberg, D-06099 Halle, Germany

Abstract

We study the "gap" between the length of a theorem and the smallest length of its proof in a given formal system $\mathbf{T}$. To this aim, we define and study f-short and f-long proofs in $\mathbf{T}$, where f is a computable function. The results show that formalisation comes with a price tag, and a long proof does not guarantee a theorem's non-triviality or importance. Applications to proof-assistants are briefly discussed.

1 Introduction

According to Spencer [24],

> Long proofs are an anathema to mathematicians.

Gödel's seminal length-of-proof paper [15] was "re-discovered" after its English translation [16, p. 396–399] and led to studies of theorems with long proofs, see [20, 21], and more generally, to Blum's (abstract) computational complexity theory [2].

But, what is a "long proof"? First, we note that the original proof of a theorem tends to be unnecessarily long (and sometimes not entirely correct), but shorter and better proofs emerge in time. For example, Abel-Ruffini Theorem, stating the impossibility of finding a solution in radicals to polynomial equations of degree five or higher with arbitrary real coefficients, was initially 500 pages long (Ruffini's proof). Still, later, Abel obtained a mere 9-page proof.

Second, as every sufficiently complex formal system, for example, a system which includes Peano arithmetic (PA), proves infinitely many theorems, we can easily deduce:

Fact 1 (Norwood [19]) *Assume that the formal theory* $\mathbf{T}$ *based on a finite alphabet proves infinitely many theorems. Then, for every positive integer* N*, there exist infinitely many theorems in* $\mathbf{T}$ *whose smallest proof lengths are larger than* N*.*

2 Notation

By $\mathbf{N}$ denote the set of non-negative integers. Fix a formal axiomatic theory with negation $\mathbf{T}$ based on a finite alphabet. The formula $\neg S$ is the negation of S.

A formula (sentence) S is a *theorem of* $\mathbf{T}$, written, $\vdash^{\mathbf{T}} S$, if there exists a proof π in $\mathbf{T}$ for S, written, $\vdash^{\mathbf{T}}_{\pi} S$.

Consider the following symbol-length measure: the *proof-length* of the proof π is the length of π (as a word on the finite alphabet of $\mathbf{T}$) and is denoted by $|\pi|$. The *minimum-length proof* of S is the shortest proof of S if S is provable in $\mathbf{T}$:

$$\pi(S) = \inf\{|\sigma| \mid \vdash^{\mathbf{T}}_{\sigma} S\}. \tag{1}$$

If there is more than one proof π satisfying the first condition in (1), then $\pi(S)$ is the lexicographically first such proof.

The following properties of a formal theory $\mathbf{T}$ are used in what follows:

- $\mathbf{T}$ is *computably enumerable* if the set of proofs (hence, theorems) in $\mathbf{T}$ is computably enumerable.
- $\mathbf{T}$ is *rich enough*[1] if a certain amount of elementary arithmetic can be carried out in it.
- $\mathbf{T}$ is *consistent* if there is no sentence S in $\mathbf{T}$ such that $\vdash S$ and $\vdash \neg S$.

[1]The minimal amount of arithmetic required will be clear in each case.

In what follows, we will use Turing machines M operating with words on a finite alphabet [23, 11]. We will assume that the space complexity of the Turing machine T, space_T, satisfies the following natural condition: $space_T(x) \geq |x|$, for every input x. A *decider* is a Turing machine that stops on every input and returns either 0 or 1.

The set Σ^* is the free monoid under concatenation generated by the finite set Σ; its elements are called words. If $u \in \Sigma^*$, by $|u|$ we denote the length of the word u and by $u\Sigma^*$ the set $\{uv \mid v \in \Sigma^*\}$.

3 Prerequisites

A famous result on Turing machines refers to the

> **Halting Problem**: Given a pair (M, x), decide whether M *halts* on x.

There are no resource limitations on the amount of memory or time required for the decider's execution. The decider is a Turing machine that stops in *finite time* and gives the *correct answer for all possible pairs* (M, x). The undecidability of the Halting Problem [9, p. 70–71] is arguably the most important result in computability theory:

Theorem 1 (Halting Theorem) *No decider solves the Halting Problem.*

Corollary 1 *There exists a computably enumerable set that is incomputable.*

A special class of computably enumerable but incomputable sets is the class of *creative* sets, i.e. computably enumerable sets such that every other computably enumerable set can be one-one reduced to it. All creative sets are recursively isomorphic [22]. The set of theorems of many interesting formal theories, including PA and Zermelo-Fraenkel set theory with choice (ZFC), are creative [22].

A more interesting result than Fact 1 was proved by Hartmanis [18] for the class of formal systems whose theorems form a creative set. *Hartmanis proof-length* is the amount of tape used by Turing machines to accept the theorems of a formal system. This measure is justified by the fact that for any reasonable formal system, one can design a Turing machine which, for any given sentence in the system, successively checks all possible proofs of increasing length until it finds a proof of the given sentence or never halts if the input is not provable in the system.

Theorem 2 (Hartmanis [18]) *Fix a formal theory* $\mathbf{T}$ *whose set of theorems is creative, a Turing machine* M *that enumerates the theorems of* $\mathbf{T}$ *and Hartmanis proof-length with respect to* M. *Then, one can effectively find infinite subsets of "trivially true" theorems which require as long proofs in* $\mathbf{T}$ *as the hardest theorems of* $\mathbf{T}$.

The proof consists in constructing a decidable infinite set of theorems S in $\mathbf{T}$ *TrivialTrue* such that their shortest Hartmanis proof-length proofs in $\mathbf{T}$ grow faster than any computable function (of the length of the theorems to be proved). The theorems $S \in$ *TrivialTrue* are called "trivially true" because there is a decider for *TrivialTrue* which decides the question $S' \in$ *TrivialTrue* with computably bounded space. The proofs in $\mathbf{T}$ of the theorems in *TrivialTrue* can be algorithmically generated by enumerating all proofs in $\mathbf{T}$ and selecting those whose corresponding theorems are in *TrivialTrue*. Theorem 2 shows that their shortest Hartmanis proof-length proofs in $\mathbf{T}$ grow faster than any computable function (of the length of the theorems to be proved).

4 Results

We start with a stronger form of Spencer Theorem [24]:

Theorem 3 *Assume* $\mathbf{T}$ *is a computably enumerable, rich enough and consistent formal theory and* $f : \mathbf{N} \to \mathbf{N}$ *a computable function. Then there exist an incomputable set of theorems* I *in* $\mathbf{T}$ *such that for every* $S \in I$:

$$|\pi(S)| \geq f(|S|). \tag{2}$$

Proof. Assume by absurdity the existence of a computable function f as in the statement of the Theorem 3 such that for every theorem S in $\mathbf{T}$ we have

$$|\pi(S)| < f(|S|). \tag{3}$$

Under this assumption, we show that the set of theorems in $\mathbf{T}$ is computable, which contradicts Gödel's First Incompleteness Theorem [5] for $\mathbf{T}$. Indeed, the following algorithm decides membership in the set of theorems of $\mathbf{T}$. Given a formula S in $\mathbf{T}$

1. Calculate $f(|S|)$.

2. Enumerate the finite set of proofs σ with $|\sigma| < f(|S|)$.

3. If for some proof σ we have $\mathop{\vdash}\limits_{\sigma}^{\mathbf{T}} S$, then return "yes" and stop; otherwise, return "no" and stop.

This algorithm stops in finite time and returns the correct answer "yes". In case no proof σ proves S, then S is not a theorem of $\mathbf{T}$ because by (3) every theorem has a proof $|\pi(S)| < f(|S|)$, hence the "no" answer is also correct.

Finally the set

$$\{S \mid \stackrel{\mathbf{T}}{\vdash} S, |\pi(S)| \geq f(|S|)\} \tag{4}$$

is incomputable because otherwise the set $\{S \mid \stackrel{\mathbf{T}}{\vdash} S\}$ would be computable as the complement of the set (4) is computable, contradicting again Gödel's First Incompleteness Theorem [5] for $\mathbf{T}$. QED

Remark 1 Note the difference between the following two sets: a) $\{\sigma \mid \mathop{\vdash}\limits_{\sigma}^{\mathbf{T}} S$, for some $S\}$, and b) $\{S \mid \mathop{\vdash}\limits_{\sigma}^{\mathbf{T}} S$, for some $\sigma\}$. The first set is computable, but, in the context of Theorem 3, the second one is not.

Remark 2 Theorem 3 applies to PA, ZFC, the first-order theory of the rational numbers with addition, multiplication and equality, and the first-order theory of groups. In contrast, Presburger arithmetic, the first-order theory of the natural numbers in the signature with equality and addition, the first-order theory of Euclidean geometry and the first-order theory of Abelian groups are each decidable. Hence Theorem 3 does not work.

Next, we give a simple affirmative answer to the following open question [19, p. 112]:

> It remains, however, an open and interesting question whether the ratio of the [minimum-] length of proofs to the size of theorems is unbounded.

Corollary 2 *Assume* $\mathbf{T}$ *is a computably enumerable, rich enough and consistent formal theory. Then, for every positive integer* N *there exists a theorem* S *in* $\mathbf{T}$ *such that* $|\pi(S)| > N \times |S|$.

Proof. Let $f(n) = n^2$. By Theorem 3, there exists an incomputable set of theorems I (depending on f) such that for each $S \in I$, $|\pi(S)| \geq f(|S|) = |S|^2$. Giving a positive integer N we can choose $S \in I$ with $|S| > N$ (because I is incomputable, hence infinite) so that $|\pi(S)| > N \times |S|$. QED

Corollary 3 *Assume* **T** *is a computably enumerable, rich enough and consistent formal theory. Then, for every positive integer N, the set.*

$$\{S \mid \vdash^{\mathbf{T}} S, |\pi(S)| > N \times |S|\} \tag{5}$$

is incomputable.

Proof. The complement of the set (5) is computable, so by the Gödel's First Incompleteness Theorem [5] for **T**, the set (5) is computably enumerable and incomputable. QED

Consider a formal theory **T** over a finite alphabet Σ containing the symbol $\neg$. In **T** we fix two sets: P_1 is a non-empty computable set of sentences not starting with $\neg$ and its set of negations $P_2 = \{\neg S \mid S \in P_1\}$, and define two sets of sentences in P_1 provable in **T**:

$$\mathrm{Prov}_1 = \{S \in P_1 \mid \vdash S\}, \mathrm{Prov}_2 = \{S \in P_1 \mid \vdash \neg S\}.$$

Theorem 4 ([5]) *Let* **T** *be a computably enumerable, rich enough and consistent formal theory such that* Prov_1 *is not computable. Then, there exist infinitely many sentences S in P_1 such that S and $\neg S$ are not provable in* **T**.

The sentence "$N(P, v)$" says that the Turing machine P never halts on input v. So, for every Turing machine P and word v, "$N(P, v)$" is a perfectly definite sentence which is either true (if P never halts) or false (if P eventually halts). The falsity of "$N(P, v)$" can always be proved by exhibiting the sequence of Turing machine instructions run by P on v which leads to termination. However, due to Theorem 1, when "$N(P, v)$" is true, no finite sequence of instructions suffices to demonstrate it.

The sentence "$N(P, v)$" can be formalised in a sufficiently complicated formal theory **T** like PA or ZFC. In such a **T** we choose P_1 to be the set of sentences "$N(P, v)$". By Theorem 1, the set Prov_1 is computably enumerable but not computable; in fact Prov_1 is creative. Hence, by Theorem 2 we get:

Corollary 4 *One can effectively construct infinite subsets of "trivially true" sentences "$N(P, v)$" that require as long proofs as the hardest theorems of* **T**.

Let $f : \mathbf{N} \to \mathbf{N}$ be a computable non-decreasing function. For example, $f(n) = n + \log n$. We say that a proof π for S is *f-short* if $|\pi| \leq f(|S|)$; otherwise, π is *f-long.*

Theorem 3 shows the existence of an incomputable set of theorems with f-long proofs in $\mathbf{T}$. Is this set "small"?

The proof of Hartmanis Theorem 2 shows that the sets of "trivially true" theorems, which require as long proofs in $\mathbf{T}$ as the hardest theorems of $\mathbf{T}$ contain an open set in the prefix topology of words on Σ^* [7, 3] (the open sets are unions of sets $u\Sigma^*$). Theorems 5 and 6 prove a similar result in terms of minimal-length proofs.

Theorem 5 *Assume* $\{S \mid \vdash_{\pi}^{\mathbf{T}} S\}$ *is creative, and* $f : \mathbf{N} \to \mathbf{N}$ *is a computable function. Then we can effectively find an infinite computable subset* $L \subseteq \{S \mid \vdash_{\pi}^{\mathbf{T}} S\}$ *which can be accepted by a Turing machine* M *such that for every* $S \in L$ *we have:*

$$|\pi(S)| \geq f(\mathrm{space}_M(S)). \tag{6}$$

Proof. We use the following Turing machine T accepting $\{S \mid \vdash_{\pi}^{\mathbf{T}} S\}$. On input $S \subseteq \Sigma^*$ the machine T enumerates in length-lexicographical order all proofs π in $\mathbf{T}$ and accepts S as soon as π is a proof for S. Then, using a suitably large tape alphabet $\Sigma' \supseteq \Sigma$, we can construct T in such a way that $\mathrm{space}_T(S) = |\pi(S)|$ ([26]).

Corollary 4 in [18] shows the existence of L and M as in the statement of Theorem. QED

Theorem 6 *Let* $L \subseteq \Sigma^*$ *be an infinite computable set. Then there is a computable bijection* $\psi_L : \Sigma^* \to \Sigma^*$ *such that* $\psi_L(L)$ *contains an open subset* $u\Sigma^*$.

Proof. If $\Sigma^* \setminus L$ is finite the assertion is obvious.

If $\Sigma^* \setminus L$ is infinite, let $u \in \Sigma^*, |u| > 0$, and fix the computable bijections $f : \mathbf{N} \to L$, $g : \mathbf{N} \to \Sigma^* \setminus L$, $h_u : \mathbf{N} \to u\Sigma^*$, and $\bar{h}_u : \mathbf{N} \to \Sigma^* \setminus u\Sigma^*$, respectively.

Define the function $\psi_L : \Sigma^* \to \Sigma^*$ as follows:

(a) if $w \in L$ then set $\psi_L(w) = h_u(f^{-1}(w))$, and

(b) if $w \notin L$ then set $\psi_L(w) = \bar{h}(g^{-1}(w))$.

By construction, the function ψ_L is a computable bijection and by (a), $\psi_L(L) = u\Sigma^*$. QED

The set of theorems in a formal theory $\mathbf{T}$ is computably enumerable; hence by Theorem 6 contains, in some suitable topology, a non-empty open subset; hence it is not "small".

Finally, we prove an analogue of Theorem 3 for theorems with long statements and short proofs:

Theorem 7 *Assume* $\mathbf{T}$ *is a computably enumerable, rich enough and consistent formal theory. Then, there exist an infinite computable set of theorems in* $\mathbf{T}$ *with* $n + \log n$*-short proofs.*

Proof. Consider the theorem $S_x =$ "2^{1x} is even", where x is a non-empty binary word. The proof π "As 2^{1x} is a positive power of 2, hence it is even" has $|\pi|+$ constant $< |S_x| + \log|S_x|$, whenever $|x|$ is long enough. By varying x we get a computable set whose elements S_x have the required property.

QED

5 Proof-assistants

The sentence [1]

> This statement has no proof in PA that contains fewer than N symbols.

can be formulated in PA (using Gödel's method [14]) but cannot be proved with less than N symbols if PA is consistent. If we take an integer N larger than the number of particles of ordinary matter in the Universe, crudely estimated to 10^{80}, this proof cannot be written down even if one could write one symbol on each particle.

From an arithmetical point of view, the above sentence is not particularly interesting. Can we give relevant examples of theorems with long proofs?[2] The answer is affirmative.

The results presented above show that formalising mathematics comes with price tags, which include unprovable statements and the existence of infinite sets of "trivially true" theorems that have very long proofs. Therefore, it is essential to search various formalisations and to explore new axioms [12].

A seemingly naive question is: Can brute-force-proof-search be improved to become a helpful tool? The answer is related, at least in part, to the problem of "automating" mathematics.

Fix a formal theory for a part of mathematics, $\mathbf{A}$. There are at least three interpretations of "automating $\mathbf{A}$":

a) We can write an algorithm that decides whether an arbitrary statement in $\mathbf{A}$ can be proved or not in $\mathbf{A}$.

[2]Examples of interesting theorems with long proofs can be found in [27].

b) We can write an algorithm that finds proofs for all provable sentences in **A**.

c) An economically-viable algorithm can perform the human activity of proving theorems in **A**.

The alternative a) is valid for some **A** (like the propositional calculus) but false for more complex theories, like PA or ZFC for the whole mathematics, because of the Halting Theorem. The weaker interpretation b) is true because it does *not* require an algorithm to decide the provability status within **A** of an arbitrary sentence. Indeed, a brute-force algorithm can find proof for every sentence which is *provable* **A**. Such an algorithm is highly inefficient and impractical when proofs are very long. However, this is *not* a limitation affecting the working mathematician because humans cannot even read, even less understand, "too long" mathematical sentences.

Brute-force-proof-searches show that b) is possible, which is one reason for developing proof-assistants.

From b), we naturally arrive at c), which can be discussed from three points of view: computational complexity, economics, and epistemology.

Following Gödel [13], if P=NP, then there is a polynomial-time algorithm that given a first-order sentence and a positive integer n (in unary), decides whether the sentence has size n proof in ZFC. It may seem that under this hypothesis, there is no computational complexity obstacle to answering the question c) affirmatively. However, this is not true because the distinction between P and NP is mathematically, but *not* practically, meaningful: P=NP only implies that problems that can be verified in polynomial time can also be solved in polynomial time; however, polynomial-time algorithms are not necessarily practical [10]. A quadratic time algorithm can check very long proofs; in fact, proofs longer than any proof a human can write; hence, if the algorithm does not find an answer, then the sentence is practically/humanly impossible to prove.

Is it enough to know that something follows from some axioms and rules of inference, or is a proof something that provides deeper insight? Understanding is a crucial point in mathematics, so what kind of statements could be "humanly interesting"? What is the meaning of such a sentence? Is a practical prover epistemologically viable, too? Fortunately, proof-assistants are not ordinary algorithms working "in their world", but algorithms used in human-machine cooperation, in which *understanding* is essential and achievable [4, 8, 25].

Incompleteness can be proved by reduction to the Halting Theorem, so one possibility of improvement is to use approximate solutions to the Halting

Problem. Anytime algorithms trade execution time for quality of results [17]. Instead of correctness, an anytime algorithm returns a result together with a "quality measure" which evaluates how close the obtained outcome is to the result returned if the algorithm ran until completion. An efficient statistical algorithm for solving the Halting Theorem is in [6].

Acknowledgement

We thank the anonymous referee for helpful comments and suggestions that have improved the presentation.

References

[1] J. Baez. Insanely long proofs `https://johncarlosbaez.wordpress.com/2012/10/19/insanely-long-proofs/`.

[2] C. Calude. *Theories of Computational Complexity*. North-Holland, Amsterdam, 1988.

[3] C. Calude, H. Jürgensen, and M. Zimand. Is independence an exception? *Appl. Math. Comput.*, 66:63–76, 1994.

[4] C. Calude and S. Marcus. Mathematical proofs at a crossroad? In J. Karhumäki, H. A. Maurer, G. Păun, and G. Rozenberg, editors, *Theory Is Forever, Essays Dedicated to Arto Salomaa on the Occasion of His 70th Birthday*, volume 3113 of *Lecture Notes in Computer Science*, pages 15–28. Springer, 2004.

[5] C. S. Calude. Incompleteness and the halting problem. *Studia Logica* 109, 5(2021), 1159–1169.

[6] C. S. Calude and M. Dumitrescu. A statistical anytime algorithm for the halting problem. *Computability*, 9(2):155–166, 2020.

[7] C. S. Calude, H. Jürgensen, and L. Staiger. Topology on words. *Theor. Comput. Sci.*, 410(24-25):2323–2335, 2009.

[8] C. S. Calude and C. Müller. Formal proof: Reconciling correctness and understanding. In J. Carette, L. Dixon, C. S. Coen, and S. M. Watt, editors, *Intelligent Computer Mathematics, 16th Symposium, Calculemus 2009, 8th International Conference, MKM 2009, Held as Part of CICM*

2009, Grand Bend, Canada, July 6-12, 2009. Proceedings, volume 5625, pages 217–232, 2009.

[9] M. Davis. *Computability and Unsolvability.* McGraw-Hill, New York, 1958.

[10] M. Davis. Interview with Martin Davis. *Notices Amer. Math. Soc.*, 55(560–571), May 2008.

[11] M. Davis. *The Universal Computer: the Road from Leibniz to Turing.* W. W. Norton Company, New York, 2017.

[12] S. Feferman, H. M. Friedman, P. Maddy, and J. R. Steel. Does mathematics need new axioms? *Bulletin of Symbolic Logic*, 6(4):401–446, 2000.

[13] K. Gödel. Letter to John von Neumann, `https://www.anilada.com/notes/godel-letter.pdf`, 1956.

[14] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik*, 38:173–198, 1931.

[15] K. Gödel. Über die Länge von Beweisen. *Ergebnisse eines Mathematischen Kolloquiums*, 7:23–24, 1936.

[16] K. Gödel. *Collected Works.* Oxford University Press, 1986.

[17] J. Grass. Reasoning about computational resource allocation. An introduction to anytime algorithms. *Magazine Crossroads*, 3(1):16–20, 1996.

[18] J. Hartmanis. On effective speed-up and long proofs of trivial theorems in formal theories. *RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications*, 10(R1):29–38, 1976.

[19] F. H. Norwood. Long proofs. *The American Mathematical Monthly*, 89(2):110–112, 1982.

[20] P. Pudlák. On the length of proofs of finitistic consistency statements in first order theories. In J. Paris, A. Wilkie, and G. Wilmers, editors, *Logic Colloquium '84*, volume 120 of *Studies in Logic and the Foundations of Mathematics*, pages 165–196. Elsevier, 1986.

[21] P. Pudlák. The lengths of proofs. In S. R. Buss, editor, *Handbook of proof theory*, volume 137 of *Stud. Logic Found. Math.*, pages 547–637. Amsterdam: Elsevier, 1998.

[22] H. Rogers, Jr. *Theory of Recursive Functions and Effective Computability.* MacGraw-Hill, New York, 1967.

[23] M. Sipser. *Introduction to the Theory of Computation.* International Thomson Publishing, 2013 (3rd ed.).

[24] J. Spencer. Short theorems with long proofs. *The American Mathematical Monthly*, 90(6):365–366, 1983.

[25] D. Thompson. Formalisation vs. understanding. In C. S. Calude and M. J. Dinneen, editors, *Uncoventional Computation and Natural Computation: 14th International Conference, UCNC 2015, Auckland, New Zealand, August 30 – September 3, 2015, Proceedings*, pages 290–300. Springer, 2015.

[26] K. Wagner and G. Wechsung. *Computational Complexity*, volume 19 of *Mathematische Monographien.* Deutscher Verlag der Wissenschaften, Berlin, 1986.

[27] List of long mathematical proofs — Wikipedia, The Free Encyclopedia, 2021, `https://en.wikipedia.org/w/index.php?title=List_of_long_mathematical_proofs&oldid=1029115674`. [Online; accessed 18-May-2022].