Encoding NP Problems in Real/Transcendental Space

1. Introduction

The realm of computational complexity grapples with the inherent difficulty of solving problems, particularly those residing within the NP (nondeterministic polynomial time) space. While traditional computational models operate within discrete domains, exploring alternative representations of these problems could yield novel insights. Real, transcendental, and irrational numbers, with their infinite and non-repeating nature, possess the capacity to encode intricate information. Investigating the potential of these continuous number systems to represent NP problems is a compelling avenue of research. The intersection of number theory and computational complexity suggests a deeper connection between the structure of numbers and the nature of computation [1]. This exploration aims to transcend the limitations of discrete computation and leverage the richness of continuous number systems, potentially leading to new approaches for analyzing and possibly solving computationally hard problems.

Encoding NP problems within real or transcendental numbers could offer fresh perspectives on the P versus NP problem, a central question in computer science [4]. Such a representation might facilitate the development of novel algorithms or complexity measures, and could forge connections to other areas of mathematics, including analysis, topology, and abstract algebra. Furthermore, this approach holds the potential for encoding computational processes within the fundamental fabric of mathematical constants [13]. The structure of this report will delve into the classical methods of encoding, explore advanced numbering systems, investigate existing attempts to represent computational problems using real numbers, propose potential construction approaches for such encodings, and finally, identify suitable publication venues for this line of research.

2. Classical Encoding: Gödel Numbering

The concept of encoding symbolic structures into numbers has a rich history, most notably with Gödel numbering, originally formulated by Kurt Gödel in 1931 [18]. Gödel's ingenious method assigned a unique natural number, known as the Gödel number, to each symbol and well-formed formula within a formal language [18]. He employed prime factorization as a key technique to encode sequences of symbols representing formulas, and further, to encode sequences of formulas representing proofs [18]. For instance, in a specific Gödel numbering, the formula "0 = 0" could be encoded as 2⁶ × 3⁵ × 5⁶ [18]. The significance of this encoding lay in its ability to represent statements about natural numbers within the theory of arithmetic itself, enabling self-reference which was crucial for the proof of his incompleteness theorems [18]. This established a foundational link between formal systems and natural numbers, demonstrating that metamathematical statements could be mirrored within the system through numerical representation.

While Gödel's original formulation utilized a specific prime factorization scheme, it is important to note that infinitely many different Gödel numberings are possible [18]. Alternative methods can be constructed using bijective base-K numeral systems, where symbols are mapped to digits [18]. A simpler example of encoding sequences as numbers is the concatenation of ASCII codes, where each character is represented by its numerical ASCII value [18]. In set theory, an analogous concept exists with Gödel sets, where formulas are encoded using sets rather than numbers, which can be equivalent to Gödel numbers in certain cases but offers advantages in representing the tree structure of formulas [18]. The existence of these variations highlights the flexibility inherent in designing encoding schemes, suggesting that different methods might be more suitable for targeting specific number spaces like real or transcendental numbers.

The applications of Gödel numbering extend beyond the proof of incompleteness theorems. It has been instrumental in establishing the limits of provability in formal systems [22] and has a direct connection to the Halting Problem, demonstrating the inherent undecidability in computation [21]. Gödel numbering also provides a way to show the countability of countable mathematical objects, such as the set of all finite-length words over a countable alphabet [30]. Furthermore, it plays a role in formalizing recursion and self-reference within logical systems [18] and has connections to other encoding methods like Church encoding [18]. While its primary use has been in foundational theoretical results, the core principle of mapping symbolic structures to numbers has influenced various areas of theoretical computer science, including automata theory and potentially cryptography [1]. The success of Gödel numbering in proving profound theorems underscores the potential of encoding strategies for exploring complex mathematical and computational concepts.

3. Expanding the Encoding Landscape: Beyond Natural Numbers

The need to represent more complex structures, such as sequences and sets, has led to the development of encoding methods that go beyond the natural numbers. For encoding sequences as real numbers, several approaches have been explored. One straightforward method involves concatenating padded ASCII codes of the sequence elements [18]. Another approach utilizes binary representations, where each element of the sequence is converted to its binary form and then concatenated, often with a prefix indicating the length of each element or the sequence itself [36]. A more sophisticated method is the variation of Ackermann encoding, which provides a recursive mapping from hereditarily finite sets to real numbers using an infinite sum [16]. This encoding leverages the binary expansion of the real number to represent the characteristic function of the set. Furthermore, choice sequences, which are fundamental in constructive mathematics, can be encoded as real numbers within the interval by employing a recursive injective and surjective pairing function and a carefully constructed real number generator [38]. Real numbers themselves can also be represented by computable sequences of rational numbers that converge to the target real with a known rate [39]. For numerical data, base-b positional encoding schemes offer ways to represent numbers as sequences of tokens [42]. These diverse methods illustrate that the infinite nature of real numbers allows for encoding a wide range of discrete sequences and structures by carefully utilizing their binary or decimal expansions.

Encoding discrete structures into real numbers encompasses a variety of techniques. Church encoding, rooted in lambda calculus, provides a way to represent natural numbers, rational numbers, and even computable real numbers as higher-order functions [43]. This functional representation offers a unique perspective on encoding mathematical objects. Sets of integers can be encoded as real numbers using their binary representations, where the presence or absence of an integer in the set is indicated by the corresponding bit in the real number's expansion [36]. Variations of Ackermann encoding also serve this purpose [16]. The Cantor pairing function, while primarily mapping pairs of natural numbers to a single natural number, can be adapted to real numbers, although the uniqueness property is generally lost in the continuous domain [44]. Notably, Okhotin's number systems utilize equations over sets of natural numbers with operations like union and addition to represent sets with significant expressive power, capable of defining hyper-arithmetical sets [49]. These methods demonstrate that the choice of encoding depends heavily on the specific discrete structure and the properties one aims to preserve or analyze within the real number representation.

The concept of computable real numbers is central to bridging the gap between computational problems and their real number representations. A real number is deemed computable if it can be approximated to any desired precision by a finite, terminating algorithm, often formalized using Turing machines or computable functions [40]. Various equivalent definitions exist, including the existence of a Turing machine to generate its decimal expansion, a computable function to approximate it, a computable sequence of rationals that converges to it, or a computable Dedekind cut [55]. Computable real numbers possess the structure of a real closed field, meaning they are closed under arithmetic operations and taking roots of polynomials [55]. However, while the set of all real numbers is uncountable, the set of computable real numbers is only countable, implying the existence of non-computable real numbers, many of which are transcendental [13]. Furthermore, a fundamental result in computable analysis is the uncomputability of equality for computable real numbers, meaning there is no algorithm to definitively determine if two computable reals are equal [60]. Understanding the nature and limitations of computable real numbers is crucial when considering encoding NP problems, as it highlights the potential need to explore beyond this class to represent the full spectrum of computational complexity.

4. The Quest for Real Number Representations of NP Problems

The endeavor to directly encode NP-complete problems, such as SAT (Boolean satisfiability), into the fabric of real numbers presents a significant challenge. One potential approach involves mapping each clause and variable assignment of a Boolean formula into the digits of a real number. The satisfiability of the formula might then be represented by specific patterns or properties within the real number's expansion. Alternatively, the existence of a satisfying assignment could potentially be linked to whether a particular real number is a root of a specific polynomial. However, there is limited existing literature on direct, effective encodings of this nature that demonstrably preserve the inherent computational complexity of NP problems.

The field of algebraic complexity theory offers another perspective by studying the complexity of computing polynomials over real or complex numbers [35]. Within this framework, the P vs NP problem has an analog (PR vs NPR), concerning polynomial decidability and verifiability for problems over the reals [76]. Geometric Complexity Theory (GCT) represents a research program that aims to resolve the P vs NP question by translating it into problems within algebraic geometry and representation theory, focusing on the symmetries of computational problems [61]. If an NP problem could be reduced to determining a property of an algebraic variety, and if that variety's characteristics could be encoded within a real number, this could provide a potential link between NP complexity and real number properties.

Transcendental numbers, which are not roots of any polynomial with rational coefficients, might be particularly relevant for encoding the difficulty of NP problems [77]. Many transcendental numbers exhibit high Kolmogorov complexity, suggesting they contain a significant amount of irreducible information [77]. The Hartmanis-Stearns conjecture proposes a connection between real-time computable numbers and the dichotomy of rational and transcendental numbers, suggesting that algebraic numbers (beyond rationals) might not be easily recognizable computationally [83]. Furthermore, Christol's theorem establishes a link between automata theory and transcendence in the realm of formal power series over finite fields, hinting at a deeper relationship between computational processes and the algebraic nature of numbers and functions [89]. This suggests that an NP-complete problem could potentially be encoded into the digits or continued fraction expansion of a transcendental number, such that solving the NP problem becomes equivalent to determining a specific, computationally difficult property of that number.

5. Laying the Groundwork: Theoretical Foundations

The field of computable analysis provides a robust theoretical foundation for discussing computation involving real numbers [56]. It defines the complexity of real functions and operators, often using models like Turing machines adapted to handle real inputs or the Blum-Shub-Smale (BSS) model, which focuses on algebraic computations over the reals [76]. Complexity classes specific to real numbers, such as PR (polynomial decidable) and NPR (polynomial verifiable), have been defined, mirroring the P and NP classes in discrete complexity [76]. Understanding these frameworks is essential for establishing a rigorous basis for encoding NP problems into real numbers and analyzing the computational implications of such encodings. If the goal is to map NP problems to real numbers, the complexity of solving the NP problem must be reflected in the complexity of working with the corresponding real number within these established models.

Geometric Complexity Theory (GCT) offers another potential avenue by attempting to address the P vs NP problem through the lens of algebraic geometry and representation theory [61]. GCT seeks to translate computational complexity questions into problems about algebraic varieties and their symmetries, with a particular focus on the permanent versus determinant problem [61]. This approach suggests that the combinatorial structure of NP problems might be related to underlying algebraic or geometric properties that could be encoded within real numbers. For instance, an NP-complete problem might be represented as an algebraic variety, and a transcendental number could encode certain invariant properties of this variety, with the difficulty of solving the NP problem corresponding to the difficulty of extracting these invariants from the number.

The P vs NP conjecture itself, despite decades of research, remains one of the most significant open problems in computer science [7]. Its connection to number theory, particularly through problems like integer factorization which are believed to be in NP but not P, hints at a potential interplay between the structure of numbers and computational complexity [4]. The lack of fundamental progress in proving P ≠ NP suggests that novel approaches, such as encoding NP problems into real or transcendental numbers, might be necessary to gain new insights into the nature of this problem. If an NP-complete problem could be shown to be inherently linked to a deep, unresolved question in number theory concerning the properties of real or transcendental numbers, it could provide a fresh perspective on its computational hardness.

6. Building the Bridge: Practical Construction Approaches

Several potential approaches exist for practically constructing encodings of NP problems as real, irrational, or transcendental numbers. One method involves utilizing continued fractions. Irrational and transcendental numbers have infinite continued fraction representations, and the sequence of partial quotients could potentially encode the steps of a non-deterministic computation or the solution to an NP problem. A key challenge here would be ensuring that the solution can be verified in polynomial time from this continued fraction representation [84]. The structure of NP-complete problems like SAT, with their variables and clauses, might be mapped to specific patterns within the sequence of partial quotients.

Another approach involves encoding information within the binary or decimal expansions of transcendental numbers, such as π, e, or Liouville numbers [84]. The digits could represent the state space of a non-deterministic Turing machine or the truth assignments of variables in an NP problem. Furthermore, it might be possible to construct new transcendental numbers specifically designed to encode NP problems, potentially by embedding the output of a non-deterministic computation into the digit sequence [13]. The main challenge lies in ensuring that this encoding allows for verification and that the complexity of solving the NP problem is reflected in the difficulty of accessing or interpreting the encoded information.

A third possibility involves using the variation of Ackermann encoding that maps sets to real numbers [16]. If computational states or problem instances of an NP problem can be represented as sets, this encoding could provide a direct mapping to a real number. The properties of the resulting real numbers, particularly whether they are algebraic or transcendental, might then be related to the complexity of the encoded NP problem. The recursive nature of the Ackermann encoding could potentially capture the recursive structure inherent in many computational processes.

The following table summarizes these potential encoding approaches:

Approach	**Target Number Space**	**Potential Encoding Strategy**	**Challenges**
Continued Fractions	Irrational/Transcendental	Sequence of partial quotients represents computational steps or solution.	Ensuring polynomial-time verifiability, relating continued fraction properties to NP

			complexity.
Transcendental Expansions	Transcendental	Digits encode states, variables, or computational paths; constructing custom transcendentals.	Ensuring reversibility/verifiability, reflecting complexity in number properties.
Real Ackermann Encoding	Real	Mapping computational structures (sets of states/solutions) to real numbers.	Relating set-based representation to NP problems, analyzing the resulting real numbers.

7. Navigating the Publication Landscape: Target Journals

For disseminating research exploring the encoding of NP problems in real/transcendental space, several reputable mathematical journals specializing in computational complexity and mathematical logic would be suitable. *ACM Transactions on Computation Theory (TOCT)* is a highly relevant journal devoted to the study of computational complexity theory and allied fields, publishing original research on the limits of feasible computation, including areas like lower bounds and algebraic complexity [106]. The *Journal of the ACM (JACM)*, a top-tier general computer science journal with a rigorous peer-review process, would also be a strong candidate if the research presents significant and broadly applicable results in theoretical computer science [107]. *Theoretical Computer Science (TCS)*, published by Elsevier, welcomes papers introducing mathematical, logic, and formal concepts motivated by computing, covering abstract complexity and automata theory, making it a good fit for this interdisciplinary topic [109]. *Information and Computation*, also published by Elsevier, focuses on the fundamental aspects of information processing and computation, including the theoretical foundations of computer science, aligning well with the nature of this research [110]. Finally, *Mathematical Structures in Computer Science (MSCS)*, published by Cambridge University Press, specifically focuses on the application of ideas from the structural side of mathematics and mathematical logic to computer science, aiming to bridge the gap between theoretical contributions and applications, and welcoming work in areas like logic and algebra [111].

8. Meeting the Standards: Submission Guidelines and Citation Practices

Submitting to these journals requires careful adherence to their specific guidelines and style requirements. *ACM Transactions on Computation Theory (TOCT)*, according to SciSpace, follows ACM formatting guidelines and uses a numbered citation style, although Author Year is also common [113]. ACM provides LaTeX templates for submissions, requiring a single-column review format initially, with the final version adhering to ACM's style for print and digital display [114]. The *Journal of the ACM (JACM)* also requires manuscripts prepared using LaTeX or Word templates according to ACM guidelines, with PDF for submission and source files and figures uploaded separately; compliance with ACM Publication Ethics is mandatory [107]. For *Theoretical Computer Science (TCS)*, published by Elsevier, authors should consult the journal's specific formatting guidelines and citation style, which are likely Author Year or Numbered [109]. *Information and Computation*, another Elsevier journal, mandates a specific manuscript structure and uses a numbered citation style where references appear in square brackets before punctuation, following the ACS style guide [110]. *Mathematical Structures in Computer Science (MSCS)*, published by Cambridge University Press, utilizes an online peer review service and requires the use of the Harvard system for referencing (Author Year), with double-spaced manuscripts and LaTeX as the preferred format [111]. It is advisable to conduct further research on the most recent citation style specifics for each journal by examining their websites and recent publications to ensure accurate formatting.

9. A Formal Exposition and Philosophical Reflections

A formal mathematical exposition of encoding NP problems in real/transcendental space would begin by precisely defining the core concepts: the NP space, real numbers, transcendental numbers, and the notion of encoding itself. Potential encoding schemes would be formulated as formal mathematical functions mapping instances of NP problems (e.g., Boolean formulas, graphs) to specific real or transcendental numbers. Theorems or propositions would then be stated regarding the properties of these encodings, such as their uniqueness, reversibility (or the ability to verify solutions), and the relationship between the computational complexity of the original NP problem and the properties of the resulting number. Rigorous proofs or sketches of proofs would be necessary to establish the validity of these claims.

Beyond the formal mathematical treatment, a brief conceptual and philosophical introduction would elaborate on the motivations for this research and its potential impact on the fields of computational complexity and number theory. Connecting the idea of encoding computation in numbers to broader themes in mathematics and philosophy, such as the fundamental nature of mathematical objects and the relationship between the discrete and continuous, would provide valuable context. Significant philosophical insights could emerge from this exploration. For instance, the encoding of NP problems into real/transcendental numbers might offer new perspectives on the P vs NP problem, potentially revealing connections that are not apparent in traditional computational models [25]. The nature of truth and provability, particularly in relation to Gödel's incompleteness theorems, could be re-examined within this framework, as the limitations of formal systems might find new interpretations through continuous encodings [21]. Furthermore, the relationship between self-reference in computation, as demonstrated in Gödel's work, and the intrinsic properties of transcendental numbers, which often defy finite algebraic description, could yield profound philosophical insights [32]. The inherent limitations of formal systems and the potential for real/transcendental encodings to offer new perspectives on these boundaries warrant careful consideration [24].

10. Conclusion

This report has explored the potential for encoding problems in the NP space as numbers in the real, transcendental, or irrational space. While classical methods like Gödel numbering provide a foundation for encoding discrete structures into natural numbers, the quest to represent the complexity of NP problems in continuous number systems presents a frontier of research. Several potential approaches have been identified, including encoding via continued fractions, transcendental expansions, and variations of Ackermann encoding using real numbers. These methods face challenges in ensuring polynomial-time verifiability and reflecting the inherent difficulty of NP problems within the properties of the encoded numbers.

The theoretical groundwork for this endeavor lies in computable analysis, which provides tools for studying computation over real numbers, and algebraic complexity theory, which examines the complexity of algebraic computations. The enduring P vs NP conjecture, with its connections to number theory, further motivates the search for novel perspectives on computational hardness.

Future research should focus on developing specific and rigorous encoding schemes, analyzing their theoretical properties within established frameworks of computable analysis and complexity theory, and exploring their potential applications. The interdisciplinary nature of this research holds the promise of not only advancing our understanding of computational complexity but also uncovering deeper connections between the seemingly disparate fields of computer science and pure mathematics, potentially leading to significant philosophical insights into the nature of computation, truth, and the limits of mathematical knowledge.

Works cited

1. www.tutorialspoint.com, accessed March 30, 2025, https://www.tutorialspoint.com/automata_theory/automata_theory_godel_numbering.htm#:~:text=G%C3%B6del%20Numbering%20provides%20a%20method,theory%20to%20questions%20of%20logic.

2. Godel Numbering in Automata Theory - TutorialsPoint, accessed March 30, 2025, https://www.tutorialspoint.com/automata_theory/automata_theory_godel_numbering.htm
3. Number Theory in Discrete Mathematics | GeeksforGeeks, accessed March 30, 2025, https://www.geeksforgeeks.org/number-theory-in-discrete-mathematics/
4. Explained: P vs. NP | MIT News | Massachusetts Institute of Technology, accessed March 30, 2025, https://news.mit.edu/2009/explainer-pnp
5. Eli5: What is P vs NP? : r/explainlikeimfive - Reddit, accessed March 30, 2025, https://www.reddit.com/r/explainlikeimfive/comments/15fciqn/eli5_what_is_p_vs_np/
6. P vs NP Problems - GeeksforGeeks, accessed March 30, 2025, https://www.geeksforgeeks.org/p-vs-np-problems/
7. P versus NP problem - Wikipedia, accessed March 30, 2025, https://en.wikipedia.org/wiki/P_versus_NP_problem
8. ELI5 the P vs NP problem in mathematics : r/explainlikeimfive - Reddit, accessed March 30, 2025, https://www.reddit.com/r/explainlikeimfive/comments/ywmr3y/eli5_the_p_vs_np_problem_in_mathematics/
9. en.wikipedia.org, accessed March 30, 2025, https://en.wikipedia.org/wiki/P_versus_NP_problem#:~:text=In%20this%20theory%2C%20the%20class,given%20the%20right%20information%2C%20or
10. The P versus NP problem - Clay Mathematics Institute, accessed March 30, 2025, https://www.claymath.org/wp-content/uploads/2022/06/pvsnp.pdf
11. Fifty Years of P vs. NP and the Possibility of the Impossible - Communications of the ACM, accessed March 30, 2025, https://cacm.acm.org/research/fifty-years-of-p-vs-np-and-the-possibility-of-the-impossible/
12. Why is "P vs. NP" necessarily relevant? - MathOverflow, accessed March 30, 2025, https://mathoverflow.net/questions/54239/why-is-p-vs-np-necessarily-relevant
13. Transcendental number - Wikipedia, accessed March 30, 2025, https://en.wikipedia.org/wiki/Transcendental_number
14. Decidability of transcendental numbers - Theoretical Computer Science Stack Exchange, accessed March 30, 2025, https://cstheory.stackexchange.com/questions/10495/decidability-of-transcendental-numbers
15. Programmer's Guide To Theory - Transcendental Numbers, accessed March 30, 2025, https://www.i-programmer.info/programming/theory/15283-programmers-guide-to-theory-transcendental-numbers.html
16. Encoding Sets as Real Numbers - CEUR-WS, accessed March 30, 2025, https://ceur-ws.org/Vol-2199/paper1.pdf
17. Concerning the rarity of provably transcendental real numbers - MathOverflow, accessed March 30, 2025, https://mathoverflow.net/questions/59638/concerning-the-rarity-of-provably-transcendental-real-numbers

18. Gödel numbering - Wikipedia, accessed March 30, 2025, https://en.wikipedia.org/wiki/G%C3%B6del_numbering
19. On Formally Undecidable Propositions of Principia Mathematica and Related Systems, accessed March 30, 2025, https://en.wikipedia.org/wiki/On_Formally_Undecidable_Propositions_of_Principia_Mathematica_and_Related_Systems
20. A simplified explanation of Gödel's incompleteness proof: 3 - Logic, accessed March 30, 2025, https://jamesrmeyer.com/ffgit/GodelSimplified3
21. How Gödel's Proof Works | Quanta Magazine, accessed March 30, 2025, https://www.quantamagazine.org/how-godels-proof-works-20200714/
22. Gödel's Incompleteness Theorems (Stanford Encyclopedia of ..., accessed March 30, 2025, https://plato.stanford.edu/entries/goedel-incompleteness/
23. Gödel's Substitution Function in his incompleteness proof - Logic, accessed March 30, 2025, https://jamesrmeyer.com/ffgit/godel_sb
24. Gödel's incompleteness theorems - Wikipedia, accessed March 30, 2025, https://en.wikipedia.org/wiki/G%C3%B6del%27s_incompleteness_theorems
25. Gödel's Incompleteness Theorems and their Implications for Computing - TOM ROCKS MATHS, accessed March 30, 2025, https://tomrocksmaths.com/wp-content/uploads/2023/06/godels-incompleteness-theorems-and-their-implications-for-computing.pdf
26. Halting problem - Wikipedia, accessed March 30, 2025, https://en.wikipedia.org/wiki/Halting_problem
27. The Turing Machine Halting Problem | by Brent Morgan - Medium, accessed March 30, 2025, https://medium.com/@BrentMorgan/the-turing-machine-halting-problem-cd5bfb7bfbc9
28. Programmer's Guide To Theory - The Halting Problem, accessed March 30, 2025, https://www.i-programmer.info/programming/theory/13330-programmers-guide-to-theory-the-halting-problem.html?start=1
29. The halting problem is decidable on a set of asymptotic probability one (2006), accessed March 30, 2025, https://news.ycombinator.com/item?id=36105717
30. Could someone explain Gödel numbers? : r/learnmath - Reddit, accessed March 30, 2025, https://www.reddit.com/r/learnmath/comments/s7wkkp/could_someone_explain_g%C3%B6del_numbers/
31. Exploring Gödel's Numbering with Easy-ISLisp | by Kenichi Sasagawa | Medium, accessed March 30, 2025, https://medium.com/@kenichisasagawa/exploring-g%C3%B6dels-numbering-with-easy-islisp-a54ba87e2b38
32. Self-Reference and Paradox - Stanford Encyclopedia of Philosophy, accessed March 30, 2025, https://plato.stanford.edu/entries/self-reference/
33. CSCE 551 — Notes on Self-Reference in Computation, accessed March 30, 2025, https://cse.sc.edu/~fenner/csce551/self-reference.pdf
34. From Self-Referential Paradoxes to Intelligence | by CP Lu, PhD | Medium, accessed March 30, 2025, https://cplu.medium.com/from-self-referential-paradoxes-to-intelligence-b073cbeb

afe2
35. ALGEBRAIC AND TRANSCENDENTAL NUMBERS FROM AN INVITATION TO MODERN NUMBER THEORY 1. Introduction 1 2. Russell's Paradox and the, accessed March 30, 2025, https://web.williams.edu/Mathematics/sjmiller/public_html/238/currentnotes/AlgTrans_Chap5.pdf
36. Representing a finite set as a real number - Mathematics Stack Exchange, accessed March 30, 2025, https://math.stackexchange.com/questions/3642553/representing-a-finite-set-as-a-real-number
37. Encoding Sets as Real Numbers - LIRMM, accessed March 30, 2025, https://www.lirmm.fr/sets2018/papers/paper_4.pdf
38. Encoding Sequences in Intuitionistic Real Algebra - Qeios, accessed March 30, 2025, https://www.qeios.com/read/7IMXXE/pdf
39. elementary set theory - The "elements" of a real number - Mathematics Stack Exchange, accessed March 30, 2025, https://math.stackexchange.com/questions/3828775/the-elements-of-a-real-number
40. Alan Turing, On computable numbers | Joel David Hamkins, accessed March 30, 2025, https://jdh.hamkins.org/alan-turing-on-computable-numbers/
41. Computable analysis, exact real arithmetic and analytic functions in Coq, accessed March 30, 2025, https://staff.aist.go.jp/reynald.affeldt/coq2019/coqws2019-steinberg-thies-slides.pdf
42. Encoding numerical data for Transformers | by Aryamaan Thakur - Medium, accessed March 30, 2025, https://medium.com/@aryamaanthakur/encoding-numerical-data-for-transformers-1bb0c884e0c1
43. Church encoding - Wikipedia, accessed March 30, 2025, https://en.wikipedia.org/wiki/Church_encoding
44. Cantor pairing function - Wikipedia, accessed March 30, 2025, https://en.m.wikipedia.org/wiki/Cantor_pairing_function
45. Pairing Functions: Cantor & Szudzik - Vertex Fragment, accessed March 30, 2025, https://www.vertexfragment.com/ramblings/cantor-szudzik-pairing-functions/
46. The Cantor pairing function, accessed March 30, 2025, https://www.math.drexel.edu/~tolya/cantorpairing.pdf
47. Cantor Pairing Function. Represent an ordered pair of integers… | by Praveen Alex Mathew - Cantor's Paradise, accessed March 30, 2025, https://www.cantorsparadise.com/cantor-pairing-function-e213a8a89c2b
48. Is the Cantor Pairing function guaranteed to generate a unique real number for all real numbers? - Mathematics Stack Exchange, accessed March 30, 2025, https://math.stackexchange.com/questions/2437916/is-the-cantor-pairing-function-guaranteed-to-generate-a-unique-real-number-for-a
49. ON EQUATIONS OVER SETS OF NUMBERS AND THEIR ..., accessed March 30, 2025, https://www.worldscientific.com/doi/abs/10.1142/S012905411100809X
50. Logarithmic asymptotics of Landau–Okhotin function | Request PDF -

ResearchGate, accessed March 30, 2025, https://www.researchgate.net/publication/368225629_Logarithmic_asymptotics_of_Landau-Okhotin_function
51. (PDF) Representing Hyper-arithmetical Sets by Equations over Sets ..., accessed March 30, 2025, https://www.researchgate.net/publication/226377612_Representing_Hyper-arithmetical_Sets_by_Equations_over_Sets_of_Integers
52. Equations over Sets of Natural Numbers with Addition Only - DROPS, accessed March 30, 2025, https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.STACS.2009.1806
53. Least and greatest solutions of equations over sets of integers, accessed March 30, 2025, https://ii.uni.wroc.pl/~aje/prace/MFCS2010.pdf
54. www.cs.virginia.edu, accessed March 30, 2025, https://www.cs.virginia.edu/~robins/Turing_Paper_1936.pdf
55. Computable number - Wikipedia, accessed March 30, 2025, https://en.wikipedia.org/wiki/Computable_number
56. Computable analysis - Wikipedia, accessed March 30, 2025, https://en.wikipedia.org/wiki/Computable_analysis
57. mathoverflow.net, accessed March 30, 2025, https://mathoverflow.net/questions/252728/hard-to-compute-real-numbers#:~:text=Here%20is%20Turing's%20definition%3A%20A,number's%20decimal%20expansion%20as%20output.
58. computability theory - Hard-to-compute real numbers - MathOverflow, accessed March 30, 2025, https://mathoverflow.net/questions/252728/hard-to-compute-real-numbers
59. Constructive vs computable real numbers - Mathematics Stack Exchange, accessed March 30, 2025, https://math.stackexchange.com/questions/3884598/constructive-vs-computable-real-numbers
60. Uncomputability of the identity relation on computable real numbers - MathOverflow, accessed March 30, 2025, https://mathoverflow.net/questions/40618/uncomputability-of-the-identity-relation-on-computable-real-numbers
61. Introduction to geometric complexity theory - DCS - Department of Computer Science, accessed March 30, 2025, https://www.dcs.warwick.ac.uk/~u2270030/teaching_sb/summer17/introtogct/gct.pdf
62. Partition problems for geometric complexity theory, and the permanent vs. determinant problem. - Undergraduate Research - Colorado School of Mines, accessed March 30, 2025, https://www.mines.edu/undergraduate-research/partition-problems-for-geometric-complexity-theory-and-the-permanent-vs-determinant-problem/
63. Wikipedia-style explanation of Geometric Complexity Theory, accessed March 30, 2025, https://cstheory.stackexchange.com/questions/17610/wikipedia-style-explanation-of-geometric-complexity-theory

64. Geometric complexity theory - Wikipedia, accessed March 30, 2025, https://en.wikipedia.org/wiki/Geometric_complexity_theory
65. Geometry and Complexity Theory (Cambridge Studies in Advanced Mathematics, Series Number 169): 9781107199231: Landsberg, J. M. - Amazon.com, accessed March 30, 2025, https://www.amazon.com/Geometry-Complexity-Cambridge-Advanced-Mathematics/dp/1107199239
66. Hodge theory, between algebraicity and transcendence - ResearchGate, accessed March 30, 2025, https://www.researchgate.net/publication/357365559_Hodge_theory_between_algebraicity_and_transcendence
67. Algebraic Complexity Theory: Where the Abstract and the Practical Meet, accessed March 30, 2025, https://www.simonsfoundation.org/2021/02/24/algebraic-complexity-theory-where-the-abstract-and-the-practical-meet/
68. Independence in Algebraic Complexity Theory, accessed March 30, 2025, https://d-nb.info/1045872121/34
69. Algebraic complexity theory / Peter Bürgisser, Michael Clausen, M. Amin Shokrollahi ; with the collaboration of Thomas Lickteig. - University of California Berkeley - UC Library Search, accessed March 30, 2025, https://search.library.berkeley.edu/discovery/fulldisplay?vid=01UCS_BER%3AUCB&search_scope=DN_and_CI&tab=Default_UCLibrarySearch&docid=alma991065176489706532&lang=en&context=L&adaptor=Local%20Search%20Engine&query=sub%2Cexact%2C%20Stochastic%20processes%2CAND&mode=advanced&offset=0
70. Algebraic computation models (Chapter 16) - Cambridge University Press & Assessment, accessed March 30, 2025, https://www.cambridge.org/core/books/computational-complexity/algebraic-computation-models/4A62FD4F8AAEB1B42F29593D3E948DF5
71. Computational Complexity - Cambridge University Press & Assessment, accessed March 30, 2025, https://www.cambridge.org/core/books/computational-complexity/3453CAFDEB0B4820B186FE69A64E1086
72. Computation over algebraic structures and a classification of undecidable problems, accessed March 30, 2025, https://www.cambridge.org/core/journals/mathematical-structures-in-computer-science/article/computation-over-algebraic-structures-and-a-classification-of-undecidable-problems/64E8E1323A5E81D8BA5EE5769DC48C35
73. Implicit computation complexity in higher-order programming languages | Mathematical Structures in Computer Science | Cambridge Core, accessed March 30, 2025, https://www.cambridge.org/core/journals/mathematical-structures-in-computer-science/article/implicit-computation-complexity-in-higherorder-programming-languages/0E9A155F520EA7294C6A64F039479D33
74. Logical characterizations of algebraic circuit classes over integral domains | Mathematical Structures in Computer Science - Cambridge University Press &

Assessment, accessed March 30, 2025, https://www.cambridge.org/core/journals/mathematical-structures-in-computer-science/article/logical-characterizations-of-algebraic-circuit-classes-over-integral-domains/94541B2F327A87136EE80DAAE763B706
75. Arithmetic complexity computations | Algorithmics, complexity, computer algebra and computational geometry - Cambridge University Press & Assessment, accessed March 30, 2025, https://www.cambridge.org/fk/universitypress/subjects/computer-science/algorithmics-complexity-computer-algebra-and-computational-g/arithmetic-complexity-computations
76. Are there established complexity classes with real numbers? - Computer Science Stack Exchange, accessed March 30, 2025, https://cs.stackexchange.com/questions/29567/are-there-established-complexity-classes-with-real-numbers
77. Transcendentals - Complexity - follow the idea - Obsidian Publish, accessed March 30, 2025, https://publish.obsidian.md/followtheidea/Content/Math/Transcendentals+-+Complexity
78. publish.obsidian.md, accessed March 30, 2025, https://publish.obsidian.md/followtheidea/Content/Math/Transcendentals+-+Complexity#:~:text=Kolmogorov%20complexity%20(the%20measure%20of,numbers%2C%20indicating%20their%20fundamental%20uncompressibility.
79. Transcendentals - Randomness - follow the idea - Obsidian Publish, accessed March 30, 2025, https://publish.obsidian.md/followtheidea/Content/Math/Transcendentals+-+Randomness
80. Programmer's Guide To Theory - Kolmogorov Complexity, accessed March 30, 2025, https://www.i-programmer.info/programming/theory/13793-programmers-guide-to-theory-kolmogorov-complexity.html
81. kolmogorov complexity - Name for real that requires infinite bits to specify?, accessed March 30, 2025, https://math.stackexchange.com/questions/4413355/name-for-real-that-requires-infinite-bits-to-specify
82. arxiv.org, accessed March 30, 2025, https://arxiv.org/pdf/1610.04026
83. Why The Hartmanis-Stearns Conjecture Is Still Open | Gödel's Lost Letter and P=NP, accessed March 30, 2025, https://rjlipton.com/2012/06/15/why-the-hartmanis-stearns-conjecture-is-still-open/
84. The Real Conjecture of Hartmanis | Gödel's Lost Letter and P=NP, accessed March 30, 2025, https://rjlipton.com/2009/02/24/a-conjecture-of-hartmanis/
85. Hartmanis-Stearns conjecture and Mahler's method - Mathematical Institute, accessed March 30, 2025, https://www.maths.ox.ac.uk/node/11012
86. Hartmanis-Stearns Conjecture on Real Time and Transcendence - ResearchGate, accessed March 30, 2025, https://www.researchgate.net/publication/221351069_Hartmanis-Stearns_Conjecture_on_Real_Time_and_Transcendence

87. 1 Introduction 2 Extremely Sparse Sets - UMD Computer Science, accessed March 30, 2025, https://www.cs.umd.edu/~gasarch/open/juris.pdf
88. math history - Which automata recognise the algebraic numbers ..., accessed March 30, 2025, https://math.stackexchange.com/questions/147763/which-automata-recognise-the-algebraic-numbers
89. people.math.rochester.edu, accessed March 30, 2025, https://people.math.rochester.edu/faculty/dthakur2/autbanffFinal.pdf
90. Automata in Number Theory ∗ Contents - Boris.Adamczewski, accessed March 30, 2025, https://adamczewski.perso.math.cnrs.fr/Chapter25.pdf
91. Automata and Transcendence, accessed March 30, 2025, https://people.math.rochester.edu/faculty/dthakur2/autamsf.pdf
92. [2308.10977] Algebraic power series and their automatic complexity I: finite fields - arXiv, accessed March 30, 2025, https://arxiv.org/abs/2308.10977
93. (PDF) Algebraic power series and their automatic complexity I: finite fields - ResearchGate, accessed March 30, 2025, https://www.researchgate.net/publication/373297602_Algebraic_power_series_and_their_automatic_complexity_I_finite_fields
94. Semi-galois Categories II: An arithmetic analogue of Christol's theorem - arXiv, accessed March 30, 2025, https://arxiv.org/pdf/1710.01021
95. A note on Christol's theorem - Boris.Adamczewski, accessed March 30, 2025, https://adamczewski.perso.math.cnrs.fr/Note_On_Christol.pdf
96. Fast coefficient computation for algebraic power series in positive characteristic - MSP, accessed March 30, 2025, https://msp.org/obs/2019/2-1/obs-v2-n1-p08-p.pdf
97. www.brics.dk, accessed March 30, 2025, https://www.brics.dk/RS/03/50/BRICS-RS-03-50.pdf
98. (PDF) On representations of real numbers and the computational ..., accessed March 30, 2025, https://www.researchgate.net/publication/370058206_On_representations_of_real_numbers_and_the_computational_complexity_of_converting_between_such_representations
99. Computational complexity theory - Wikipedia, accessed March 30, 2025, https://en.wikipedia.org/wiki/Computational_complexity_theory
100. Computability & Complexity in Analysis Schedule - CCA Net, accessed March 30, 2025, http://www.cca-net.de/vasco/cca/tutorial.pdf
101. Complexity Theory for Operators in Analysis - Department of Computer Science, accessed March 30, 2025, https://www.cs.toronto.edu/~sacook/homepage/stoc.2010.pdf
102. On the Complexity of Real Functions - Mark Braverman, accessed March 30, 2025, https://mbraverm.princeton.edu/files/realFunctions05.pdf
103. Computing over the Reals: Where Turing Meets Newton1, accessed March 30, 2025, https://janos.cs.technion.ac.il/COURSES/238900-13/Papers/TuringMeetsNewton.pdf
104. The Existential Theory of the Reals as a Complexity Class: A Compendium -

ResearchGate, accessed March 30, 2025, https://www.researchgate.net/publication/382560131_The_Existential_Theory_of_the_Reals_as_a_Complexity_Class_A_Compendium
105. Recounting the History of Math's Transcendental Numbers - Quanta Magazine, accessed March 30, 2025, https://www.quantamagazine.org/recounting-the-history-of-maths-transcendental-numbers-20230627/
106. ACM Transactions on Computation Theory Submit your manuscript: http://www.editorialmanager.com/toct - ACM STOC Conference, accessed March 30, 2025, https://acm-stoc.org/stoc2020/acm-journals/toct-cfp-12-2019.pdf
107. Submissions - Applied and Computational Mechanics, accessed March 30, 2025, https://acm.kme.zcu.cz/acm/about/submissions
108. Guidelines and Criteria for Evaluation of Submissions for ACM Publications, accessed March 30, 2025, https://www.acm.org/publications/policies/pre-publication-evaluation
109. Theoretical Computer Science template - For Authors - SciSpace, accessed March 30, 2025, https://scispace.com/formats/elsevier/theoretical-computer-science/495d0f7a8ec681ae689c5a0b3db98aa0?key=hn5lwtwybtkhxtgzn9xjs1krzs6nj8txs3omkp53gumm3zdflwl0sewhivbwoq8n
110. Computation | Instructions for Authors - MDPI, accessed March 30, 2025, https://www.mdpi.com/journal/computation/instructions
111. Preparing your materials - Cambridge University Press & Assessment, accessed March 30, 2025, https://www.cambridge.org/core/journals/mathematical-structures-in-computer-science/information/author-instructions/preparing-your-materials
112. Mathematical Structures in Computer Science - Cambridge University Press & Assessment, accessed March 30, 2025, https://www.cambridge.org/core/journals/mathematical-structures-in-computer-science/information/author-instructions
113. ACM Transactions on Computation Theory (TOCT) Template ..., accessed March 30, 2025, https://scispace.com/formats/association-for-computing-machinery/acm-transactions-on-computation-theory-toct/5f66dd32971e422fa6c9b28e3cec95e6
114. Association for Computing Machinery (ACM) - Small Standard Format Template - Overleaf, accessed March 30, 2025, https://www.overleaf.com/latex/templates/association-for-computing-machinery-acm-small-standard-format-template/sksvmbxyfhnw
115. ACM TOPLAS Journal Submission Instructions - UCLA Compilers Group ·, accessed March 30, 2025, http://compilers.cs.ucla.edu/toplas/submissions.html
116. Submitting Articles to ACM Journals - Association for Computing ..., accessed March 30, 2025, https://www.acm.org/publications/authors/submissions
117. Mathematical Structures in Computer Science: Volume 25 - | Cambridge Core, accessed March 30, 2025, https://core-prod.cambridgecore.org/core/journals/mathematical-structures-in-computer-science/volume/4E459D03187FD8CB198C1484A2BDE7F1?pageNum=4

118. History Of Gödel Numbering Part 1 - Durable Scope, accessed March 30, 2025, https://blog.durablescope.com/post/HistoryOfGodelNumberingPart1/
119. dmi.unibas.ch, accessed March 30, 2025, https://dmi.unibas.ch/fileadmin/user_upload/dmi/Studium/Computer_Science/Vorlesung_HS20/Seminar_Turing_Award_Winners_and_Their_Contributions/report-example.pdf
120. How Gödel Proved Math's Inherent Limitations - Infinity Plus One - WordPress.com, accessed March 30, 2025, https://infinityplusonemath.wordpress.com/2017/09/04/how-godel-proved-maths-inherent-limitations/
121. Gödel and the limits of logic - plus.maths.org - Millennium Mathematics Project, accessed March 30, 2025, https://plus.maths.org/content/godel-and-limits-logic