Primary Historical Sources in the Classroom: Discrete Mathematics and Computer Science

Janet Barnett, Guram Bezhanishvili, Hing Leung, Jerry Lodder,
David Pengelley, Inna Pivkina, Desh Ranjan, Maria Zack

1 Introduction

A traditional method of instruction in the sciences and mathematics involves lecturing, solving template problems within the paradigm, and assigning parallel routine homework questions, as if the logical structure of the subject had always existed, awaiting memorization by our students and ourselves. Definitions are the point of departure, and properties of these defined objects follow without a detailed discussion of the motivating problems and intellectual struggle behind their origin. For example, simply presenting the truth table of an implication (an "if-then" statement in propositional logic) does little justice to the ancient Greek debate about when a hypothetical proposition (an "if-then" statement) should be true. Nor is justice done to the concerted attempts to render logic to a suitable computational form in the work of George Boole (1815–1864), Gottlob Frege (1848–1925), Alfred North Whitehead (1861–1947), Bertrand Russell (1872–1970), Ludwig Wittgenstein (1889–1951) or Emil Post (1897–1954), to name a few. Without a study of this historical backdrop, no wonder students are bewildered or bored by the modern proof, via truth tables, that an implication is logically equivalent to a certain inclusive "or" statement. More than two millennia passed before Russell and Whitehead were able to define an implication in terms of a statement of equivalent deductive power. Many other topics in computer science and discrete mathematics are introduced via announcement, such as the definition of a "tree" in graph theory as a connected graph with no cycles, or the formula for the sum of squares as

$$\sum_{i=1}^{n} i^2 = \frac{n^3}{3} + \frac{n^2}{2} + \frac{n}{6}$$

and the unenlightening proof of this equality by formal mathematical induction.

As an alternative form of instruction, this book of projects offers primary historical source material to provide context, motivation and direction to the development of certain topics in discrete mathematics and computer science. Students learn by reading the original work of the pioneers, studying the problems behind modern streamlined definitions, and, together with the instructor, arrive at an understanding of the mathematical concepts and methods that have taken shape over the centuries. Some projects, such as "Discovery of Huffman Codes," are centered around only a few sources and highlight a focused topic of recent development. Other projects, such as "Deduction Through the Ages: A History of Truth," begin with sources from antiquity and conclude with readings from the twentieth century. Each project contains excerpts from one or several historical sources, a discussion of the mathematical significance of each selection, and student exercises designed to illuminate the source. Readings and selected exercises could be assigned before the source is discussed in class. In-class activities could include an interpretation of the excerpt, an analysis of how the author's original writing resolves a particular issue, or a discussion of what mathematical techniques have evolved from the source. More challenging student exercises could be assigned after an initial discussion of the source. Instructors should familiarize themselves with the contents of a project, including the exercises, before assignment. Each project contains a section "Notes to the Instructor" with a summary of the project along with suggestions about class activities. The instructor should feel free to fashion their own exercises, assign further reading from the source material, if desired, and ask students to write summary papers of their learning, if a writing component is part of the course.

An entire course could be organized around several projects, allowing instruction from historical material without a modern textbook. We have taught several courses in this fashion, including introductory discrete mathematics, upper-divisional undergraduate combinatorics, and courses in upper-divisional mathematical logic. In this way, students witness the development of an entire subject area from primary historical sources and gain a sense of direction to the material. At the end of this introduction we offer several groupings of projects that could be used to teach these courses entirely from historical modules.

What are the advantages of learning from primary historical sources, especially when textbooks present a highly polished view of the subject? First, the source adds context. The historical author is often keenly motivated to solve a particular problem or present a novel view on problems with previous ad-hoc or fragmented solutions. We read what the problem was and how it was solved. Second, we see the motivation for the development of the subject, and are ourselves motivated to further study. Third, we gain a sense of direction to the subject matter, observing where the historical author begins, how a problem is solved, and what subsequent work builds on the solution. By reading the historical source, often before streamlined notation was developed, the student and instructor are forced to grapple with the verbal meaning of the passage, consider abstract formulations of ideas, and ask "What is an appropriate system of notation for this problem?" "What are the key properties to a solution to this problem?" Answers to these questions often find resolution in present-day definitions, theorems, and algorithms. Readers of a primary source may experience a bit of cognitive dissonance when presented with a historical passage and are unable to see modern mathematics. The thought process required to bridge the gap between the historical and the modern offers an invaluable learning experience that solidifies the concept in the learner's mind. On the other hand, some readers may find it refreshing to learn from a verbal description of the problem without the axiomatic constraints of a pre-defined system of calculation. In this way, original sources offer alternative learning strategies, and require no previous knowledge of the subject.

We learn more than just the conceptual roots of computer science and mathematics by studying historical sources, but also gain insight into the process of discovery itself by following the thoughts of some of the greatest minds throughout history. We gain an appreciation of the cultural and intellectual setting in which the author was writing. Concerning the benefits of learning from historical sources, students themselves claim the following on end-of-the-semester questionnaires, after studying from projects:

> "You get answers to questions like 'Where did all of this come from?' "
>
> "It helps me understand the reason why things were put together like they are."
>
> "I like to see where everything comes from and how it works."
>
> "You learn the concept."

"It makes me care about learning."

What topics are covered in this compendium of projects? Certainly almost all topics in an introductory undergraduate discrete mathematics course, several topics in algorithm design and data structure for computer science, topics in automata theory, combinatorics and abstract algebra for upper-divisional undergraduate courses, and topics in logic from elementary to the beginning graduate level. To present the unity of mathematical and algorithmic thought, and encourage the interdisciplinary use of these curricular materials, the projects are not arranged into chapters, but listed in a vaguely increasing level of sophistication. Here then is the list of project titles, specific topics contained in each, and the traditional courses in which the project could be used.

1. "Deduction through the Ages: A History of Truth"

   Topics: Elementary propositional logic, truth tables, the truth table of an implication.

   Courses: Introductory Discrete Mathematics, A course in discrete mathematics for beginning computer science majors.

2. "Sums of Numerical Powers in Discrete Mathematics: Archimedes Sums Squares in the Sand"

   Topics: Summations, closed formulas, Riemann sums, areas, inequalities, telescoping sums, mathematical induction (optional), systems of linear equations (optional), recursive computation (optional.)

   Courses: Introductory Discrete Mathematics, Calculus.

3. "Euclid's Algorithm for the Greatest Common Divisor"

   Topics: Basic division properties of natural numbers, divisors and common divisors, greatest common divisor, basic reasoning, arguments and proofs, formal proof by induction (optional), basic algorithm design and formal description (pseudo-code), reasoning about correctness of algorithms.

   Courses: Introductory Discrete Mathematics, Introduction to Computer Science.

4. "An Introduction to Symbolic Logic"

   Topics: Propositional logic, logical connectives, logical equivalence, truth tables, tautologies, inference rules, predicate logic, individual variables and predicates, universal and existential quantifiers, logical equivalence of quantified statements, inference rules in predicate logic.

   Courses: Introductory Discrete Mathematics, A course in discrete mathematics for beginning computer science majors.

5 "An Introduction to Elementary Set Theory"

   Topics: Sets, subsets, set equality, set union and intersection, set complement, the empty set, power sets, Cartesian products, Russell's paradox, functions, function image and inverse image, composition of functions, one-to-one functions and correspondences, set equivalence, Cardinal numbers, finite and infinite sets, countable and uncountable sets.

   Courses: Introductory Discrete Mathematics, A course in discrete mathematics for beginning computer science majors.

6. “Computing the Determinant Through the Looking Glass”

   Topics: Determinants, systems of linear equations, row reduction, elementary operations with matrices.

   Courses: Linear Algebra, Introductory Discrete Mathematics.

7. “Origins of Boolean Algebra in the Logic of Classes: George Boole, John Venn and C. S. Peirce”

   Topics: Elementary set operations (union, intersection, relative difference, symmetric difference) and their (algebraic) properties, Venn diagrams, relation of set operations to standard language use, relation of elementary set operations to Boolean algebra, inverse operations (optional), issues related to mathematical notation (optional), changing standards of rigor and proof (optional).

   Courses: Introductory Discrete Mathematics.

8. “Boolean Algebra as an Abstract Structure: Edward V. Huntington and Axiomatization”

   Topics: Axioms for Boolean algebra, Boolean algebra properties (with proofs), two-valued model of Boolean algebra, axiomatization, consistency, independence.

   Courses: Discrete Mathematics (introductory or intermediate), Model Theory, Abstract Algebra.

9. “Applications of Boolean Algebra: Claude Shannon and Circuit Design”

   Topics: Boolean algebra axioms and properties, application of Boolean algebra to design and simplification of circuits, simplification of Boolean expressions, disjunctive normal form, two-valued model of Boolean algebra.

   Courses: A course in discrete mathematics for computer science majors, Introductory Discrete Mathematics.

10. “Figurate Numbers and Sums of Numerical Powers: Fermat, Pascal, Bernoulli”

    Topics: Sums of powers, figurate numbers, binomial coefficients, combination numbers, Pascal’s triangle, Bernoulli numbers, recursive definitions and computation, summations and inequalities, counting and geometry, mathematical induction, interplay between discrete sums and calculus.

    Courses: Combinatorics, Discrete Mathematics (upper-divisional).

11. “Gabriel Lamé’s Counting of Triangulations”

    Topics: Triangulations of a convex polygon, the Catalan numbers, enumeration of triangulations, binomial coefficients.

    Courses: Combinatorics, Discrete Mathematics (upper-divisional), Analysis of Algorithms.

12. “Networks and Spanning Trees”

    Topics: Graphs, trees, enumeration of labeled trees, minimal spanning trees, Cayley’s formula and Prüfer’s symbols for labeled trees, Borůvka’s algorithm for finding a minimal spanning tree.

Courses: Graph Theory, Combinatorics, Analysis of Algorithms.

13. “Striving for Efficiency in Algorithms: Sorting”

    Topics: Sorting, quicksort, insertion sort, efficiency of algorithms, stack data structure.

    Courses: Data Structures and Algorithms.

14. “Discovery of Huffman Codes”

    Topics: Huffman encoding, greedy algorithm, prefix-free code, optimal code, unit of information, amount of information, binary tree.

    Courses: Data Structures and Algorithms.

15. “Project on Program Correctness”

    Topics: Correctness, partial correctness, algorithms.

    Courses: Algorithms, Data Structures, Programming Languages.

16. “Regular Languages and Finite Automata”

    Topics: Regular Languages, finite automata, regular expressions, regular events, Kleene’s theorem, Kleene’s star operation.

    Courses: Automata and Formal Languages, Theory of Computation.

17. “Henkin’s Method and the Completeness Theorem”

    Topics: Syntax of first-order logic, axioms and rules of inference, formal concept of proof, deduction theorem, satisfiability and validity, soundness, Henkin’s method, completeness, Loewenheim-Skolem theorem, first-order logic with equality.

    Courses: Mathematical Logic (advanced).

18. “Peano Arithmetic”

    Topics: Peano postulates, consistency and independence of Peano postulates, recursive definition of addition and multiplication, properties of addition and multiplication, first-order arithmetic .

    Courses: Mathematical Logic (advanced).

19. “Goedel’s Incompleteness Theorems”

    Topics: Self-referential statements and paradoxes, first-order languages and first-order theories, models, Peano arithmetic (PA), diagonalization lemma, incompleteness of PA (Goedel’s first incompleteness theorem), unprovability of consistency (Goedel’s second incompleteness theorem).

    Courses: Mathematical Logic (advanced).

20. “Undecidability of First-Order Logic”

    Topics: Different formulations of the decision problem, models, isomorphic models, intended model, one-way infinite line, two-way infinite line, upper half-plane, Turing machines, halting problem, undecidability of first-order satisfiability.

Courses: Mathematical Logic (advanced).

The following subject areas are the primary (unique) groupings of the projects by traditional courses:

Discrete Mathematics

1. "Deduction through the Ages: A History of Truth"
4. "An Introduction to Symbolic Logic"
5. "An Introduction to Elementary Set Theory"
7. "Origins of Boolean Algebra in the Logic of Classes: George Boole, John Venn and C. S. Peirce"
8. "Boolean Algebra as an Abstract Structure: Edward V. Huntington and Axiomatization"

Linear Algebra

6. "Computing the Determinant Through the Looking Glass"

Combinatorics

2. "Sums of Numerical Powers in Discrete Mathematics: Archimedes Sums Squares in the Sand"
10. "Figurate Numbers and Sums of Numerical Powers: Fermat, Pascal, Bernoulli"
11. "Gabriel Lamé's Counting of Triangulations"
12. "Networks and Spanning Trees"

Computer Science

3. "Euclid's Algorithm for the Greatest Common Divisor"
9. "Applications of Boolean Algebra: Claude Shannon and Circuit Design"
13. "Striving for Efficiency in Algorithms: Sorting"
14. "Discovery of Huffman Codes"
15. "Project on Program Correctness"
16. "Regular Languages and Finite Automata"

Advanced Logic

17. "Henkin's Method and the Completeness Theorem"
18. "Peano Arithmetic"
19. "Goedel's Incompleteness Theorems"
20. "Undecidability of First-Order Logic"

The following projects could be used to teach entire courses from historical modules.

An Introductory Discrete Mathematics Course

1. “Deduction through the Ages: A History of Truth”
4. “An Introduction to Symbolic Logic”
5. “An Introduction to Elementary Set Theory”
2. “Sums of Numerical Powers in Discrete Mathematics: Archimedes Sums Squares in the Sand”

   “Pascal’s Treatise on the Arithmetical Triangle,” [1, 2]

An Upper-Divisional Combinatorics Course

10. “Figurate Numbers and Sums of Numerical Powers: Fermat, Pascal, Bernoulli”
11. “Gabriel Lamé’s Counting of Triangulations”
12. “Networks and Spanning Trees”

An Upper-Divisional Course in Mathematical Logic

17. “Henkin’s Method and the Completeness Theorem”
18. “Peano Arithmetic”
19. “Goedel’s Incompleteness Theorems”

Further information about these projects, along with the pedagogical approach of teaching from historical sources, can be found from our web resource [3]. This compendium of projects builds on those already in print [1], where the reader will find additional projects related to logic, mathematical induction, combinatorics, graph theory, Turing machines and automata theory. These previous projects can be viewed on a separate web site [2].

2 Acknowledgment

The development of curricular materials for discrete mathematics has been partially supported by the National Science Foundation’s Course, Curriculum and Laboratory Improvement Program under grants DUE-0717752 and DUE-0715392 for which the authors are most appreciative. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Barnett J., Bezhanishvili G., Leung H., Lodder J., Pengelley D., Ranjan D., “Historical Projects in Discrete Mathematics and Computer Science,” in *Resources for Teaching Discrete Mathematics* (Hopkins, B., editor), Mathematical Association of America, Washington DC, 2009.

[2] Barnett J., Bezhanishvili G., Leung H., Lodder J., Pengelley D., Ranjan D., *Teaching Discrete Mathematics via Primary Historical Sources*, `http://www.math.nmsu.edu/hist_projects/`.

[3] Barnett J., Bezhanishvili G., Leung H., Lodder J., Pengelley D., Pivkina I., Ranjan D., *Learning Discrete Mathematics and Computer Science via Primary Historical Sources*, `http://www.cs.nmsu.edu/historical-projects/`.