International Journal of Education and Humanities
ISSN: 2770-6702 | Vol. 21, No. 2, 2025

Instructional Prelude of Theory of Computation: The History of Algebra

Honglin Wu*, Fangling Leng

School of Computer Science and Engineering, Northeastern University, Shenyang, China

Abstract: This paper examines the pedagogical benefits of introducing the history of algebra at the outset of a Theory of Computation course. By tracing the evolution from the abstraction of numbers and the invention of computational tools to the development of abstract concepts such as imaginary numbers, students are guided to recognize that the essence of computation resides in the abstraction and symbolic representation of human thought. This historical approach provides an intuitive and intellectually rich foundation for understanding core topics, including formal languages, computability, and computational complexity, while also encouraging students to reflect on the origins, development, and future trajectory of computational science.

Keywords: Theory of Computation, History of Algebra, Number Abstraction, Instructional Prelude.

1. Introduction: Why Begin with the History of Algebra?

The Theory of Computation course investigates computational models, algorithmic capabilities, and the fundamental limits of computation. Its content is inherently abstract, encompassing topics such as formal languages, automata, and Turing machines. For students encountering these concepts for the first time, immediate immersion into such abstractions can produce feelings of difficulty and intellectual distance. Therefore, an engaging introduction that connects intuition with conceptual understanding is essential [1].

Mathematics, and in particular algebra, represents the most direct historical and intellectual origin of computation theory [2]. At its core, the discipline seeks to understand what is computable and how computation can be performed efficiently. This endeavor begins with humanity's capacity to abstract real-world problems into symbols and manipulate them systematically, in other words, to compute.

Revisiting the history of algebra provides students with an accessible and intuitive entry point into these ideas [3]. It illustrates that the intellectual foundations of contemporary computational systems can be traced back to the earliest acts of quantification, such as counting with one's fingers, highlighting the enduring human pursuit of abstraction and symbolic reasoning.

2. The Leap of Thought: Number Abstraction and the Dawn of Computation

The first step in computation is abstraction. Numbers are not only tools for measurement but also a higher form of human thought. When early humans recognized the commonality among "two chickens," "two horses," and "two pigs," they achieved a fundamental cognitive leap: abstracting the idea of quantity from concrete entities [4].

This process of abstraction led to the emergence of arithmetic operations. Various civilizations contributed to the development of symbolic numerical frameworks [2]. Babylon employed a base-60 system, while India developed the base-10 system, which was later transmitted globally through the Arab world and became the international standard. The diversity of numeral systems, including binary and hexadecimal representations in modern computer science, demonstrates how the choice of representation directly influences computational efficiency, reflecting an early manifestation of the concept of encoding in computation theory.

The abstraction of numbers provided the objects of computation. Conversely, the need to manipulate these objects efficiently drove the invention of computational tools.

3. The Evolution of Computational Tools

As societies grew more complex, the demand for speed and accuracy in computation increased, prompting continuous innovation in computational instruments [2, 5].

3.1. Manual Era

The abacus, a representative manual computing tool, exemplifies the early integration of algorithmic rules with physical implements. It enabled systematic manipulation of numbers through structured operations on beads and rods.

3.2. Mechanical Era

Slide Rule (1622): Transformed complex operations such as multiplication, division, and logarithms into simple linear motions, providing an early example of analog computation.

Pascal's Calculator (1642): Mechanically implemented addition and subtraction, demonstrating the feasibility of replacing mental computation with machinery.

Babbage's Difference Engine (1822): A monumental step toward modern computing, this steam-powered device could perform automated calculations. Babbage is regarded as the "father of the computer" because his Analytical Engine already incorporated core principles of modern computing architecture, including input, processing, storage, and output.

The historical development of computational tools illustrates humanity's persistent effort to externalize, automate, and mechanize intellectual work. From Pascal's gears to Babbage's steam engine, and subsequently to vacuum tubes, transistors, and integrated circuits, the underlying objective has remained constant. Understanding this evolution provides essential historical context for discussions in the Theory of Computation, particularly concerning which computational models can be executed mechanically.

4. Deepening of Concepts: The Birth of Imaginary Numbers and the Expansion of Mathematical Thought

Computation theory addresses not only how to compute but also what can be computed, which requires an expansion of the mathematical objects themselves [2]. The history of imaginary numbers exemplifies how mathematical thought evolves by transcending traditional boundaries [6].

4.1. Confusion and Emergence

In the 16th century, Italian mathematician Gerolamo Cardano, while solving cubic equations, was compelled to consider the square roots of negative numbers. Although he regarded them as “false numbers,” he faithfully recorded them, introducing them into mathematical literature for the first time. Cardano’s complex and often tragic life, including personal struggles to preserve his reputation, provides a compelling teaching example, illustrating that scientific discovery frequently coexists with confusion, controversy, and human challenges.

4.2. Definition and Systematization

In the 17th century, engineer Rafael Bombelli encountered numerous practical problems requiring solutions to equations. Dissatisfied with the notion of “no solution,” he systematically defined new mathematical entities to resolve these issues. In Algebra, he introduced the imaginary unit and its operational rules, demonstrating how practical computational needs can directly drive conceptual innovation in mathematics.

4.3. Unification and Application

In the 18th century, Euler unified imaginary numbers, exponential functions, and trigonometric functions through his celebrated formula, revealing profound interconnections among seemingly unrelated branches of mathematics. From that point onward, imaginary numbers were no longer regarded as eccentric notions confined to theoretical mathematics; they became indispensable tools for analyzing alternating current circuits, describing quantum wave functions, and constructing control systems.

The history of imaginary numbers illustrates that the boundaries of computation are determined by the boundaries of our conceptual frameworks. Concepts once dismissed as illusory can ultimately become the most powerful means of describing reality. This development directly parallels computation theory, where the invention of the Turing machine, by defining an idealized mechanical process, clarified the fundamental limits of computability [8, 9].

5. Pedagogical Insights

Integrating the history of algebra into the Theory of Computation course provides multiple pedagogical benefits.

5.1. Establishing Intuitive Understanding

Situating abstract computational concepts within the broader narrative of human civilization renders them more tangible and relatable [7]. By tracing the progression from counting systems to symbolic arithmetic, mechanical calculation tools, and the eventual acceptance of imaginary numbers, students can see computation as a natural extension of human cognitive evolution rather than a set of isolated formal constructs. This approach aids in internalizing concepts such as formal languages, algorithmic procedures, and symbolic manipulation.

5.2. Revealing Inner Logic

Students gain insight into the intellectual progression from number abstraction to the development of computational tools and the creation of new mathematical entities. This historical sequence demonstrates the underlying logic of computational science, showing how modern computer science builds on centuries of systematic reasoning. Recognizing this continuity helps students understand why the field is structured as it is and how different topics-such as formal languages, Turing machines, and complexity theory-relate to one another.

5.3. Inspiring Creativity

The challenges faced by Cardano, the formalization achieved by Bombelli, and the unifying vision of Euler exemplify the courage and creativity required to transcend conventional thinking. Presenting these historical examples encourages students to approach fundamental questions in computation theory, such as the P versus NP problem, undecidability, or algorithmic optimization, with imaginative reasoning and intellectual boldness. Understanding that even foundational mathematical concepts emerged from human creativity fosters confidence in tackling complex and abstract computational problems.

5.4. Connecting Course Themes

The abstraction of numbers parallels formal languages and symbolic systems. The evolution of computational tools mirrors the development of computational models, such as Turing machines. The acceptance and utilization of imaginary numbers reflect the study of computability and decidability. In this way, history provides vivid and meaningful context for otherwise abstract theoretical constructions.

6. Conclusion

In summary, introducing the Theory of Computation through the historical development of algebra provides more than narrative enrichment; it constitutes a robust intellectual framework. By engaging with the conceptual challenges faced by historical mathematicians, students develop intuition, structural understanding, and creative problem-solving skills. This pedagogical strategy demonstrates that computation theory, like algebra, is both a technical discipline and a humanistic endeavor, shaped by centuries of exploration, abstraction, and conceptual innovation. Ultimately, this approach equips learners to navigate the abstract landscape of computation with historical perspective, intellectual rigor, and imaginative insight.

Acknowledgments

We would like to thank “The 2024 Graduate Curriculum Construction Project of Northeastern University (DBDXYKC2024104: Text Information Processing and Machine Translation)” and “The 2021 3rd Demonstration Courses for Thought of Northeastern University (Machine Translation)”.

References

[1] R.P. Perry, J.C. Smart: The Scholarship of Teaching and Learning in Higher Education: An Evidence-Based Perspective (Springer, Dordrecht 2007).

[2] W.S. Jiang: Mathematical elements (Tsinghua University Press, Beijing 2023).

[3] J.D. Novak: A Theory of Education (Cornell University Press, Ithaca 1977).

[4] Mickaël Launay: Great Mathematical Novels: From Prehistory to the Present (Le grand roman des maths: de la préhistoire à nos jours) (Flammarion Press, Paris 2016).

[5] C.X. Liu: Heard the Truth in the Morning (Jiangsu Phoenix Literature and Art Publishing, Nanjing 2019).

[6] T. Buzan, B. Buzan: The Mind Map Book: How to Use Radiant Thinking to Maximize Your Brain's Untapped Potential (Plume Press, New York 1996).

[7] D.L. Stufflebeam, C.L.S. Coryn: Evaluation Theory, Models, and Applications, 2nd Edition (Jossey-Bass, Indianapolis 2014).

[8] G.D. Phye: Handbook of Classroom Assessment: Learning, Achievement and Adjustment (Academic Press, Cambridge 1996).

[9] J.D. Novak, D.B. Gowin: Learning How to Learn (Cambridge University Press, Cambridge 1984).