

Historical Projects in Discrete Mathematics and Computer Science

Janet Barnett, Guram Bezhanishvili, Hing Leung,
Jerry Lodder, David Pengelley, Desh Ranjan

1 Introduction

A course in discrete mathematics is a relatively recent addition, within the last 30 or 40 years, to the modern American undergraduate curriculum, born out of a need to instruct computer science majors in algorithmic thought. The roots of discrete mathematics, however, are as old as mathematics itself, with the notion of counting a discrete operation, usually cited as the first mathematical development in ancient cultures [38]. By contrast, a course in finite mathematics is sometimes presented as a fast-paced news reel of facts and formulae, often memorized by the students, with the text offering only passing mention of the motivating problems and original work that eventually found resolution in the modern concepts of induction, recursion and algorithm. This chapter focuses on the pedagogy of historical projects, which offer excerpts from original sources, place the material in context, and provide direction to the subject matter.

Each historical project is centered around a publication of mathematical significance, such as Blaise Pascal's "Treatise on the Arithmetical Triangle" [53, vol. 30] from the 1650s or Alan Turing's 1936 paper "On Computable Numbers with an Application to the Entscheidungsproblem" [66]. The projects are designed to introduce or provide supplementary material for topics in the curriculum, such as induction in a discrete mathematics course, or compilers and computability for a computer science course. Each project provides a discussion of the historical exigency of the piece, a few biographical comments about the author, excerpts from the original work, and a sequence of questions to help the student appreciate the source. The main pedagogical idea is to teach and learn certain course topics from the primary historical source, thus recovering motivation for studying the material.

For use in the classroom, allow several weeks per project with one or two projects per course. Each project should count for a significant portion of the course grade (about 20%) and may take the place of an in-class examination. Begin early in the course with a discussion of the relevance of the historical piece, its relation to the course curriculum, and how modern textbook techniques owe their development to problems often posed centuries earlier. While a project is assigned, several class activities are possible. Students could be encouraged to work on the project in class, either individually or in small groups, as the instructor monitors their progress and explores meaning in language from time past. A comparison with modern techniques could begin as soon as the students have read the related historical passages. For example, after reading Pascal's verbal description of what today is recognized as induction, the instructor could lead a discussion comparing this to the axiomatic formulation of induction found in the textbook. Finally, the historical source can be used to provide discovery exercises for related course material. In his 1703 publication "An Explanation of Binary Arithmetic" [20], Gottfried Leibniz introduces the binary system of numeration, states its advantages in terms of efficiency of calculation, and claims that this system allows for the discovery of other properties of numbers, such as patterns in the base two expansion of the perfect squares.

An engaging in-class exercise is to examine patterns in a table of perfect squares (base two) and conjecture corresponding divisibility properties of the integers. The pattern of zeroes in the binary equivalent of n^2 , n odd, leads to the conjecture that $8|(n^2 - 1)$, n odd, where the vertical bar denotes “divides.” Construct the table! Time spent working on the project is time for explanation, exploration, and discovery, for both the instructor and the student.

Instructors are encouraged to adapt each project to their particular course. Add or rephrase some questions, or delete others to reflect what is actually being covered. Be familiar with all details of a project before assignment. The source file for each project together with its bibliographic references can be downloaded and edited from the web resource [2]. The topics covered by the projects include set theory, mathematical induction, binary arithmetic, computability, graph theory, and the combinatorics of the Catalan numbers. They range in level from beginning undergraduate courses in discrete mathematics to advanced undergraduate courses in logic, graph theory, and computer science. Most are independent of each other, although a few projects build on the same source or present closely related topics via different sources. Each historical source is one which either solved an outstanding problem, inaugurated a mathematical technique, or offered a novel point of view on existing material. For example, although Leibniz was not the first to experiment with base two numeration [22], his paper “An Explanation of Binary Arithmetic” presents the topic as a confluence of order, harmony, and ease of calculation.

Here is a brief description of the projects appearing in this chapter, an indication of their mathematical sophistication, and a statement of topics covered in each. The project *Are All Infinities Created Equal?* touches on naive set theory, countability, and one-to-one correspondences between sets using excerpts from George Cantor’s original work [6] in the late 1890s. The project is well suited for a beginning undergraduate course in discrete mathematics. The idea of computability is raised in the project *An Introduction to Turing Machines*, based on Alan Turing’s 1936 paper [66] in which he solves the decision problem (das Entscheidungsproblem). The project remains at an introductory level for a first course in programming or discrete mathematics, and asks the student to design machines that perform simple tasks. Building on this, the project *Turing Machines, Induction and Recursion*, requires the student to identify a pattern in the output of a machine and use induction to prove that the pattern persists. The project continues with the construction of a machine to compute the product of two positive integers recursively in unitary notation. A variant of this used recently in a beginning discrete mathematics course requires the computation of the product in binary notation, and illustrates the efficiency of base two calculations, particularly when compared to base ten. At the intermediate to advanced undergraduate level are two additional projects, *The Universal Computing Machine*, and *The Decision Problem*, that sketch the main results of Turing’s paper. The universal machine is a Turing machine that accepts as its input any other machine, T , and computes the same output as T . This foreshadows the development of a compiler or interpreter in computer science. The decision problem asks whether there is a decision procedure that can be applied to any well-formulated mathematical statement and determine whether the statement is true or false. Turing’s negative solution to this problem forms the historical legacy of his paper, not to mention the concept of “Turing computable” to refer to that which can be computed via a Turing machine.

Computability forms the topic of two additional advanced projects, *Church’s Thesis*, and *Two-Way Deterministic Finite Automata*, both of which are written to be independent of the four Turing projects, although they could be used in any combination. Alonzo Church examines a general logical notion of computability in terms of specific types of functions, which then forms the basis of “Church’s thesis.” The project contains excerpts from the original work of Kurt Gödel [23] and Stephen Kleene [39] from the 1930s in addition to that of Turing. In computer science today a Turing machine is modeled by an automaton. A key result of John Shepherdson [62] from 1959

states that whatever can be computed via an automaton reading a tape in two directions (forwards and backwards) can be computed via an automaton reading a tape in just one direction.

The base two number system is explored in the beginning-level projects *Binary Arithmetic: From Leibniz to von Neumann*, and *Arithmetic Backwards from Shannon to the Chinese Abacus*. The first of these introduces the reader to binary numeration from Gottfried Leibniz's 1703 publication "An Explanation of Binary Arithmetic" [20], and compares this to John von Neumann's use of base two calculations in some of the first electronic digital computers from 1945 [69]. The second project begins with Claude Shannon's analysis of the circuitry necessary to perform base two addition from his 1938 paper "A Symbolic Analysis of Relay and Switching Circuits" [60]. Taking a step backwards chronologically, the project examines addition and subtraction on a Chinese abacus, which, when used to its full potential, provides an excellent device for base sixteen (hexadecimal) arithmetic, since the numerical values from zero to fifteen inclusive can be represented on each bar of such an abacus. This use of the abacus does not follow the historical record, but is presented as an enrichment exercise, particularly for two-power base arithmetic. A final project at the introductory level is *Treatise on the Arithmetical Triangle*, which presents the concept of mathematical induction from the pioneering work of Blaise Pascal [53, vol. 30] in the 1650s. After arranging the figurate numbers in one table, forming "Pascal's triangle," the French scholar notices several patterns in the table, which he would like to claim continue indefinitely. Exhibiting unusual rigor for his day, Pascal offers a condition for the persistence of a pattern, stated verbally in his Twelfth Consequence, a condition known today as mathematical induction. Moreover, the Twelfth Consequence results in the modern formula for the combination numbers or binomial coefficients. An upper-level project on combinatorics is *Counting Triangulations of a Polygon*, which presents Gabriel Lamé's 1838 derivation [47] for the number of triangulations of a convex n -sided polygon in terms of a simple recursion relation. From this follows easily the modern formula for the "Catalan numbers" in terms of binomial coefficients.

The compendium of historical projects is concluded with three projects on graph theory for an upper-level mathematics or computer science course. The project *Euler Circuits and the Königsberg Bridge Problem* offers Leonhard Euler's 1736 solution [3] to what today is phrased as finding a closed path in a graph that traverses each edge exactly once. The project *Topological Connections from Graph Theory* studies the idea of flow around a network (graph) and the resulting linear equations, one for each circuit in the graph. The problem of finding linearly independent equations for a given graph is studied from the work of Oswald Veblen [3] of 1922. A final project on graph theory is *Hamiltonian Circuits and Icosian Game*, which presents the work of William Hamilton [3] from the late 1850s on what modern mathematics describes as a path in a graph that touches each vertex exactly once, i.e., a Hamiltonian path, or a Hamiltonian cycle if the beginning vertex is the same as the ending vertex.

After completion of a course using historical projects, students write the following about the benefits of history: "See how the concepts developed and understand the process." "Learn the roots of what you've come to believe in." "Appropriate question building." "Helps with English-math conversion." "It leads me to my own discoveries."

ACKNOWLEDGMENT

The development of curricular materials for discrete mathematics and computer science has been partially supported by the National Science Foundation's Course, Curriculum and Laboratory Improvement Program under grant DUE-0231113, 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.

2 Are All Infinities Created Equal?

Guram Bezhanishvili^{1 2}

Georg Ferdinand Ludwig Philip Cantor (1845–1918), the founder of set theory, and considered by many as one of the most original minds in the history of mathematics, was born at St. Petersburg, Russia in 1845. His parents, who were of Jewish descent, moved the family to Frankfurt, Germany in 1856. Georg entered the Wiesbaden Gymnasium at the age of 15, and two years later began his university career at Zürich. In 1863 he moved to the University of Berlin, which during Cantor’s time was considered the world’s center of mathematical research. Four years later Cantor received his doctorate from the great Karl Weierstrass (1815–1897). In 1869 Cantor obtained an unpaid lecturing post, which ten years later flourished into a full professorship, at the minor University of Halle. However, he never achieved his dream of holding a Chair of Mathematics at Berlin. It is believed that one of the main reasons was nonacceptance of his theories of infinite sets by the leading mathematicians of that time, most noticeably by Leopold Kronecker (1823–1891), a professor at the University of Berlin and a very influential figure in German mathematics, both mathematically and politically.

Cantor married in 1874 and had two sons and four daughters. Ten years later Georg suffered the first of the mental breakdowns that were to plague him for the rest of his life. He died in 1918 in a mental hospital at Halle. By that time his revolutionary ideas were becoming accepted by some of the leading figures of the new century. For example, one of the greatest mathematicians of the twentieth century, David Hilbert (1862–1943), described Cantor’s new mathematics as “the most astonishing product of mathematical thought” [34, p. 359], and claimed that “no one shall ever expel us from the paradise which Cantor has created for us” [34, p. 353].

In this project we will learn Cantor’s treatment of infinite sets. We will discuss the cardinality of a set, the notion of equivalence of two sets, and study how to compare infinite sets with each other. We will introduce countable sets and show that many sets are countable, including the set of integers and the set of rational numbers. We will also discuss Cantor’s diagonalization method which allows us to show that not every infinite set is countable. In particular, we will show that the set of real numbers is not countable. We will also examine the cardinal number \aleph_0 , the first in the hierarchy of transfinite cardinal numbers, and obtain a method that allows us to create infinitely many transfinite cardinal numbers.

We will learn much of this by studying and working with the historical source [6], which is an English translation of two papers by Cantor [4, 5] that appeared in 1895 and 1897. More on Georg Cantor can be found in [15, 34, 48] and in the literature cited therein.

We begin by reading Cantor’s definition of the cardinal number of a given set. Note that in his writings Cantor uses “aggregate” instead of the more familiar “set.”

Every aggregate M has a definite “power,” which we also call its “cardinal number.”

We will call by the name “power” or “cardinal number” of M the general concept which, by means of our active faculty of thought, arises from the aggregate M when we make abstraction of the nature of its various elements m and of the order in which they are given.

We denote the result of this double act of abstraction, the cardinal number or power of M , by

$\overline{\overline{M}}$.

¹Department of Mathematical Sciences, New Mexico State University, Las Cruces, NM 88003; gbezhani@nmsu.edu.

²With thanks to Joel Lucero-Bryan.

1. What do you think Cantor means by “cardinal number”? Why? Given a set M consisting of ten round marbles, each of a different color, what is $\overline{\overline{M}}$?

We say that two aggregates M and N are “equivalent,” in signs

$$M \sim N \quad \text{or} \quad N \sim M,$$

if it is possible to put them, by some law, in such a relation to one another that to every element of each one of them corresponds one and only one element of the other.

2. In modern terminology describe what it means for two sets to be equivalent.

Every aggregate is equivalent to itself:

$$M \sim M.$$

3. Prove the above claim of Cantor.

If two aggregates are equivalent to a third, they are equivalent to one another, that is to say:

$$\text{from } M \sim P \quad \text{and} \quad N \sim P \quad \text{follows} \quad M \sim N.$$

4. Prove the above claim of Cantor.

Of fundamental importance is the theorem that two aggregates M and N have the same cardinal number if, and only if, they are equivalent: thus,

$$\text{from } M \sim N, \text{ we get } \overline{\overline{M}} = \overline{\overline{N}},$$

and

$$\text{from } \overline{\overline{M}} = \overline{\overline{N}}, \text{ we get } M \sim N.$$

Thus the equivalence of aggregates forms the necessary and sufficient condition for the equality of their cardinal numbers.

5. Explain in your own words what Cantor means in the above.

6. Let \mathbf{P} be the set of all perfect squares

$$\{0, 1, 4, 9, 16, 25, \dots\},$$

and let \mathbf{N} denote the set of all natural numbers

$$\{0, 1, 2, 3, 4, 5, \dots\}.$$

From Cantor’s statement above, do \mathbf{P} and \mathbf{N} have the same cardinality? Justify your answer.

7. Let \mathbf{Z} denote the set of all integers. Do \mathbf{N} and \mathbf{Z} have the same cardinality? Justify your answer.

8. Let \mathbf{Q} denote the set of all rational numbers. Do \mathbf{N} and \mathbf{Q} have the same cardinality? Justify your answer. Hint: Can you establish a 1-1 correspondence between \mathbf{N} and \mathbf{Q} ?

If for two aggregates M and N with the cardinal numbers $\mathfrak{a} = \overline{\overline{M}}$ and $\mathfrak{b} = \overline{\overline{N}}$, both the conditions:

- (a) There is no part³ of M which is equivalent to N ,
- (b) There is a part N_1 of N , such that $N_1 \sim M$,

are fulfilled, it is obvious that these conditions still hold if in them M and N are replaced by two equivalent aggregates M' and N' . Thus they express a definite relation of the cardinal numbers \mathfrak{a} and \mathfrak{b} to one another.

Further, the equivalence of M and N , and thus the equality of \mathfrak{a} and \mathfrak{b} , is excluded; for if we had $M \sim N$, we would have, because $N_1 \sim M$, the equivalence $N_1 \sim N$, and then, because $M \sim N$, there would exist a part M_1 of M such that $M_1 \sim M$, and therefore we should have $M_1 \sim N$; and this contradicts the condition (a).

Thirdly, the relation of \mathfrak{a} to \mathfrak{b} is such that it makes impossible the same relation of \mathfrak{b} to \mathfrak{a} ; for if in (a) and (b) the parts played by M and N are interchanged, two conditions arise which are contradictory to the former ones.

We express the relation of \mathfrak{a} to \mathfrak{b} characterized by (a) and (b) by saying: \mathfrak{a} is "less" than \mathfrak{b} or \mathfrak{b} is "greater" than \mathfrak{a} ; in signs

$$\mathfrak{a} < \mathfrak{b} \text{ or } \mathfrak{b} > \mathfrak{a}.$$

9. Describe in modern terminology when two cardinals $\mathfrak{a} = \overline{\overline{M}}$ and $\mathfrak{b} = \overline{\overline{N}}$ are in the relation $\mathfrak{a} < \mathfrak{b}$.

We can easily prove that,

$$\text{if } \mathfrak{a} < \mathfrak{b} \text{ and } \mathfrak{b} < \mathfrak{c}, \text{ then we always have } \mathfrak{a} < \mathfrak{c}.$$

10. Prove the above claim of Cantor.

Aggregates with finite cardinal numbers are called "finite aggregates," all others we will call "transfinite aggregates" and their cardinal numbers "transfinite cardinal numbers."

The first example of a transfinite aggregate is given by the totality of finite cardinal numbers ν ; we call its cardinal number "Aleph-zero," and denote it by \aleph_0 ;

11. In the modern terminology, a set whose cardinal number is \aleph_0 is called "countable." What symbol is used today to denote the "totality of finite cardinal numbers ν "?

The number \aleph_0 is greater than any finite number μ :

$$\aleph_0 > \mu.$$

12. Prove the above claim of Cantor.

On the other hand, \aleph_0 is the least transfinite cardinal number. If \mathfrak{a} is any transfinite cardinal number different from \aleph_0 , then

$$\aleph_0 < \mathfrak{a}.$$

³The modern terminology is "subset".

13. Prove the above claim of Cantor. Hint: Let $\mathfrak{a} = \overline{\overline{A}}$. Can you define a 1-1 map from \mathbf{N} into A ? What can you deduce from this?

14. Let $[0, 1]$ denote the set of all real numbers between 0 and 1. Show that $\aleph_0 < \overline{\overline{[0, 1]}}$. We outline what is now known as Cantor's diagonalization method as one way to prove this. Represent real numbers in $[0, 1]$ as infinite decimals (which do not end in infinitely repeating 9's). Assume that $\mathbf{N} \sim [0, 1]$. Then to each infinite decimal one can assign a unique natural number, so the infinite decimals can be enumerated as follows:

$$\begin{array}{c} .a_{11}a_{12} \dots a_{1n} \dots \\ .a_{21}a_{22} \dots a_{2n} \dots \\ \vdots \\ .a_{n1}a_{n2} \dots a_{nn} \dots \\ \vdots \end{array}$$

Can you construct an infinite decimal $.b_1b_2 \dots b_n \dots$ such that $a_{nn} \neq b_n$ for each positive n ? What can you conclude from this?

15. Let \mathbf{R} denote the set of all real numbers. Is $\overline{\overline{\mathbf{R}}}$ strictly greater than \aleph_0 ? Justify your answer.

16. For a set M , let $\mathcal{P}(M)$ denote the set of all subsets of M ; that is $\mathcal{P}(M) = \{N : N \subseteq M\}$. Prove the following claim of Cantor:

$$\overline{\overline{\mathcal{P}(M)}} > \overline{\overline{M}}.$$

Hint: Employ a generalized version of Cantor's diagonalization method. Assume that $M \sim \mathcal{P}(M)$. Then there is a 1-1 and onto function $f : M \rightarrow \mathcal{P}(M)$. Consider the set $N = \{m \in M : m \notin f(m)\}$. Can you deduce that $N \subseteq M$ is not in the range of f ? Does this imply a contradiction?

17. Using the previous exercise, give an infinite increasing sequence of transfinite cardinal numbers.

3 A Study of Logic and Programming via Turing Machines

Jerry M. Lodder⁴

3.1 An Introduction to Turing Machines

During the International Congress of Mathematicians in Paris in 1900 David Hilbert (1862–1943), one of the leading mathematicians of the last century, proposed a list of problems for following generations to ponder [29, p. 290–329] [30]. On the list was whether the axioms of arithmetic are consistent, a question which would have profound consequences for the foundations of mathematics. Continuing in this direction, in 1928 Hilbert proposed the decision problem (das Entscheidungsproblem) [31, 32, 33], which asked whether there was a standard procedure that can be applied to decide whether a given mathematical statement is true. Both Alonzo Church (1903–1995) [9, 10] and Alan Turing (1912–1954) [66] published papers in 1936 demonstrating that the decision problem has no solution, although it is the algorithmic character of Turing’s paper “On Computable Numbers, with an Application to the Entscheidungsproblem” [66] that forms the basis for the modern programmable computer. Today his construction is known as a *Turing machine*.

Let’s first study a few excerpts from Turing’s original paper [66, p. 231–234], and then design a few machines to perform certain tasks.

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO
THE ENTSCHIEDUNGSPROBLEM

By A. M. Turing

1. *Computing Machines.*

We have said that the computable numbers are those whose decimals are calculable by finite means. This requires more explicit definition. No real attempt will be made to justify the definitions given until we reach §9. For present I shall only say that the justification lies in the fact that the human memory is necessarily limited.

We may compare a man in the process of computing a real number to a machine which is only capable of a finite number of conditions q_1, q_2, \dots, q_R , which will be called the “ m -configurations”. The machine is supplied with a “tape” (the analogue of paper) running through it, and divided into sections (called “squares”) each capable of bearing a “symbol”. At any moment there is just one square, say the r -th, bearing the symbol $\mathcal{S}(r)$ which is “in the machine”. We may call this square the “scanned square”. The symbol on the scanned square may be called the “scanned symbol”. The “scanned symbol” is the only one of which the machine is, so to speak, “directly aware”. However, by altering its m -configuration the machine can effectively remember some of the symbols it has “seen” (scanned) previously. The possible behaviour of the machine at any moment is determined by the m -configuration q_n and the scanned symbol $\mathcal{S}(r)$. This pair $q_n, \mathcal{S}(r)$ will be called the “configuration”; thus the configuration determines the possible behaviour of the machine. In some of the configurations in which the scanned square is blank (i.e. bears no symbol) the machine writes down a new symbol on the scanned square; in other configurations it erases the scanned symbol. The machine may also change the square which is being scanned, but only by shifting it one place to right or left. In addition to any of these operations the m -configuration may be changed. Some of the symbols written down will form the sequence

⁴Mathematical Sciences; Dept. 3MB, Box 30001; New Mexico State University; Las Cruces, NM 88003; jlodder@nmsu.edu.

of figures which is the decimal of the real number which is being computed. The others are just rough notes to “assist the memory”. It will only be these rough notes which will be liable to erasure.

It is my contention that these operations include all those which are used in the computation of a number. The defense of this contention will be easier when the theory of the machines is familiar to the reader. In the next section I therefore proceed with the development of the theory and assume that it is understood what is meant by “machine”, “tape”, “scanned”, etc.

2. Definitions.

Automatic machines.

If at each stage the motion of a machine (in the sense of §1) is *completely* determined by the configuration, we shall call the machine an “automatic machine” (or *a*-machine).

For some purposes we might use machines (choice machines or *c*-machines) whose motion is only partially determined by the configuration (hence the use of the word “possible” in §1). When such a machine reaches one of these ambiguous configurations, it cannot go on until some arbitrary choice has been made by an external operator. This would be the case if we were using machines to deal with axiomatic systems. In this paper I deal only with automatic machines, and will therefore often omit the prefix *a*-.

Computing machines.

If an *a*-machine prints two kinds of symbols, of which the first kind (called figures) consists entirely of 0 and 1 (the others being called symbols of the second kind), then the machine will be called a computing machine. If the machine is supplied with a blank tape and set in motion, starting from the correct initial *m*-configuration the subsequence of the symbols printed by it which are of the first kind will be called the *sequence computed by the machine*. The real number whose expression as a binary decimal is obtained by prefacing this sequence by a decimal point is called the *number printed by the machine*.

At any stage of the motion of the machine, the number of the scanned square, the complete sequence of all symbols on the tape, and the *m*-configuration will be said to describe the *complete configuration* at that stage. The changes of the machine and tape between successive complete configurations will be called the *moves* of the machine. ...

3. Examples of computing machines.

I. A machine can be constructed to compute the sequence 010101 The machine is to have the four *m*-configurations “*b*”, “*c*”, “*f*”, “*e*” and is capable of printing “0” and “1”. The behaviour of the machine is described in the following table [Example 1] in which “*R*” means “the machine moves so that it scans the square immediately on the right of the one it was scanning previously”. Similarly for “*L*”. “*E*” means “the scanned symbol is erased and “*P*” stands for “prints”. This table (and all succeeding tables of the same kind) is to be understood to mean that for a configuration described in the first two columns the operations in the third column are carried out successively, and the machine then goes over into the *m*-configuration described in the last column. When the second column is blank, it is understood that the behaviour of the third and fourth columns applies for any symbol and for no symbol. The machine starts in the *m*-configuration *b* with a blank tape.

| Configuration | | Behaviour | |
|---------------|--------|-----------|-----------------|
| m-config. | symbol | operation | final m-config. |
| b | none | P0, R | c |
| c | none | R | e |
| e | none | P1, R | f |
| f | none | R | b |

If (contrary to the description §1) we allow the letters L, R to appear more than once in the operations column we can simplify the table considerably.

| Configuration | | Behaviour | |
|---------------|--------|-----------|-----------------|
| m-config. | symbol | operation | final m-config. |
| b | none | P0 | b |
| b | 0 | R, R, P1 | b |
| b | 1 | R, R, P0 | b |

1.1. Describe the workings of a Turing machine (referred to as a “computing machine” in the original paper).

1.2. What is the precise output of the machine in Example 1? Certain squares may be left blank. Be sure to justify your answer.

1.3. Design a Turing machine which generates the following output. Be sure to justify your answer.

010010100101001 ...

1.4. Describe the behavior of the following machine, which begins with a blank tape, with the machine in configuration α .

| Configuration | | Behavior | |
|---------------|--------|-----------|-----------------|
| m-config. | symbol | operation | final m-config. |
| α | none | R P1 | β |
| α | 1 | R P0 | β |
| α | 0 | HALT | (none) |
| β | 1 | R P1 | α |
| β | 0 | R P0 | α |

1.5. Given finite, non-empty, sets A and B , design a Turing machine which tests whether $A \subseteq B$. Suppose that the first character on the tape is a 0, simply to indicate the beginning of the tape. To the right of 0 follow the (distinct, non-blank) elements of A , listed in consecutive positions, followed by the symbol $\&$. To the right of $\&$ follow the (distinct, non-blank) elements of B , listed in consecutive positions, followed by the symbol Z to indicate the end of the tape:

| | | | | | | | | | | | |
|---|--|--|-----|--|---|--|--|-----|--|--|---|
| 0 | | | ... | | & | | | ... | | | Z |
|---|--|--|-----|--|---|--|--|-----|--|--|---|

The symbols 0, &, Z are neither elements of A nor B . The machine starts reading the tape in the right most position, at Z. If $A \subseteq B$, have the machine erase all the elements of A and return a tape with blanks for every square which originally contained an element of A . You may use the following operations for the behavior of the machine:

- R: Move one position to the right.
- L: Move one position to the left.
- S: Store the scanned character in memory. Only one character can be stored at a time.
- C: Compare the currently scanned character with the character in memory. The only operation of C is to change the final configuration depending on whether the scanned square matches what is in memory.
- E: Erase the currently scanned square.
- P(): Print whatever is in parentheses in the current square.

You may use multiple operations for the machine in response to a given configuration. Also, for a configuration q_n , you may use the word “other” to denote all symbols $\mathcal{S}(r)$ not specifically identified for the given q_n . Be sure that your machine halts.

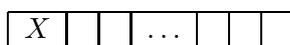
3.2 Turing Machines, Induction and Recursion

The logic behind the modern programmable computer owes much to Turing’s “computing machines,” discussed in the first section of the project. Since the state of the machine, or m -configuration as called by Turing, can be altered according to the symbol being scanned, the operation of the machine can be changed depending on what symbols have been written on the tape, and affords the machine a degree of programmability. The program consists of the list of configurations of the machine and its behavior for each configuration. Turing’s description of his machine, however, did not include memory in its modern usage for computers, and symbols read on the tape could not be stored in any separate device. Using a brilliant design feature for the tape, Turing achieves a limited type of memory for the machine, which allows it to compute many arithmetic operations. The numbers needed for a calculation are printed on every other square of the tape, while the squares between these are used as “rough notes to ‘assist the memory.’ It will only be these rough notes which will be liable to erasure” [66, p. 232].

Turing continues [66, p. 235]:

The convention of writing the figures only on alternate squares is very useful: I shall always make use of it. I shall call the one sequence of alternate squares F -squares, and the other sequence E -squares. The symbols on E -squares will be liable to erasure. The symbols on F -squares form a continuous sequence. . . . There is no need to have more than one E -square between each pair of F -squares: an apparent need of more E -squares can be satisfied by having a sufficiently rich variety of symbols capable of being printed on E -squares.

Let’s examine the Englishman’s use of these two types of squares. Determine the output of the following Turing machine, which begins in configuration a with the tape



and the scanner at the far left, reading the symbol X .

| Configuration | | Behavior | |
|---------------|--------|------------------------------|-----------------|
| m-config. | symbol | operation | final m-config. |
| a | X | R | a |
| a | 1 | R, R | a |
| a | blank | P(1), R, R, P(1), R, R, P(0) | b |
| b | X | E, R | c |
| b | other | L | b |
| c | 0 | R, P(X), R | a |
| c | 1 | R, P(X), R | d |
| d | 0 | R, R | e |
| d | other | R, R | d |
| e | blank | P(1) | b |
| e | other | R, R | e |

- 2.1. What is the precise output of the machine as it just finishes configuration a and enters configuration b for the first time? Justify your answer.
- 2.2. What is the precise output of the machine as it just finishes configuration a and enters configuration b for the second time? Justify your answer.
- 2.3. What is the precise output of the machine as it just finishes configuration a and enters configuration b for the third time? Justify your answer.
- 2.4. Guess what the output of the machine is as it just finishes configuration a and enters configuration b for the n -th time. Use induction to prove that your guess is correct. Be sure to write carefully the details of this proof by induction.
- 2.5. Design a Turing machine, which when given two arbitrary natural numbers, n and m , will compute the product $n \cdot m$. Suppose that the machine begins with the tape

| | | | | | | | | | | | | |
|-----|---|---|-----|---|---|-----|---|---|-----|---|---|-----|
| A | 1 | 1 | ... | 1 | 1 | B | 1 | 1 | ... | 1 | 1 | C |
|-----|---|---|-----|---|---|-----|---|---|-----|---|---|-----|

where the number of ones between A and B is n , the number of ones between B and C is m , and the machine begins scanning the tape at the far left, reading the symbol A . The output of the machine should be:

| | | | | | | | | | | | | | |
|-----|---|-----|---|-----|---|-----|---|-----|---|---|-----|---|-----|
| A | 1 | ... | 1 | B | 1 | ... | 1 | C | 1 | 1 | ... | 1 | D |
|-----|---|-----|---|-----|---|-----|---|-----|---|---|-----|---|-----|

where the number of ones between C and D is $n \cdot m$. Use induction to verify that the machine produces the correct output.

Letting T denote the Turing machine which multiplies n and m together, so that the value of $T(n, m)$ is $n \cdot m$, design T so that for $n \in \mathbf{N}$,

$$T(n, 1) = n$$

and for $m \in \mathbf{N}$, $m \geq 2$, we have

$$T(n, m) = T(n, m - 1) + n.$$

Such an equation provides an example of a recursively defined function, an important topic in computer science. In our case, the algorithm for multiplication, T , is defined in terms of addition, a more elementary operation.

3.3 The Universal Computing Machine

The decision problem of Hilbert asked whether there is a standard procedure, an algorithm in modern terminology, which can be invoked to decide whether an arbitrary statement (within some system of logic) is valid. In answering this question, Turing introduced several fundamental concepts, certainly of importance to logic, but also pivotal to the development of the modern programmable computer. The first of these is a “computing machine,” called a “Turing machine” today, which is the forerunner of a modern computer program. The next concept is the “universal computing machine,” which is in fact a particular type of Turing machine that accepts the instructions of some other machine M in standard form, and the outputs the same sequence as M . Turing writes [66, p. 241–242]:

It is possible to invent a single machine which can be used to compute any computable sequence. If this machine U is supplied with a tape on the beginning of which is written the $S.D$ [standard description] of some computing machine M , then U will compute the same sequence as M .

The reader should review §3 *Examples of computing machines* reprinted above under “An Introduction to Turing Machines,” and have the table from Example 1 at hand before reading the following excerpts [66, p. 239–241]:

5. Enumeration of computable sequences.

A computable sequence γ is determined by a description of a machine which computes γ . Thus the sequence 001011011101111... is determined by the table on p. 234, and, in fact, any computable sequence is capable of being described in terms of such a table.

It will be useful to put these tables into a kind of standard form. In the first place let us suppose that the table is given in the same form as the first table, for example, 1 on p. 233. That is to say, that the entry in the operations column is always one of the form E : E, R : E, L : $P\alpha$: $P\alpha, R$: $P\alpha, L$: R : L : or no entry at all. The table can always be put into this form by introducing more m -configurations. Now let us give numbers to the m -configurations, calling them q_1, \dots, q_R as in §1. The initial m -configuration is always to be called q_1 . We also give numbers to the symbols S_1, \dots, S_m and, in particular, blank = S_0 , $0 = S_1$, $1 = S_2$. The lines of the table are now of form

| m -config. | Symbol | Operations | Final m -config. | |
|--------------|--------|------------|--------------------|---------|
| q_i | S_j | PS_k, L | q_m | (N_1) |
| q_i | S_j | PS_k, R | q_m | (N_2) |
| q_i | S_j | PS_k | q_m | (N_3) |

Lines such as

| | | | |
|-------|-------|--------|-------|
| q_i | S_j | E, R | q_m |
|-------|-------|--------|-------|

are to be written as

| | | | |
|-------|-------|-----------|-------|
| q_i | S_j | PS_0, R | q_m |
|-------|-------|-----------|-------|

and lines such as

| | | | |
|-------|-------|-----|-------|
| q_i | S_j | R | q_m |
|-------|-------|-----|-------|

to be written as

$$q_i \quad S_j \quad PS_j, R \quad q_m$$

In this way we reduce each line of the table to a line of one of the forms (N_1) , (N_2) , (N_3) .

From each line of form (N_1) let us form an expression $q_i S_j S_k L q_m$; from each line of form (N_2) we form an expression $q_i S_j S_k R q_m$; and from each line of form (N_3) we form an expression $q_i S_j S_k N q_m$.

Let us write down all expressions so formed from the table for the machine and separate them by semi-colons. In this way we obtain a complete description of the machine. In this description we shall replace q_i by the letter "D" followed by the letter "A" repeated i times, and S_j by "D" followed by "C" repeated j times. This new description of the machine may be called the *standard description* (S.D). It is made up entirely from the letters "A", "C", "D", "L", "R", "N", and from ";". ...

Let us find a description number for the machine I of §3. When we rename the m -configurations its table becomes:

| m-config. | Symbol | Operations | Final m-config. |
|-----------|--------|------------|-----------------|
| q_1 | S_0 | PS_1, R | q_2 |
| q_2 | S_0 | PS_0, R | q_3 |
| q_3 | S_0 | PS_2, R | q_4 |
| q_4 | S_0 | PS_0, R | q_1 |

Other tables could be obtained by adding irrelevant lines such as

$$q_1 \quad S_1 \quad PS_1, R \quad q_2$$

Our first standard form would be

$$q_1 S_0 S_1 R q_2; q_2 S_0 S_0 R q_3; q_3 S_0 S_2 R q_4; q_4 S_0 S_0 R q_1; .$$

The standard description is

$$DADDCRDAA; DAADDRDAAA; DAAADDCCRDAAAA; DAAAADDRDA;$$

Continuing from [66, p. 243], we read:

Each instruction consists of five consecutive parts

- (i) "D" followed by a sequence of letters "A". This describes the relevant m -configuration.
- (ii) "D" followed by a sequence of letters "C". This describes the scanned symbol.
- (iii) "D" followed by another sequence of letters "C". This describes the symbol into which the scanned symbol is to be changed.
- (iv) "L", "R", or "N", describing whether the machine is to move to left, right, or not at all.
- (v) "D" followed by a sequence of letters "A". This describes the final m -configuration.

3.1. What is the output of the following machine, T , if T begins in configuration a with a blank tape, scanning the blank at the far left?

| Configuration | | Behavior | |
|---------------|--------|-----------|-----------------|
| m-config. | symbol | operation | final m-config. |
| a | 1 | R | c |
| a | blank | P(1), R | b |
| b | 0 | R | a |
| b | blank | P(1) | a |
| c | blank | P(0) | b |

3.2. Rewrite the output of machine T using the “standard description” for the output symbols, i.e., $S_0 = \text{blank}$, $S_1 = 0$, $S_2 = 1$, and then replace each S_j with D followed by C repeated j times.

3.3. What is the standard description ($S.D$) of machine T ? Be sure that every instruction, including the last one, is followed by a semi-colon. Make sure that you have the correct answer to this before continuing.

3.4. Suppose that the number of configurations for a given machine M is limited to nine, while the number of symbols which M can recognize or write is limited to four. The machine M begins in its first listed configuration (a) with a blank tape, reading the blank at the far left. Now suppose that the standard description of M is written on every other square (in fact the F -squares) of a second tape, with each instruction followed by a semi-colon (on an F -square as well), with the last semi-colon followed by the symbol “:.” on an F -square. Initially all other squares on this second tape are blank. If the standard description for machine I of §3 in Turing’s paper is entered on tape in this way, what is the output of the following machine U , which begins in configuration one reading the tape at the far left? Note that 20R is shorthand for move 20 squares to the right, and similarly for 10R. How does this compare with the actual output of machine I, §3?

| Configuration | | Behavior | |
|---------------|--------|----------------------------------|-----------------|
| m-config. | symbol | operation | final m-config. |
| 1 | D | R | 1 |
| 1 | A | R | 2 |
| 1 | :: | none | none |
| 1 | other | R | 1 |
| 2 | blank | R | 2 |
| 2 | A | R | 4 |
| 2 | D | R, R | 3 |
| 3 | D | R, P(X) | 5 |
| 3 | C | R | 4 |
| 4 | ; | R | 1 |
| 4 | :: | none | none |
| 4 | other | R | 4 |
| 5 | :: | 20R, P(Y), R, R, P(D), 10R, P(Z) | 6 |
| 5 | other | R | 5 |
| 6 | X | E, R | 7 |
| 6 | other | L | 6 |
| 7 | C | R, P(X) | 8 |
| 7 | R | R, P(X) | 10 |
| 7 | L | none | none |
| 7 | N | R, P(X) | 11 |
| 8 | Y | 4R | 9 |
| 8 | other | R | 8 |
| 9 | blank | P(C) | 6 |
| 9 | C | R, R | 9 |
| 10 | Z | E, P(T) | 12 |
| 10 | other | R | 10 |
| 11 | Y | R, P(T) | 12 |
| 11 | other | R | 11 |
| 12 | X | E, 4R, P(X) | 13 |
| 12 | other | L | 12 |
| 13 | :: | R, R | 14 |
| 13 | other | R | 13 |
| 14 | blank | P(A) | 15 |
| 14 | Y | none | none |
| 14 | other | R, R | 14 |
| 15 | X | E, R | 16 |
| 15 | other | L | 15 |
| 16 | ; | none | 17 |
| 16 | A | R, P(X) | 13 |

3.5. If the standard description of machine T from question 3.1 is written on tape as described in 3.4, what is the output of U when applied to this tape? How does this compare to the actual output of T ?

3.6. Describe in words the operation of configuration one from machine U . Describe separately the operations of configurations two, three, and four.

3.7. Note that only the first 16 configurations of U are listed. From this point, if U was originally supplied with the standard description of a machine M (as specified in 3.4), then U should output the same sequence as M , except coded according to the standard description of the output symbols. Moreover, U keeps track of the current configuration of M in the first 18 squares immediately following “:”. The position of the scanner for M is recorded via the symbol “T” on the tape which U processes. Beginning with configuration 17, outline the remaining operations of U . You may use phrases such as “match m -configuration,” “match scanned symbol,” or “move scanner for M ,” etc. Be sure to include a written explanation of these and any other operations you decide to use in your outline. For this part, you do not need to design an actual Turing machine to perform these tasks. Does recursion occur in your outline? How?

3.8. Beginning with configuration 17, find the actual machine instructions of U so that U finds a match between an arbitrary configuration stored on the 18 squares to the right of “:” and a configuration at the beginning of a coded instruction to the left of “:”. Suppose that the standard description entered on tape is that of a machine M for which there is always a well-defined configuration to follow every move of M . Use only the operations “R,” “L,” “E,” “P(),” “none,” and be sure to explain the new steps of U . What is the present-day terminology used to describe U ?

Extra Credit: Write a computer program for a universal computing machine (in the language of your choice). Demonstrate with several examples that your universal machine functions properly.

3.4 The Decision Problem, *Das Entscheidungsproblem*

Turing’s paper “On Computable Numbers with an Application to the Entscheidungsproblem” proved most influential not only for mathematical logic, but also for the development of the programmable computer, and together with work of Alonzo Church (1903–1995) [9, 10] and others [26], inaugurated a new field of study known today as computability. Recall that Turing’s original motivation for writing the paper was to answer the decision problem of David Hilbert, posed in 1928 along with a list of other problems [31, 32] dealing with the consistency, completeness and independence of the axioms of a logical system in general. The solutions to these problems, in particular Kurt Gödel’s (1906–1978) demonstration of the incompleteness of arithmetic with the existence of statements that are not provable (as true or false) [23], had profound consequences for mathematics, and brought mathematical logic to the fore as a separate field of study [26]. In this section, however, we deal primarily with the problem of deciding whether a given statement is valid within a logical system, the decision problem, which Hilbert expressed as [33, p. 112–113]:

... [T]here emerges the fundamental importance of determining whether or not a given formula of the predicate calculus is universally valid. ... A formula ... is called *satisfiable* if the sentential variables can be replaced with the values truth and falsehood ... in such a way that the formula [becomes] a true sentence. ... It is customary to refer to the equivalent problems of *universal validity* and *satisfiability* by the common name of the *decision problem*.

Following Gödel’s results, the decision problem remained, although it must be reinterpreted as meaning whether there is a procedure by which a given proposition can be determined to be either “provable” or “unprovable”. In the text *Introduction to Mathematical Logic* [11, p. 99], Church formulated this problem as:

The decision problem of a logistic system is the problem to find an effective procedure or algorithm, a *decision procedure*, by which for an arbitrary well-formed formula of the system, it is possible to determine whether or not it is a theorem

To be sure, Church proves that the decision problem has no solution [9, 10], although it is the algorithmic character of Turing’s solution that is pivotal to the logical underpinnings of the programmable computer. Moreover, the simplicity of a Turing machine provides a degree of accessibility to logic and computability ideal for readers new to this material.

The concept of a universal computing machine, studied above, has evolved into what now is known as a compiler or interpreter in computer science, and is indispensable for the processing of any programming language. The question then arises, does the universal computing machine provide a solution to the decision problem? The universal machine is the standard procedure for answering all questions that can in turn be phrased in terms of a computer program.

First, study the following excerpts from Turing’s paper [66, p. 232–233]:

Automatic machines.

If at each stage the motion of a machine is *completely* determined by the configuration, we shall call the machine an “automatic machine” (or *a-machine*). . . .

Computing machines.

If an *a-machine* prints two kinds of symbols, of which the first kind (called figures) consists entirely of 0 and 1 (the others being called symbols of the second kind), then the machine will be called a computing machine. If the machine is supplied with a blank tape and set in motion, starting from the correct initial *m*-configuration, the subsequence of symbols printed by it which are of the first kind will be called the *sequence computed by the machine*. . . .

Circular and circle-free machines.

If a computing machine never writes down more than a finite number of symbols of the first kind, it will be called *circular*. Otherwise it is said to be *circle-free*. . . .

A machine will be circular if it reaches a configuration from which there is no possible move, or if it goes on moving and possibly printing symbols of the second kind, but cannot print any more symbols of the first kind.

Computable sequences and numbers.

A sequence is said to be computable if it can be computed by a circle-free machine. A number is computable if it differs by an integer from the number computed by a circle-free machine. . . .

4.1. Consider the following machine, T_1 , which begins in *m*-configuration *a* with a blank tape, reading the blank at the far left. Is T_1 circle-free? Justify your answer.

| Configuration | | Behavior | | |
|---------------|----------|-----------|-------------------|----------|
| m-config. | symbol | operation | final m-config. | |
| T_1 : | <i>a</i> | blank | <i>R, P(1)</i> | <i>b</i> |
| | <i>a</i> | 0 | <i>R</i> | <i>b</i> |
| | <i>b</i> | 1 | <i>R, R, P(0)</i> | <i>a</i> |
| | <i>b</i> | blank | (none) | <i>a</i> |

4.2. Consider the following machine, T_2 , which begins in *m*-configuration *a* with a blank tape, reading the blank at the far left. Is T_2 circle-free? Justify your answer.

| Configuration | | Behavior | |
|---------------|--------|--------------|-----------------|
| m-config. | symbol | operation | final m-config. |
| a | blank | $R, P(1)$ | b |
| a | 0 | R | b |
| b | 1 | $R, R, P(0)$ | a |
| b | 0 | R | a |

4.3. Describe in your own words the key feature which distinguishes a circle-free machine from a circular machine.

4.4. Is the sequence 101001000100001 ... computable? If so, find a circle-free machine (with a finite number of m -configurations) that computes this sequence on every other square (the F -squares) of a tape which is originally blank. If not, prove that there is no circle-free machine that computes the above sequence.

Turing's insight into the decision problem begins by listing all computable sequences in some order:

$$\phi_1, \phi_2, \phi_3, \dots, \phi_n, \dots,$$

where ϕ_n is the n -th computable sequence. Moreover, let $\phi_n(k)$ denote the k -th figure (0 or 1) of ϕ_n . For example, if

$$\phi_2 = 101010 \dots,$$

then $\phi_2(1) = 1$, $\phi_2(2) = 0$, $\phi_2(3) = 1$, etc. Turing then considers the sequence β' defined by $\beta'(n) = \phi_n(n)$. If the decision problem has a solution, then [66, p. 247]:

We can invent a machine D which, when supplied with the $S.D$ [standard description] of any computing machine M will test this $S.D$ and if M is circular will mark the $S.D$ with the symbol "u" [unsatisfactory] and if it is circle-free will mark it with "s" [satisfactory]. By combining the machines D and U [the universal computing machine] we could construct a machine H to compute the sequence β' .

4.5. Is the number of computable sequences finite or infinite? If finite, list the computable sequences. If infinite, find a one-to-one correspondence between the natural numbers, \mathbf{N} , and a subset of the computable sequences. Use the result of this question to carefully explain why H must be circle-free.

4.6. Since H is circle-free, the sequence computed by H must be listed among the ϕ_n 's. Suppose this occurs for $n = N_0$. In a written paragraph, explain how $\beta'(N_0)$ should be computed. Is it possible to construct a machine H that computes β' ? If so, find the configuration table for H . If not, what part of H , i.e., D or U , cannot be constructed? Justify your answer.

4.7. Does the universal computing machine solve the decision problem? Explain.

4.8. By what name is the decision problem known today in computer science? Support your answer with excerpts from outside sources.

4 Binary Arithmetic: From Leibniz to von Neumann

Jerry M. Lodder⁵

The Era of Leibniz

Gottfried Wilhelm Leibniz (1646–1716) is often described as the last universalist, having contributed to virtually all fields of scholarly interest of his time, including law, history, theology, politics, engineering, geology, physics, and perhaps most importantly, philosophy, mathematics and logic [1, 34, 37]. The young Leibniz began to teach himself Latin at the age of 8, and Greek a few years later, in order to read classics not written in his native language, German. Later in life, he wrote:

Before I reached the school-class in which logic was taught, I was deep into the historians and poets, for I began to read the historians almost as soon as I was able to read at all, and I found great pleasure and ease in verse. But as soon as I began to learn logic, I was greatly excited by the division and order in it. I immediately noticed, to the extent that a boy of 13 could, that there must be a great deal in it [19, p. 516].

His study of logic and intellectual quest for order continued throughout his life and became a basic principle to his method of inquiry. At the age of 20 he published *Dissertatio de arte combinatoria* (Dissertation on the Art of Combinatorics) in which he sought a *characteristica generalis* (general characteristic) or a *lingua generalis* (general language) that would serve as a universal symbolic language and reduce all debate to calculation. Leibniz maintained:

If controversies were to arise, there would be no more need of disputation between two philosophers than between two accountants. For it would suffice to take their pencils in their hands, to sit down to their slates, and to say to each other: . . . Let us calculate [58, p. 170].

The Leipzig-born scholar traveled extensively with diplomatic visits to Paris and London, and extended trips to Austria and Italy to research the history of the House of Brunswick. The years 1672–1676 were spent in Paris in an attempt to persuade King Louis XIV (1638–1715) not to invade Germany, but Egypt instead, although Leibniz was never granted an audience with the French king. During this time in Paris, however, the young German became acquainted with several of the leading philosophers of the day, acquired access to unpublished manuscripts of Blaise Pascal (1623–1662) and René Descartes (1596–1650), and met the renowned Christiaan Huygens (1629–1695), from whom he learned much about contemporary mathematics. During these years he laid the foundation of his calculus and the core of what would become his philosophical legacy.

Leibniz’s invention of the differential and integral calculus is, in part, a product of his search for a universal language. Questions in the calculus can be reduced to the rules of calculation which the symbols for derivative, d , and integral, \int , satisfy. Sadly a priority dispute with Isaac Newton (1642–1727) over the invention of calculus cast a pall over Leibniz’s later years. Moreover, he became a subject of ridicule with his philosophy that this is the best of all possible worlds bitterly satirized in Voltaire’s (1694–1778) play *Candide*.

Let’s turn to the universal genius’s 1703 publication “Explication de l’arithmétique binaire, qui se sert des seuls caractères 0 et 1, avec des remarques sur son utilité, et sur ce qu’elle donne le sens des anciennes figures Chinoises de Fohy” [20, p. 223–227] (An Explanation of Binary Arithmetic Using only the Characters 0 and 1, with Remarks about its Utility and the Meaning it Gives to the Ancient Chinese Figures of Fuxi), which originally appeared in the journal *Memoires de*

⁵Mathematical Sciences; Dept. 3MB, Box 30001; New Mexico State University; Las Cruces, NM 88003; jlodder@nmsu.edu.

l'Académie Royale des Sciences [49]. Here again, with the reduction of arithmetic to expressions involving only zeroes and ones, we see a possible candidate for Leibniz's *characteristica generalis*. Of binary numeration, he writes "it permits new discoveries [in] . . . arithmetic . . . in geometry, because when the numbers are reduced to the simplest principles, like 0 and 1, a wonderful order appears everywhere." Concerning the binary calculations themselves " . . . these operations are so easy that we shall never have to guess or apply trial and error, as we must do in ordinary division. Nor do we need to learn anything by rote." Certainly Leibniz was not the first to experiment with binary numbers or the general concept of a number base [22]. However, with base 2 numeration, Leibniz witnessed the confluence of several intellectual ideas of his world view, not just the *characteristica generalis*, but also theological and mystical ideas of order, harmony and creation [65]. Additionally his 1703 paper [49] contains a striking application of binary numeration to the ancient Chinese text of divination, the *Yijing* (*I-Ching* or *Book of Changes*).

Early in life Leibniz developed an interest in China, corresponded with Catholic missionaries there, and wrote on questions of theology concerning the Chinese. Surprisingly he believed that he had found an historical precedent for his binary arithmetic in the ancient Chinese lineations or 64 hexagrams of the *Yijing*. This, he thought, might be the origin of a universal symbolic language. A hexagram consists of six lines atop one another, each of which is either solid or broken, forming a total of 64 possibilities, while a grouping of only three such lines is called a trigram. Leibniz lists the eight possible trigrams in his exposition on binary arithmetic, juxtaposed with their binary equivalents.

He had been in possession of his ideas concerning binary arithmetic well before his 1703 publication. In 1679 Leibniz outlined plans for a binary digital calculating machine, and in 1697 he sent a congratulatory birthday letter to his patron Duke Rudolph August of Brunswick, in which he discusses binary numeration and the related creation theme with 0 denoting nothing and 1 denoting God [65]. Furthermore, Leibniz sent the French Jesuit Joachim Bouvet (1656–1730) an account of his binary system while Bouvet was working in China. The Jesuits are an educational order of Catholic priests, who, while in China, sought the conversion of the Chinese to Christianity, hopefully by the identification of an ancient theology common to both religions. Bouvet began a study of the *Yijing*, viewing this text as the possible missing link between the two religions [65]. It was from this Jesuit priest that Leibniz received the hexagrams attributed to Fuxi, the mythical first Emperor of China and legendary inventor of Chinese writing. In actuality, the hexagrams are derived from the philosopher Shao Yong's (1011–1077) *Huangji jingshi shu* (Book of Sublime Principle Which Governs All Things Within the World). Shortly after receiving Bouvet's letter containing the hexagrams and Bouvet's identification of a relation between them and binary numeration, Leibniz submitted for publication his 1703 paper "Explanation of Binary Arithmetic" [7, p. 44].

1. Concerning the utility of the binary system, Leibniz cites an application to weighing masses. Suppose that a two-pan balance is used for weighing stones. A stone of unknown (integral) weight is placed on the left pan, while standard weights are placed only on the right pan until both sides balance. For example, if standard weights of 1, 4, 6 are used, then a stone of weight 7 on the left pan would balance the standard weights 1 and 6 on the right. Two standard weights with the same value cannot be used. Leibniz claims that all stones of integral weight between 1 and 15 inclusive can be weighed with just four standard weights. What are these four standard weights? Explain how each stone of weight between 1 and 15 inclusive can be weighed with the four standard weights. Make a table with one column for each of the four standard weights and another column for the stone of unknown weight. For each of the 15 stones, place an "X" in the columns for the standard weights used to weigh the stone.

Let's now read from an "Explanation of Binary Arithmetic," using a modified version of the Ching-Oxtoby translation [7, p. 81–86].

**An Explanation of Binary Arithmetic
Using only the Characters 0 and 1, with Remarks
about its Utility and the Meaning it Gives to the
Ancient Chinese Figures of Fuxi**

Ordinary arithmetical calculations are performed according to a progression of tens. We use ten characters, which are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, that signify zero, one and the following numbers up to nine, inclusive. After reaching ten, we begin again, writing *ten* with 10, and ten times ten or *one hundred* with 100, and ten times one hundred or *one thousand* with 1000, and ten times one thousand with 10000, and so on.

2. Write the numbers 1, 10, 100, 1000 and 10000 as powers of ten. Express your answer in complete sentences or with equations. What pattern do you notice in the exponents?

But instead of the progression by tens, I have already used for several years the simplest of all progressions, that by twos, having found that this contributes to the perfection of the science of numbers. Thus I use no characters other than 0 and 1, and then, reaching two, I begin again. This is why *two* is written here as 10, and two times two or *four* as 100, and two times four or *eight* as 1000, and two times eight or *sixteen* as 10000, and so on.

3. Write the numbers 1, 2, 4, 8 and 16 as powers of two. Express your answer in complete sentences or with equations. What pattern do you notice in the exponents? How do the exponents compare with those in question 2? How does the progression by twos compare with the standard weights in question 1?

Here is the *Table of Numbers* according to this pattern, which we can continue as far as we wish.

4. Compare the entries from 1 to 15 in Leibniz's "Table of Numbers" with the table for weighing stones that you constructed previously in question 1. Today a number written only with the characters 0 and 1 according to Leibniz's "progression of twos" is said to be written in binary notation, or base 2. Find the binary equivalents of the (base 10) numbers

34, 64, 100, 1015.

Be sure to explain your work.

At a glance we see the reason for the *famous property of the double geometric progression* in whole numbers, which states that given only one of these numbers in each degree, we can form all other whole numbers below the double of the highest degree. Since, as we would say, for example, 111 or 7 is the sum of four, two and one, and that 1101 or 13 is the sum of eight, four and one.

$$\begin{array}{r}
 1 \ 0 \ 0 \ || \ 4 \\
 \ 1 \ 0 \ || \ 2 \\
 \ 1 \ || \ 1 \\
 \hline
 1 \ 1 \ 1 \ || \ 7
 \end{array}
 \qquad
 \begin{array}{r}
 1 \ 0 \ 0 \ 0 \ || \ 8 \\
 \ 1 \ 0 \ 0 \ || \ 4 \\
 \ 1 \ || \ 1 \\
 \hline
 1 \ 1 \ 0 \ 1 \ || \ 13
 \end{array}$$

This property is useful to investigators for weighing all kinds of masses with just a few weights or could be used in monetary systems to provide a range of change with just a few coins.

Table of Numbers

| | | | | | | | |
|---|---|---|---|---|---|---|----|
| ○ | ○ | ○ | ○ | ○ | ○ | 0 | 0 |
| ○ | ○ | ○ | ○ | ○ | ○ | 1 | 1 |
| ○ | ○ | ○ | ○ | ○ | 1 | 0 | 2 |
| ○ | ○ | ○ | ○ | ○ | 1 | 1 | 3 |
| ○ | ○ | ○ | 1 | 0 | 0 | | 4 |
| ○ | ○ | ○ | 1 | 0 | 1 | | 5 |
| ○ | ○ | ○ | 1 | 1 | 0 | | 6 |
| ○ | ○ | ○ | 1 | 1 | 1 | | 7 |
| ○ | ○ | 1 | 0 | 0 | 0 | | 8 |
| ○ | ○ | 1 | 0 | 0 | 1 | | 9 |
| ○ | ○ | 1 | 0 | 1 | 0 | | 10 |
| ○ | ○ | 1 | 0 | 1 | 1 | | 11 |
| ○ | ○ | 1 | 1 | 0 | 0 | | 12 |
| ○ | ○ | 1 | 1 | 0 | 1 | | 13 |
| ○ | ○ | 1 | 1 | 1 | 0 | | 14 |
| ○ | ○ | 1 | 1 | 1 | 1 | | 15 |
| ○ | 1 | 0 | 0 | 0 | 0 | | 16 |
| ○ | 1 | 0 | 0 | 0 | 1 | | 17 |
| ○ | 1 | 0 | 0 | 1 | 0 | | 18 |
| ○ | 1 | 0 | 0 | 1 | 1 | | 19 |
| ○ | 1 | 0 | 1 | 0 | 0 | | 20 |
| ○ | 1 | 0 | 1 | 0 | 1 | | 21 |
| ○ | 1 | 0 | 1 | 1 | 0 | | 22 |
| ○ | 1 | 0 | 1 | 1 | 1 | | 23 |
| ○ | 1 | 1 | 0 | 0 | 0 | | 24 |
| ○ | 1 | 1 | 0 | 0 | 1 | | 25 |
| ○ | 1 | 1 | 0 | 1 | 0 | | 26 |
| ○ | 1 | 1 | 0 | 1 | 1 | | 27 |
| ○ | 1 | 1 | 1 | 0 | 0 | | 28 |
| ○ | 1 | 1 | 1 | 0 | 1 | | 29 |
| ○ | 1 | 1 | 1 | 1 | 0 | | 30 |
| ○ | 1 | 1 | 1 | 1 | 1 | | 31 |
| 1 | 0 | 0 | 0 | 0 | 0 | | 32 |

etc.

5. Using modern notation, Leibniz’s “double geometric progression” or “progression by twos” would be written $2^0, 2^1, 2^2, 2^3, \dots, 2^n$. Guess a simple formula for

$$2^0 + 2^1 + 2^2 + 2^3 + \dots + 2^n$$

based on Leibniz’s verbal description “that given only one of these numbers in each degree,” we should be able to achieve all whole numbers “below the double of the highest degree.” Prove that your guess is correct using only algebra (addition and multiplication). Hint: Multiply $(2^0 + 2^1 + 2^2 + 2^3 + \dots + 2^n)$ by 1, where 1 is written as $2 - 1$.

This expression of numbers, once established, facilitates all kinds of operations.

For example, addition (1)

$$\begin{array}{r|l}
 1 & 1 & 0 & & 6 \\
 1 & 1 & 1 & & 7 \\
 \hline
 1 & 1 & 0 & 1 & 13
 \end{array}
 \quad
 \begin{array}{r|l}
 & 1 & 0 & 1 & 5 \\
 & 1 & 0 & 1 & 11 \\
 \hline
 1 & 0 & 0 & 0 & 0 & 16
 \end{array}
 \quad
 \begin{array}{r|l}
 & 1 & 1 & 1 & 0 & 14 \\
 1 & 0 & 0 & 0 & 1 & 17 \\
 \hline
 1 & 1 & 1 & 1 & 1 & 31
 \end{array}$$

For subtraction

$$\begin{array}{r|l}
 1 & 1 & 0 & 1 & 13 \\
 1 & 1 & 1 & & 7 \\
 \hline
 1 & 1 & 0 & & 6
 \end{array}
 \quad
 \begin{array}{r|l}
 1 & 0 & 0 & 0 & 0 & 16 \\
 1 & 0 & 1 & 1 & & 11 \\
 \hline
 & 1 & 0 & 1 & & 5
 \end{array}
 \quad
 \begin{array}{r|l}
 1 & 1 & 1 & 1 & 1 & 31 \\
 1 & 0 & 0 & 0 & 1 & 17 \\
 \hline
 & 1 & 1 & 1 & 0 & 14
 \end{array}$$

For multiplication (2)

$$\begin{array}{r|l}
 1 & 1 & & & 3 \\
 1 & 1 & & & 3 \\
 \hline
 1 & 1 & & & \\
 1 & 1 & & & \\
 \hline
 1 & 0 & 0 & 1 & 9
 \end{array}
 \quad
 \begin{array}{r|l}
 1 & 0 & 1 & & 5 \\
 & 1 & 1 & & 3 \\
 \hline
 1 & 0 & 1 & & \\
 1 & 1 & 1 & 1 & 15
 \end{array}
 \quad
 \begin{array}{r|l}
 1 & 0 & 1 & & 5 \\
 1 & 0 & 1 & & 5 \\
 \hline
 1 & 0 & 1 & 0 & \\
 1 & 1 & 0 & 0 & 1 & 25
 \end{array}$$

For division

$$\begin{array}{r|l}
 15 & \cancel{1} & \cancel{1} & 1 & 1 \\
 3 & \cancel{1} & \cancel{1} & \cancel{1} & 1 \\
 & & \cancel{1} & 1 & \\
 \hline
 & & & &
 \end{array}
 \quad
 1 \ 0 \ 1 \ || \ 5$$

All these operations are so easy that we shall never have to guess or apply trial and error, as we must do in ordinary division. Nor do we need to learn anything by rote here, as must be done in ordinary calculation, where, for example, it is necessary to know that 6 and 7 taken together makes 13, and that 5 multiplied by 3 gives 15, following the so-called Pythagorean table⁶ that one times one is one. But here everything is found and proven from the source, just as we see in the preceding examples under the signs (1) and (2).

6. Using your knowledge of base 10 addition, explain the examples of base 2 addition given by Leibniz. What is the likely meaning of the dot Leibniz includes in certain columns for addition? Using binary arithmetic, compute $1101 + 1110$, without converting these numbers to base 10. Explain the examples for binary subtraction, multiplication and division above. Keep in mind that these should be base 2 analogues of base 10 procedures. Since Leibniz's example for division may be incomplete by today's standards, you may wish to supplement his work with additional steps, indicating clearly what multiples of 3 are subtracted from 15 in base 2. Finally, using binary arithmetic, compute the following.

$$11010 - 1101, \quad (1101) \cdot (11), \quad 1101 \div 101.$$

Be sure to explain your work. For the division problem, you may state what the remainder is in terms of a binary whole number, without writing it as a fraction.

⁶This is likely a reference to the multiplication table.

However, I am not at all recommending this manner of counting as a replacement for the ordinary practice of tens. For aside from the fact that we are accustomed to this, there is no need to learn what we have already memorized; the practice of tens is shorter, the numbers not as long. If we were accustomed to proceed by twelves or by sixteens, there would be even more of an advantage. As compensation for its length, however, calculation by twos, that is by 0 and by 1, is most basic for science; it permits new discoveries which become useful even in the practice of arithmetic, and especially in geometry, because when the numbers are reduced to the simplest principles, like 0 and 1, a wonderful order appears everywhere. For example, even in the Table of Numbers, we see in each column those periods which always reappear. In the first column it is 01, in the second 0011, in the third 00001111, in the fourth 0000000111111111, and so on. Small zeroes are put into the table to fill the void at the beginning of the column, and to mark these periods better. Lines are also traced in the table indicating that what these lines enclose always reoccurs below them. The square numbers, cubes and other powers, as well as the triangular numbers,⁷ pyramidal numbers,⁸ and other figurate numbers, also have similar periods, so that one can immediately write the tables without even calculating. A certain tedium at the beginning, which later serves to spare us calculation and to allow us to go by rule infinitely far, is extremely advantageous.

What is surprising in this calculation is that this arithmetic of 0 and 1 contains the mystery of lines of an ancient king and philosopher named Fuxi, who is believed to have lived more than four thousand years ago and whom the Chinese regard as the founder of their empire and of their sciences. There are several figures of lines that are attributed to him; they all go back to this arithmetic. But it is enough to place here the so-called figures of the eight Cova [trigrams], which are basic, and to add to these an explanation which is manifest, so that it is understood that a whole line — signifies unity or one, and that a broken line — — signifies zero or 0.

| | | | | | | | |
|----|----|----|----|-----|-----|-----|-----|
| -- | — | -- | — | -- | — | -- | — |
| -- | -- | — | — | -- | -- | — | — |
| -- | -- | -- | -- | — | — | — | — |
| 0 | ↖ | 0 | ↖ | 0 | ↖ | 0 | ↖ |
| 0 | 0 | ↖ | ↖ | 0 | 0 | ↖ | ↖ |
| 0 | 0 | 0 | 0 | ↖ | ↖ | ↖ | ↖ |
| 0 | 1 | 10 | 11 | 100 | 101 | 110 | 111 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

The Chinese have lost the signification these Cova or of the lineations of Fuxi, perhaps for more than a millennium, and they have made commentaries on these, seeking for I don't know what distant meaning, so that it has now become necessary for them to have the true explanation from the Europeans. This is how it happened: scarcely two years ago I sent to the Reverend Father Bouvet, the famous French Jesuit living in Peking, my manner of counting by 0 and 1, and it was all he needed to recognize that this holds the key to Fuxi's figures. So he wrote to me on November 14, 1701, sending me the great figure of this princely philosopher which goes to 64. This leaves no more room for doubting the truth of our interpretation, so that one can indeed say that this Father has deciphered the

⁷The sequence 1, 3, 6, 10, 15, . . . , giving the number of dots in certain triangles [38, p. 49] forms the triangular numbers.

⁸The sequence 1, 4, 10, 20, 35, . . . , giving the number of dots in certain pyramids [70, p. 76] forms the pyramidal numbers.

enigma of Fuxi, with the help of what I had communicated to him. And since these figures constitute perhaps the most ancient monument of science that still remains in the world, this restoration of their meaning after such a great interval of time appears all the more fascinating.

The Electronic Age

John von Neumann (1903–1957) was a leading mathematician, physicist and engineer of the twentieth century, having contributed significantly to the foundations of quantum mechanics, the development of the atomic bomb, and the logical structure of the electronic digital computer [25, 36]. Born in Budapest Hungary, the young von Neumann showed a gift for mathematics, received a doctorate in the subject from the University of Budapest and a degree in chemical engineering from the *Eidgenössische Technische Hochschule* (Swiss Federal Polytechnic) in Zurich. He met the renowned David Hilbert (1862–1943) on a visit to Göttingen in 1926, after which he was offered the position of a *Privatdozent* (an un-salaried lecturer) at the University of Berlin and then at the University of Hamburg. In 1930 he visited the United States, accepting a salaried lectureship at Princeton University, a move which would shape the rest of his life.

Becoming a Professor of Mathematics at the prestigious Institute for Advanced Study (Princeton, New Jersey) in 1933, von Neumann was able to devote his time to the study of analysis, continuous geometry, fluid dynamics, wave propagation and differential equations. In 1943 he became a member of the Los Alamos Laboratory and helped develop the atomic bomb. The particular problem he faced, the implosion problem, was how to produce an extremely fast reaction in a small amount of the uranium isotope U^{235} in order to cause a great amount of energy to be released. In conjunction with Seth Neddermeyer, Edward Teller and James Tuck, this problem was solved with a high explosive lens designed to produce a spherical shock wave to cause the implosion necessary to detonate the bomb. Von Neumann's strength was his ability to model theoretical phenomena mathematically and solve the resulting equations numerically [25, p. 181], which required adroit skills in calculation.

Meanwhile in 1941 John William Mauchly (1907–1980), as a newly appointed Assistant Professor at the University of Pennsylvania's Moore School of Electrical Engineering, began discussions with graduate student John Presper Eckert (1919–1995) and others about the possibility of an electronic digital computing device that would be faster and more accurate than any existing machine, designed in part to meet the computing needs of the Ballistics Research Laboratory (BRL) of the Army Ordnance Department in Aberdeen, Maryland. With the help of mathematician and First Lieutenant Herman Goldstine (1913–), Mauchly's proposal for a high-speed vacuum-tube computer received funding from the BRL in 1943. The device was dubbed the Electronic Numerical Integrator and Computer (ENIAC). Tested in late 1945, and unveiled in 1946, the ENIAC was a behemoth containing 18,000 vacuum tubes and requiring 1,800 square feet of floor space for the computer alone [64, p. 133]. Arithmetic on the ENIAC was performed using the base 10 decimal system, requiring the ability to store ten different values for each digit of a numerical quantity. The multiplication table for all digits between zero and nine was also stored on the machine. The ENIAC was not programmable in the modern sense of a coded program, and contained no sub-unit similar to a present-day compiler. To alter its function, i.e., to implement a different numerical algorithm, external switches and cables had to be repositioned. Designs for a more robust machine may have been in place as the ENIAC went into production, but the rush to meet the needs of the war effort took precedence.

By serendipity, in the summer of 1944 Goldstine met von Neumann at a railway station in Aberdeen, both working on separate highly classified projects. Goldstine writes: "Prior to that

time I had never met this great mathematician, but I knew much about him of course and had heard him lecture on several occasions” [25, p. 182]. After a discussion of the computing power of the ENIAC, von Neumann became keenly interested in this machine, and in late 1945 tested it on computations needed for the design of the hydrogen bomb. Von Neumann quickly became involved with the logical structure of the next generation of computing machinery, the Electronic Discrete Variable Automatic Computer (EDVAC), designed around the “stored program” concept. The instructions of an algorithm could be stored electronically on the EDVAC and then executed in sequential order. In this way, von Neumann had outlined a “universal computing machine” in the sense of Alan Turing (1912–1954), with the universal character referring to the machine’s ability to execute any algorithmic procedure that could be reduced to simple logical steps. Turing first introduced a logical description of his universal computing machine in 1936 [66] as the solution to a problem posed by David Hilbert. Von Neumann, who had studied logic early in his career, was certainly aware of Turing’s work, and in 1938 had offered Turing an assistantship at the Institute for Advanced Study [25, p. 271]. In 1945 von Neumann issued his white paper “First Draft of a Report on the EDVAC” [69] under the auspices of the University of Pennsylvania and the United States Army Ordnance Department. Although this draft was never revised, the ideas therein soon became known as von Neumann architecture in computer design. Let’s read a few excerpts from this paper [69] related to binary arithmetic.

First Draft of a Report on the EDVAC

2.2 First: Since the device is primarily a computer, it will have to perform the elementary operations of arithmetic most frequently. There are addition, subtraction, multiplication and division: $+$, $-$, \times , \div . It is therefore reasonable that it should contain specialized organs for just these operations. At any rate a *central arithmetical* part of the device will probably have to exist, and this constitutes *the first specific part: CA*.

4.3 It is clear that a very high speed computing device should ideally have vacuum tube elements. Vacuum tube aggregates like counters and scalars have been used and found reliable at reaction times (synaptic delays) as short as a microsecond ($= 10^{-6}$ seconds).

5.1 Let us now consider certain functions of the first specific part: the central arithmetical part CA.

The element in the sense of 4.3, the vacuum tube used as a current valve or *gate*, is an all-or-none device, or at least it approximates one: According to whether the grid bias is above or below cut-off; it will pass current or not. It is true that it needs definite potentials on all its electrodes in order to maintain either state, but there are combinations of vacuum tubes which have perfect equilibria: Several states in each of which the combination can exist indefinitely, without any outside support, while appropriate outside stimuli (electric pulses) will transfer it from one equilibrium into another. These are the so called *trigger circuits*, the basic one having two equilibria. The trigger circuits with more than two equilibria are disproportionately more involved.

Thus, whether the tubes are used as gates or as triggers, the all-or-none, two equilibrium arrangements are the simplest ones. Since these tube arrangements are to handle numbers by means of their digits, it is natural to use a system of arithmetic in which the digits are also two valued. This suggests the use of the binary system.

5.2 A consistent use of the binary system is also likely to simplify the operations of multiplication and division considerably. Specifically it does away with the decimal multiplication table. In other words: Binary arithmetic has a simpler and more one-piece logical structure than any other, particularly than the decimal⁹ one.

7. Let a and b denote binary variables with one digit each. Using only the logical connectives \wedge (and), \vee (or) and \sim (not), find a logical expression which gives the digit in the one's place (the right-hand digit) of $a + b$. Find a logical expression which gives the digit in the two's place (the left-hand digit) of $a + b$. Explain your answer.

Extra Credit A: Find a pattern in the binary representation of the square numbers 1, 4, 9, 16, 25, Leibniz claims to have found such patterns.

Extra Credit B: If in question 1 of the main project, standard weights can be placed on both sides of the balance, what four standard weights should be used in order to weigh all stones of integral weight between 1 and 40 inclusive?

⁹base 10

5 Arithmetic Backwards from Shannon to the Chinese Abacus

Jerry M. Lodder¹⁰

Recall that in the 1945 white paper “First Draft of a Report on the EDVAC” (Electronic Discrete Variable Automatic Computer), John von Neumann (1903–1957) advocated the use of binary arithmetic for the digital computers of his day. Vacuum tubes afforded these machines a speed of computation unmatched by other calculational devices, with von Neumann writing: “Vacuum tube aggregates . . . have been found reliable at reaction times as short as a microsecond . . .” [69, p. 188].

Predating this, in 1938 Claude Shannon (1916–2001) published a ground-breaking paper “A Symbolic Analysis of Relay and Switching Circuits” [60] in which he demonstrated how electronic circuits can be used for binary arithmetic, and more generally for computations in Boolean algebra and logic. These relay contacts and switches performed at speeds slower than vacuum tubes. Shannon, as von Neumann, identified an economy of representing numbers electronically in binary notation as well as an ease for arithmetic operations, such as addition. Shannon [60] writes:

A circuit is to be designed that will automatically add two numbers, using only relays and switches. Although any numbering base could be used the circuit is greatly simplified by using the scale of two. Each digit is thus either 0 or 1; the number whose digits in order are $a_k, a_{k-1}, a_{k-2}, \dots, a_2, a_1, a_0$ has the value $\sum_{j=0}^k a_j 2^j$.

1. Explain how the base 10 number 95 can be written in base 2 using the above formula. In particular, compute $a_k, a_{k-1}, a_{k-2}, \dots, a_2, a_1, a_0$ for the number 95. What is k in this case? Write $\sum_{j=0}^k a_j 2^j$ in terms of addition symbols using the above value for k and each value of a_j .

Claude Elwood Shannon was a pioneer in electrical engineering, mathematics and computer science, having founded the field of information theory, and discovered key relationships between Boolean algebra and computer circuits [63]. Born in the state of Michigan in 1916, he showed an interest in mechanical devices, and studied both electrical engineering and mathematics at the University of Michigan. Having received Bachelor of Science degrees in both of these subjects, he then accepted a research assistantship in the Department of Electrical Engineering at the Massachusetts Institute of Technology. The fundamental relation between Boolean logic and electrical circuits formed the topic of his master’s thesis at MIT, and became his first published paper [60], for which he received an award from the combined engineering societies of the United States. In 1940 he earned his doctorate in mathematics from MIT with the dissertation “An Algebra for Theoretical Genetics.” He spent the academic year 1940–41 visiting the Institute for Advanced Study in Princeton, New Jersey, where he began to formulate his ideas about information theory and efficient communication systems. The next fifteen years were productively spent at Bell Laboratories, and in 1948 he launched a new field of study, information theory, with the paper “A Mathematical Theory of Communication” [61]. He published extensively in communication theory, cryptography, game theory and computer science. In 1956 Dr. Shannon accepted a professorship at MIT, and retired in 1978.

Let’s read a few excerpts from “A Symbolic Analysis of Relay and Switching Circuits” [60] with an eye toward understanding the circuitry behind binary arithmetic.

¹⁰Mathematical Sciences; Dept. 3MB, Box 30001; New Mexico State University; Las Cruces, NM 88003; jlodder@nmsu.edu.

A Symbolic Analysis of Relay and Switching Circuits

Claude E. Shannon

I. Introduction

In the control and protective circuits of complex electrical systems it is frequently necessary to make intricate interconnections of relay contacts and switches. Examples of these circuits occur in automatic telephone exchanges, industrial motor-control equipment, and in almost any circuits designed to perform complex operations automatically. In this paper a mathematical analysis of certain of the properties of such networks will be made. . . .

II. Series-Parallel Two-Terminal Circuits

Fundamental Definitions and Postulates

We shall limit our treatment of circuits containing only relay contacts and switches, and therefore at any given time the circuit between two terminals must be either open (infinite impedance) or closed (zero impedance). Let us associate a symbol X_{ab} or more simply X with the terminals a and b . This variable, a function of time, will be called the hindrance of the two-terminal circuit $a-b$. The symbol 0 (zero) will be used to represent the hindrance of a closed circuit and the symbol 1 (unity) to represent the hindrance of an open circuit. Thus when the circuit $a-b$ is open $X_{ab} = 1$ and when closed $X_{ab} = 0$ Now let the symbol + (plus) be defined to mean the series connection of the two-terminal circuits whose hindrances are added together. Thus $X_{ab} + X_{cd}$ is the hindrance of the circuit $a-d$ when b and c are connected together. Similarly the product of two hindrances $X_{ab} \cdot X_{cd}$ or more briefly $X_{ab}X_{cd}$ will be defined to mean the hindrance of the circuit formed by connecting the circuits $a-b$ and $c-d$ in parallel. A relay contact or switch will be represented in a circuit by the symbol in Figure 1, the letter being the corresponding hindrance function. Figure 2 shows the interpretation of the plus sign and Figure 3 the multiplication sign.

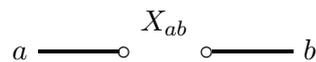


Figure 1. Symbol for hindrance function.



Figure 2. Interpretation of addition.

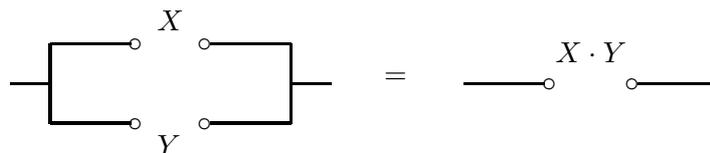
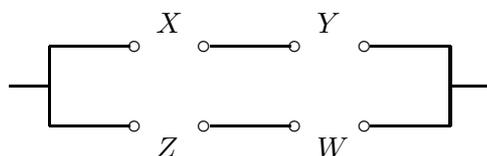


Figure 3. Interpretation of multiplication.

2. Consider X_{ab} as a Boolean variable with only two possible values 0 or 1. If $X_{ab} = 0$, then the switch in Figure 1 is closed, and current flows from a to b . If $X_{ab} = 1$, then the switch is open, and current does not flow from a to b . Now let X and Y be two Boolean variables. In a table listing all possible values for X and Y , record the results for X and Y joined in series, i.e., $X + Y$. Be sure to justify your answer by discussing whether current flows in the series circuit. Note that if current flows from left to right in Figure 2, then $X + Y = 0$, while if current does not flow, then $X + Y = 1$. To what extent does $X + Y$ represent a usual notation of addition? To what extent does $X + Y$ represent a construction in logic?

3. In a table listing all possible values for X and Y , record the results for X and Y joined in parallel, i.e., $X \cdot Y$. Justify your answer by discussing whether current flows in the parallel circuit. To what extent does $X \cdot Y$ represent a usual notion of multiplication? To what extent does $X \cdot Y$ represent a construction in logic?

4. These basic series and parallel circuits may be combined in any combination, using any finite number of switches. let X, Y, Z and W be Boolean variables, used as switches in the picture below. In a table listing all possible values for X, Y, Z and W , compute the value (0 or 1) of the circuit:



Explain your result in terms of a simple logical construction involving the results for the basic circuits $X + Y$ and $Z + W$:



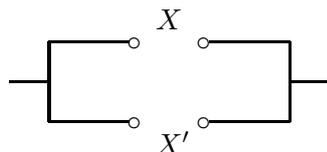
Shannon continues:

We shall now define a new operation to be called negation. The negative of a hindrance X will be written X' and is defined to be a variable which is equal to 1 when X equals 0 and equal to 0 when X equals 1.

5. In a table listing both values for X , compute the value of the circuit:



In another table, compute the value of the circuit:



Can you interpret these tables in terms of simple logical constructions?

6. Now let a and b be binary variables with one digit. Recall from the project “Binary Arithmetic: From Leibniz to von Neumann” that the digit in the one’s place (the right-hand digit) of the *arithmetic sum* $a + b$ can be expressed using the “exclusive or” operation. Find a circuit which gives this digit. Justify your answer with a table that lists all possible values of a and b . Note that the arithmetic sum $a + b$ is the result of adding the binary values of a and b , not a and b combined in series.

7. Let a and b be binary variables with with digit as in question 6. Find a circuit which gives the digit in the two’s place (the left-hand digit) of the arithmetic sum $a + b$. Justify your answer using a table listing all possible values of a and b .

8. Let a and b be binary variables with two possible digits. In Shannon’s notation,

$$a = a_12^1 + a_02^0, \quad b = b_12^1 + b_02^0.$$

The digits of a are a_1, a_0 , and the digits of b are b_1, b_0 . Let c_0 be the result of the carried digit from $a_0 + b_0$. Either $c_0 = 0$ or $c_0 = 1$. In terms of the values for a_1, b_1 and c_0 , when is the digit in the two’s place for the arithmetic sum $a + b$ equal to zero? equal to one? Using a_1, b_1 and c_0 as switches, find a circuit which gives the digit in the two’s place for $a + b$. Justify your answer using a table that lists all possible values for a_1, b_1 and c_0 .

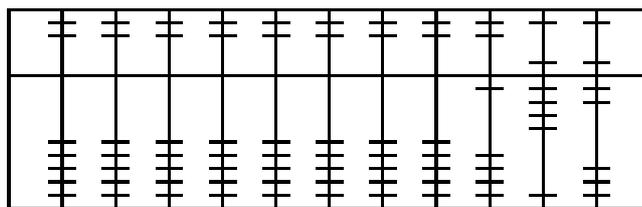
9. Let a and b be binary variables with two digits as in part question 8. Using a_1, b_1 and c_0 as switches, find a circuit which gives the digit in the four’s place (the left-most digit) of $a + b$. Justify your answer using a table listing all possible values for a_1, b_1 and c_0 .

Recall that of binary numeration, Leibniz [20, p. 225] writes:

However, I am not at all recommending this manner of counting as a replacement for the ordinary practice of tens. . . . [The] practice of tens is shorter, the numbers not as long. If we were accustomed to proceed by twelves or by sixteens, there would be even more of an advantage.

Writing large numbers by hand in binary notation easily results in transcription errors, since there are often many digits in a number base 2. However, entering base 10 numbers on a computer requires a conversion to base 2 at some level, a conversion which is not readily made, since 10 is not an integral power of 2. Let’s now examine the Chinese abacus, and remember that Leibniz’s intellectual curiosity had led him to the study of Chinese culture and religion, with an interpretation of the ancient text *Yijing* (the *I-ching* or *Book of Changes*) in terms of binary numeration.

The Chinese abacus (*suan pan*) consists of bars set in a rectangular frame, with the number of bars being 9, 11, 13, 17 or more [50, p. 211]. Each bar contains two upper beads and five lower beads, separated by a crossbar. Each upper bead counts as five units, and each lower bead counts as one unit. Traditionally numbers are represented positionally using base 10. A decimal point could be arbitrarily chosen between two bars of the abacus, and the digits are then arranged from left to right in decreasing powers of ten, so that the one’s place is to the right of the ten’s place, the ten’s place is to the right of the hundred’s place [52, p. 74–75]. Before representing a number on the abacus, all beads are moved away from the central crossbar so that they rest against the frame. Placing the decimal point at the far right of the frame, the base 10 number 197 would be displayed by moving one upper bead and two lower beads against the crossbar of the right-most bar, one upper bead and four lower beads against the crossbar of the bar immediately to the left of that, and one lower bead against the crossbar of the next bar.



The number 197 set on a Chinese abacus.

Go to the library or the world wide web and research how (base 10) addition is performed on an abacus. Pay particular attention to the operation known today as “carrying.” Notice that representing a number in base 10 requires a minimum of ten distinct values for bead arrangements along a given bar, which includes the value zero.

10. On a Chinese abacus, how many distinct numerical values can be represented along a given bar, including the value zero? Note that certain numbers greater than ten can be constructed by using two five beads on the same bar. Let N denote this number of distinct values. If the full range of values for bead arrangements is employed on each bar, what number base is represented on a Chinese abacus?
11. Using N as in question 10, write the base 10 numbers from 1 to 20 inclusive in base N . Although you may invent any symbols that you wish for additional digits in this new base, be sure to explain what your symbols mean. For now, call the new digits n_1, n_2, n_3 , etc., where $n_1 = 10, n_2 = 11, n_3 = 12$, etc.
12. In a table list the base 10 numbers 1 through 32 inclusive, their binary equivalents and their base N equivalents. Is there a pattern between the base 2 and base N representations? Explain.
13. Consider the number $9n_25n_4$ in base N , where n_4 is in the one’s place, 5 in the N ’s place, n_2 in the N^2 ’s place, and 9 in the N^3 ’s place. Explain how to perform the addition

$$9n_25n_4 + n_172n_2$$

in base N on a Chinese abacus. What is the value of this sum in base N ? Convert the final sum to base 10, and explain the conversion process.

Extra Credit: What is the sum $9n_25n_4 + n_172n_2$ in base 2? Justify your answer.

6 Pascal’s Treatise on the Arithmetical Triangle: Mathematical Induction, Combinations, the Binomial Theorem and Fermat’s Theorem

David Pengelley^{11 12}

Introduction

Blaise Pascal (1623–1662) was born in Clermont-Ferrand in central France. Even as a teenager his father introduced him to meetings for mathematical discussion in Paris run by Marin Mersenne, who served as a primary conduit for transmitting mathematical ideas widely at that time, before the existence of any research journals. He quickly became involved in the development of projective geometry, the first in a sequence of highly creative mathematical and scientific episodes in his life, punctuated by periods of religious fervor. Around age twenty-one he spent several years developing a mechanical addition and subtraction machine, in part to help his father in tax computations as a local administrator. It was the first of its kind ever to be marketed. Then for several years he was at the center of efforts to understand vacuum, which led to an understanding of barometric pressure. In fact the scientific unit of pressure is named the *pascal*. He is also known for Pascal’s Law on the behavior of fluid pressure.

Around 1654 Pascal conducted his studies on the Arithmetical Triangle (“Pascal’s Triangle”) and its relationship to probabilities. His correspondence with Pierre de Fermat (1601–1665) in that year marks the beginning of probability theory. Several years later, Pascal refined his ideas on area problems via the method of indivisibles already being developed by others, and solved various problems of areas, volumes, centers of gravity, and lengths of curves. Later in the seventeenth century, Gottfried Leibniz, one of the two inventors of the infinitesimal calculus which supplanted the method of indivisibles, explicitly credited Pascal’s approach as stimulating his own ideas on the so-called characteristic triangle of infinitesimals in his fundamental theorem of calculus. After only two years of work on the calculus of indivisibles, Pascal fell gravely ill, abandoned almost all intellectual work to devote himself to prayer and charitable work, and died three years later at age thirty-nine. In addition to his work in mathematics and physics, Pascal is prominent for his *Provincial Letters* defending Christianity, which gave rise to his posthumously published *Pensées* (Thoughts) on religious philosophy [16, 21]. Pascal was an extremely complex person, and one of the outstanding scientists of the mid-seventeenth century, but we will never know how much more he might have accomplished with more sustained efforts and a longer life.

Pascal’s *Traité du Triangle Arithmétique* (in English translation in [53, vol. 30]) makes a systematic study of the numbers in his triangle. They have simultaneous roles in mathematics as figurate numbers¹³, combination numbers, and binomial coefficients, and he elaborates on all these. Given their multifaceted nature, it is no wonder that these ubiquitous numbers had already been in use for over 500 years, in places ranging from China to the Islamic world [38]. Pascal, however, was the first to connect binomial coefficients with combinatorial coefficients in probability. In fact, a major motivation for Pascal was a question from the beginnings of probability theory, about the equitable division of stakes in an interrupted game of chance. The question had been posed to Pascal around 1652 by Antoine Gombaud, the Chevalier de Méré, who wanted to improve his

¹¹Mathematical Sciences; Dept. 3MB, Box 30001; New Mexico State University; Las Cruces, NM 88003; davidp@nmsu.edu.

¹²With thanks to Joel Lucero-Bryan and Jerry Lodder.

¹³Figurate numbers count the number of equally spaced dots in geometric figures. Especially important to Pascal were the numbers of dots in equilateral triangles, triangular pyramids, and so forth in higher dimensions.

chances at gambling: Suppose two players are playing a fair game, to continue until one player wins a certain number of rounds, but the game is interrupted before either player reaches the winning number. How should the stakes be divided equitably, based on the number of rounds each player has won [38, p. 431, 451ff]? The solution requires the combinatorial properties inherent in the numbers in the Arithmetical Triangle, as Pascal demonstrated in his treatise, since they count the number of ways various occurrences can combine to produce a given result. The Arithmetical Triangle overflows with fascinating patterns and applications, and we will see several of these in reading his treatise. We will study parts of Pascal's explanation of the connections between the numbers in his triangle and combination numbers. The reader is encouraged to read his entire treatise to see its many other aspects and connections.

From Pascal's treatise we will also learn the principle of mathematical induction. Pascal explains this in the specific context of proofs about the numbers in the triangle. The basic idea of mathematical induction had occurred in the mathematics of the Islamic world during the Middle Ages, and in southern Europe in the fourteenth century [38], but Pascal's was perhaps the first text to make a complete explicit statement and justification of this extremely powerful method of proof in modern mathematics. Mathematical induction is an astonishingly clever technique that allows us to prove claims about infinitely many interlinked phenomena all at once, even when proving just a single one of them in isolation would be very difficult! It will be a challenging technique to master, but will provide tremendous power for future mathematical work.

Learning about the connections of the Arithmetical Triangle to the binomial theorem in algebra will also allow an application to proving a famous and extremely important theorem on prime numbers discovered by Pascal's correspondent Pierre de Fermat (1601–1665) of Toulouse, on congruence remainders and prime numbers. This prepares one to understand the RSA cryptosystem, which today is at the heart of securing electronic transactions. We'll see how all these things are interconnected, and along the way we'll also acquire important mathematical tools, like notations for general indexing, summations, and products, and learn how to work with recurrence relations.

Note: An alternative start for the project is to explore the beginning of Part Three first, in which you will develop experimental evidence, based on calculations in congruence arithmetic, which will lead you to conjecture Fermat's theorem about prime numbers. The process of *experimentation* and *conjecture* are two of the key steps in the creation of new mathematics, followed by attempting to *prove* one's conjectures, and then the possibility of *generalization* and interaction with other areas for further research.

Part One: The Arithmetical Triangle and Mathematical Induction

Let us begin reading Blaise Pascal's

TREATISE ON THE ARITHMETICAL TRIANGLE

DEFINITIONS

I call *arithmetical triangle* a figure constructed as follows:

From any point, G, I draw two lines perpendicular to each other, GV, Gζ in each of which I take as many equal and contiguous parts as I please, beginning with G, which I number 1, 2, 3, 4, etc., and these numbers are the *exponents* of the sections of the lines.

Next I connect the points of the first section in each of the two lines by another line, which is the base of the resulting triangle.

In the same way I connect the two points of the second section by another line, making a second triangle of which it is the base.

| | | | | | | | | | | | |
|----|----------------|---------------|---------------|----------------|-------------|---------------|--------------|----|---|---|----|
| Z | 1 | 2 | 3 | 4 | 5 | 6 | 7 | L | 8 | 9 | 10 |
| 1 | G 1 | σ 1 | π 1 | λ 1 | μ 1 | δ 1 | ζ 1 | 1 | 1 | 1 | |
| 2 | φ 1 | ψ 2 | θ 3 | R 4 | S 5 | N 6 | 7 | 8 | 9 | | |
| 3 | A 1 | B 3 | C 6 | ω 10 | ξ 15 | 21 | 28 | 36 | | | |
| 4 | D 1 | E 4 | F 10 | ρ 20 | Y 35 | 56 | 84 | | | | |
| 5 | H 1 | M 5 | K 15 | 35 | 70 | 126 | | | | | |
| 6 | P 1 | Q 6 | 21 | 56 | 126 | | | | | | |
| 7 | V 1 | 7 | 28 | 84 | | | | | | | |
| 8 | T 1 | 8 | 36 | | | | | | | | |
| 9 | 1 | 9 | | | | | | | | | |
| 10 | 1 | | | | | | | | | | |

And in this way connecting all the points of section with the same exponent, I construct as many triangles and bases as there are exponents.

Through each of the points of section and parallel to the sides I draw lines whose intersections make little squares which I call *cells*.

Cells between two parallels drawn from left to right are called *cells of the same parallel row*, as, for example, cells *G*, σ , π , etc., or φ , ψ , θ , etc.

Those between two lines are drawn from top to bottom are called *cells of the same perpendicular row*, as, for example, cells *G*, φ , *A*, *D*, etc., or σ , ψ , *B*, etc.

Those cut diagonally by the same base are called *cells of the same base*, as, for example, *D*, *B*, θ , λ , or *A*, ψ , π .

Cells of the same base equidistant from its extremities are called *reciprocals*, as, for example, *E*, *R* and *B*, θ , because the parallel exponent of one is the same as the perpendicular exponent of the other, as is apparent in the above example, where *E* is in the second perpendicular row and in the fourth parallel row and its reciprocal, *R*, is in the second parallel row and in the fourth perpendicular row, reciprocally. It is very easy to demonstrate that cells with exponents reciprocally the same are in the same base and are equidistant from its extremities.

It is also very easy to demonstrate that the perpendicular exponent of any cell when added to is parallel exponent exceeds by unity the exponent of its base.

For example, cell *F* is in the third perpendicular row and in the fourth parallel row and in the sixth base, and the exponents of rows 3 and 4, added together, exceed by unity the

exponent of base 6, a property which follows from the fact that the two sides of the triangle have the same number of parts; but this is understood rather than demonstrated.

Of the same kind is the observation that each base has one more cell than the preceding base, and that each has as many cells as its exponent has units; thus the second base, $\phi\sigma$, has two cells, the third, $A\psi\pi$, has three, etc.

Now the numbers assigned to each cell are found by the following method:

The number of the first cell, which is at the right angle, is arbitrary; but that number having been assigned, all the rest are determined, and for this reason it is called the *generator* of the triangle. Each of the others is specified by a single rule as follows:

The number of each cell is equal to the sum of the numbers of the perpendicular and parallel cells immediately preceding. Thus cell F , that is, the number of cell F , equals the sum of cell C and cell E , and similarly with the rest.

Whence several consequences are drawn. The most important follow, wherein I consider triangles generated by unity, but what is said of them will hold for all others.

FIRST CONSEQUENCE

In every arithmetical triangle all the cells of the first parallel row and of the first perpendicular row are the same as the generating cell.

For by definition each cell of the triangle is equal to the sum of the immediately preceding perpendicular and parallel cells. But the cells of the first parallel row have no preceding perpendicular cells, and those of the first perpendicular row have no preceding parallel cells; therefore they are all equal to each other and consequently to the generating number.

Thus $\varphi = G + 0$, that is, $\varphi = G$,
 $A = \varphi + 0$, that is, φ ,
 $\sigma = G + 0$, $\pi = \sigma + 0$,

And similarly of the rest.

SECOND CONSEQUENCE

In every arithmetical triangle each cell is equal to the sum of all the cells of the preceding parallel row from its own perpendicular row to the first, inclusive.

Let any cell, ω , be taken. I say that it is equal to $R + \theta + \psi + \varphi$, which are the cells of the next higher parallel row from the perpendicular row of ω to the first perpendicular row.

This is evident if we simply consider a cell as the sum of its component cells.

For ω equals $R + C$
 $\theta + B$
 $\psi + A$
 φ ,

for A and φ are equal to each other by the preceding consequence.

Therefore $\omega = R + \theta + \psi + \varphi$.

THIRD CONSEQUENCE

In every arithmetical triangle each cell is equal to the sum of all the cells of the preceding perpendicular row from its own parallel row to the first, inclusive.

Let any cell, C , be taken. I say that it is equal to $B + \psi + \sigma$, which are the cells of the preceding perpendicular row from the parallel row of cell C to the first parallel row.

This is also apparent, as above, simply by the interpretation of cells.

$$\text{For } C = B + \underbrace{\psi + \pi}_{\sigma},$$

for $\pi = \sigma$ by the first consequence.

Therefore $C = B + \psi + \sigma$.

FOURTH CONSEQUENCE

In every arithmetical triangle each cell exceeds by unity the sum of all the cells within its parallel and perpendicular rows, exclusive.

Let any cell, ξ , be taken. I say that $\xi - G = R + \theta + \psi + \varphi + \lambda + \pi + \sigma + G$, which are all the numbers between row $\xi\omega CBA$ and row $\xi S\mu$ exclusive.

This is also apparent by interpretation.

$$\text{For } \xi = \lambda + R + \underbrace{\omega}_{\pi + \theta + C} + \underbrace{\sigma + \psi + B}_{G + \varphi + A} + \underbrace{G.}$$

Therefore $\xi = \lambda + R + \pi + \theta + \sigma + \psi + G + \varphi + G$.

N.B. I have written in the enunciation *each cell exceeds by unity* because the generator is unity. If it were some other number, the enunciation should read: *each cell exceeds by the generating number*.

1. Pascal's Triangle and its numbers

- (a) Let us use the notation $T_{i,j}$ to denote what Pascal calls the number assigned to the cell in *parallel row* i (which we today call just *row* i) and *perpendicular row* j (which we today call *column* j). We call the i and j by the name *indices* (plural of *index*) in our notation. Using this notation, explain exactly what Pascal's rule is for determining all the numbers in all the cells. Be sure to give full details. This should include explaining for exactly which values of the indices he defines the numbers.
- (b) In terms of our notation $T_{i,j}$, explain his terms *exponent*, *base*, *reciprocal*, *parallel row*, *perpendicular row*, and *generator*.

- (c) Rewrite Pascal's first two "Consequences" entirely in the $T_{i,j}$ notation.
- (d) Rewrite his proofs of these word for word in our notation also.
- (e) Do you find his proofs entirely satisfactory? Explain why or why not.
- (f) Improve on his proofs using our notation. In other words, make them apply for arbitrary prescribed situations, not just the particular examples he lays out.

2. Modern mathematical notation

Read in a modern textbook about index, summation, and product notations, and recurrence relations. Do some exercises.

FIFTH CONSEQUENCE

In every arithmetical triangle each cell is equal to its reciprocal.

For in the second base, $\phi\sigma$, it is evident that the two reciprocal cells, φ, σ , are equal to each other and to G .

In the third base, A, ψ, π , it is also obvious that the reciprocals, π, A , are equal to each other and to G .

In the fourth base it is obvious that the extremes, D, λ , are again equal to each other and to G .

And those between, B, θ , are obviously equal since $B = A + \psi$ and $\theta = \pi + \psi$. But $\pi + \psi = A + \psi$ by what has just been shown. Therefore, etc.

Similarly it can be shown for all the other bases that reciprocals are equal, because the extremes are always equal to G and the rest can always be considered as the sum of cells in the preceding base which are themselves reciprocals.

3. Symmetry in the triangle: first contact with mathematical induction

Write the Fifth Consequence using our index notation. Use index notation and the ideas in Pascal's proof to prove the Consequence in full generality, not just for the example he gives. Explain the conceptual ideas behind the general proof.

4. Mathematical induction: gaining more familiarity

- (a) Read in a modern textbook about mathematical induction.
- (b) Prove Pascal's First Consequence by mathematical induction. (Hint: for a proof by mathematical induction, always first state very clearly exactly what the n -th mathematical statement $P(n)$ says. Then state and prove the base step. Then state the inductive step very clearly before you prove it.)
- (c) Write the general form of Pascal's Second Consequence, and give a general proof using summation notation, but following his approach.
- (d) Now prove the Second Consequence by mathematical induction, i.e., a different proof.
- (e) **Optional:** More patterns.

- i. Write the Fourth Consequence using summation notation. Hint: You can write it using a sum of sums. Try writing Pascal's proof in full generality, using summation notation to help. If you don't complete it his way, explain why it is difficult.
- ii. Prove the Fourth Consequence by mathematical induction.

SEVENTH CONSEQUENCE

In every arithmetical triangle the sum of the cells of each base is double that of the preceding base.

Let any base, $DB\theta\lambda$, be taken. I say that the sum of its cells is double the sum of the cells of the preceding base, $A\psi\pi$.

For the extremes $\underbrace{D,}_{A,}$ $\underbrace{\lambda,}_{\pi,}$
 are equal to the extremes
 and each of the rest $\underbrace{B,}_{A + \psi,}$ $\underbrace{\theta,}_{\psi + \pi,}$
 is equal to two cells of the other base ...

Therefore $D + \lambda + B + \theta = 2A + 2\psi + 2\pi$.

The same thing is demonstrated in the same way of all other bases.

EIGHTH CONSEQUENCE

In every arithmetical triangle the sum of the cells of each base is a number of the double progression beginning with unity whose exponent is the same as that of the base.

For the first base is unity.

The second is double the first; therefore it is 2.

The third is double the second; therefore it is 4.

And so on to infinity.

N.B. If the generator were not unity but some other number, such as 3, the same thing would be true. But we should have to take not the numbers of the double progression beginning with unity, that is, 1, 2, 4, 8, 16, etc., but those of the double progression beginning with the generator 3, that is, 3, 6, 12, 24, 48, etc.

5. Sums of bases in the triangle: a geometric progression
 - (a) Use our index notation $T_{i,j}$ to explain exactly which are the numbers in the n -th base.
 - (b) In full generality, write the Seventh Consequence and its proof, using our $T_{i,j}$ notation.
 - (c) Write the statement of the Eighth Consequence in our notation, using modern exponential notation to describe his double progression. Use summation notation as needed, and introduce additional new notation if helpful. Then prove the Eighth Consequence by mathematical induction.

The next consequence is the most important and famous in the whole treatise. Pascal derives a formula for the ratio of consecutive numbers in a base. From this he will obtain an elegant and efficient formula for all the numbers in the triangle.

TWELFTH CONSEQUENCE

In every arithmetical triangle, of two contiguous cells in the same base the upper is to the lower as the number of cells from the upper to the top of the base is to the number of cells from the lower to the bottom of the base, inclusive.

Let any two contiguous cells of the same base, E, C , be taken. I say that

$E : C :: 2 : 3$
 the lower cell because there are two cells from E to the bottom, namely E, H ,
 the upper cell because there are three cells from C to the top, namely C, R, μ .

Although this proposition has an infinity of cases, I shall demonstrate it very briefly by supposing two lemmas:

The first, which is self-evident, that this proportion is found in the second base, for it is perfectly obvious that $\varphi : \sigma :: 1 : 1$;

The second, that if this proportion is found in any base, it will necessarily be found in the following base.

Whence it is apparent that it is necessarily in all the bases. For it is in the second base by the first lemma; therefore by the second lemma it is in the third base, therefore in the fourth, and to infinity.

It is only necessary therefore to demonstrate the second lemma as follows: If this proportion is found in any base, as, for example, in the fourth, $D\lambda$, that is, if $D : B :: 1 : 3$, and $B : \theta :: 2 : 2$, and $\theta : \lambda :: 3 : 1$, etc., I say the same proportion will be found in the following base, $H\mu$, and that, for example, $E : C :: 2 : 3$.

For $D : B :: 1 : 3$, by hypothesis.

Therefore $\underbrace{D + B} : B :: \underbrace{1 + 3} : 3$
 $E : B :: 4 : 3$

Similarly $B : \theta :: 2 : 2$, by hypothesis

Therefore $\underbrace{B + \theta} : B :: \underbrace{2 + 2} : 2$
 $C : B :: 4 : 2$

But $B : E :: 3 : 4$

Therefore, by compounding the ratios, $C : E :: 3 : 2$.

Q.E.D.

The proof is the same for all other bases, since it requires only that the proportion be found in the preceding base, and that each cell be equal to the cell before it together with the cell above it, which is everywhere the case.

6. Pascal's Twelfth Consequence: the key to our modern factorial formula

- (a) Rewrite Pascal's Twelfth Consequence as a generalized modern formula, entirely in our $T_{i,j}$ terminology. Also verify its correctness in a couple of examples taken from his table in the initial definitions section.
- (b) Adapt Pascal's proof by example of his Twelfth Consequence into modern generalized form to prove the formula you obtained above. Use the principle of mathematical induction to create your proof.

Now Pascal is ready to describe a formula for an arbitrary number in the triangle.

PROBLEM

Given the perpendicular and parallel exponents of a cell, to find its number without making use of the arithmetical triangle.

Let it be proposed, for example, to find the number of cell ξ of the fifth perpendicular and of the third parallel row.

All the numbers which precede the perpendicular exponent, 5, having been taken, namely 1, 2, 3, 4, let there be taken the same number of natural numbers, beginning with the parallel exponent, 3, namely 3, 4, 5, 6.

Let the first numbers be multiplied together and let the product be 24. Let the second numbers be multiplied together and let the product be 360, which, divided by the first product, 24, gives as quotient 15, which is the number sought.

For ξ is to the first cell of its base, V , in the ratio compounded of all the ratios of the cells between, that is to say, $\xi : V$

in the ratio compounded of $\xi : \rho, \rho : K, K : Q, Q : V$
 or by the twelfth consequence $3 : 4 \quad 4 : 3 \quad 5 : 2 \quad 6 : 1$

Therefore $\xi : V :: 3 \cdot 4 \cdot 5 \cdot 6 : 4 \cdot 3 \cdot 2 \cdot 1$.

But V is unity; therefore ξ is the quotient of the division of the product of $3 \cdot 4 \cdot 5 \cdot 6$ by the product of $4 \cdot 3 \cdot 2 \cdot 1$.

N.B. If the generator were not unity, we should have had to multiply the quotient by the generator.

7. Pascal's formula for the numbers in the Arithmetical Triangle

- (a) Write down the general formula Pascal claims in solving his "Problem." Your formula should read $T_{i,j} =$ "some formula in terms of i and j ." Also write your formula entirely in terms of factorials.
- (b) Look at the reason Pascal indicates for his formula for a cell, and use it to make a general proof for your formula above for an arbitrary $T_{i,j}$. You may try to make your proof just like Pascal is indicating, or you may prove it by mathematical induction.

VARIOUS USES OF THE ARITHMETICAL TRIANGLE WHOSE GENERATOR IS UNITY

Having given the proportions obtaining between the cells and the rows of arithmetical triangles, I turn in the following treatises to various uses of those triangles whose generator is unity. But I leave out many more than I include; it is extraordinary how fertile in properties this triangle is. Everyone can try his hand. I only call your attention here to the fact that in everything that follows I am speaking exclusively of arithmetical triangles whose generator is unity.

Part Two: Combinations and the Arithmetical Triangle

We continue reading Pascal's *Treatise on the Arithmetical Triangle*:

USE OF THE ARITHMETICAL TRIANGLE FOR COMBINATIONS

The word *combination* has been used in several different senses, so that to avoid ambiguity I am obliged to say how I understand it.

When of many things we may choose a certain number, all the ways of taking as many as we are allowed out of all those offered to our choice are here called the *different combinations*.

For example, if of four things expressed by the four letters, *A, B, C, D*, we are permitted to take, say any two, all the different ways of taking two out of the four put before us are called *combinations*.

Thus we shall find by experience that there are six different ways of choosing two out of four; for we can take *A* and *B*, or *A* and *C*, or *A* and *D*, or *B* and *C*, or *B* and *D*, or *C* and *D*.

I do not count *A* and *A* as one of the ways of taking two; for they are not different things, they are only one thing repeated.

Nor do I count *A* and *B* and *B* and *A* as two different ways; for in both ways we take only the same two things but in a different order, and I am not concerned with the order; so that I could make myself understood at once by those who are used to considering combinations, simply by saying that I speak only of combinations made without changing the order.

We shall also find by experience that there are four ways of taking three things out of four; for we can take *ABC* or *ABD* or *ACD* or *BCD*.

Finally we shall find that we can take four out of four in one way only, *ABCD*.

I shall speak therefore in the following terms:

1 in 4 can be combined 4 times.

2 in 4 can be combined 6 times.

3 in 4 can be combined 4 times.

4 in 4 can be combined 1 time.

Or:

the number of combinations of 1 in 4 is 4.

the number of combinations of 2 in 4 is 6.

the number of combinations of 3 in 4 is 4.

the number of combinations of 4 in 4 is 1.

But the sum of all the combinations in general that can be made in 4 is 15, because the number of combinations of 1 in 4, of 2 in 4, of 3 in 4, of 4 in 4, when joined together, is 15. After this explanation I shall give the following consequences in the form of lemmas:

LEMMA 1.

There are no combinations of a number in a smaller number; for example, 4 cannot be combined in 2.

...

PROPOSITION 2

The number of any cell is equal to the number of combinations of a number less by unity than its parallel exponent in a number less by unity than the exponent of its base.

Let any cell be taken, say F in the fourth parallel row and in the sixth base. I say that is equal to the number of combinations of 3 in 5, less by unity than 4 and 6, for it is equal to the cells $A + B + C$. Therefore by the preceding proposition, etc.

1. Combinations according to Pascal

- (a) Explain in your own words what Pascal says about how many combinations there are for choosing two things out of four things.
- (b) Write Pascal's Proposition 2 using our $T_{i,j}$ notation for numbers in the triangle. In other words, fill in a sentence saying " $T_{i,j}$ is the number of combinations of choosing ____ things from ____ things." Pascal's justification for his Proposition 2 is based on his Lemma 4 and Proposition 1, which are not included in this project. However, the reader is encouraged to study and understand them, to wit:
- (c) **Optional:** From Pascal's treatise [53, vol. 30], rewrite his statements and explanations of his Lemma 4 and Proposition 1 in your own words. State and prove Lemma 4 in the general case; that is, show that the number of combinations of k in n is the sum of the combinations of $k - 1$ in $n - 1$ and the combinations of k in $n - 1$. Also explain why Proposition 2 follows from Proposition 1.

2. Combinations and Pascal's recursion formula

- (a) The modern symbol $\binom{n}{r}$ means the number of ways ("combinations") of choosing r things from amongst n things. Explain how this is related to what we have been learning about the Arithmetical Triangle from reading Pascal. In particular, explain how the numbers $T_{i,j}$ are related to the numbers $\binom{n}{r}$. Do this by writing an equation expressing $T_{i,j}$ in $\binom{n}{r}$ notation, and also writing an equation expressing $\binom{n}{r}$ in $T_{i,j}$ notation. Now use the formula we learned earlier, from Pascal's solution to his *Problem*,¹⁴ to write a formula for the combination number $\binom{n}{r}$, and manipulate it to express it entirely in terms of factorials.

¹⁴Given the perpendicular and parallel exponents of a cell, to find its number without making use of the arithmetical triangle.

- (b) Now read in a modern textbook about the multiplication rule for counting possibilities, about permutations, and about combinations. Explain how a combination is different from a permutation.
- (c) Read in a modern textbook about the algebra of combinations, Pascal’s recursion formula, and how the text presents Pascal’s Triangle. How is it different from Pascal’s presentation?

Part Three: The Binomial Theorem and Fermat’s Theorem

Now we can put together all of what we have learned from Pascal to prove an extremely important result in number theory, called *Fermat’s Little Theorem*, which is at the heart of today’s encryption methods in digital communications. The ingredients will be the binomial theorem, proof by mathematical induction as learned from Pascal, Pascal’s formula for the numbers in his triangle (solved in his *Problem*), and uniqueness of prime factorization.

1. The binomial theorem and combinations

Read in a modern textbook about the binomial theorem. Write an explanation of the proof of the binomial theorem using the idea of counting combinations.

2. Discovering Fermat’s “Little” Theorem: prime numbers and congruence remainders

- (a) Make a table of the remainders of a^n upon division by n for positive integer values of both a and n ranging up to 14. To do this you should learn about congruence arithmetic, and figure out how to do these calculations quickly and easily without a calculator.
- (b) Based on your table, make a conjecture of the form $a^p \equiv ? \pmod{p}$ for p a prime number and a any integer. This is called Fermat’s “Little” Theorem; it is one of the most important phenomena in number theory. Also make some other interesting conjectures from patterns in your table, and try to prove them, perhaps using the binomial theorem.
- (c) Write up the details of proving Fermat’s Theorem by mathematical induction on a , with p held fixed. Use the binomial theorem, our knowledge of Pascal’s “factorial” formula for binomial coefficients, and the Fundamental Theorem of Arithmetic (uniqueness of prime factorization) to analyze divisibility of the binomial coefficients by a prime p .
- (d) **Optional:** Read about what Fermat was trying to do when he discovered his Theorem [48, p. 159ff]. Describe what you find in your own words.

3. **Optional:** The RSA cryptosystem

Read and study the RSA cryptosystem and its applications to digital security, including how it works, which follows from Fermat’s Theorem. Write up the details in your own words, with some example calculations.

7 Counting Triangulations of a Convex Polygon

Desh Ranjan^{15 16}

Introduction

In a 1751 letter to Christian Goldbach (1690–1764), Leonhard Euler (1707–1783) talks about the problem of counting the number of triangulations of a convex polygon. Euler, one of the most prolific mathematicians of all times, and Goldbach, who was a Professor of Mathematics and historian at St. Petersburg and later served as a tutor for Tsar Peter II, carried out extensive correspondence, mostly on mathematical matters. In his letter, Euler provides a “guessed” method for computing the number of triangulations of a polygon that has n sides but does not provide a proof of his method. The method, if correct, leads to a formula for calculating the number of triangulations of an n -sided polygon which can be used to quickly calculate this number [17, p. 339-350] [18]. Later, Euler communicated this problem to the Hungarian mathematician Jan Andrej Segner (1704–1777). Segner, who spent most of his professional career in Germany (under the German name Johann Andreas von Segner!), was the first Professor of Mathematics at the University of Göttingen, becoming the chair in 1735. Segner “solved” the problem by providing a proven correct method for computing the the number of triangulations of a convex n -sided polygon using the number of triangulations for polygons with fewer than n sides [59]. However, this method did not establish the validity (or invalidity) of Euler’s guessed method. Segner communicated his result to Euler in 1756 and in his communication he also calculated the number of triangulations for the n -sided polygons for $n = 1 \dots 20$ [59]. Interestingly enough, he made simple arithmetical errors in calculating the number of triangulations for polygons with 15 and 20 sides! Euler corrected these mistakes and also calculated the number of triangulations for polygons with up to 25 sides. It turns out that with the corrections, Euler’s guessed method gives the right number triangulations of polygons with up to 25 sides.

Was Euler’s guessed method correct? It looked like it was but there was no proof. The problem was posed as an open challenge to the mathematicians by Joseph Liouville (1809–1882) in the late 1830’s. He received solutions or purported solutions to the problem by many mathematicians (including one by Belgian mathematician Catalan which was correct but not so elegant!), some of which were later published in the Liouville journal, one of the primary journals of mathematics at that time and for many decades. The most elegant of these solutions was communicated to him in a paper by Gabriel Lamé (1795-1870) in 1838. French mathematician, engineer and physicist Lamé was educated at the prestigious Ecole Polytechnique and later at the Ecole des Mines[21, p. 601–602]. From 1832 to 1844 he served as the chair of physics at the Ecole Polytechnique, and in 1843 joined the Paris Academy of Sciences in the geometry section. He contributed to the fields of differential geometry, number theory, thermodynamics and applied mathematics. Among his publications are textbooks in physics and papers on heat transfer, where he introduced the rather useful technique of curvilinear coordinates. In 1851 he was appointed Professor of Mathematical Physics and Probability at the University of Paris, and resigned eleven years later after becoming deaf. Gauss considered Lamé the foremost French mathematician of his day [21, p. 601–602].

A translated version of this paper by Lamé is the key historical source for this project. This translation is presented in its entirety before the description of the project tasks in the next section. Interestingly and perhaps somewhat ironically, today these numbers are called Catalan numbers.

¹⁵Department of Computer Science; New Mexico State University; Las Cruces, NM 88003; dranjan@cs.nmsu.edu.

¹⁶With thanks to David Pengelley, Inna Pivkina and Karen Villaverde.

Understanding Triangulations

A *diagonal* in a (convex) polygon is a straight line that connects two non-adjacent vertices of the polygon. Two diagonals are different if they have at least one different endpoint. A *triangulation* of a polygon is a division of the polygon into triangles by drawing *non-intersecting* diagonals. For example, the 6-sided polygon $ABCDEF$ below is triangulated into 4 triangles by using the diagonals AD, AE, BD .

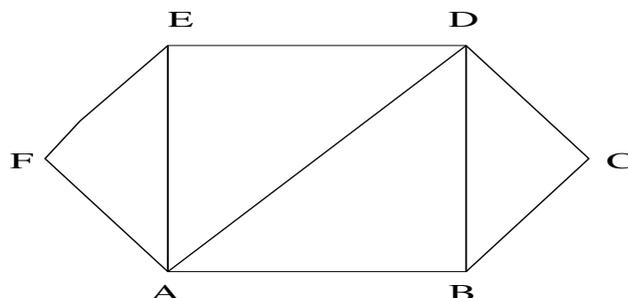


Figure 1: A Triangulation of $ABCDEF$

Two triangulations are different if at least one of the diagonals in a triangulation is different from all diagonals in the other triangulation.

Let's now read from Lamé's letter to Liouville [47].

Excerpt from a letter of Monsieur Lamé to Monsieur Liouville on the question: Given a convex polygon, in how many ways can one partition it into triangles by means of diagonals?¹⁷

The formula that you communicated to me yesterday is easily deduced from the comparison of two methods leading to the same goal.

Indeed, with the help of two different methods, one can evaluate the number of decompositions of a polygon into triangles: by consideration of the sides, or of the vertices.

I.

Let $ABCDEF \dots$ be a convex polygon of $n + 1$ sides, and denote by the symbol P_k the total number of decompositions of a polygon of k sides into triangles. An arbitrary side AB of $ABCDEF \dots$ serves as the base of a triangle, in each of the P_{n+1} decompositions of the polygon, and the triangle will have its vertex at C , or D , or $F \dots$; to the triangle CBA there will correspond P_n different decompositions; to DBA another group of decompositions,

¹⁷See a Memoir of Segner (*Novi Commentarii Acad. Petrop.*, vol. VII, p. 203). The author found equation (1) of M. Lamé; but formula (3) presents a much simpler solution. Formula (3) is no doubt due to Euler. It is pointed out without proof on page 14 of the volume cited above. The equivalence of equations (1) and (3) is not easy to establish. M. Terquem proposed this problem to me, achieving it with the help of some properties of factorials. I then communicated it to various geometers: none of them solved it; M. Lamé has been very successful: I am unaware of whether others before him have obtained such an elegant solution. J. LIOUVILLE

represented by the product P_3P_{n-1} ; to EBA the group P_4P_{n-2} ; to FBA , P_5P_{n-3} ; and so forth, until the triangle ZAB , which will belong to a final group P_n . Now, all these groups are completely distinct: their sum therefore gives P_{n+1} . Thus one has

$$P_{n+1} = P_n + P_3P_{n-1} + P_4P_{n-2} + P_5P_{n-3} + \cdots + P_{n-3}P_5 + P_{n-2}P_4 + P_{n-1}P_3 + P_n. \quad (1)$$

II.

Let $abcde \dots$ be a polygon of n sides. To each of the $n - 3$ diagonals, which end at one of the vertices a , there will correspond a group of decompositions, for which this diagonal will serve as the side of two adjacent triangles: to the first diagonal ac corresponds the group P_3P_{n-1} ; to the second ad corresponds P_4P_{n-2} ; to the third ae , P_5P_{n-3} , and so forth until the last ax , which will occur in the group P_3P_{n-1} . These groups are not totally different, because it is easy to see that some of the partial decompositions, belonging to one of them, is also found in the preceding ones. Moreover they do not include the partial decompositions of P_n in which none of the diagonals ending in a occurs.

But if one does the same for each of the other vertices of the polygon, and combines all the sums of the groups of these vertices, by their total sum

$$n(P_3P_{n-1} + P_4P_{n-2} + \cdots + P_{n-2}P_4 + P_{n-1}P_3)$$

one will be certain to include all the partial decompositions of P_n ; each of these is itself repeated therein a certain number of times.

Indeed, if one imagines an arbitrary such decomposition, it contains $n - 2$ triangles, having altogether $3n - 6$ sides; if one removes from this number the n sides of the polygon, and takes half of the remainder, which is $n - 3$, one will have the number of diagonals appearing in the given decomposition. Now, it is clear that this partial decomposition is repeated, in the preceding total sum, as many times as these $n - 3$ diagonals have ends, that is $2n - 6$ times: since each end is a vertex of the polygon, and in evaluating the groups of this vertex, the diagonal furnished a group including the particular partial decomposition under consideration.

Thus, since each of the partial decompositions of the total group P_n is repeated $2n - 6$ times in $n(P_3P_{n-1} + P_4P_{n-2} + \cdots + P_{n-2}P_4 + P_{n-1}P_3)$, one obtains P_n upon dividing this sum by $2n - 6$. Therefore one has

$$P_n = \frac{n(P_3P_{n-1} + P_4P_{n-2} + \cdots + P_{n-2}P_4 + P_{n-1}P_3)}{2n - 6}. \quad (2)$$

III.

The first formula (1) gives

$$P_3P_{n-1} + P_4P_{n-2} + \cdots + P_{n-2}P_4 + P_{n-1}P_3 = P_{n+1} - 2P_n,$$

and the second (2) gives

$$P_3P_{n-1} + P_4P_{n-2} + \cdots + P_{n-2}P_4 + P_{n-1}P_3 = \frac{2n - 6}{n}P_n;$$

so finally

$$P_{n+1} - 2P_n = \frac{2n - 6}{n}P_n,$$

or

$$P_{n+1} = \frac{4n - 6}{n}P_n. \quad (3)$$

This is what was to be proven.

Paris, 25 August, 1838

TASKS:

- 1.1 Draw a triangulation of $ABCDEF$ that is different from the triangulation in Figure 1. How many diagonals does your triangulation have? How many triangles does it divide $ABCDEF$ into?
- 1.2 Consider an n -sided polygon $A_1A_2 \dots A_n$. How many different possible diagonals does this polygon have? **Note:** We are talking about all possible diagonals, not just diagonals in a triangulation.
- 1.3 In Section II of Lamé's paper there is a statement that any triangulation of an n sided polygon has $n - 2$ triangles and $n - 3$ diagonals. Prove that these statements are true using mathematical induction.

Optimal Triangulation and Counting Triangulations

The *Optimal Polygon Triangulation* Problem is the following: Given an n -sided polygon $A_1A_2 \dots A_n$ and a weight $w_{i,j}$ for each diagonal A_iA_j , find a triangulation of the polygon such that the sum of the weights of the diagonals in the triangulations is minimized. A naïve way to solve the problem is to generate all possible triangulations one by one, calculate their weight (i.e. sum of weights of all the diagonals in the triangulation) and keep the best. The efficiency of this naïve method depends on the number of possible triangulations of a polygon with n sides. Thus, we would like to count how many different triangulations an n -sided polygon has. As mentioned in the introduction, the problem of counting the number of triangulations of an n -sided convex polygon was already being discussed in the mid eighteenth century by well-known figures in mathematics like Euler and Segner and an elegant solution was provided by Lamé in 1838.

TASKS:

- 2.1 Read Section I in Lamé's paper. Explain what Lamé is saying in your own words and derive the general recursive formula for P_{n+1} , i.e., formula (1) in Lamé's paper.
- 2.2 Use the recursive formula to calculate P_i for $i = 2, 3, 4, 5, 6, 7, 8$ by hand and display it as a table.
- 2.3 Draw all triangulations of polygons with n sides for $n = 4, 5$.
- 2.4 Lamé's recurrence relation in his section 1 for $n = 5$ yields

$$P_6 = P_5 + P_3P_4 + P_4P_3 + P_5.$$

Draw all triangulations of a 6-sided polygon classified into groups according to the idea of the recurrence relation, i.e., the triangulations should be classified into four groups with each group corresponding to a term on the right-hand side of the recurrence above.

- 2.5 Write a simple recursive function $SRCAT(n)$ (for "Simple Recurrence CATalan") in Java that given an input n calculates P_n using the recurrence relation (1) in Lamé's paper directly.
- 2.6 Write another Java program that repeatedly uses $SRCAT$ to calculate P_i for $i = 3, 4, 5 \dots$. Restrict the total time your program uses to 10 minutes. What is the largest value N_0 of i for which your program calculates P_i ? Print out a table with i and the time required in seconds by $SRCAT$ to calculate each of the P_i values. Your table should have a row for each $i = 3, 4, 5 \dots, N_0$.
- 2.7 From your calculations you may observe that it seems that for all n , if $n \geq 3$ then $P_{n+1} \geq 2 * P_n$. Give a simple mathematical argument that establishes the truth of this statement.
- 2.8 Prove that for all n , if $n \geq 3$ then $P_n \geq 2^n / 8$.
- 2.9 What does this tell you about the efficiency of the naïve algorithm for solving the optimal polygon triangulation problem?
- 2.10 Write a Java program that repeatedly uses the recurrence given in formula (1) in Lamé's paper to calculate P_i for $i = 3, 4, 5 \dots$ but that stores the computed values in an array systematically and uses them as needed. Restrict the total time your program uses to 10 minutes. What is the largest value M_0 of i for which your program calculates P_i ?
- 2.11 Extend your program to print out a table of values of i and time required in seconds to compute P_i for $i = 3, 4, \dots, M_0$.
- 2.12 Graph the tables obtained in 2.6 and 2.11. Analyse these graphs and write down your observations.

Lamé's Method for deriving a formula for P_n

Section II of Lamé's paper gives an alternative way of counting triangulations of a polygon. Read this section carefully.

TASKS:

Consider a 6-sided polygon $ABCDEF$.

3.1 Draw all triangulations of the polygon where:

- AC is one of the diagonals in the triangulation.
- AD is one of the diagonals in the triangulation.
- AE is one of the diagonals in the triangulation.

How many total triangulations did you draw?

3.2 Repeat the same with vertex B as the "special" vertex, i.e., draw all triangulations where:

- BD is one of the diagonals in the triangulation.
- BE is one of the diagonals in the triangulation.
- BF is one of the diagonals in the triangulation.

How many total triangulations did you draw?

- 3.3 Do the same with vertices C, D, E, F being “special”.
- 3.4 Consider the triangulation of $ABCDEF$ in figure 1 (of section 1). How many times is that triangulation repeated in all the triangulations that you drew for $ABCDEF$ in this section? Identify the diagonals in whose group it was drawn.
- 3.5 Do the same for the different triangulations of $ABCDEF$ that you drew in section 1.
- 3.6 What would you guess about the number of times any triangulation of $ABCDEF$ is repeated? Argue why your guess is correct.
- 3.7 Consider the n -sided polygon $A_1A_2 \dots A_n$. Let P_i denote the number of different triangulations of a polygon with i sides.
- Calculate, in terms of P_i 's, the number of triangulations of this polygon that have A_1A_3 as a diagonal, that have A_1A_4 as a diagonal, that have A_1A_j as a diagonal.
 - Consider drawing triangulations treating A_1 as the “special” vertex. That is, draw all triangulations where A_1A_3 is a diagonal, then draw all triangulations where A_1A_4 is a diagonal, etc. all the way up to where A_1A_{n-1} is a diagonal. What is the number of triangulations you draw (in terms of P_i 's) when A_1 is treated as a special vertex?
 - Suppose we repeat the above process with with another vertex (say A_2) being the special vertex instead of A_1 . What can you say about the number of triangulations drawn as compared to the number of triangulations drawn when A_1 was chosen as the special vertex? Explain in your own words why this is true.
 - Consider doing what you did for A_1 in (b) successively for each vertex. That is, enumerate all triangulations treating A_1 as a special vertex, treating A_2 as a special vertex, ... treating A_n as a special vertex. Now consider the specific triangulation of $A_1, A_2 \dots A_n$ obtained by drawing the diagonals $A_1A_3, A_1A_4, \dots A_1A_{n-1}$. How many times is this triangulation enumerated? What about the triangulation obtained by drawing the diagonals $A_1A_4, A_1A_5 \dots A_1A_{n-2}$ and the two diagonals $A_2A_4, A_{n-2}A_n$? Justify your answer.
 - What is your guess as to how many times any specific triangulation is enumerated? Explain in your own words why this is the case.
- 3.8 Combine (b) and (e) to derive the formula (2) in Lamé’s paper. Explain in your own words how this formula is obtained.
- 3.9 Combine formulas (1) and (2) in Lamé’s paper to obtain the formula (3) in Lamé’s paper. Show all the steps in your calculation. Explain why this formula is better for calculating P_n .
- 3.10 Using formula (3) in Lamé’s paper, show that $P_{n+2} = \frac{1}{n+1} \binom{2n}{n}$ where $\binom{2n}{n} = \frac{(2n)!}{n!n!}$.
- 3.11 Write a simple recursive function $ASRCAT(n)$ (for “Another Simple Recurvisc CATalan”) in Java that given an input n calculates P_n using the recurrence relation (3) in Lamé’s paper directly.
- 3.12 Write another Java program that repeatedly uses $ASRCAT$ to calculate P_i for $i = 3, 4, 5 \dots$. Restrict the total time your program uses to 10 minutes. What is the largest value L_0 of i for which your program calculates P_i ?

- 3.13 Extend your program to print out a table of values of i and time required in seconds by $ASRCAT(n)$ to compute each of the P_i values for $i = 3, 4, \dots, L_0$. Your table should have a row for each $i = 3, 4, \dots, L_0$.
- 3.14 Write a better Java program using the ideas from dynamic programming (“store and re-use”) that repeatedly calculates P_i for $i = 3, 4, \dots$. Restrict the total time your program uses to 10 minutes. What is the largest value L_1 of i for which your program calculates P_i .
- 3.15 Extend your program to print out a table of values of i and the time required in seconds to calculate each of the P_i values. Your table should have a row for each $i = 3, 4, \dots, L_1$.
- 3.16 Graph the tables obtained in 3.13 and 3.15. Analyse all four graphs obtained and write down your observations. How do the results for the second two programs compare with your first two programs? How fast does the running time of the last two programs grow?
- 3.17 Discuss how the choice of Lamé’s formulas (1) or (3), or using dynamic versus naïve recursive programming influences the effectiveness of computation.

8 Two-Way Deterministic Finite Automata

Hing Leung¹⁸

8.1 Introduction

In 1943, McCulloch and Pitts [51] published a pioneering work on a model for studying the behavior of the nervous systems. Following up on the ideas of McCulloch and Pitts, Kleene [44] wrote the first paper on finite automata, which proved a theorem that we now call Kleene's theorem. A finite automaton can be considered as the simplest machine model in that the machine has a finite memory; that is, the memory size is independent of the input length. In a 1959 paper [55], Michael Rabin and Dana Scott presented a comprehensive study on the theory of finite automata, for which they received the Turing award in 1976, the highest award in computer science. The citation for the Turing Award states that the award was granted:

For their joint paper "Finite Automata and Their Decision Problem," which introduced the idea of nondeterministic machines, which has proved to be an enormously valuable concept. Their (Scott & Rabin) classic paper has been a continuous source of inspiration for subsequent work in this field.

In this project, we will not discuss nondeterministic machines. We consider two-way finite automata which is another concept that was introduced in the seminal paper by Rabin and Scott [55].

In an early stage, the theory of finite automata has been developed as a mathematical theory for sequential circuits. A sequential circuit maintains a current state from a finite number of possible states. The circuit logic (which is a finite state control) decides the new state based on the current state of the circuit and the given input symbol. Once an input symbol is processed, the circuit will not be able to read it again.

A main application of finite automata is text processing. In compiler design, finite automata are used to capture the logic of lexical analysis. Other applications include string matching, natural language processing, text compression, etc.

A one-way deterministic finite automata (DFA) is defined as the mathematical model of a machine with a finite amount of memory where the input is processed once from left to right. After an input has been read, the DFA decides whether the input is accepted or rejected.

8.1.1 Two-Way Finite Automata

A Turing machine is an abstract mathematical model of a computer. In a previous project, we have already studied Turing machines. Recall that a Turing machine can move back and forth in the working tape while reading and/or writing.

In fact, there are several variants of Turing machines in the literature. Our variant of a Turing machine consists of a finite-state control (also called the program), a read-only input tape where the input is given and a working tape (also called memory) where computations are performed.

In computing an answer, intermediate results are computed and kept in the working tape so that it can be referenced later for further computations.

Unlike a real computer, a Turing machine has no limit to the amount of memory that it can use. As the input becomes more complicated (say longer or of larger value), a Turing machine may use more memory to compute.

¹⁸Department of Computer Science, New Mexico State University, Las Cruces, NM 88003; hleung@cs.nmsu.edu.

Consider the problem of primality testing. That is, given a positive integer, we want to test if it is a prime. As the given integer becomes larger, we need to use more memory to test for primality. We say that the primality testing problem requires an unbounded amount of memory to solve.

However, there are other problems that may require only a finite amount of memory to solve. Consider the problem of computing the parity of a binary sequence. The parity is the number of occurrences of 1's in the binary sequence modulo 2. (Note: The concept defined is not the same as whether the binary sequence, when considered as a binary number, is odd or even.) One can keep reading the binary sequence bit by bit, and maintain the intermediate result of the current parity in the working tape. That is, the memory usage is one bit (thus, finite) in the working tape no matter how long the binary sequence is. Note that the length of the input is not counted in the memory use. We assume that the input tape allows only read operations. The input tape cannot be over-written.

One may wonder what computational problems can be solved using finite memory Turing machines. Another interesting question is whether we can simplify the model of Turing machines using finite memory.

Since the memory usage is finite (independent of the size of the input), we can incorporate¹⁹ the finite memory into the finite state control so that the Turing machine no longer uses a working tape. The resulting model is called the two-way deterministic finite automaton²⁰ (2DFA) in that it consists of a finite state control and a read-only input tape that allows an input to be read back and forth. As in the case of DFA, the 2DFA decides whether a given input is accepted or rejected.

One can see that a 2DFA can be equivalently defined as a read only Turing machine (without the ability to write symbols on the tape).

8.1.2 DFA and 2DFA

In comparison, DFA and 2DFA differ in that an input can be read only once from left to right by a DFA, whereas a 2DFA can read the input back and forth with no limit on how many times an input symbol can be read.

In computer science, DFA has been studied heavily, since many problems are found to be solvable by DFA. Almost all textbooks in discrete mathematics discuss the DFA model. However, no textbooks in discrete mathematics discuss the model of 2DFA.

Are 2DFA unrealistic? As we have discussed before, 2DFA is a very interesting model in that it captures/solves problems that are solvable by a computer using finite memory. In fact, in most real life computing tasks performed by a computer, the input has been saved into the computer's hard disk, or is kept in a CD-ROM or DVD-ROM. Thus, there is no reason why a program cannot read the input more than once using two-way processing. So, 2DFA is indeed a very meaningful and arguably more realistic model than DFA.

How does DFA compare to 2DFA? Clearly, DFA is a restricted version of 2DFA. Therefore, 2DFA can solve any problems that are solvable by DFA. Next, are there problems that can be solved by 2DFA but cannot be solved by DFA?

This question is one of many fundamental questions answered in Rabin and Scott's paper [55] on the theory of finite automata. It is proved that 2DFA can be simulated by DFA. That is, whatever problems can be solved by 2DFA can also be solved by DFA.

One may wonder why we have to study 2DFA given that DFA, being a simpler model, can do the

¹⁹The technique involved, which we omit, is not really direct or immediate. But it is not difficult either.

²⁰There is another machine model called two-way nondeterministic finite automata. But we are not going to discuss nondeterministic automata in this project.

same job. Are there advantages of 2DFA over DFA? It turns out that 2DFA can be significantly²¹ simpler in design for solving the same problem than DFA. In this project, we are going to illustrate the advantages of 2DFA using a number of examples.

It is difficult to explain why most textbooks²² in automata theory are not covering 2DFA. One possible reason is that the equivalence proof (that 2DFA can be simulated by the simpler model DFA) given by Rabin and Scott is too difficult.

John C. Shepherdson [62] was able to offer another proof of this important result. It is a very clean proof that we want to present in this project. Instead of going through the proof steps, we emphasize the technique for constructing²³ a DFA from a 2DFA.

Shepherdson is a retired professor in mathematics at the University of Bristol, Great Britain. He published many papers in symbolic logic, computability, logic programming and fuzzy logic.

In fact, Rabin and Scott referred the readers to Shepherdson's proof in their pioneering paper, and decided to give only a sketch of their proof of the equivalence result. Following is an excerpt from Rabin and Scott's paper about Shepherdson's proof:

The result, with its original proof, was presented to the Summer Institute of Symbolic Logic in 1957 at Cornell University. Subsequently J. C. Shepherdson communicated to us a very elegant proof which also appears in this Journal. In view of this we confine ourselves here to sketching the main ideas of our proof.

We hope that the students using this project will find Shepherdson's construction not tricky, but that it indeed makes a lot of sense and is the logical way to go for proving the equivalence result. Moreover, students are required to read from the verbal explanations given by Shepherdson, and derive from it computer programs for solving problems in this project.

8.2 Project

We assume that students are familiar with the concept and formal definition of DFA.

Following are the definitions of one-way and two-way deterministic finite automata (adapted) from the paper by Shepherdson [62].

Definition 1. A one-way finite automaton (DFA) over a finite alphabet²⁴ Σ is a system $A = (Q, \delta, q_0, F)$, where Q is a finite non-empty set (the internal states of A), δ is a function from $Q \times \Sigma$ into Q (the table of moves of A), q_0 is an element of Q (the initial state of A), and F is a subset of Q (the designated final states of A). The class $T(A)$ of tapes accepted²⁵ by A is the class of all finite sequences $\sigma_1, \dots, \sigma_n$ of symbols of Σ for which the sequence q_0 (initial state), q_1, \dots, q_n defined by $q_{i+1} = \delta(q_i, \sigma_{i+1})$ ($i = 0, \dots, n-1$) satisfies $q_n \in F$. A set of tapes is said to be definable by a one-way automaton if it is equal to $T(A)$ for some A .

Definition 2. A two-way finite automaton (2DFA) over Σ is a system $A = (Q, \delta, q_0, F)$ as in Definition 1 with the difference that now δ is a function from $Q \times \Sigma$ into $Q \times D$ where

²¹Technically (and, more accurately), we say that 2DFA can be *exponentially* more succinct in descriptonal size than DFA for solving the same problems.

²²Two textbooks ([35], [46]) cover 2DFA. One [35] follows the approach by Rabin and Scott, and the other [46] follows Shepherdson's ideas.

²³In contrast, it is difficult to construct mechanically a DFA from a 2DFA based on the proof of Rabin and Scott.

²⁴ Σ is the set of input symbols. In the examples considered in this project, the input is always a binary sequence with $\Sigma = \{0, 1\}$.

²⁵The word 'accepted' is used in a technical sense. It has the same meaning as the word 'defined'.

$D = \{L, S, R\}$. A operates as follows: It starts on the leftmost square of the given tape in state q_0 . When its internal state is q and it scans the symbol σ , then if $\delta(q, \sigma) = (q', d)$ it goes into the new state q' and moves one square to the left, stays where it is, or moves one square to the right according as $d = L, S$, or R . The class $T(A)$ of tapes accepted by A is the class of those tapes t such that A eventually moves off the right-hand edge of t in a state belonging to F .

We want to design a 2DFA over $\Sigma = \{0, 1\}$ that accepts tapes containing two 1's separated by 4 symbols in between them. That is, the 2DFA accepts finite sequences $\sigma_1 \dots \sigma_n$ of symbols of Σ such that $\sigma_i = \sigma_{i+5} = 1$ for some $i \in \{1, \dots, n - 5\}$. For example, the sequence 001010101100 should be accepted as the 5th and 10th symbols are both 1.

The following 10-state 2DFA A_1 , where q_0 is the starting state and q_9 is the only final state, systematically checks every possible 6-symbol subsequence to see if it begins and ends with 1's.

| current state | symbol | new state | go to |
|---------------|--------|-----------|-------|
| q_0 | 0 | q_0 | R |
| q_0 | 1 | q_1 | R |
| q_1 | 0 or 1 | q_2 | R |
| q_2 | 0 or 1 | q_3 | R |
| q_3 | 0 or 1 | q_4 | R |
| q_4 | 0 or 1 | q_5 | R |
| q_5 | 0 | q_6 | L |
| q_5 | 1 | q_9 | R |
| q_6 | 0 or 1 | q_7 | L |
| q_7 | 0 or 1 | q_8 | L |
| q_8 | 0 or 1 | q_0 | L |
| q_9 | 0 or 1 | q_9 | R |

1.2.1 Demonstrate the steps performed by A_1 in accepting the tape 11001010.

1.2.2 Demonstrate the steps performed by A_1 in processing the tape 11001001. Conclude that the tape is not accepted by A_1 .

Another example 2DFA A_2 (taken from Example 2.14 of the textbook by Hopcroft and Ullman [35]) is given as follows, where q_0 is the starting state and $F = \{q_0, q_1, q_2\}$:

| current state | symbol | new state | go to |
|---------------|--------|-----------|-------|
| q_0 | 0 | q_0 | R |
| q_0 | 1 | q_1 | R |
| q_1 | 0 | q_1 | R |
| q_1 | 1 | q_2 | L |
| q_2 | 0 | q_0 | R |
| q_2 | 1 | q_2 | L |

1.2.3 Demonstrate that the input 101001 is accepted by A_2 .

1.2.4 Demonstrate that the input 10111 is not accepted by A_2 . Indeed, A_2 will run into an infinite loop.

Given the description of a 2DFA, Shepherdson explained how to derive the description of an equivalent DFA that accepts the same set of inputs. The following is an excerpt (modified slightly) from Shepherdson [62] describing the construction:

The only way an initial portion t of the input tape can influence the future behaviour of the two-way machine A when A is not actually scanning this portion of the tape is via the state

transitions of A which it causes. The external effect of t is thus completely determined by the transition function, or “table”, τ_t which gives (in addition to the state in which the machine originally exits from t), for each state q of A in which A might re-enter t , the corresponding state q' which A would be in when it left t again. This is all the information the machine can ever get about t however many times it comes back to refer to t ; so it is all the machine needs to remember about t . But there are only a finite number of different such transition tables (since the number of states of A is finite), so the machine has no need to use the input tape to supplement its own internal memory; a one-way machine \bar{A} with a sufficiently large number of internal states could store the whole transition table τ_t of t as it moved forward, and would then have no need to reverse and refer back to t later. If we think of the different states which A could be in when it re-entered t as the different questions A could ask about t , and the corresponding states A would be in when it subsequently left A again, as the answers, then we can state the result more crudely and succinctly thus: A machine can spare itself the necessity of coming back to refer to a piece of tape t again, if, before it leaves t , it thinks of all the possible questions it might later come back and ask about t , answers these questions now and carries the table of question-answer combinations forward along the tape with it, altering the answers where necessary as it goes along.

To summarize Shepherdson’s idea, we need to maintain for each prefix t two pieces of information: (1) the state in which the machine exits from t (when starting at q_0 in the leftmost square of the input), and (2) the external effect τ_t .

Let us refer to the processing of the input tape 101001 by A_2 . Consider the first symbol of the input which is a 1. That is, let $t = 1$.

To answer the first question, we observe that when A_2 begins with the initial state q_0 at the symbol 1, it will exit t to its right at state q_1 .

Next, we summarize the external effect τ_t where $t = 1$ by answering the following questions:

1. Suppose later in the processing of the input, the first symbol 1 is revisited again by a left move from the right. What will be the effect if it is revisited with a state q_0 ?
2. Suppose later in the processing of the input, the first symbol 1 is revisited again by a left move from the right. What will be the effect if it is visited with a state q_1 ?
3. Suppose later in the processing of the input, the first symbol 1 is revisited again by a left move from the right. What will be the effect if it is visited with a state q_2 ?

Verify that the answers to the 3 questions are: (1) move right at state q_1 , (2) move left at state q_2 and (3) move left at state q_2 . In short the external effect can be succinctly summarized as a 3-tuple $[(q_1, R), (q_2, L), (q_2, L)]$.

We combine the two pieces of information computed in a 4-tuple $[(q_1, R), (q_1, R), (q_2, L), (q_2, L)]$. Let us call this 4-tuple the effect of t where $t = 1$. Note that the first entry is the answer to the first question, whereas the next 3 entries are the external effect of t . That is, the effect of t is the total effect, which is more than just the external effect.

Next, given the effect computed for the first symbol 1, we want to compute the effect of the first two symbols 10 of the input 101001.

From the effect $[(q_1, R), (q_1, R), (q_2, L), (q_2, L)]$ of the symbol 1, we know that the machine exits 1 to its right with the state q_1 . Thus, the machine is at state q_1 when visiting the second symbol 0. According to the transition table, the machine will again move to the right of the second symbol at state q_1 .

To compute the external effect of $t = 10$, we have to answer the following questions:

1. Suppose later in the processing of the input, the second symbol 0 is revisited again by a left move from the right. What will be the effect if it is revisited with a state q_0 ?
2. Suppose later in the processing of the input, the second symbol 0 is revisited again by a left move from the right. What will be the effect if it is visited with a state q_1 ?
3. Suppose later in the processing of the input, the second symbol 0 is revisited again by a left move from the right. What will be the effect if it is visited with a state q_2 ?

The answers are as follows:

1. When the second symbol 0 is revisited again with a state q_0 , the machine according to the transition table will move to the right with the state q_0 .
2. When the second symbol 0 is revisited again with a state q_1 , the machine according to the transition table will move to the right with the state q_1 .
3. When the second symbol 0 is revisited again with a state q_2 , the machine according to the transition table will move to the right with the state q_0 .

Therefore, the effect of $t = 10$ is summarized in the 4-tuple $[(q_1, R), (q_0, R), (q_1, R), (q_0, R)]$.

Note that in the above computations, we do not make use of the external effect computed for the first prefix 1 to compute the effect for the prefix 10. This is not usually the case. In general, the external effect for t is needed in computing the new effect when t is extended by one more symbol.

1.2.5 Next, with the answers $[(q_1, R), (q_0, R), (q_1, R), (q_0, R)]$ for the effect for 10, compute the effect when the input is extended by the third symbol 1.

From the effect for 10, we know that the machine arrives at the third symbol 1 at state q_1 . According to the transition table, the machine will move to the left to the 2nd symbol at state q_2 . Next, consulting the last entry of the effect for 10, we know the machine will leave 10 to its right at state q_0 . Thus, the machine revisits the third symbol 1 at state q_0 . Again, from the transition table, the machine moves to the right at state q_1 . Verify that the external effect of 101 is $[(q_1, R), (q_1, R), (q_1, R)]$. That is, the effect for 101 is $[(q_1, R), (q_1, R), (q_1, R), (q_1, R)]$.

In answering question 1.2.5, you should provide the steps involved in computing the external effect for 101.

Note that given the effect for t and a new symbol 1, we can compute the effect for $t' = t1$ by referring to the transition table for the machine. There is no need to know what t is. Only the effect for t is needed in computing the new effect for t' .

1.2.6 Repeatedly, compute the effects by extending the current input 101 with symbols 0, 0, 1. From the effect computed for 101001, conclude that the input is accepted. Recall that 101001 was shown in 1.2.3 to be accepted by A_2 .

1.2.7 Repeat the whole exercise with the input 10111 as in 1.2.4. Conclude that the input 10111 is not accepted.

To summarize, in simulating A_2 by a DFA, we need only remember the effect of the sequence of input symbols that has been processed so far.

1.2.8 How many different effects can there be in simulating A_2 by a DFA? How many states are needed for a DFA to accept the same set of inputs as A_2 ? Note that there are only finitely many different effects even though we have an infinite number of inputs of finite lengths.

1.2.9 Applying Shepherdson's method to A_1 , how many states are there in the DFA constructed?

The work involved is very tedious. It is impossible to do it by hand. You should write a program to perform the computation.

Note that the smallest DFA²⁶ accepting the same set of inputs as A_1 has 33 states.

In computer programming, we can detect if the end-of-input has been reached. For example, in C programming, we write

```
while ((input=getChar()) != EOF)
```

Thus, it is very reasonable to extend the 2DFA model so that the machine can detect the two ends of an input tape.

Read the following excerpt (modified slightly) from Shepherdson's paper [62] regarding extending the 2DFA model by providing each input tape with special marker symbols b , e at the beginning and end of the input. Let us call this new model 2DFA-with-endmarkers.

At first sight it would appear that with a two-way automaton more general sets of tapes could be defined if all tapes were provided with special marker symbols b , e (not in Σ) at the beginning and end, respectively. For then it would be possible, e.g., to design a two-way machine which, under certain conditions, would go back to the beginning of a tape, checking for a certain property, and then return again. This would appear to be impossible for an unmarked tape because of the danger of inadvertently going off the left-hand edge of the tape in the middle of the computation. In fact, the machine has no way of telling when it has returned to the first symbol of the tape. However, the previous construction result implies that this is not so; that the addition of markers does not make any further classes of tapes definable by two-way automata. For if the set $\{b\}U\{e\}$ (of all tapes of the form bte for $t \in U$) is definable by a two-way automaton then it is definable by a one-way automaton; and it is easy to prove that $\{b\}U\{e\}$ is definable by a one-way automaton if and only if U is.

We assume that a 2DFA-with-endmarkers starts on the left endmarker b at the initial state.

Suppose we want to design a 2DFA-with-endmarkers over $\Sigma = \{0, 1\}$ to accept input tapes that has a symbol 1 in the sixth position from the right end. For example, the input 10101011 has 8 symbols with the third symbol being a 1, which is the sixth position from the last (eighth) position. The following 2DFA-with-endmarkers A_3 , where q_0 is the starting state and q_9 is the accepting state, accepts the input tapes described. Observe that an accepted input must begin with b and end with e . Furthermore, b and e do not appear in other positions of the string accepted.

| current state | symbol | new state | go to |
|---------------|--------|-----------|-------|
| q_0 | b | q_1 | R |
| q_1 | 0 or 1 | q_1 | R |
| q_1 | e | q_2 | L |
| q_2 | 0 or 1 | q_3 | L |
| q_3 | 0 or 1 | q_4 | L |
| q_4 | 0 or 1 | q_5 | L |
| q_5 | 0 or 1 | q_6 | L |
| q_6 | 0 or 1 | q_7 | L |
| q_7 | 1 | q_8 | R |
| q_8 | 0 or 1 | q_8 | R |
| q_8 | e | q_9 | R |

1.2.10 Following the discussion by Shepherdson and based on the definition of A_3 , explain the construction of a DFA equivalent to A_3 accepting the same set of input tapes where the inputs are

²⁶There is a mechanical method for computing the number of states of a smallest DFA.

delimited by b and e . How many states does the DFA have? Next, modify the DFA constructed to accept inputs with the endmarkers b and e removed. What is the change in the number of states of the DFA?

As in 1.2.9, you should write a program to perform the computation.

Note that it can be shown that the smallest DFA accepting the same set of inputs (without the endmarkers b and e) as A_3 has 64 states.

Consider another 2DFA-with-endmarkers A_4 , where q_0 is the starting state and q_9 is the accepting state, as defined below.

| current state | symbol | new state | go to |
|---------------|--------|-----------|-----------------------|
| q_0 | b | q_1 | R |
| q_1 | 0 or 1 | q_1 | R |
| q_1 | e | q_2 | L |
| q_i | 0 | q_{i+1} | $L \quad i = 2, 3, 4$ |
| q_i | 1 | q_i | $L \quad i = 2, 3, 4$ |
| q_5 | 0 | q_7 | L |
| q_5 | 1 | q_6 | L |
| q_6 | 0 | q_6 | L |
| q_6 | 1 | q_7 | L |
| q_7 | 0 | q_7 | L |
| q_7 | 1 | q_5 | L |
| q_7 | b | q_8 | R |
| q_8 | 0 or 1 | q_8 | R |
| q_8 | e | q_9 | R |

1.2.11 Re-do part 1.2.10 for A_4 .

Note that it can be shown that the smallest DFA accepting the same set of inputs (without the endmarkers b and e) as A_4 has 64 states.

9 Church's Thesis

Guram Bezhanishvili^{27 28}

Introduction

In this project we will learn about primitive recursive and general recursive functions. We will also learn about Turing computable functions, and will discuss why the class of general recursive functions coincides with the class of Turing computable functions. We will introduce the effectively calculable functions, and the ideas behind Alonzo Church's (1903–1995) proposal to identify the class of effectively calculable functions with the class of general recursive functions, known as “Church's thesis.” We will analyze Kurt Gödel's (1906–1978) initial rejection of Church's thesis, together with the work of Alan Turing's (1912–1954) that finally convinced Gödel of the validity of Church's thesis. We will learn much of this by studying and working with primary historical sources by Gödel, Stephen Cole Kleene (1909–1994), and Turing.

We begin by asking the following question: What does it mean for a function f to be *effectively calculable*? Obviously if we can find an algorithm to calculate f , then f is effectively calculable. For example, the famous Euclidean algorithm tells us that the binary function producing the greatest common divisor of two integers is effectively calculable. But what if we can not find an algorithm that calculates f ? The reason could be that there is no algorithm calculating f ; or it could be that f is effectively calculable but we were not successful in finding an algorithm. Thus, it is evident that we need better means to identify effectively calculable functions.

The problem of identifying the effectively calculable functions (of natural numbers) was at the center stage of mathematical research in the twenties and thirties of the twentieth century. In the early thirties at Princeton, Church and his two gifted students Kleene and John Barkley Rosser (1907–1989) were developing the theory of λ -definable functions. Church proposed to identify the effectively calculable functions with the λ -definable functions. Here is Kleene's description of these events, taken from page 59 of [45]:

The concept of λ -definability existed full-fledged by the fall of 1933 and was circulating among the logicians at Princeton. Church had been speculating, and finally definitely proposed, that the λ -definable functions are all the effectively calculable functions—what he published in [9], and which I in [43] Chapter XII (or almost in [42]) called “Church's thesis”.

When Church proposed this thesis, I sat down to disprove it by diagonalizing out of the class of the λ -definable functions. But, quickly realizing that the diagonalization cannot be done effectively, I became overnight a supporter of the thesis.

Though Kleene became an “overnight” supporter of the thesis, it was a different story with Gödel. Gödel arrived at Princeton in the fall of 1933. Church proposed his thesis to Gödel early in 1934, but, according to a November 29, 1935, letter from Church to Kleene, Gödel regarded it “as thoroughly unsatisfactory.” Instead, in his lectures during the spring of 1934 at Princeton [24], Gödel generalized the notion of *primitive recursive* functions, which was introduced by him in his epoch-making paper on undecidable propositions [23].²⁹ He did this by modifying a suggestion made by Jacques Herbrand (1908–1931) in a 1931 letter, to obtain the notion of general recursive

²⁷Department of Mathematical Sciences, New Mexico State University, Las Cruces, NM 88003; gbezhani@nmsu.edu.

²⁸With thanks to Joel Lucero-Bryan.

²⁹It has to be noted that what we now call primitive recursive functions Gödel simply called recursive. The term primitive recursive was introduced by Kleene in [39].

functions (also known as the Herbrand-Gödel general recursive functions). Below we give an excerpt from the abovementioned letter from Church to Kleene (taken from [14]) that gives an account of his discussion of effective calculability with Gödel:

In regard to Gödel and the notions of recursiveness and effective calculability, the history is the following. In discussion [sic] with him the notion of lambda-definability, it developed that there was no good definition of effective calculability. My proposal that lambda-definability be taken as a definition of it he regarded as thoroughly unsatisfactory. I replied that if he would propose any definition of effective calculability which seemed even partially satisfactory I would undertake to prove that it was included in lambda-definability. His only idea at the time was that it might be possible, in terms of effective calculability as an undefined notion, to state a set of axioms which would embody the generally accepted properties of this notion, and to do something on that basis. Evidently it occurred to him later that Herbrand's definition of recursiveness, which has no regard to effective calculability, could be modified in the direction of effective calculability, and he made this proposal in his lectures. At that time he did specifically raise the question of the connection between recursiveness in this new sense and effective calculability, but said he did not think that the two ideas could be satisfactorily identified "except heuristically."

It appears that Gödel's rejection of λ -definability as a possible "definition" of effective calculability was the main reason behind Church's announcement of his thesis in terms of general recursive functions [8]. Church made his announcement at a meeting of the American Mathematical Society in New York City on April 19, 1935. Below is an excerpt from his abstract:

Following a suggestion of Herbrand, but modifying it in an important respect, Gödel has proposed (in a set of lectures at Princeton, N. J., 1934) a definition of the term *recursive function*, in a very general sense. In this paper a definition of *recursive function of positive integers* which is essentially Gödel's is adopted. And it is maintained that the notion of an effectively calculable function of positive integers should be identified with that of a recursive function, since other plausible definitions of effective calculability turn out to yield notions which are either equivalent to or weaker than recursiveness.

Note that in the abstract Church relegated λ -definability to "other plausible definitions of effective calculability" that were "either equivalent to or weaker than recursiveness," which indicates that, at the time, Church was not yet certain whether λ -definability was equivalent to general recursiveness. Kleene filled in this gap in [40] by showing that these two notions were indeed equivalent. Thus, in the full version of his paper [9], Church was already fully aware that the two notions of general recursiveness and λ -definability coincide.

Kleene's theorem that identified general recursive and λ -definable functions, together with Kleene's famous Normal Form Theorem³⁰ were beginning to convince Gödel of the validity of Church's thesis. However, it wasn't until the work of Turing that he finally accepted Church's thesis.

Turing's famous paper [66] appeared in 1936 (a correction to it was published in 1937). Turing introduced what we now call *Turing machines*, and defined a function to be *computable* if it can be computed on a Turing machine. His work was entirely independent of the related research being done in Princeton. According to [45], page 61:

³⁰The Normal Form Theorem appeared in [39] and considerably simplified the notion of general recursive functions.

Turing learned of the work at Princeton on λ -definability and general recursiveness just as he was ready to send off his manuscript, to which he then added an appendix outlining a proof of the equivalence of his computability to λ -definability. In [67] he gave a proof of the equivalence in detail.

Thus, Turing introduced his notion of computability in 1936–1937 and, using some of the results of Kleene, showed that the three notions of Turing computable, general recursive, and λ -definable functions coincide.

On page 72 of Gödel’s “postscriptum” to his 1934 lecture notes which he prepared in 1964 for [13], Gödel states:

Turing’s work gives an analysis of the concept of “mechanical procedure” (alias “algorithm” or “computation procedure” or “finite combinatorial procedure”). This concept is shown to be equivalent with that of a “Turing machine”.

Thus, Gödel made it clear that, in his view, Turing’s work was of fundamental importance in establishing the validity of Church’s thesis. In particular, it influenced Gödel to accept it.

Our account of effective calculability would be incomplete if we did not mention that around the same time and independently of Turing, but not of the work in Princeton, Emil Leon Post (1897–1954) formulated yet another equivalent version of computability [54]. However, his work was less detailed than Turing’s.

Lastly we mention that *partial* recursive functions were introduced by Kleene in [41]. In [43] he also generalized the notion of Turing computable functions to partial functions and showed that a partial function is Turing computable if, and only if, it is partial recursive. The importance of his work is underlined in footnote 20 of [14] quoted below:

It is difficult for those who have learned about recursive functions via a treatment that emphasized partial functions from the outset to realize just how important Kleene’s contribution was. Thus Rogers’ excellent and influential treatise [57], p. 12, contains an historical account which gives the impression that the subject had been formulated in terms of partial functions from the beginning.

To summarize, in the mid-thirties there were several versions proposed to formalize the intuitive concept of effective calculability. These were λ -definability (Church), general recursiveness (Gödel), Turing computability (Turing), and Post computability (Post). All these concepts were seen to be equivalent to each other, thus producing evidence for general acceptance of Church’s thesis.

This project will only scratch the surface of the subject. Instead of working with partial functions, we will restrict our attention to total functions. We will learn about primitive recursive and general recursive functions from the work of Gödel and Kleene. We will also learn about Turing machines, Turing computable sequences (of natural numbers), and Turing computable real numbers (in the interval $[0, 1]$) from the work of Turing and Kleene. We will examine Turing’s and Kleene’s definitions of computable functions (of natural numbers), and show that every general recursive function is computable on a Turing machine. The fact that every Turing computable function is general recursive requires the relatively advanced technique of using Gödel numbers and will not be addressed in this project. Instead we refer the interested reader to either [23], [24] or [43] for the definition of Gödel numbers, and [67] or [43] for the proof that every Turing computable function is general recursive.

The four sources by Gödel, Turing, and Kleene that will be used in the project are [24], [66], [42], and [43]. Edited versions of the first three with forewords have been reprinted in [13].

Part one. Primitive recursive functions

Note: For the reading in part I we will use excerpts of Gödel's 1934 lecture notes [24] reprinted in Davis [13] and edited by Gödel himself. We decided to choose the reprinted version over the original since its definition of rule (1) is more convenient for our purposes.

1.(a) Read carefully the following excerpt of Gödel's 1934 lecture notes [24] reprinted in Davis [13].

The function $\phi(x_1, \dots, x_n)$ shall be *compound* with respect to $\psi(x_1, \dots, x_m)$ and $\chi_i(x_1, \dots, x_n)$ ($i = 1, \dots, m$) if, for all natural numbers x_1, \dots, x_n ,

$$(1) \quad \phi(x_1, \dots, x_n) = \psi(\chi_1(x_1, \dots, x_n), \dots, \chi_m(x_1, \dots, x_n)).$$

$\phi(x_1, \dots, x_n)$ shall be said to be *recursive* with respect to $\psi(x_1, \dots, x_{n-1})$ and $\chi(x_1, \dots, x_{n+1})$ if, for all natural numbers k, x_2, \dots, x_n ,

$$(2) \quad \begin{aligned} \phi(0, x_2, \dots, x_n) &= \psi(x_2, \dots, x_n) \\ \phi(k+1, x_2, \dots, x_n) &= \chi(k, \phi(k, x_2, \dots, x_n), x_2, \dots, x_n). \end{aligned}$$

In both (1) and (2), we allow the omission of each of the variables in any (or all) of its occurrences on the right side (e.g. $\phi(x, y) = \psi(\chi_1(x), \chi_2(x, y))$ is permitted under (1))³¹. We define the class of *recursive* functions to be the totality of functions which can be generated by substitution, according to the scheme (1), and recursion, according to the scheme (2), from the successor function $x + 1$, constant functions $f(x_1, \dots, x_n) = c$, and identity functions $U_j^n(x_1, \dots, x_n) = x_j$ ($1 \leq j \leq n$). In other words, a function ϕ shall be recursive if there exists a finite sequence of functions ϕ_1, \dots, ϕ_n which terminates with ϕ such that each function of the sequence is either the successor function $x + 1$ or a constant function $f(x_1, \dots, x_n) = c$, or an identity function $U_j^n(x_1, \dots, x_n) = x_j$, or is compound with respect to preceding functions, or is recursive with respect to preceding functions.

1.(b) Rewrite rule (1) for $n = 1$ and $m = 2$. Also rewrite rule (2) for $n = 1$. Explain in your own words what the rules express.

1.(c) Give a definition of primitive recursive functions in your own words.

1.(d) On page 44 of [13] Gödel states:

The functions $x + y$, xy , x^y and $x!$ are clearly [primitive] recursive.

Give an argument for why this is so. Hint: Review your answer to 1.(c) and make sure that you have a good grasp of Gödel's definition of primitive recursive functions.

1.(e) Is every function of natural numbers primitive recursive? Hint: Use a cardinality argument.

1.(f) Read carefully the excerpts from Church's 1935 letter to Kleene and Church's 1935 abstract appearing above. Also read carefully the following excerpt of Gödel's 1934 lecture notes [24] reprinted in Davis [13].

Recursive functions have the important property that, for each given set of values of the arguments, the value of the function can be computed by a finite procedure³².

Do you see any connection between Gödel's writing and Church's thesis? Explain your answer.

³¹[This sentence could have been] omitted, since the removal of any of the occurrences of variables on the right may be effected by means of the function U_j^n . This footnote occurs in the original source.

³²The converse seems to be true, if, besides recursions according to the scheme (2), recursions of other forms (e.g., with respect to two variables simultaneously) are admitted. This cannot be proved, since the notion of finite computation is not defined, but it serves as a heuristic principle. This footnote occurs in the original source.

Part two. General recursive functions

2.(a) Read carefully the following excerpt of section 1 of Kleene [42].

We consider the following schemata as operations for the definition of a function ϕ from given functions appearing in the right members of the equations (c is any constant natural number):

$$\begin{aligned}
 \text{(I)} \quad & \phi(x) = x', \\
 \text{(II)} \quad & \phi(x_1, \dots, x_n) = c, \\
 \text{(III)} \quad & \phi(x_1, \dots, x_n) = x_i, \\
 \text{(IV)} \quad & \phi(x_1, \dots, x_n) = \theta(\chi(x_1, \dots, x_n), \dots, \chi_m(x_1, \dots, x_n)), \\
 \text{(Va)} \quad & \begin{cases} \phi(0) = c \\ \phi(y') = \chi(y, \phi(y)), \end{cases} \\
 \text{(Vb)} \quad & \begin{cases} \phi(0, x_1, \dots, x_n) = \psi(x_1, \dots, x_n) \\ \phi(y', x_1, \dots, x_n) = \chi(y, \phi(y, x_1, \dots, x_n), x_1, \dots, x_n), \end{cases}
 \end{aligned}$$

Schema (I) introduces the successor function, Schema (II) the constant functions, and Schema (III) the identity functions. Schema (IV) is the schema of definition by substitution, and Schema (V) the schema of primitive recursion. Together we may call them (and more generally, schemata reducible to a series of applications of them) the *primitive recursive* schemata.

A function ϕ that can be defined from given functions ψ_1, \dots, ψ_k by a series of applications of these schemata we call *primitive recursive* in the given functions; and in particular, a function ϕ definable ab initio³³ by these means, *primitive recursive*.

Is your answer to 1.(c) equivalent to Kleene's definition of primitive recursive functions? Explain why.

2.(b) Give a definition of *total* and *partial* functions. Discuss briefly the difference between the two. Read carefully the following definition of the μ -operator taken from the beginning of section 3 of Kleene [42].

Consider the operator: μy (the least y such that). If this operator is applied to a predicate $R(x_1, \dots, x_n, y)$ of the $n + 1$ variables x_1, \dots, x_n, y , and if this predicate satisfies the condition

$$(2) \quad (\forall x_1) \cdots (\forall x_n) (\exists y) R(x_1, \dots, x_n, y),$$

we obtain a function $\mu y R(x_1, \dots, x_n, y)$ of the remaining n free variables x_1, \dots, x_n .

Thence we have a new schema,

$$(VI_1) \quad \phi(x_1, \dots, x_n) = \mu y [\rho(x_1, \dots, x_n, y) = 0],$$

for the definition of a function ϕ from a given function ρ which satisfies the condition

$$(3) \quad (\forall x_1) \cdots (\forall x_n) (\exists y) [\rho(x_1, \dots, x_n, y) = 0].$$

³³From the beginning.

Give an example of a function obtainable by the application of the μ -operator that is not total. What does condition (3) guarantee about the function obtained by using schema (VI₁)?

2.(c) Read carefully excerpts of theorem III and the corollary to it taken from page 51 of Kleene [42].

THEOREM III. The class of general recursive functions is closed under applications of Schemata (I)–(VI) with (3) holding for applications of (VI).

COROLLARY. Every function obtainable by applications of Schemata (I)–(VI) with (3) holding for applications of (VI) is general recursive.

Based on the corollary, give a definition of general recursive functions. Explain whether or not every primitive recursive function is general recursive.

2.(d) Is every function of natural numbers general recursive? Explain your answer.

2.(e) (Extra Credit) Give a reasonable argument for why the class of general recursive functions is *strictly* larger than the class of primitive recursive functions.

Part three. Turing machines

3.(a) Read carefully the following excerpt of section 1 of Turing [66].

We may compare a man in the process of computing a real number to a machine which is only capable of a finite number of conditions q_1, q_2, \dots, q_R which will be called “ m -configurations”. The machine is supplied with a “tape” (the analogue of paper) running through it, and divided into sections (called “squares”) each capable of bearing a “symbol”. At any moment there is just one square, say the r -th, bearing the symbol $\mathfrak{S}(r)$ which is “in the machine”. We may call this square the “scanned square”. The symbol on the scanned square may be called the “scanned symbol”. The “scanned symbol” is the only one of which the machine is, so to speak, “directly aware”. However, by altering its m -configuration the machine can effectively remember some of the symbols which it has “seen” (scanned) previously. The possible behaviour of the machine at any moment is determined by the m -configuration q_n and the scanned symbol $\mathfrak{S}(r)$. This pair $q_n, \mathfrak{S}(r)$ will be called the “configuration”: thus the configuration determines the possible behaviour of the machine. In some of the configurations in which the scanned square is blank (i.e. bears no symbol) the machine writes down a new symbol on the scanned square: in other configurations it erases the scanned symbol. The machine may also change the square which is being scanned, but only by shifting it one place to right or left. In addition to any of these operations the m -configuration may be changed. Some of the symbols written down will form the sequence of figures which is the decimal of the real number which is being computed. The others are just rough notes to “assist the memory”. It will only be these rough notes which will be liable to erasure.

Also read carefully the following excerpt of section 2 of Turing [66].

Automatic machines.

If at each stage the motion of a machine (in the sense of §1) is *completely* determined by the configuration, we shall call the machine an “automatic machine” (or a-machine).

For some purposes we might use machines (choice machines or c-machines) whose motion is only partially determined by the configuration (hence the use of the word “possible” in §1). When such a machine reaches one of these ambiguous configurations, it cannot go on until some arbitrary choice has been made by an external operator. This would be the case if we were using machines to deal with axiomatic systems. In this paper I deal only with automatic machines, and will therefore often omit the prefix a-.

Computing machines.

If an a-machine prints two kinds of symbols, of which the first kind (called figures) consists entirely of 0 and 1 (the others being called symbols of the second kind), then the machine will be called a computing machine. If the machine is supplied with a blank tape and set in motion, starting from the correct initial m -configuration, the subsequence of the symbols printed by it which are of the first kind will be called the *sequence computed by the machine*. The real number whose expression as a binary decimal is obtained by prefacing this sequence by a decimal point is called the *number computed by the machine*.

At any stage of the motion of the machine, the number of the scanned square, the complete sequence of all symbols on the tape, and the m -configuration will be said to describe the *complete configuration* at that stage. The changes of the machine and tape between successive complete configurations will be called the *moves* of the machine.

Circular and circle-free machines.

If a computing machine never writes down more than a finite number of symbols of the first kind, it will be called *circular*. Otherwise it is said to be *circle-free*.

A machine will be circular if it reaches a configuration from which there is no possible move, or if it goes on moving, and possibly printing symbols of the second kind, but cannot print any more symbols of the first kind...

Computable sequences and numbers.

A sequence is said to be computable if it can be computed by a circle-free machine. A number is computable if it differs by an integer from the number computed by a circle-free machine.

Note: The machines that Turing designs will be called Turing machines.

- 3.(b) What are the primary components of a Turing machine? Describe m -configurations, configurations, and complete configurations of a Turing machine, and their differences.
- 3.(c) Formulate in your own words Turing’s definition of a computing machine, a computable sequence (of natural numbers), and a computable real number (in the interval $[0, 1]$).
- 3.(d) Read carefully the following excerpt of section 3 of Turing [66] where it is shown that the sequence 010101... is computable.

A machine can be constructed to compute the sequence 010101... The machine is to have the four configurations “b”, “c”, “f”, “e” and is capable of printing “0” and “1”. The behaviour of the machine is described in the following table in which “ R ” means “the machine moves so that it scans the square immediately on the right of the one it was scanning previously”. Similarly for “ L ”. “ E ” means “the scanned symbol is erased” and “ P ” stands for “prints”. This table (and all succeeding tables of the same kind) is to be understood to mean that for a configuration described in the first two columns the operations in the third column are carried out successively, and the machine then goes over into the

m -configuration described in the last column. When the second column is left blank, it is understood that the behaviour of the third and fourth columns applies for any symbol and for no symbol. The machine starts in the m -configuration b with a blank tape.

| Configuration | | Behaviour | |
|---------------|--------|------------|--------------------|
| m -config. | symbol | operations | final m -config. |
| b | None | $P0, R$ | c |
| c | None | R | e |
| e | None | $P1, R$ | f |
| f | None | R | b |

If (contrary to the description in §1) we allow the letters L, R to appear more than once in the operations column we can simplify the table considerably.

| m -config. | symbol | operations | final m -config. |
|--------------|--------|------------|--------------------|
| b | None | $P0$ | b |
| | 0 | $R, R, P1$ | b |
| | 1 | $R, R, P0$ | b |

What can you conclude about the real number $\frac{1}{3}$? Design a Turing machine that computes the real number $\frac{1}{7}$. Give an argument for why the machine you just designed does what it is supposed to do.

3.(e) (Extra Credit) Read carefully the following excerpt of section 3 of Turing [66] where it is shown that the sequence $00101101110111101111\dots$ is computable.

As a slightly more difficult example we can construct a machine to compute the sequence $00101101110111101111\dots$. The machine is to be capable of five m -configurations, viz. " o ", " q ", " p ", " f ", " b " and of printing " ∂ ", " x ", " 0 ", " 1 ". The first three symbols on the tape will be " $\partial\partial\partial$ "; the other figures follow on alternate squares. On the intermediate squares we never print anything but " x ". These letters serve to "keep the place" for us and are erased when we have finished with them. We also arrange that in the sequence of figures on alternate squares there shall be no blanks.

| Configuration | | Behaviour | |
|---------------|---|--|--------------------|
| m -config. | symbol | operations | final m -config. |
| b | | $P\partial, R, P\partial, R, P0, R, R, P0, L, L$ | o |
| o | $\left\{ \begin{array}{l} 1 \\ 0 \end{array} \right.$ | R, Px, L, L, L | o |
| | | | q |
| q | $\left\{ \begin{array}{l} \text{Any (0 or 1)} \\ \text{None} \end{array} \right.$ | R, R | q |
| | | $P1, L$ | p |
| p | $\left\{ \begin{array}{l} x \\ \partial \end{array} \right.$ | E, R | q |
| | | R | f |
| f | $\left\{ \begin{array}{l} \text{None} \\ \text{Any} \\ \text{None} \end{array} \right.$ | L, L | p |
| | | R, R | f |
| | | $P0, L, L$ | o |

To illustrate the working of this machine a table is given below of the first few complete configurations. These complete configurations are described by writing down the sequence of symbols which are on the tape, with the m -configuration written below the scanned symbol. The successive complete configurations are separated by colons.

```

      : ∂∂0 0 : ∂∂0 0 : ∂∂0 0 : ∂∂0 0      : ∂∂0 0 1
    b   o       q       q       q       p
    ∂∂0 0 1 : ∂∂0 0 1 : ∂∂0 0 1 : ∂∂0 0 1 : ∂∂0 0 1 :
          p       p       f       f
    ∂∂0 0 1 : ∂∂0 0 1      : ∂∂0 0 1 0 :
          f       f       o
    ∂∂0 0 1x0 : . . . .
          o
  
```

In this example Turing uses the symbols “∂” and “x”, which are the symbols of the second kind. Explain the need for these symbols, and their use in this particular Turing machine. Give an argument for why the machine described above does what it is supposed to do.

Part four. Turing computable functions

4.(a) Read carefully Turing’s definition of computable functions of natural numbers taken from page 254 of [66].

If γ is a computable sequence in which 0 appears infinitely³⁴ often, and n is an integer, then let us define $\xi(\gamma, n)$ to be the number of figures 1 between the n -th and the $(n + 1)$ -th figure 0 in γ . Then $\phi(n)$ is computable if, for all n and some γ , $\phi(n) = \xi(\gamma, n)$.

Note: We will call these functions Turing computable.

4.(b) In your own words explain what it means for a function of natural numbers of one variable to be Turing computable. Extra Credit. Generalize the concept of Turing computable functions to functions of two variables. Hint: Can you code every function of two variables by a function of one variable? Generalize the concept of Turing computable functions to functions of multiple variables.

4.(c) In your own words explain Kleene’s definition of a Turing machine by reading carefully the following excerpt from section 67 of Kleene [43].

The machine is supplied with a linear *tape*, (potentially) infinite in both directions (say to the *left* and *right*). The tape is divided into *squares*. Each square is capable of being *blank*, or of having *printed* upon it any one of a finite list s_1, \dots, s_j ($j \geq 1$) of *symbols*, fixed for a particular machine. If we write “ s_0 ” to stand for “blank”, a given square can thus have any one of $j + 1$ *conditions* s_0, \dots, s_j . The tape will be so employed that in any “situation” only a finite number (≥ 0) of squares will be printed.

The tape will pass through the machine so that in a given “situation” the machine *scans* just one square (the *scanned square*). The symbol on this square, or s_0 if it is blank, we call the *scanned symbol* (even though s_0 is not properly a symbol).

The machine is capable of being in any one of a finite list q_0, \dots, q_k ($k \geq 1$) of (*machine*) *states* (called by Turing “machine configurations” or “ m -configurations”). We call q_0 the

³⁴If \mathcal{M} computes γ , then the problem whether \mathcal{M} prints 0 infinitely often is of the same character as the problem whether \mathcal{M} is circle-free. This footnote occurs in the original source.

passive (or *terminal*) state; and q_1, \dots, q_k we call *active states*. The list q_0, \dots, q_k is fixed for a particular machine.

A (*tape vs. machine*) *situation* (called by Turing “complete configuration”) consists in a particular printing on the tape (i.e. which squares are printed, and each with which of the j symbols), a particular position of the tape in the machine (i.e. which square is scanned), and a particular state (i.e. which of the $k + 1$ states the machine is in). If the state is active, we call the situation *active*; otherwise, *passive*.

Given an active situation, the machine performs an (*atomic*) *act* (called a “move” by Turing). The act performed is determined by the scanned symbol s_a and the machine state q_c in the given situation. This pair (s_a, q_c) we call the *configuration*. (It is *active* in the present case that q_c is active; otherwise *passive*.) The act alters the three parts of the situation to produce a resulting situation, thus. First, the scanned symbol s_a is changed to s_b . (But $a = b$ is permitted, in which case the “change” is identical.) Second, the tape is shifted in the machine (or the machine shifts along the tape) so that the square scanned in the resulting situation is either one square to the left of, or the same square as, or one square to the right of, the square scanned in the given situation. Third, the machine state q_c is changed to q_d . (But $c = d$ is permitted.)

No act is performed, if the given situation is passive.

The machine is used in the following way. We choose some active situation in which to start the machine. We call this the *initial situation* or *input*. Our notation will be chosen so that the state in this situation (the *initial state*) is q_1 . The machine then performs an atomic act. If the situation resulting from this act is active, the machine acts again. The machine continues in this manner, clicking off successive acts, as long and only as long as active situations result. If eventually a passive situation is reached, the machine is said then to *stop*. The situation in which it stops we call the *terminal situation* or *output*.

The change from the initial situation to the terminal situation (when there is one) may be called the *operation* performed by the machine.

To describe an atomic act, we use an expression of one of the three following forms:

$$s_b L q_d, \quad s_b C q_d, \quad s_b R q_d.$$

The “*L*”, “*C*”, “*R*”, indicate that the resulting scanned square is to the left of, the same as (“center”), or to the right of, respectively, the given scanned square.

The first part of the act (i.e. the change of s_a to s_b) falls into four cases: when $a = 0$ and $b > 0$, it is “prints s_b ”; when $a > 0$ and $b = 0$, “erases s_a ”; when $a, b > 0$ and $a \neq b$, “erases s_a and prints s_b ” or briefly “overprints s_b ”; when $a = b$, “no change”. We often describe this part of the act as “prints s_b ” without regard to the case.

To define a particular machine, we must list the symbols s_1, \dots, s_j and the active states q_1, \dots, q_k , and for each active configuration (s_a, q_c) we must specify the atomic act to be performed. These specifications may be given by displaying the descriptions of the required acts in the form of a (*machine*) *table* with k rows for the active states and $j + 1$ columns for the square conditions.

EXAMPLE I. The following table defines a machine (“Machine \mathfrak{A} ”) having only one symbol

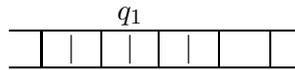
s_1 and only one active state q_1 .

| Name of machine | Machine state | Scanned symbol s_0 | symbol s_1 |
|-----------------|---------------|----------------------|--------------|
| \mathfrak{A} | q_1 | $s_1 C q_0$ | $s_1 R q_1$ |

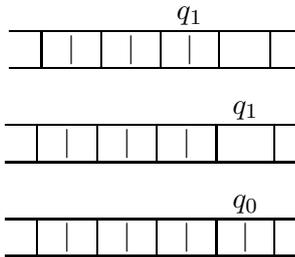
Suppose the symbol s_1 is actually a tally mark “|”. Let us see what the machine does, if a tape of the following appearance is placed initially in the machine so that the square which we identify by writing the machine state q_1 over it is the scanned square. The conditions of all squares not shown will be immaterial, and will not be changed during the action.



The machine is in the state q_1 , and is scanning a square on which the symbol s_1 is printed. In this configuration, the atomic act ordered by the table is $s_1 R q_1$; i.e. no change is made in the condition of the scanned square, the machine shifts right, and again assumes state q_1 . The resulting situation appears as follows.



The next three acts lead successively to the following situations, in the last of which the machine stops.



Machine \mathfrak{A} performs the following operation: It seeks the first blank square at or to the right of the scanned square, prints a | there, and stops scanning that square.

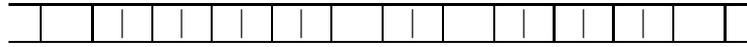
- 4.(d) Explain the similarities and differences of Turing’s and Kleene’s definitions. Which definition of Turing machines do you prefer? Explain why.
- 4.(e) The following is Kleene’s definition of Turing computable functions of multiple variables (see [43, p. 359]).

Now we define how a machine shall ‘compute’ a partial number-theoretic function ϕ of n variables (cf. §63). The definition for an ordinary (i.e. completely defined) number-theoretic function is obtained by omitting the reference to the possibility that $\phi(x_1, \dots, x_n)$ may be undefined.

We begin by agreeing to represent the natural numbers $0, 1, 2, \dots$ by the sequence of tallies |, ||, |||, . . . , respectively, the tally “|” being the symbol s_1 . There are $y + 1$ tallies in the representation of the natural number y .

Then to represent an m -tuple y_1, \dots, y_m ($m \geq 1$) of natural numbers on the tape, we print the corresponding numbers of tallies, leaving a single blank between each two groups of tallies and before the first and after the last.

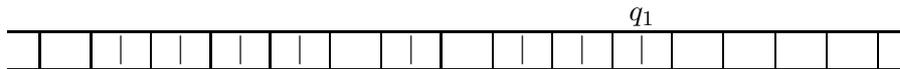
EXAMPLE 2. The triple 3, 0, 2 is represented thus:



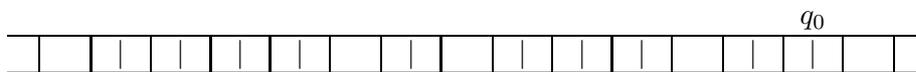
We say that (the representation of) a number y (or of any m -tuple y_1, \dots, y_m) on the tape is (*scanned*) *in the standard position*, when the scanned square is the one bearing the last tally in the representation of y (or of y_m).

Now we say that a given machine \mathfrak{A} *computes* a given partial function ϕ of n variables ($n \geq 1$), if the following holds for each n -tuple x_1, \dots, x_n of natural numbers. (For the case $n = 0$, cf. Remark 1 below.) Let x_1, \dots, x_n be represented on the tape, with the tape blank elsewhere, i.e. outside of the $x_1 + \dots + x_n + 2n + 1$ squares required for the representation. Let \mathfrak{A} be started scanning the representation of x_1, \dots, x_n in standard position. Then \mathfrak{A} will eventually stop with the $n + 1$ -tuple x_1, \dots, x_n, x represented on the tape and scanned in standard position, if and only if $\phi(x_1, \dots, x_n)$ is defined and $\phi(x_1, \dots, x_n) = x$. (If $\phi(x_1, \dots, x_n)$ is undefined, \mathfrak{A} may fail to stop. It may stop but without an $n + 1$ -tuple x_1, \dots, x_n, x scanned in standard position.)

Example 2 (concluded). If $\phi(3, 0, 2) = 1$ and \mathfrak{A} computes ϕ , then when \mathfrak{A} is started in the situation



with all squares other than those shown blank, it must eventually stop in the situation



where the condition of the squares other than those shown is immaterial.

Although only one symbol s_1 or “|” is used in stating the arguments and in receiving the function value, others may be used in the progress of the computation. For each $n \geq 1$, each machine (with its first symbol s_1 serving as the tally) computes a certain partial function of n variables.

A partial function ϕ is *computable*, if there is a machine \mathfrak{A} which computes it.

Remark 1 that Kleene refers to is on page 363 of Kleene [43]. Below we give an excerpt from it.

REMARK 1. In this chapter, outside the present remark and passages referring to it, we shall understand that we are dealing with functions of $n \geq 1$ variables. Since we have not provided for representing n -tuples of natural numbers on the tape for $n = 0$, we say a machine computes a function ϕ of 0 variables, if it computes the function $\phi(x)$ of 1 variable such that $\phi(x) \simeq \phi$.

Describe in your own words Kleene’s definition of Turing computable functions of multiple variables. Kleene mentions that the representation of the tuple x_1, \dots, x_n on a tape requires $x_1 + \dots + x_n + 2n + 1$ squares. Explain why.

4.(f) Are Turing's and Kleene's definitions of computable functions of one variable equivalent? Explain why.

4.(g) (Extra Credit) Is your generalization of Turing computability to functions of multiple variables equivalent to Kleene's definition of computable functions of multiple variables? Explain why.

Part five. Turing computable and general recursive functions

5.(a) Explain why the successor function $s(x) = x + 1$ is Turing computable.

5.(b) Explain why the constant functions $f(x_1, \dots, x_n) = c$ are Turing computable.

5.(c) Explain why the identity functions $U_j^n(x_1, \dots, x_n) = x_j$ are Turing computable.

5.(d) Explain why whenever $\psi(x_1, \dots, x_m)$ and $\chi_1(x_1, \dots, x_n), \dots, \chi_m(x_1, \dots, x_n)$ are Turing computable, then $\phi(x_1, \dots, x_n)$ defined by

$$\phi(x_1, \dots, x_n) = \psi(\chi_1(x_1, \dots, x_n), \dots, \chi_m(x_1, \dots, x_n))$$

is also Turing computable.

5.(e) Explain why whenever $\psi(x_1, \dots, x_{n-1})$ and $\chi(x_1, \dots, x_{n+1})$ are Turing computable, then $\phi(x_1, \dots, x_n)$ defined by

$$\phi(0, x_2, \dots, x_n) = \psi(x_2, \dots, x_n)$$

and

$$\phi(k + 1, x_2, \dots, x_n) = \chi(k, \phi(k, x_2, \dots, x_n), x_2, \dots, x_n)$$

is also Turing computable. In other words, explain why the schema of primitive recursion preserves Turing computability.

5. (f) Explain why whenever $\rho(x_1, \dots, x_n, y)$ is Turing computable and

$$(\forall x_1) \dots (\forall x_n) (\exists y) [\rho(x_1, \dots, x_n, y) = 0],$$

then $\phi(x_1, \dots, x_n)$ defined by

$$\phi(x_1, \dots, x_n) = \mu y [\rho(x_1, \dots, x_n, y) = 0]$$

is also Turing computable. In other words, explain why Kleene's μ -operator preserves Turing computability.

5.(g) Using exercises 5.(a)–5.(f) make a deduction concerning primitive recursive and general recursive functions. How is your conclusion related to Church's thesis?

10 Early Writings on Graph Theory: Euler Circuits and the Königsberg Bridge Problem

Janet Heine Barnett³⁵



Figure 2: The Königsberg Bridges.

In a 1670 letter to Christian Huygens (1629 - 1695), the celebrated philosopher and mathematician Gottfried W. Leibniz (1646 - 1716) wrote as follows:

I am not content with algebra, in that it yields neither the shortest proofs nor the most beautiful constructions of geometry. Consequently, in view of this, I consider that we need yet another kind of analysis, geometric or linear, which deals directly with position, as algebra deals with magnitude. [3, p. 30]

Known today as the field of ‘topology,’ Leibniz’s study of position was slow to develop as a mathematical field. As C. F. Gauss (1777 - 1855) noted in 1833,

Of the geometry of position, which Leibniz initiated and to which only two geometers, Euler and Vandermonde, have given a feeble glance, we know and possess, after a century and a half, very little more than nothing. [3, p. 30]

The ‘feeble glance’ which Leonhard Euler (1707 - 1783) directed towards the geometry of position consists of a single paper now considered to be the starting point of modern graph theory. Within the history of mathematics, the eighteenth century itself is commonly known as ‘The Age of Euler’ in recognition of the tremendous contributions that Euler made to mathematics during this period. Born in Basel, Switzerland, Euler studied mathematics under Johann Bernoulli (1667 - 1748), then one of the leading European mathematicians of the time and among the first — along with his brother Jakob Bernoulli (1654 - 1705) — to apply the new calculus techniques developed by Leibniz in the late seventeenth century to the study of curves. Euler soon surpassed his early teacher, and made important contributions to an astounding variety of subjects, ranging from number theory and analysis to astronomy and optics to mapmaking, in addition to graph theory and topology. His work was particularly important in re-defining calculus as the study of analytic functions, in contrast to the seventeenth century view of calculus as the study of curves. Amazingly, nearly half of Euler’s nearly 900 books, papers and other works were written after he became almost totally blind in 1771.

³⁵Department of Mathematics; Colorado State University - Pueblo; Pueblo, CO 81001 - 4901; janet.barnett@colostate-pueblo.edu.

The paper we examine in this project appeared in *Commentarii Academiae Scientiarum Imperialis Petropolitanae* in 1736. In it, Euler undertakes a mathematical formulation of the now-famous Königsberg Bridge Problem: is it possible to plan a stroll through the town of Königsberg which crosses each of the town's seven bridges once and only once? Like other early graph theory work, the Königsberg Bridge Problem has the appearance of being little more than an interesting puzzle. Yet from such deceptively frivolous origins, graph theory has grown into a powerful and deep mathematical theory with applications in the physical, biological, and social sciences. The resolution of the Four Color Problem — one of graph theory's most famous historical problems — even raised new questions about the notion of mathematical proof itself. First formulated by Augustus De Morgan in a 1852 letter to Hamilton, the Four Color Problem asks whether four colors are sufficient to color every planar map in such a way that regions sharing a boundary are colored in different colors. After a long history of failed attempts to prove this is the case, Kenneth Appel (1932 -) and Wolfgang Haken (1928 -) published a computer-assisted proof in 1976 which many mathematicians were unwilling to accept as valid. At the heart of the issue is a question that could be asked of any computer-assisted proof: should an argument that can not be directly checked by any member of the mathematical community be considered to be a valid proof?

This modern controversy highlights the historical fact that standards of proof have always varied from century to century, and from culture to culture. This project will highlight one part of this historical story by examining the differences in precision between an eighteenth century proof and a modern treatment of the same result. In particular, we wish to contrast Euler's approach to the problem of finding necessary and sufficient conditions for the existence of what is now known as an 'Euler circuit' to a modern proof of the main result of the paper.

In what follows, we take our translation from [3, pp. 3 - 8], with some portions eliminated in order to focus only on those most relevant to Euler's reformulation of the 'bridge crossing problem' as a purely mathematical problem. Definitions of modern terminology are introduced as we proceed through Euler's paper; modern proofs of two lemmas used in the proof of the main result are also included in an appendix.

SOLUTIO PROBLEMATIS AD GEOMETRIAM SITUS PERTINENTIS

- 1 In addition to that branch of geometry which is concerned with magnitudes, and which has always received the greatest attention, there is another branch, previously almost unknown, which Leibniz first mentioned, calling it the *geometry of position*. This branch is concerned only with the determination of position and its properties; it does not involve measurements, nor calculations made with them. It has not yet been satisfactorily determined what kind of problems are relevant to this geometry of position, or what methods should be used in solving them. Hence, when a problem was recently mentioned, which seemed geometrical but was so constructed that it did not require the measurement of distances, nor did calculation help at all, I had no doubt that it was concerned with the geometry of position — especially as its solution involved only position, and no calculation was of any use. I have therefore decided to give here the method which I have found for solving this kind of problem, as an example of the geometry of position.
- 2 The problem, which I am told is widely known, is as follows: in Königsberg in Prussia, there is an island *A*, called *the Kneiphof*; the river which surrounds it is divided into two branches, as can be seen in Fig. [1.2], and these branches are crossed by seven

bridges, a, b, c, d, e, f and g . Concerning these bridges, it was asked whether anyone could arrange a route in such a way that he would cross each bridge once and only once. I was told that some people asserted that this was impossible, while others were in doubt: but nobody would actually assert that it could be done. From this, I have formulated the general problem: whatever be the arrangement and division of the river into branches, and however many bridges there be, can one find out whether or not it is possible to cross each bridge exactly once?

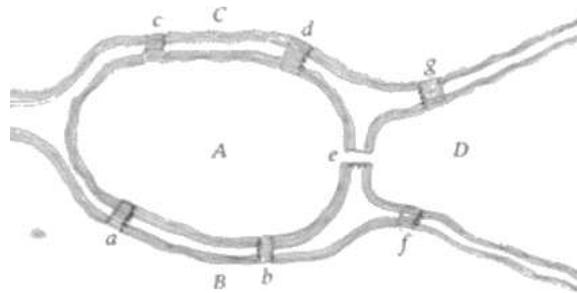
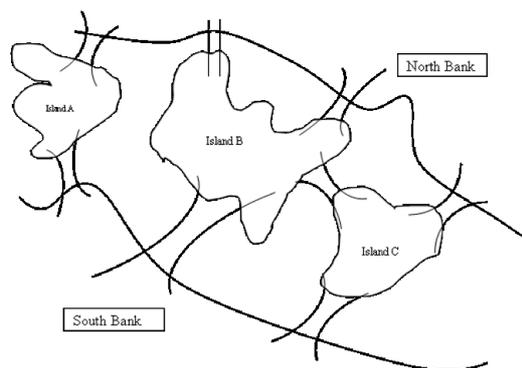


FIG. 1.2

Notice that Euler begins his analysis of the ‘bridge crossing’ problem by first replacing the map of the city by a simpler diagram showing only the main feature. In modern graph theory, we simplify this diagram even further to include only points (representing land masses) and line segments (representing bridges). These points and line segments are referred to as *vertices* (singular: vertex) and *edges* respectively. The collection of vertices and edges together with the relationships between them is called a *graph*. More precisely, a graph consists of a set of vertices and a set of edges, where each edge may be viewed as an ordered pair of two (usually distinct) vertices. In the case where an edge connects a vertex to itself, we refer to that edge as a *loop*.

1. Sketch the diagram of a graph with 5 vertices and 8 edges to represent the following bridge problem.



- 3 As far as the problem of the seven bridges of Königsberg is concerned, it can be solved by making an exhaustive list of all possible routes, and then finding whether or not any

route satisfies the conditions of the problem. Because of the number of possibilities, this method of solution would be too difficult and laborious, and in other problems with more bridges it would be impossible. Moreover, if this method is followed to its conclusion, many irrelevant routes will be found, which is the reason for the difficulty of this method. Hence I rejected it, and looked for another method concerned only with the problem of whether or not the specified route could be found; I considered that such a method would be much simpler.

- 4 My whole method relies on the particularly convenient way in which the crossing of a bridge can be represented. For this I use the capital letters A, B, C, D , for each of the land areas separated by the river. If a traveler goes from A to B over bridge a or b , I write this as AB — where the first letter refers to the area the traveler is leaving, and the second refers to the area he arrives at after crossing the bridge. Thus, if the traveler leaves B and crosses into D over bridge f , this crossing is represented by BD , and the two crossing AB and BD combined I shall denote by the three letters ABD , where the middle letter B refers to both the area which is entered in the first crossing and to the one which is left in the second crossing.
- 5 Similarly, if the traveler goes on from D to C over the bridge g , I shall represent these three successive crossings by the four letters $ABDC$, which should be taken to mean that the traveler, starting in A , crosses to B , goes on to D , and finally arrives in C . Since each land area is separated from every other by a branch of the river, the traveler must have crossed three bridges. Similarly, the successive crossing of four bridges would be represented by five letters, and in general, however many bridges the traveler crosses, his journey is denoted by a number of letters one greater than the number of bridges. Thus the crossing of seven bridges requires eight letters to represent it.

After rejecting the impractical strategy of solving the bridge-crossing problem by making an exhaustive list of all possible routes, Euler again reformulates the problem in terms of sequences of letters (vertices) representing land masses, thereby making the diagram itself unnecessary to the solution of the problem. Today, we say that two vertices joined by an edge in the graph are *adjacent*, and refer to a sequence of adjacent vertices as a *walk*. Technically, a walk is a sequence of alternating (adjacent) vertices and edges $v_0e_1v_1e_2 \dots e_nv_n$ in which both the order of the vertices and the order of the edges used between adjacent vertices are specified. In the case where no edge of the graph is repeated (as required in a bridge-crossing route), the walk is known as a *path*. If the initial and terminal vertex are equal, the path is said to be a *circuit*. If *every* edge of the graph is used *exactly once* (as desired in a bridge-crossing route), the path (circuit) is said to be a *Euler path (circuit)*.

2. For the bridge problem shown in Question A above, how many capital letters (representing graph vertices) will be needed to represent an Euler path?

Having reformulated the bridge crossing problem in terms of sequences of letters (vertices) alone, Euler now turns to the question of determining *whether* a given bridge crossing problem admits of a solution. As you read through Euler's development of a procedure for deciding this question in paragraphs 7 - 13 below, pay attention to the style of argument employed, and how this differs from that used in a modern textbook.

- 7 The problem is therefore reduced to finding a sequence of eight letters, formed from the four letters A , B , C and D , in which the various pairs of letters occur the required number of times. Before I turn to the problem of finding such a sequence, it would be useful to find out whether or not it is even possible to arrange the letters in this way, for if it were possible to show that there is no such arrangement, then any work directed toward finding it would be wasted. I have therefore tried to find a rule which will be useful in this case, and in others, for determining whether or not such an arrangement can exist.



FIG. 1.3

- 8 In order to try to find such a rule, I consider a single area A , into which there lead any number of bridges a , b , c , d , etc. (Fig. [1.3]). Let us take first the single bridge a which leads into A : if a traveler crosses this bridge, he must either have been in A before crossing, or have come into A after crossing, so that in either case the letter A will occur once in the representation described above. If three bridges (a , b and c , say) lead to A , and if the traveler crosses all three, then in the representation of his journey the letter A will occur twice, whether he starts his journey from A or not. Similarly, if five bridges lead to A , the representation of a journey across all of them would have three occurrences of the letter A . And in general, if the number of bridges is any odd number, and if it is increased by one, then the number of occurrences of A is half of the result.
3. In paragraph 8, Euler deduces a rule for determining how many times a vertex must appear in the representation of the route for a given bridge problem for the case where an odd number of bridges leads to the land mass represented by that vertex. **Before reading further**, use this rule to determine how many times each of the vertices A , B , C and D would appear in the representation of a route for the Königsberg Bridge Problem. Given Euler's earlier conclusion (paragraph 5) that a solution to this problem requires a sequence of 8 vertices, is such a sequence possible? Explain.
- 9 In the case of the Königsberg bridges, therefore, there must be three occurrences of the letter A in the representation of the route, since five bridges (a , b , c , d , e) lead to the area A . Next, since three bridges lead to B , the letter B must occur twice; similarly, D must occur twice, and C also. So in a series of eight letters, representing the crossing of seven bridges, the letter A must occur three times, and the letters B , C and D twice each - but this cannot happen in a sequence of eight letters. It follows that such a journey cannot be undertaken across the seven bridges of Königsberg.

- 10 It is similarly possible to tell whether a journey can be made crossing each bridge once, for any arrangement of bridges, whenever the number of bridges leading to each area is odd. For if the sum of the number of times each letter must occur is one more than the number of bridges, then the journey can be made; if, however, as happened in our example, the number of occurrences is greater than one more than the number of bridges, then such a journey can never be accomplished. The rule which I gave for finding the number of occurrences of the letter A from the number of bridges leading to the area A holds equally whether all of the bridges come from another area B, as shown in Fig. [1.3], or whether they come from different areas, since I was considering the area A alone, and trying to find out how many times the letter A must occur.
- 11 If, however, the number of bridges leading to A is even, then in describing the journey one must consider whether or not the traveler starts his journey from A; for if two bridges lead to A, and the traveler starts from A, then the letter A must occur twice, once to represent his leaving A by one bridge, and once to represent his returning to A by the other. If, however, the traveler starts his journey from another area, then the letter A will only occur once; for this one occurrence will represent both his arrival in A and his departure from there, according to my method of representation.
- 12 If there are four bridges leading to A, and if the traveler starts from A, then in the representation of the whole journey, the letter A must occur three times if he is to cross each bridge once; if he begins his walk in another area, then the letter A will occur twice. If there are six bridges leading to A, then the letter A will occur four times if the journey starts from A, and if the traveler does not start by leaving A, then it must occur three times. So, in general, if the number of bridges is even, then the number of occurrences of A will be half of this number if the journey is not started from A, and the number of occurrences will be one greater than half the number of bridges if the journey does start at A.
- 13 Since one can start from only one area in any journey, I shall define, corresponding to the number of bridges leading to each area, the number of occurrences of the letter denoting that area to be half the number of bridges plus one, if the number of bridges is odd, and if the number of bridges is even, to be half of it. Then, if the total of all the occurrences is equal to the number of bridges plus one, the required journey will be possible, and will have to start from an area with an odd number of bridges leading to it. If, however, the total number of letters is one less than the number of bridges plus one, then the journey is possible starting from an area with an even number of bridges leading to it, since the number of letters will therefore be increased by one.

Notice that Euler's definition concerning 'the number of occurrences of the letter denoting that area' depends on whether the the number of bridges (edges) leading to each area (vertex) is even or odd. In contemporary terminology, the number of edges incident on a vertex v is referred to as the *degree of vertex v* .

4. Let $deg(v)$ denote the degree of vertex v in a graph G . Euler's definition of 'the number of occurrences of v ' can then be re-stated as follows:
- If $deg(v)$ is even, then v occurs $\frac{1}{2}deg(v)$ times.
 - If $deg(v)$ is odd, then v occurs $\frac{1}{2}[deg(v) + 1]$ times.

Based on Euler's discussion in paragraphs 9 - 12, how convinced are you that this definition gives a correct description of the Königsberg Bridge Problem? How convincing do you find Euler's claim (in paragraph 13) that the required route can be found in the case where 'the total of all the occurrences is equal to the number of bridges plus one'? Comment on how a proof of this claim in a modern textbook might differ from the argument which Euler presents for it in paragraphs 9 - 12.

- 14 So, whatever arrangement of water and bridges is given, the following method will determine whether or not it is possible to cross each of the bridges: I first denote by the letters A, B, C, etc. the various areas which are separated from one another by the water. I then take the total number of bridges, add one, and write the result above the working which follows. Thirdly, I write the letters A, B, C, etc. in a column, and write next to each one the number of bridges leading to it. Fourthly, I indicate with an asterisk those letters which have an even number next to them. Fifthly, next to each even one I write half the number, and next to each odd one I write half the number increased by one. Sixthly, I add together these last numbers, and if this sum is one less than, or equal to, the number written above, which is the number of bridges plus one, I conclude that the required journey is possible. It must be remembered that if the sum is one less than the number written above, then the journey must begin from one of the areas marked with an asterisk, and it must begin from an unmarked one if the sum is equal. Thus in the Königsberg problem, I set out the working as follows:

Number of bridges 7, which gives 8

| | | |
|----|----------------|---|
| | <i>Bridges</i> | |
| A, | 5 | 3 |
| B, | 3 | 2 |
| C, | 3 | 3 |
| D, | 3 | 2 |

Since this gives more than 8, such a journey can never be made.

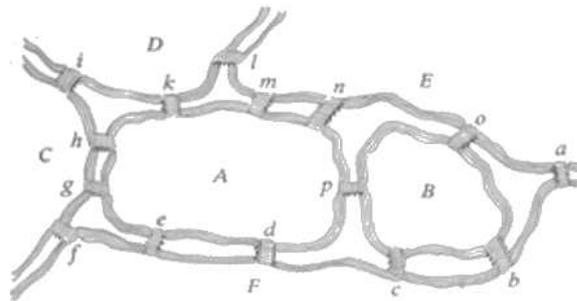


FIG. 1.4

- 15 Suppose that there are two islands A and B surrounded by water which leads to four rivers as shown in Fig. [1.4]. Fifteen bridges (a, b, c, d, etc.) cross the rivers and the water surrounding the islands, and it is required to determine whether one can arrange a journey which crosses each bridge exactly once. First, therefore, I name all the areas

separated by water as A, B, C, D, E, F, so that there are six of them. Next, I increase the number of bridges (15) by one, and write the result (16) above the working which follows.

| | |
|-------|----|
| | 16 |
| A*, 8 | 4 |
| B*, 4 | 2 |
| C*, 4 | 2 |
| D, 3 | 2 |
| E, 5 | 3 |
| F*, 6 | 3 |
| | 16 |

Thirdly, I write the letters A, B, C, etc. in a column, and write next to each one the number of bridges which lead to the corresponding area, so that eight bridges lead to A, four to B, and so on. Fourthly, I indicate with an asterisk those letters which have an even number next to them. Fifthly, I write in the third column half the even numbers in the second column, and then I add one to the odd numbers and write down half the result in each case. Sixthly, I add up all the numbers in the third column in turn, and I get the sum 16; since this is equal to the number (16) written above, it follows that the required journey can be made if it starts from area D or E, since these are not marked with an asterisk. The journey can be done as follows:

E a F b B c F d A e F f C g A h C i D k A m E n A p B o E I D,

where I have written the bridges which are crossed between the corresponding capital letters.

5. Apply Euler's procedure to determine whether the graph representing the 'bridge-crossing' question in question 1 above contains an Euler path. If so, find one.

In paragraphs 16 and 17, Euler makes some observations intended to simplify the procedure for determining whether a given bridge-crossing problem has a solution. As you read these paragraphs, consider how to reformulate these observations in terms of degree.

- 16 In this way it will be easy, even in the most complicated cases, to determine whether or not a journey can be made crossing each bridge once and once only. I shall, however, describe a much simpler method for determining this which is not difficult to derive from the present method, after I have first made a few preliminary observations. First, I observe that the numbers of bridges written next to the letters A, B, C, etc. together add up to twice the total number of bridges. The reason for this is that, in the calculation where every bridge leading to a given area is counted, each bridge is counted twice, once for each of the two areas which it joins.

17 It follows that the total of the numbers of bridges leading to each area must be an even number, since half of it is equal to the number of bridges. This is impossible if only one of these numbers is odd, or if three are odd, or five, and so on. Hence if some of the numbers of bridges attached to the letters A, B, C, etc. are odd, then there must be an even number of these. Thus, in the Königsberg problem, there were odd numbers attached to the letters A, B, C and D, as can be seen from Paragraph 14, and in the last example (in Paragraph 15), only two numbers were odd, namely those attached to D and E.

6. The result described in Paragraph 16 is sometimes referred to as ‘The Handshake Theorem,’ based on the equivalent problem of counting the number of handshakes that occur during a social gathering at which every person present shakes hands with every other person present exactly once. A modern statement of the Handshake Theorem would be: *The sum of the degree of all vertices in a finite graph equals twice the number of edges in the graph.* Locate this theorem in a modern textbook, and comment on how the proof given there compares to Euler’s discussion in paragraph 16.
7. The result described in Paragraph 17 can be re-stated as follows: *Every finite graph contains an even number of vertices with odd degree.* Locate this theorem in a modern textbook, and comment on how the proof given there compares to Euler’s discussion in paragraph 17.

Euler now uses the above observations to develop simplified rules for determining whether a given bridge-crossing problem has a solution. Again, consider how you might reformulate this argument in modern graph theoretic terms; we will consider a modern proof of the main results below.

- 18 Since the total of the numbers attached to the letters A, B, C, etc. is equal to twice the number of bridges, it is clear that if this sum is increased by 2 and then divided by 2, then it will give the number which is written above the working. If, therefore, all of the numbers attached to the letters A, B, C, D, etc. are even, and half of each of them is taken to obtain the numbers in the third column, then the sum of these numbers will be one less than the number written above. Whatever area marks the beginning of the journey, it will have an even number of bridges leading to it, as required. This will happen in the Königsberg problem if the traveler crosses each bridge twice, since each bridge can be treated as if it were split in two, and the number of bridges leading into each area will therefore be even.
- 19 Furthermore, if only two of the numbers attached to the letters A, B, C, etc. are odd, and the rest are even, then the journey specified will always be possible if the journey starts from an area with an odd number of bridges leading to it. For, if the even numbers are halved, and the odd ones are increased by one, as required, the sum of their halves will be one greater than the number of bridges, and hence equal to the number written above. It can further be seen from this that if four, or six, or eight. . . odd numbers appear in the second column, then the sum of the numbers in the third column will be greater by one, two, three. . . than the number written above, and the journey will be impossible.

- 20 So whatever arrangement may be proposed, one can easily determine whether or not a journey can be made, crossing each bridge once, by the following rules:

If there are more than two areas to which an odd number of bridges lead, then such a journey is impossible.

If, however, the number of bridges is odd for exactly two areas, then the journey is possible if it starts in either of these areas.

If, finally, there are no areas to which an odd number of bridges leads, then the required journey can be accomplished starting from any area.

With these rules, the given problem can always be solved.

- 21 When it has been determined that such a journey can be made, one still has to find how it should be arranged. For this I use the following rule: let those pairs of bridges which lead from one area to another be mentally removed, thereby considerably reducing the number of bridges; it is then an easy task to construct the required route across the remaining bridges, and the bridges which have been removed will not significantly alter the route found, as will become clear after a little thought. I do not therefore think it worthwhile to give any further details concerning the finding of the routes.

A complete modern statement of Euler's main result requires one final definition: a graph is said to be *connected* if for every pair of vertices u, v in the graph, there is a walk from u to v . Notice that a graph which is not connected will consist of several components, or subgraphs, each of which is connected. With this definition in hand, the main results of Euler's paper can be stated as follow:

Theorem: A finite graph G contains an Euler circuit if and only if G is connected and contains no vertices of odd degree.

Corollary: A finite graph G contains an Euler path if and only if G is connected and contains at most two vertices of odd degree.

8. Illustrate why the modern statement specifies that G is connected by giving an example of a disconnected graph which has vertices of even degree only and contains no Euler circuit. Explain how you know that your example contains no Euler circuit.
9. Comment on Euler's proof of this theorem and corollary as they appear in paragraphs 16 - 19. How convincing do you find his proof? Where and how does he make use of the assumption that the graph is connected in his proof?
10. Below is the sketch of a modern proof of the 'if' direction of the main theorem. The first published proof of this direction is due to the German mathematician Carl Hierholzer (1840 - 1871); following Hierholzer's premature death, this proof was prepared for publication by a colleague and appeared in 1873 [[28]]. **Complete the proof sketch below** by filling in the missing details. (*Specific questions that you will need to address in your completed proof are indicated in italics.*)

Note: You may make use of the lemmas that are provided (with proofs) in the appendix of this project to do so.

CLAIM:

If G is connected and has no vertices of odd degree, then G contains an Euler circuit.

PROOF:

Suppose G is connected and has no vertices of odd degree.

We show that G contains an Euler circuit as follows:

CASE I Consider the case where every edge in G is a loop.

- Since every edge in G is a loop, G must contain only one vertex.
How do we know a connected graph in which every edge in G is a loop contains only one vertex?
- Since every edge in G is a loop on the single vertex v , the graph G must contain an Euler circuit.
What will an Euler circuit in a connected graph on the single vertex v look like as a sequence of alternating vertices and edges?

CASE II Consider the case where at least one edge in G is not a loop.

- Choose any vertex v in G that is incident on at least one edge that is not a loop.
- Let u and w be any vertices adjacent to v .
How do we know two such vertices exist?
- Let W be a simple path from v to w that does not use the edge $\{vw\}$.
How do we know there is a walk from v to w that does not use this edge?
(You may wish to consider what happens in the case where every walk from v to w uses the edge $\{vw\}$; what happens to the graph when the edge $\{vw\}$ is removed?)
Why can we assume that this walk is, in fact, a simple path?
- Use W to obtain a circuit C starting and ending at v .
How is this done?
- Consider the two cases:
 - C uses every edge of G .
Why are we now done?
 - C does not use every edge in G .
 - * Consider the graph G' obtained by removing the edges of C from the graph G along with any vertices that are isolated by doing so. Note that G' is connected and has only vertices of even degree.
How do we know that G' is connected and has only vertices of even degree?
 - * Select a vertex v' in G' which appears in C .
How do we know that such a vertex exists?
 - * Repeat the process outlined above to obtain a circuit C' in G' , and combine C with C' to obtain a new circuit C_1 .
How do we combine the circuits C and C' from our construction into a single circuit? How do we know that the combined walk C_1 is a circuit? How do we know that the combined circuit C_1 does not contain any repeated edges?

- * Repeat this process as required until a circuit is obtained that includes every edge of G .

How do we know this process will eventually terminate?

- 11 Now write a careful (modern) proof of the ‘only if’ direction. Begin by assuming that G is a connected graph which contains an Euler circuit. Then prove that G has no vertices of odd degree.
- 12 Finally, give a careful (modern) proof of the corollary.

APPENDIX: Lemmas used in proving Euler’s Theorem

LEMMA I

For every graph G , if W is a walk in G that has repeated edges, then W has repeated vertices.

PROOF

Let G be a graph and W a walk in G that has a repeated edge e . Let v and w be the endpoint vertices of e . If e is a loop, note that $v = w$, and v is a repeated vertex of W since the sequence ‘ vev ’ must appear somewhere in W . Thus, we need only consider the case where e is not a loop and $v \neq w$. In this case, one of following must occur:

1. The edge e is immediately repeated in the walk W . That is, W includes a segment of the form ‘ $vewev$ ’ a segment of the form ‘ $wewew$ ’.
2. The edge e is not immediately repeated, but occurs later in the walk W and in the same order. That is, either W includes a segment of the form ‘ $vew \dots vew$ ’ or W includes a segment of the form ‘ $wew \dots wev$ ’.
3. The edge e is not immediately repeated, but occurs later in the walk W in the reverse order. That is, either W includes a segment of the form ‘ $vew \dots wev$ ’ or W includes a segment of the form ‘ $wew \dots vew$ ’.

Since one of the vertices v or w is repeated in the first case, while both the vertices v and w are repeated in the latter two cases, this completes the proof.

COROLLARY

For every graph G , if W is a walk in G that has no repeated vertices, then W has no repeated edges.

PROOF

This is the contrapositive of Lemma I.

LEMMA II

If G is a connected graph, then every pair of vertices of G is connected by a simple path.

PROOF

Let G be a connected graph. Let u and w be any arbitrary vertices in G . Since G is connected, we know G contains a walk W from u to w . Denote this walk by the sequence ' $v_0e_0v_1e_1 \dots v_n e_n v_{n+1}$ ', where e_0, e_1, \dots, e_n denote edges, v_0, \dots, v_{n+1} denote vertices with $v_0 = u$ the starting vertex and $v_{n+1} = w$ the ending vertex.

Note that W may include repeated vertices. If so, construct a new walk W' from u to w as follows:

- Let v be the first repeated vertex in the walk W . Then $v = v_i$ and $v = v_j$ for some $i < j$. To construct the new walk W' , delete the segment of the original walk between the first occurrence of v and its next occurrence, including the second occurrence of v . That is, replace

$$\underbrace{v_0}_u e_1 v_1 e_2 \dots v_{i-1} e_{i-1} \overbrace{v_i}^v \underbrace{e_i v_{i+1} e_{i+1} \dots e_{j-1} v_{j-1} e_j}_{\text{delete}} \overbrace{v_j}^v e_{j+1} v_{j+1} \dots v_{n-1} e_{n-1} v_n e_n \underbrace{v_{n+1}}_w$$

by

$$u e_1 v_1 e_2 \dots v_{i-1} e_{i-1} \overbrace{v_i}^v e_{j+1} v_{j+1} \dots v_{n-1} e_{n-1} v_n e_n w$$

Since ' $v e_{j+1}$ ' appeared in the original walk W , we know the edge e_{j+1} is incident on the vertex $v = v_i$. Thus, the new sequence of alternating edges and vertices is also a walk from $u = v_0$ to $w = v_{n+1}$.

(Also note that if $j = n + 1$, then the repeated vertex was $w = v_{n+1}$ and the walk now ends at v_i , where we know that $v_i = v_j = v_{n+1} = w$; thus, the new walk also ends at w .)

- If the new walk W' contains a repeated vertex, we repeat the above process. Since the sequence is finite, we know that we will obtain a walk with no repeated vertices after a finite number of deletions.

In this way, we obtain a new walk S from u to w that contains no repeated vertices. By the corollary to Lemma I, it follows that S contains no repeated edges. Thus, by definition of simple path, S is a simple path from u to w . Since u and w were arbitrary, this completes the proof.

11 Early Writings on Graph Theory: Hamiltonian Circuits and The Icosian Game

Janet Heine Barnett³⁶

Problems that are today considered to be part of modern graph theory originally appeared in a variety of different connections and contexts. Some of these original questions appear little more than games or puzzles. In the instance of the ‘Icosian Game’, this seems quite literally true. Yet for the game’s inventor, the Icosian Game encapsulated deep mathematical ideas which we will explore in this project.

Sir William Rowan Hamilton (1805 - 1865) was a child prodigy with a gift for both languages and mathematics. His academic talents were fostered by his uncle James Hamilton, an Anglican clergyman with whom he lived from the age of 3. Under his uncle’s tutelage, Hamilton mastered a large number of languages — including Latin, Greek, Hebrew, Persian, Arabic and Sanskrit — by the age of 10. His early interest in languages was soon eclipsed by his interests in mathematics and physics, spurred in part by his contact with an American calculating prodigy. Hamilton entered Trinity College in Dublin in 1823, and quickly distinguished himself. He was appointed Astronomer Royal of Ireland at the age of 22 based on his early work in optics and dynamics. Highly regarded not only by his nineteenth century colleagues, Hamilton is today recognized as a leading mathematician and physicist of the nineteenth century.

In mathematics, Hamilton is best remembered for his creation of a new algebraic system known as the ‘quarternions’ in 1843. The system of quarternions consists of ‘numbers’ of the form $Q = a + bi + cj + dk$ subject to certain basic ‘arithmetic’ rules. The project that led Hamilton to the discovery of quarternions was the search for an algebraic system that could be reasonably interpreted in the three-dimensional space of physics, in a manner analogous to the interpretation of the algebra of complex numbers $a + bi$ in a two-dimensional plane. Although this geometrical interpretation of the complex numbers is now standard, it was discovered by mathematicians only in the early 1800’s and thus was relatively new in Hamilton’s time. Hamilton was one of several nineteenth century British mathematicians interested in developing a purely *algebraic* foundation for complex numbers that would capture the essence of this geometrical interpretation. His algebraic development of the complex numbers as ordered pairs of real numbers (a, b) subject to certain operations appeared in a landmark 1837 essay entitled “Theory of Conjugate Functions, or Algebraic Couples; with a Preliminary and Elementary Essay on Algebra as the Science of Pure Time.”

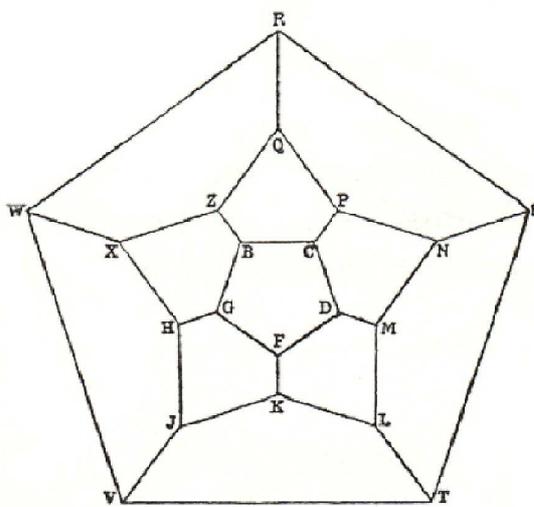
Hamilton concluded his 1837 essay with a statement concerning his hope that he would soon publish a similar work on the algebra of triplets. After years of unsuccessful work on this problem, Hamilton was able to solve it in 1843 only by abandoning the property of commutativity. For example, two of the basic multiplication rules of the quarternion system are $ij = k$ and $ji = -k$, so that $ij \neq ji$. Hamilton also replaced ‘triplets’ by the ‘four-dimensional’ quarternion $a + bi + cj + dk$. Soon after Hamilton’s discovery, physicists realized that only the ‘vector part’ $bi + cj + dk$ of a quarternion was needed to represent three-dimensional space. Although vectors replaced the use of quarternions in physics by the end of the nineteenth century, the algebraic system of vectors retains the non-commutativity of quarternions.

Today’s students of mathematics are familiar with a variety of non-commutative algebraic operations, including vector cross-product and matrix multiplication. In their day, however, Hamilton’s quarternions constituted a major breakthrough comparable to the discovery of non-Euclidean geometry. Immediately following Hamilton’s 1843 announcement of his discovery, at least seven other

³⁶Department of Mathematics; Colorado State University - Pueblo; Pueblo, CO 81001 - 4901; janet.barnett@colostate-pueblo.edu.

new numbers systems were discovered by several other British algebraists. In the 'Icosian Game', Hamilton himself developed yet another example of a non-commutative algebraic system. In this project, we explore both the algebra of that system and the graph theoretical notion of 'Hamiltonian circuit' on which Hamilton's interpretation of this algebra is based. The idea for the game was first exhibited by Hamilton at an 1857 meeting of the British Association in Dublin [27], and later sold for 25 pounds to 'John Jacques and Son,' a wholesale dealer in games. We begin with the preface to the instructions pamphlet which Hamilton prepared for marketing of the game in 1859 [3, pp. 32 - 35].

THE ICOSIAN GAME



In this new Game (invented by Sir WILLIAM ROWAN HAMILTON, LL.D., &c., of Dublin, and by him named *Icosian* from a Greek word signifying 'twenty') a player is to place the whole or part of a set of twenty numbered pieces or men upon the points or in the holes of a board, represented by the diagram above drawn, in such a manner as always to proceed *along the lines* of the figure, and also to fulfill certain *other* conditions, which may in various ways be assigned by another player. Ingenuity and skill may thus be exercised in *proposing* as well as in *resolving* problems of the game. For example, the first of the two players may place the first five pieces in any five consecutive holes, and then require the second player to place the remaining fifteen men consecutively in such a manner that the succession may be *cyclical*, that is, so that No. 20 may be adjacent to No. 1; and it is always possible to answer any question of this kind. Thus, if B C D F G be the five given initial points, it is allowed to complete the succession by following the alphabetical order of the twenty consonants, as suggested by the diagram itself; but after placing the piece No. 6 in hole H, as before, it is *also* allowed (by the supposed conditions) to put No. 7 in X instead of J, and then to conclude with the succession, W R S T V J K L M N P Q Z. Other Examples of Icosian Problems, with solutions of some of them, will be found in the following page.

In graph theoretic terminology, the holes of the game board are referred to as *vertices* (singular: *vertex*) and the lines that join two holes (vertices) are called *edges*. The collection of vertices and

edges in a given relationship (as represented by a diagram such as the game board) is called a *graph*. Two vertices that are joined by an edge in the graph are said to be *adjacent*. Thus, the instruction ‘always to proceed *along the lines* of the figure’ requires the player to find a sequence of adjacent vertices; such a sequence is known as a *path*. In the case where no vertex is repeated in the sequence, the path is said to be a *simple path*. In the case where every vertex of the graph is used exactly once in the sequence, the path is said to be a *Hamiltonian path*. The term *cycle* is now used to describe what Hamilton referred to as a ‘cyclical’ path.

1. Explain why the rules of the Icosian Game require players to always find a simple path.
2. Use modern terminology to formally define the terms *cycle* and *Hamiltonian cycle*.

Following the preface, Hamilton includes several examples of Icosian Problems in the Instruction Pamphlet. We consider only the first two problems as a means of familiarizing ourselves with the concepts of Hamiltonian cycle and Hamiltonian path.

EXAMPLES OF ICOSIAN PROBLEMS

FIRST PROBLEM

*Five initial points are given; cover the board, and finish cyclically. (As hinted in the preceding page, a succession is said to be *cyclical* when the *last* piece is adjacent to the *first*.)*

[This problem is always possible in at least two, and sometimes in four, different ways. Two examples have been assigned: the following are a few others.]

Example 3. Given B C P N M as initial: two solutions exist; one is the succession, D F K L T S R Q Z X W V J H G; the other is D F G H X W V J K L T S R Q Z.

Example 4. Five initials, L T S R Q. Four solutions.

Example 5. Five initials, J V T S R. Two solutions.

3. Explain why Hamilton’s first problem is equivalent to the problem of finding a Hamilton circuit beginning from a given initial sequence of five vertices.
4. In *Example 3*, Hamilton specifies *BCPNM* as the first five vertices in the desired circuit. He then claims that the two solutions listed in the example are the *only* two solutions of this particular problem. Prove that these are in fact the only two solutions by completing the details of the following argument. Include copies of the diagram illustrating each step of the argument in a different color as part of your proof.
 - (a) Explain why the initial conditions for this example imply that the solution to the problem must include either the sequence *RST* or the sequence *TSR*.
 - (b) Explain why the initial conditions for this example imply that the solution to the problem must include either the sequence *RQZ* or the sequence *ZQR*.
 - (c) Explain why we can now conclude that the solution to this problem must include either the sequence *XWV* or the sequence *VWX*.
 - (d) Explain why the initial conditions for this example imply that the solution to the problem must include either the sequence *FD M* or the sequence *MDF*.

- (e) Explain why we can now conclude that the solution to this problem must include either the sequence KLT or the sequence TLK .
 - (f) Use the information from above concerning which edges and vertices we know must be part of the solution to prove that the two circuits Hamilton lists are the only solutions to the problem.
5. In *Example 4*, Hamilton claims there are four Hamiltonian circuits that begin with the vertices $LTSRQ$. Find them. (You do not need to prove these are the only four.)

Although Hamilton claims that every initial sequence of five vertices will lead to at least two solutions (and possibly four) within the Icosian Game, he does not offer a proof of this claim. Nor does he claim that this is true of all graphs.

6. Show that it is not true of every graph that any initial sequence of five vertices will lead to at least one Hamiltonian circuit by finding an example of a graph with at least 5 vertices that has no Hamiltonian circuit. Prove that your graph does not contain a Hamiltonian circuit.

The next question pertains to Hamilton's second problem, which Hamilton describes in the pamphlet as follows.

EXAMPLES OF ICOSIAN PROBLEMS (continued)

SECOND PROBLEM

Three initial points are given; cover the board non-cyclically. (A succession is said to be *non-cyclical* when the *last* piece is *not* adjacent to the first.)

[This problem is sometimes soluble in only *one* way; sometimes in only *two* ways; sometimes in *four* ways; and sometimes it is not soluble at all, as will be seen in the following examples.]

Example 6. Three initial points, $B\ C\ D$; cover, and end with T . There is in this case only one solution, namely, $F\ G\ H\ X\ Z\ Q\ P\ N\ M\ L\ K\ J\ V\ W\ R\ S\ T$.

Example 7. Same initials; cover, and end with W . Two solutions.

Example 8. Same initials; cover, and end with J . Two solutions.

[The same number of solutions exists, if it be required, having the same three initials, to end with K , or L , or N , or V .]

Example 9. Same initials; cover, and end with R . Four solutions.

Example 10. Same initials; cover, and end with M . Impossible.

[The same result, if it be required to end with F , or H , or Q , or S , or X .]

7. In *Example 10*, Hamilton claims the problem of finding a 'non-cyclical' path that uses all vertices beginning with BCD and ending with M is impossible. Prove that he is correct.

The Icosian Game and Non-Commutative Algebra

We now turn to the portion of Hamilton's pamphlet which links the Icosian Game to a non-commutative algebra.

HINTS ON THE ICOSIAN CALCULUS, OF WHICH THE ICOSIAN GAME IS DESIGNED TO BE AN ILLUSTRATION.

I. In a "MEMORANDUM respecting a New System of Roots of Unity," which appeared in the *Philosophical magazine* for December 1856, Sir W. R. Hamilton expressed himself nearly as follows (a few words only being here omitted):

'I have lately been led to the conception of a new system, or rather *family of systems*, of *non-commutative roots of unity*, which are entirely distinct from the *i j k* of quaternions, though having some general analogy thereto; and which admit, even more easily than the quaternion symbols do, of geometrical interpretation. In the system which seems at present to be the most interesting one among those included in this new family, I assume three symbols, ι, κ, λ , such that $\iota^2 = 1, \kappa^3 = 1, \lambda^5 = 1, \lambda = \iota\kappa$; where $\iota\kappa$ must be *distinguished* from $\kappa\iota$, since otherwise we should have $\lambda^6 = 1, \lambda = 1$. As a very simple *specimen* of the symbolical conclusions deduced from these fundamental assumptions I may mention that if we make $\mu = \iota\kappa^2 = \lambda\iota\lambda$, we shall have also $\mu^5 = 1, \lambda = \mu\iota\mu$; so that μ is a new fifth root of reciprocity. A long train of such symbolical deductions is found to follow; and every one of the results may be *interpreted* as having reference to the passage from *face to face* (or from corner to corner) of the *icosahedron* (or of the dodecahedron): on which account, I am at present disposed to give the name of 'Icosian Calculus' to this new system of symbols, and of rules for their operations.'

The system of '*non-commutative roots of unity*' described above employs three symbols ι, κ, λ subject to the following (non-commutative) rules:

$$\iota^2 = 1, \kappa^3 = 1, \lambda^5 = 1, \lambda = \iota\kappa, \iota\kappa \neq \kappa\iota$$

The symbol '1' represents the identity, so that $1\iota = \iota 1 = \iota$, $1\kappa = \kappa 1 = \kappa$, and $1\lambda = \lambda 1 = \lambda$. In Part II of Hints on the Icosian Calculus, Hamilton describes in detail how to interpret his system of '*non-commutative roots of unity*' within the Icosian Game. First, consider only the symbolic action of ι, κ, λ as defined by the above multiplication rules to complete Questions H and I.

8. Prove symbolically that $\kappa = \iota\lambda$.

9. Prove symbolically that $\iota\kappa^2 = \lambda\iota\lambda$.

(This shows that it makes sense to define the new symbol μ by $\mu = \iota\kappa^2 = \lambda\iota\lambda$.)

Extra Credit Question: Show symbolically that $\mu^5 = 1$.

We now consider Hamilton's interpretation of this algebraic system within the Icosian Game.

HINTS ON THE ICOSIAN CALCULUS (continued)

II. In a LITHOGRAPH, which was distributed in Section A of the British Association, during its Meeting at Dublin in 1857, Sir W. R. H. pointed out a few other symbolical results of

the same kind: especially the equations $\lambda\mu^2\lambda = \mu\lambda\mu$, $\mu\lambda^2\mu = \lambda\mu\lambda$, $\lambda\mu^3\lambda = \mu^2\mu\lambda^3\mu = \lambda^2$; and the formula $(\lambda^3\mu^3(\lambda\mu)^2)^2 = 1$, which serves as a *common mathematical type* for the solution of *all cases* of the First Problem of the Game. He also gave at the same time an oral (and hitherto unprinted) account of his rules of *interpretation* of the principal symbols; which rules, with reference to the present Icosian Diagram (or ICOSIAN), may be briefly stated as follows:

1. The operation ι reverses (or reads backwards) a *line* of the figure; changing, for example, BC to CB.
 2. The operation κ causes a line to *turn* in a particular direction round its final point; changing, for instance, BC to DC.
 3. The operation λ changes a line considered as a *side* of a pentagon to the *following side* thereof, proceeding always *right-handedly* for every pentagon except the large or outer one; thus λ changes BC to CD, but SR to RW.
 4. The operation μ is *contrasted* with λ , and changes a line considered as a side of a *different pentagon*, and in the *opposite order* or rotation, to the consecutive side of that *other* pentagon; thus μ changes BC to CP, and SR to RQ; but it changes also RS to ST, whereas λ would change RS to SN.
 5. The only operations employed in the *game* are those marked λ and μ ; but another operation, $\omega = \lambda\mu\lambda\mu\lambda = \mu\lambda\mu\lambda\mu$, having the property that $\omega^2 = 1$, was also mentioned in the Lithograph above referred to; and to complete the present statement of *interpretations*, it may be added that the effect of this operation ω is to change an *edge* of a pentagonal *dodecahedron* to the *opposite edge* of that *solid*; for example, in the diagram, BC to TV.
10. Use the interpretation of ι in (1) to explain why $\iota^2 = 1$.
Begin by looking at the effect of applying the operation ι *twice in succession*, beginning with the edge *BC*. Then explain in general.
 11. Use the interpretation of κ in (2) to explain why $\kappa^3 = 1$.
Begin by looking at the effect of applying the operation κ *three times in succession*, beginning with the edge *BC*. Then look at the effect of applying the operation κ *three times in succession*, beginning with the edge *PN*. Finally, explain in general.
 12. Use the interpretation of λ in (3) to explain why $\lambda^5 = 1$.
Begin by looking at the effect of applying the operation λ *five times in succession*, beginning with the edge *BC*. Then look at the effect of applying the operation λ *five times in succession*, beginning with the edge *SR*. Finally, explain in general.
 13. Use the interpretation of μ in (4) to explain why $\mu^5 = 1$.
Begin by looking at the effect of applying the operation μ *five times in succession*, beginning with the edge *BC*. Then look at the effect of applying the operation μ *five times in succession*, beginning with the edge *RS*. Finally, explain in general. (Note that this provides a geometric solution of the extra credit question stated above, immediately following question 9.)
 14. Beginning with the edge *BC*, use the interpretations given for the four symbols $\iota, \kappa, \lambda, \mu$ to illustrate that $\mu = \iota\kappa^2 = \lambda\iota\lambda$.

Some Closing Remarks

Notice Hamilton's claim that an *algebraic proof* using equations of the type $(\lambda^3\mu^3(\lambda\mu)^2)^2 = 1$ can be used to find all Hamiltonian cycles beginning with a specified initial sequence of five vertices. Notice also the contrast between the graph theoretical proof that you completed in question 4 above, and the proof that we would get of this same result by interpreting this degree twenty equation within the context of the Icosian Game Board. In general, however, it is not viable to associate an algebraic system with an arbitrary graph as a means to find all Hamiltonian circuits within that graph. In fact, as you demonstrated in question 6 above, a graph may contain no Hamiltonian circuits at all. Unlike the situation for other kinds of circuits (e.g., Euler circuits), there is no simple condition on a graph which allows one to determine in all cases whether a Hamiltonian circuit exists or not. In the case that a graph does contain a Hamiltonian circuit, we say the graph is *Hamiltonian*.

The more general question of determining a condition under which a graph is Hamiltonian was first studied by Thomas Penyngton Kirkman (1840 - 1892). Unlike Hamilton, who was primarily interested in the algebraic connections of one specific graph, Kirkman was interested in the general study of 'Hamiltonian circuits' in arbitrary graphs. The rector of a small and isolated English parish, Kirkman presented a paper on this subject to the Royal Society on 6 August 1855. Regrettably, his solution of the problem was incorrect. He did, however, present a second paper in 1856 in which he described a general class of graphs which do not contain such a circuit. Kirkman also studied the existence of Hamiltonian circuits on the dodecahedron, a variation of the Icosian Game which Hamilton also studied. In fact, the two men met once in 1861 when Hamilton visited Kirkman at his rectory. That Hamilton's name became associated with the circuits, and not Kirkman's, appears to be one of the accidents of history, or perhaps a credit to the fame of Hamilton's quaternions and work in mathematical physics.

12 Early Writings on Graph Theory: Topological Connections

Janet Heine Barnett³⁷

The earliest origins of graph theory can be found in puzzles and game, including Euler’s Königsberg Bridge Problem and Hamilton’s Icosian Game. A second important branch of mathematics that grew out of these same humble beginnings was the study of position (“analysis situs”), known today as *topology*³⁸. In this project, we examine some important connections between algebra, topology and graph theory that were recognized during the years from 1845 - 1930.

The origin of these connections lie in work done by physicist Gustav Robert Kirchhoff [1824 - 1887] on the flow of electricity in a network of wires. Kirchhoff showed how the current flow around a network (which may be thought of as a graph) leads to a set of linear equations, one for each circuit in the graph. Because these equations are not necessarily independent, the question of how to determine a complete set of mutually independent equations naturally arose. Following Kirchhoff’s publication of his answer to this question in 1847, mathematicians slowly began to apply his mathematical techniques to problems in topology. The work done by the French mathematician Henri Poincaré [1854 - 1912] was especially important, and laid the foundations of a new subject now known as “algebraic topology.”

This project is based on excerpts from a 1922 paper in which the American mathematician Oswald Veblen [1880 - 1960] shows how Poincaré formalized the ideas of Kirchhoff. An American mathematician born in Iowa, Veblen’s father Andrew Veblen was also a mathematician who taught mathematics and physics at the State University of Iowa. At that time, graduate programs in mathematics were relatively young in the United States. A member of the first generation of American mathematicians to complete their advanced work in the United States rather than Europe, Veblen completed his Ph.D. at the University of Chicago in 1903. He remained there for two years before joining the mathematics faculty at Princeton. In 1930, he became the first faculty member of the newly founded Institute for Advanced Study at Princeton. A talented fund-raiser and organizer, Veblen also served on the Institute’s Board of Trustees in its early years. During the Nazi years, he was instrumental in assisting European mathematicians to find refuge in the United States. Although some American mathematicians, including George Birkhoff (1844 - 1944), voiced opposition to these efforts – fearing that talented young American mathematicians would lose academic positions to the immigrants – the Rockefeller Foundation and other philanthropic bodies provided financial support to these efforts as a means of recruiting world-class mathematicians and scientists to the United States³⁹. Veblen was also instrumental in the establishment of the American Mathematical Society’s *Mathematical Reviews*, a publication aimed at providing researchers with reviews of recent mathematical papers in a timely fashion. Founded during the late 1930’s when the well-known German review journal *Zentralblatt für Mathematik und ihre Grenzgebiete*

³⁷Department of Mathematics; Colorado State University - Pueblo; Pueblo, CO 81001 - 4901; janet.barnett@colostate-pueblo.edu.

³⁸According to Euler, the first person to discuss “analysis situs” was the mathematician and philosopher Gottfried Leibniz [1646 - 1716]. In a 1679 letter to Christian Huygens [1629 - 1695], Leibniz wrote:

I am not content with algebra, in that it yields neither the shortest proofs nor the most beautiful constructions of geometry. Consequently, in view of this, I consider that we need yet another kind of analysis, geometric or linear, which deals directly with position, as algebra deals with magnitude. [3, p. 30]

Although Leibniz himself did not appear to make contributions to the development of *analysis situs*, he did make important contributions the development of another kind of analysis. Today, Leibniz is recognized alongside the mathematician and physicist Isaac Newton [1642 - 1727] as an independent co-inventor of calculus.

³⁹The celebrated physicist Albert Einstein (1879 - 1955), for example, joined the Institute in 1931.

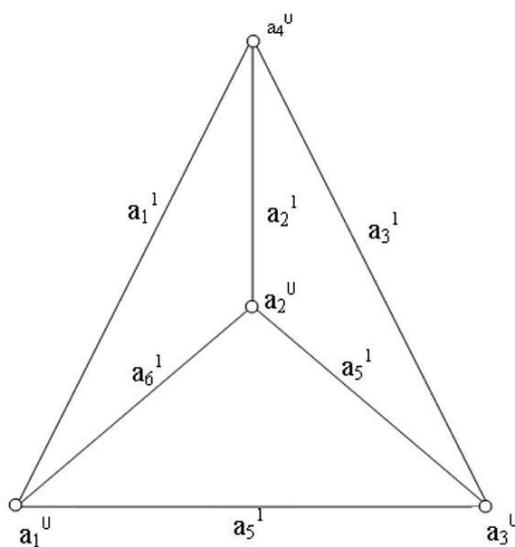
was refusing to publish reviews written by Soviet and Jewish scholars, the *Mathematical Reviews* continues to play an important role in disseminating research results and promoting communication within the mathematical community.

In addition to his administrative and philanthropic work, Veblen was an active researcher who made important contributions in projective and differential geometry in addition to topology, authoring influential books in all three areas. In this project, we examine extracts from his *Analysis Situs*, the first textbook to be written on combinatorial topology. Veblen first presented this work in a series of invited Colloquium Lectures of the American Mathematical Society in 1916. Although he remained interested in topology afterwards, he published little research in this area following the 1922 publication of *Analysis Situs* [68]. The extracts we examine are taken from [3, pp. 136 - 141].

Note: This project assumes the reader is familiar with basic notions of graph theory, including the definition of *isomorphism* and *isomorphism invariant*. Parts of the project (clearly marked as such) also assume familiarity with the basic linear algebra concepts of *rank*, *kernel* and *linear independence*. As needed, the instructor may delete these questions or refer students to a standard linear algebra textbook as needed for review.

Analysis Situs

American Mathematical Society Colloquium Lectures 1916
Symbols for Sets of Cells



[FIG. 8.3.]

- 14 Let us denote the 0-cells of a one-dimensional complex C_1 by $a_1^0, a_2^0, \dots, a_{\alpha_0}^0$ and the 1-cells by $a_1^1, a_2^1, \dots, a_{\alpha_1}^1$.

Any set of 0-cells of C_l may be denoted by a symbol $(x_1, x_2, \dots, x_{\alpha_0})$ in which $x_i = 1$ if a_i^0 is in the set and $x_i = 0$ if a_i^0 is not in the set. Thus, for example, the pair of points a_1^0, a_4^0 in Fig. [8.3] is denoted by $(1, 0, 0, 1)$. The total number of symbols $(x_1, x_2, \dots, x_{\alpha_0})$ is 2^{α_0} . Hence the total number of sets of 0-cells, barring the null-set, is $2^{\alpha_0} - 1$. The symbol for a null-set, $(0, 0, \dots, 0)$ will be referred to as zero and denoted by 0. The marks 0 and 1 which appear in the symbols just defined, may profitably be regarded as residues, modulo 2, i.e., as symbols which may be combined algebraically according to the rules

$$\begin{aligned} 0 + 0 &= 1 + 1 = 0, & 0 + 1 &= 1 + 0 = 1, \\ 0 \times 0 &= 0 \times 1 = 1 \times 0 = 0, & 1 \times 1 &= 1 \end{aligned}$$

Under this convention the sum (mod. 2) of two symbols, or of the two sets of points which correspond to the symbols $(x_1, x_2, \dots, x_{\alpha_0}) = X$ and $(y_1, y_2, \dots, y_{\alpha_0}) = Y$, may be defined as $(x_1 + y_1, x_2 + y_2, \dots, x_{\alpha_0} + y_{\alpha_0}) = X + Y$.

Geometrically, $X + Y$ is the set of all points which are in X or in Y but not in both. For example, if $X = (1, 0, 0, 1)$ and $Y = (0, 1, 0, 1)$, $X + Y = (1, 1, 0, 0)$; i.e., X represents a_1^0 and a_4^0 , Y represents a_2^0 and a_4^0 , and $X + Y$ represents a_1^0 and a_2^0 . Since a_4^0 appears in both X and Y , it is suppressed in forming the sum, modulo 2. This type of addition has the obvious property that if two sets contain each an even number of 0-cells, the sum (mod. 2) contains an even number of 0-cells.

- 15 Any set, S , of 1-cells in C_1 may be denoted by a symbol $(x_1, x_2, \dots, x_{\alpha_1})$ in which $x_i = 1$ if a_i^1 is in the set and $x_i = 0$ if a_i^1 is not in the set. The 1-cells in the set may be thought of as labeled with 1's and those not in the set as labeled with 0's. The symbol is also regarded as representing the one-dimensional complex composed of the 1-cells of S and the 0-cells which bound them. Thus, for example, in Fig. [8.3] the boundaries of two of the faces are $(1, 0, 1, 0, 1, 0)$ and $(1, 1, 0, 0, 0, 1)$. The sum (mod. 2) of two symbols $(x_1, x_2, \dots, x_{\alpha_1})$ is defined in the same way as for the case of symbols representing 0-cells. Correspondingly if C'_1 and C''_1 are one-dimensional complexes which have a certain number (which may be zero) of 1-cells in common and have no other common points except the ends of these 1-cells, the sum $C'_1 + C''_1$ (mod. 2) is defined as the one-dimensional complex obtained by suppressing all 1-cells common to C' and C'' and retaining all 1-cells which appear only in C'_1 or in C''_1 . For example, in Fig. [8.3], the sum of the two curves represented by $(1, 0, 1, 0, 1, 0)$ and $(1, 1, 0, 0, 0, 1)$ is $(0, 1, 1, 0, 1, 1)$ which represents the curve composed of $a_2^1, a_4^1, a_5^1, a_6^1$ and their ends.

1. The following questions are based on Section 14 of Veblen's paper, in which Veblen discusses symbols for sets of 0 - cells.
 - (a) In graph terminology, what is a '0-cell'? What is a '1-cell'?
 - (b) How would the set of points $\{a_1^0, a_3^0, a_4^0\}$ in Fig. 8.3 be represented by Veblen?
 - (c) Identify the set of points in Fig. 8.3 that are represented by the following 4-tuples:
$$W = (0, 1, 1, 0) \qquad Z = (1, 1, 1, 1)$$
 - (d) Find $W + Z$ for $W = (0, 1, 1, 0)$ and $Z = (1, 1, 1, 1)$.
Identify the set of points in Fig. 8.3 that is represented by $W + Z$.

- (e) In the last paragraph of Section 14, Veblen asserts the addition modulo 2 has a certain ‘obvious property.’ What property is this? How obvious is this property? Give a formal proof of the property, and interpret it in terms of sets of ‘0-cells’.
2. The following questions are based on Section 15 of Veblen’s paper, in which Veblen discusses symbols for sets of 1 - cells.

Referring to Fig. 8.3 in Veblen’s paper, let

M be the circuit defined by edges $a_1^1, a_3^1, a_4^1, a_6^1$

N be the circuit defined by edges a_2^1, a_3^1, a_4^1

- (a) How would Veblen represent M and N as 6 - tuples?
 (b) Find $M + N \text{ mod. } 2$, and identify the circuit represented by this sum in Fig. 8.3.

Veblen’s article continues with definitions of two important matrices associated with a graph.

The Matrices H_0 and H_1

- 16** Any one-dimensional complex falls into R_0 sub-complexes each of which is connected. Let us denote these sub-complexes by $C_1^1, C_1^2, \dots, C_1^{R_0}$ and let the notation be assigned in such a way that a_i^0 ($i = 1, 2, \dots, m_1$) are the 0-cells of C_1^1 , a_i^0 ($i = m_1 + 1, m_1 + 2, \dots, m_2$) those of C_1^2 , and so on.

With this choice of notation, the sets of vertices of $C_1^1, C_1^2, \dots, C_1^{R_0}$, respectively, are represented by the symbols $(x_1, x_2, \dots, x_{\alpha_0})$ which constitute the rows of the following matrix.

$$H_0 = \left\| \begin{array}{ccc} \overbrace{11 \dots 1}^{m_1} & \overbrace{00 \dots 0}^{m_2 - m_1} & \overbrace{00 \dots 0}^{\alpha_0 - m_{R_0} - 1} \\ 00 \dots 0 & 11 \dots 1 & 00 \dots 0 \\ \vdots & & \vdots \\ \vdots & & \vdots \\ 00 \dots 0 & 00 \dots 0 & 11 \dots 1 \end{array} \right\| = \|n_{ij}^0\|$$

For most purposes it is sufficient to limit attention to connected complexes. In such cases $R_0 = 1$, and H_0 consists of one row all of whose elements are 1.

- 17** By the definition ... a 0-cell is incident with a 1-cell if it is one of the ends of the 1-cell, and under the same conditions the 1-cell is incident with the 0-cell. The incidence relations between the 0-cells and the 1-cells may be represented in a table or matrix of α_0 rows and α_1 columns as follows: The 0-cells of C_1 having been denoted by a_i^0 , ($i = 1, 2, \dots, \alpha_0$) and the 1-cells by a_j^1 , ($j = 1, 2, \dots, \alpha_1$), let the element of the i th row and the j th column of the matrix be 1 if a_i^0 is incident with a_j^1 and let it be 0 if a_i^0 is not incident with a_j^1 . For example, the table for the linear graph of Fig. [8.3] formed by the vertices and edges of a tetrahedron is as follows:

| | α_1^1 | α_2^1 | α_3^1 | α_4^1 | α_5^1 | α_6^1 |
|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| α_1^0 | 1 | 0 | 0 | 0 | 1 | 1 |
| α_2^0 | 0 | 1 | 0 | 1 | 0 | 1 |
| α_3^0 | 0 | 0 | 1 | 1 | 1 | 0 |
| α_4^0 | 1 | 1 | 1 | 0 | 0 | 0 |

In the case of the complex used ... to define a simple closed curve the incidence matrix is

$$\begin{vmatrix} 1 & 1 \\ 1 & 1 \end{vmatrix}$$

We shall denote the element of the i th row and j th column of the matrix of incidence relations between the 0-cells and 1-cells by η_{ij}^1 and the matrix itself by

$$\|\eta_{ij}^1\| = H_1$$

The i th row of H_1 is the symbol for the set of all 1-cells incident with a_i^0 and the j th column is the symbol for the set of all 0-cells incident with a_j^1 .

The condition which we have imposed on the graph, that both ends of every 1-cell shall be among the α_0 0-cells, implies that every column of the matrix contains exactly two 1's. Conversely, any matrix whose elements are 0's and 1's and which is such that each column contains exactly two 1's can be regarded as the incidence matrix of a linear graph. For to obtain such a graph it is only necessary to take α_0 points in a 3-space, denote them arbitrarily by $a_1^0, a_2^0, \dots, a_{\alpha_0}^0$, and join the pairs which correspond to 1's in the same column successively by arcs not meeting the arcs previously constructed.

- 20 Denoting the connected sub-complexes of C_1 by $C_1^1, C_1^2, \dots, C_1^{R_0}$ as in 16 let the notation be so assigned that $a_1^1, a_2^1, \dots, a_{m_1}^1$ are the 1-cells in C_1 , $a_{m_1+1}^1, a_{m_1+2}^1, \dots, a_{m_2}^1$ the 1-cells in C_2 ; and so on. The matrix H_1 then must take the form

$$\begin{vmatrix} \text{I} & 0 & 0 & 0 \\ 0 & \text{II} & 0 & 0 \\ 0 & 0 & \text{III} & 0 \\ \vdots & \vdots & \vdots & \vdots \end{vmatrix}$$

where all the non-zero elements are to be found in the matrices I, II, III, etc., and I is the matrix of C_1 , II of C_2 , etc. This is evident because no element of one of the complexes C_1^i is incident with any element of any of the others.

There are two non-zero elements in each column of H_1 . Hence if we add the rows corresponding to any of the blocks I, II, etc. the sum is zero (mod. 2) in every column. Hence the rows of H_1 are connected by R_0 linear relations.

Any linear combination (mod. 2) of the rows of H_1 corresponds to adding a certain number of them together. If this gave zeros in all the columns it would mean that there were two or no 1's in each column of the matrix formed by the given rows, and this would mean that any 1-cell incident with one of the 0-cells corresponding to these rows would also be incident with another such 0-cell. These 0-cells and the 1-cells incident with them would therefore form a sub-complex of C_1 which was not connected with any of the remaining 0-cells and 1-cells of C_1 . Hence it would consist of one or more of the complexes C_1^i ($i = 1, 2, \dots, R_0$) and the linear relations with which we started would be dependent on the R_0 relations already found. Hence there are exactly R_0 linearly independent linear relations among the rows of H_1 , so that if ρ_1 is the rank of H_1 ,

$$\rho_1 = \alpha_1 - R_0.$$

3. The following questions are based on Section 16 of Veblen's paper, in which Veblen defines the matrix H_0 .

- (a) In the first paragraph of this section, Veblen assumes that we may label the '0-cells of a one-dimensional complex' in a particular way. What graph isomorphism invariant allows him to make this assumption?
- (b) Find the matrix H_0 for graphs G_3 and G_4 in the appendix. Using Veblen's notation to label the vertices and edges of each graph so that the correspondence to the associated matrix H_0 is clear.

4. The following questions are based on Section 17 of Veblen's paper, in which Veblen defines the incidence matrix H_1 .

- (a) Find the incidence matrix H_1 for graphs G_1 in the appendix.

(b) Sketch and label a graph with incidence matrix $H_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$

(c) Give two reasons why no graph can have incidence matrix $H_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$

- (d) Explain why isomorphic graphs do not necessarily have the same incidence matrix. Is there some way in which we could use incidence matrices to determine if two graphs are isomorphic? Explain.

5. The following questions are based on Section 20 of Veblen's paper, in which Veblen discusses the matrix H_1 for graphs that are not connected.

- (a) In the first paragraph of this section, Veblen asserts that the matrix H_1 must take the form of a block diagonal matrix. Illustrate this by finding the matrix H_1 for graphs G_3 and G_4 in the appendix. Then explain in general why this must be true.
- (b) In the second paragraph of this section, Veblen asserts that the rows of matrix H_1 are related by R_0 linear relations. What does the value R_0 represent? Determine the R_0 linear relations for the incidence matrices H_1 of graphs G_3 and G_4 from part (a) above. Suggestions: Denote the i^{th} row of H_1 by z_i . You may also wish to review Section 14 of Veblen's paper in which he explains the notation 0 as it applies to representing sets of 0 - cells.
- (c) Verify the relationship $\rho_1 = \alpha_0 - R_0$ for the incident matrices of graphs G_3 and G_4 .

Note: You will need to know how to find the rank of a matrix to complete this question; as required, review this concept in a linear algebra textbook, or omit. Since all sums are modulo 2, you will also need to reduce the matrices to determine their rank by hand, rather than using the matrix utility on your calculator or some other computing device.

Returning to Veblen's paper, the next excerpt introduces the concept of a 'one-dimensional circuit'.

One-dimensional Circuits

- 22** A connected linear graph each vertex of which is an end of two and only two 1-cells it called a one-dimensional circuit or a 1-circuit. Any closed curve is decomposed by any finite set of points on it into a 1-circuit. Conversely, it is easy to see that the set of all points on a 1-circuit is a simple closed curve. It is obvious, further, that any linear graph such that each vertex is an end of two and only two 1-cells is either a 1-circuit or a set of 1-circuits no two of which have a point in common.

Consider a linear graph C_1 such that each vertex is an end of an even number of edges. Let us denote by $2n_i$ the number of edges incident with each vertex at a_i^0 . The edges incident with each vertex at may be grouped arbitrarily in n_i pairs no two of which have an edge in common; let these pairs of edges be called the pairs associated with the vertex a_i^0 . Let C'_1 be a graph coincident with C_1 in such a way that (1) there is one and only one point of C'_1 on each point of C_1 which is not a vertex and (2) there are n_i vertices of C'_1 on each vertex a_i^0 of C_1 each of these vertices of C'_1 being incident only with the two edges of C'_1 which coincide with a pair associated with at a_i^0 .

The linear graph C'_1 has just two edges incident with each of its vertices and therefore consists of a number of 1-circuits. Each of these 1-circuits is coincident with a 1-circuit of C_1 , and no two of the 1-circuits of C_1 thus determined have a 1-cell in common. Hence C_1 consists of a number of 1-circuits which have only a finite number of 0-cells in common.

It is obvious that a linear graph composed of a number of closed curves having only a finite number of points in common has an even number of 1-cells incident with each vertex. Hence a necessary and sufficient condition that C_1 consist of a number of 1-circuits having only 0-cells in common is that each 0-cell of C_1 be incident with an even number of 1-cells. A set of 1-circuits having only 0-cells in common will be referred to briefly as a set of 1-circuits.

6. The following questions are based on Section 22, in which Veblen discusses one-dimensional circuits.
- (a) First paragraph: Note that Veblen is now only interested in connected graphs. According to his definition of '1-circuits', can 1-circuits repeat edges? vertices?
 - (b) Second paragraph: Veblen outlines a method for constructing a new graph C'_1 from a given linear graph C_1 .
 - i. What conditions on the graph C_1 are required for this construction?
 - ii. The resulting graph C'_1 will depend on how we pair up the edges at each vertex. Illustrate this fact using graph G_2 from the appendix. That is, apply Veblen's construction method *twice* to graph G_2 , using different pairings of edges each time.
 - (c) In the final paragraph of Section 22, Veblen states the conclusion of this section in the form of a 'necessary and sufficient' statement. To what (familiar) theorem from graph theory is this conclusion related? Explain, and comment on Veblen's proof.

Proceeding to Section 24 of Veblen's paper, an algebraic representation of one-dimensional circuits is introduced.

- 24** Let us now inquire under what circumstances a symbol $(x_1, x_2, \dots, x_{\alpha_1})$ for a one-dimensional complex contained in C_1 will represent a 1-circuit or a system of 1-circuits. Consider the sum

$$\eta_{i1}^1 x_1 + \eta_{i2}^1 x_2 + \dots + \eta_{i\alpha_1}^1 x_{\alpha_1}$$

where the coefficients η_{ij}^1 are the elements of the i th row of H_1 . Each term $\eta_{ij}^1 x_j$ of this sum is 0 if a_j^1 is not in the set of 1-cells represented by $(x_1, x_2, \dots, x_{\alpha_1})$ because in this case $x_j = 0$; it is also zero if a_j^1 is not incident with a_i^0 because $\eta_{ij}^1 = 0$ in case. The term $\eta_{ij}^1 x_j = 1$ if a_j^1 is incident with a_i^0 and in the set represented by $(x_1, x_2, \dots, x_{\alpha_1})$ because in this case $\eta_{ij}^1 = 1$ and $x_j = 1$. Hence there are as many non-zero terms in the sum as there are 1-cells represented by $(x_1, x_2, \dots, x_{\alpha_1})$ which are incident with a_i^0 . Hence by §22 the required condition is that the number of non-zero terms in the sum must be even. In other words if the x 's and η_{ij}^1 's are reduced modulo 2 as explained in §14 we must have

$$(H_1) \quad \sum_{j=1}^{\alpha_1} \eta_{ij}^1 x_j = 0 \quad (i = 1, 2, \dots, \alpha_0)$$

if and only if $(x_1, x_2, \dots, x_{\alpha_1})$ represents a 1-circuit or set of 1-circuits. The matrix of this set of equations (or congruences, mod. 2) is H_1 .

7. Note that, in the algebraic representation of one-dimensional circuits discussed in Section 24, the number of non-zero terms in the sum $\eta_{i1}^1 x_1 + \eta_{i2}^1 x_2 + \dots + \eta_{i\alpha_1}^1 x_{\alpha_1}$ corresponds to the degree of the vertex a_i^0 .
- (a) Veblen's conclusion in the penultimate sentence of Section 24 could be re-stated in terms of solutions to the matrix-vector equation $H_1 \vec{v} = \vec{0}$, where H_1 is the incidence matrix of the graph. Complete the following example to illustrate this conclusion:
Let H_1 be the incidence matrix for graph G_1 from the appendix. (See part a of question 4 above.) Let $\vec{X} = (1, 0, 1, 1, 0)$ and $\vec{Y} = (1, 0, 0, 1, 1)$. Determine the matrix-vector products (modulo 2) $H_1 \vec{X}$ and $H_1 \vec{Y}$. Use these results to determine whether (i) \vec{X} represents a set of 1-circuits in the graph G_1 ; and (ii) \vec{Y} represents a set of 1-circuits in the graph G_1 . Explain.
- (b) What advantage might there be in representing graphs and circuits in terms of matrices and linear equations?

In the final excerpt from Veblen's paper below, a connection is made between the matrix H_1 and the problem of determining whether there exists a complete set of 1-circuits that will generate all possible 1-circuits for a given graph.

- 25** If the rank of the matrix H_1 of the equations (H_1) be ρ_1 the theory of linear homogeneous equations (congruences, mod. 2) tells us that there is a set of $\alpha_1 - \rho_1$ linearly independent solutions of (H_1) upon which all other solutions are linearly dependent. This means geometrically that *there exists a set of $\alpha_1 - \rho_1$ 1-circuits or systems of 1-circuits from which all others can be obtained by repeated applications of the operation of adding (mod. 2) described in §14.* We shall call this a complete set of 1-circuits or systems of 1-circuits.

Since $\rho_1 = \alpha_0 - R_0$ (§20), the number of solutions of (H_1) in a complete set is

$$\mu = \alpha_1 - \alpha_0 + R_0,$$

where μ is the cyclomatic number For the sake of uniformity with a notation used later on we shall also denote μ by $R_1 - 1$. Thus we have

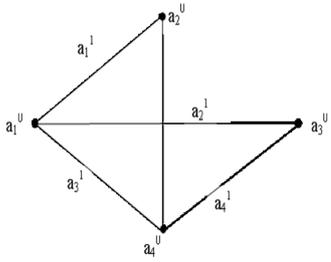
$$\alpha_0 - \alpha_1 = 1 + R_0 - R_1.$$

8. The following questions are based primarily on Section 25, in which Veblen discusses how to use the matrix H_1 to determine if there is a complete set of 1-circuits that will generate all possible 1-circuits for a given graph.

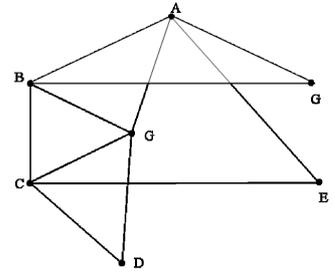
Note: You will need to know about null space, basis sets and rank to complete this question; as required, review this concept in a linear algebra textbook, or omit.

- (a) Explain Veblen's conclusion in terms of null space of the matrix H_1 . You may find it helpful to review section 24 of Veblen's paper, and project question 7 above.
- (b) Consider the incidence matrix H_1 for graph G_5 on the attached graph sheet.
 - Find a basis for the null space of that matrix, again using modulo 2 sums.
 - Use your null-space basis to identify a complete system of 1-circuits for this graph.
 - Write the circuit $C = (0, 0, 1, 1, 1, 0, 1)$ as the sum of circuits in your complete system of 1-circuits for this graph.

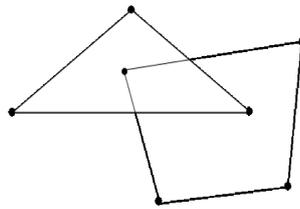
APPENDIX: Graphs for Veblen Project Questions



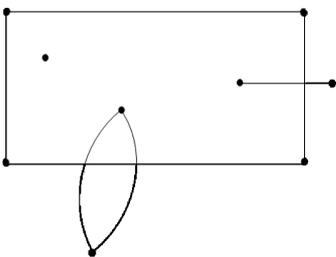
Graph G_1



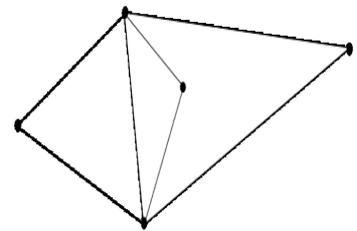
Graph G_1



Graph G_3



Graph G_4



Graph G_5

References

- [1] Aiton, E. J., *Leibniz: A Biography*, Adam Hilger, Boston, 1985.
- [2] Bezhanishvili, G., Leung, H., Lodder, J., Pengelley, D., Ranjan, D., “Teaching Discrete Mathematics via Primary Historical Sources,” www.math.nmsu.edu/hist_projects/
- [3] Biggs, N., Lloyd, E., Wilson, R., *Graph Theory: 1736–1936*, Clarendon Press, Oxford, 1976.
- [4] Cantor, G., *Beiträge zur Begründung der transfiniten Mengenlehre. I*, *Mathematische Annalen* **46** (1895), 481–512.
- [5] Cantor, G., *Beiträge zur Begründung der transfiniten Mengenlehre. II*, *Mathematische Annalen* **49** (1897), 207–246.
- [6] Cantor, G., *Contributions to the Founding of the Theory of Transfinite Numbers*, Philip Jourdain (translator), Dover Publications Inc., New York, 1952.
- [7] Ching, J., Oxtoby, W. G., *Moral Enlightenment: Leibniz and Wolff on China*, Steyler Verlag, Nettetal, 1992.
- [8] Church, A., “An Unsolvable Problem of Elementary Number Theory, Preliminary Report (abstract),” *Bull. Amer. Math. Soc.*, **41** (1935), 332–333.
- [9] Church, A., “An Unsolvable Problem of Elementary Number Theory,” *Amer. Journal Math.*, **58** (1936), 345–363. This paper with a short foreword by Davis was reprinted on pages 88–107 of [13].
- [10] Church, A., “A Note on the Entscheidungsproblem,” *Journal of Symbolic Logic*, **1** (1936), 40–41.
- [11] Church, A., *Introduction to Mathematical Logic*, Princeton University Press, Princeton, New Jersey, 1996.
- [12] Crowe, M. J., *A History of Vector Analysis: The Evolution of the Idea of a Vectorial System*, Dover Publications, New York, 1994.
- [13] Davis, M., *The Undecidable. Basic Papers on Undecidable Propositions, Unsolvable Problems and Computable Functions*, Martin Davis (editor), Raven Press, Hewlett, N.Y., 1965.
- [14] Davis, M., “Why Gödel Didn’t Have Church’s Thesis,” *Inform. and Control*, **54** (1982), no. 1-2, 3–24.
- [15] Dunham, W., *Journey Through Genius. The Great Theorems of Mathematics*, John Wiley & Sons Inc., New York, 1990.
- [16] *Encyclopædia Britannica*, Chicago, 1986.
- [17] Euler, L., *Leonhard Euler und Christian Goldbach, Briefwechsel 1729–1764*, Juskevic, A. P., Winter, E. (editors), Akademie Verlag, Berlin, 1965.
- [18] Euler, L., *Novi Commentarii Academiae Scientiarum Imperialis Petropolitane* **7** (1758–59), p. 9–28.

- [19] Gerhardt, C. I., (editor) *Die Philosophischen Schriften von Leibniz*, vol. VII, Olms, Hildesheim, 1965.
- [20] Gerhardt, C. I., (editor) *G. W. Leibniz Mathematische Schriften*, Vol. VII, Olms, Hildesheim, 1962.
- [21] Gillispie, C. C., Holmes, F. L., (editors) *Dictionary of Scientific Biography*, Scribner, New York, 1970.
- [22] Glaser, A., *History of Binary and Other Nondecimal Numeration*, Anton Glaser, Southampton, PA, 1971.
- [23] Gödel, K., “Über Formal Unentscheidbare Sätze der Principia Mathematica und Verwandter Systeme I,” *Monatsh. Math. Phys.*, **38** (1931), 173–198. The English translation of this paper by Mendelson with foreword by Davis was reprinted on pages 4–38 of [13].
- [24] Gödel, K., *On Undecidable Propositions of Formal Mathematical Systems*, Mimeographed lecture notes by S. C. Kleene and J. B. Rosser, Institute for Advanced Study, Princeton, N.J., 1934. This lecture note with foreword by Davis and postscriptum by Gödel were reprinted in [13], pages 39–74.
- [25] Goldstine, H. H., *The Computer from Pascal to von Neumann*, Princeton University Press, Princeton, New Jersey, 1972.
- [26] Grattan-Guinness, I., *The Search for Mathematical Roots, 1870–1940: Logics, Set Theories and the Foundations of Mathematics from Cantor through Russell to Gödel*, Princeton University Press, Princeton, New Jersey, 2000.
- [27] Hamilton, W. R., “Account of the Icosian Game,” *Proc. Roy. Irish. Acad.* **6** (1853-7), 415–416.
- [28] Hierholzer, C. , “Über die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechnung zu umfahren,” *Math. Ann.* **6** (1873), 30–32.
- [29] Hilbert, D., *Gesammelte Abhandlungen*, Vol. III, Chelsea Publishing Co., New York, 1965.
- [30] Hilbert, D., “Mathematical Problems,” Newson M., (translator) *Bulletin of the American Mathematical Society*, **8** (1902), 437–439.
- [31] Hilbert, D., “Probleme der Grundlegung der Mathematik,” *Mathematische Annalen*, **102**, (1930), 1–9.
- [32] Hilbert, D., Ackermann, W., *Grundzüge der Theoretischen Logik*, Dover Publications, New York, 1946.
- [33] Hilbert, D., Ackermann, W., *Principles of Mathematical Logic*, L. Hammond, G. Leckie, F. Steinhardt, translators, Chelsea Publishing Co., New York, 1950.
- [34] Hollingdale, S., *Makers of Mathematics*, Penguin Books, New York, 1994.
- [35] Hopcroft, J. E., and Ullman, J. D., *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, Reading, MA, 1979.
- [36] James, I., *Remarkable Mathematicians: From Euler to von Neumann*, Cambridge University Press, Cambridge, 2002.

- [37] Jolley, N., (editor) *The Cambridge Companion to Leibniz*, Cambridge University Press, Cambridge, 1995.
- [38] Katz, V., *A History of Mathematics: An Introduction*, Second Edition, Addison-Wesley, New York, 1998.
- [39] Kleene, S. C., “General Recursive Functions of Natural Numbers,” *Math. Ann.*, **112** (1936), 727–742. This paper with a short foreword by Davis was reprinted on pages 236–253 of [13].
- [40] Kleene, S. C., “ λ -Definability and Recursiveness,” *Duke Math. Journal*, **2** (1936), 340–353.
- [41] Kleene, S. C., “On Notation for Ordinal Numbers,” *Journal of Symbolic Logic*, **3** (1938), 150–155.
- [42] Kleene, S. C., “Recursive Predicates and Quantifiers,” *Trans. Amer. Math. Soc.*, **53** (1943), 41–73. This paper with foreword by Davis was reprinted on pages 254–287 of [13].
- [43] Kleene, S. C., *Introduction to Metamathematics*, D. Van Nostrand Co., Inc., New York, 1952.
- [44] Kleene, S. C., “Representation of Events in Nerve Nets and Finite Automata”, in *Automata Studies*, Shannon, S. C., McCarthy, J. (editors) Princeton University Press, NJ, 1956, 3–41.
- [45] Kleene, S. C., “Origins of Recursive Function Theory,” *Ann. Hist. Comput.*, **3** (1981), no. 1, 52–67.
- [46] Kozen, D. C., *Automata and Computability*, Springer-Verlag, New York, 1997.
- [47] Lamé, G., “Un polygone convexe étant donné, de combien de manières peut-on le partager en triangles au moyen de diagonales?,” *Journal de Mathématiques Pures et Appliquées*, **3** (1838), 505–507.
- [48] Laubenbacher, R., Pengelley, D., *Mathematical Expeditions: Chronicles by the Explorers*, Springer Verlag, New York, 1999.
- [49] Leibniz, G. W., “Explication de l’arithmétique binaire, qui se sert des seuls caractères 0 et 1, avec des remarques sur son utilité, et sur ce qu’elle donne le sens des anciennes figures Chinoises de Fohy,” *Memoires de l’Académie Royale des Sciences*, **3** (1703), 85–89.
- [50] Martzloff, J.-L., *A History of Chinese Mathematics*, Wilson, S.S. (translator), Springer Verlag, Berlin, 1997.
- [51] McCulloch, W. S., and Pitts, W., “A Logical Calculus of Ideas Immanent in Nervous Activity”, *Bull. Math. BioPhys.*, **5** (1943), 115–133.
- [52] Needham, J., *Science and Civilisation in China*, vol. 3, Cambridge University Press, Cambridge, 1959.
- [53] Pascal, B., “Treatise on the Arithmetical Triangle,” in *Great Books of the Western World*, Mortimer Adler (editor), Encyclopædia Britannica, Inc., Chicago, 1991.
- [54] Post, E. L., “Finite Combinatory Processes, Formulation I,” *Journal of Symbolic Logic*, **1** (1936), 103–105. This paper with a short foreword by Davis was reprinted on pages 288–291 of [13].

- [55] Rabin, M. O., Scott, D., “Finite Automata and Their Decision Problems”, *IBM Journal of Research and Development*, **3** (1959), 114–125.
- [56] Robertson, N., Sanders, D. P., Seymour P. D., Thomas, R. “The Four Colour Theorem,” *J. Combin. Theory Ser. B.* **70** (1997), 2–44.
- [57] Rogers, H., Jr., *Theory of Recursive Functions and Effective Computability*, McGraw-Hill Book Co., New York, 1967.
- [58] Russell, B., *A Critical Exposition of the Philosophy of Leibniz*, second ed., Allen and Unwin, London, 1937.
- [59] Segner, A., “Enumeratio Modorum Quibus Figurae Planae Rectilinae per Diagonales Dividuntur in Triangula”, *Novi Commentarii Academiae Scientiarum Imperialis Petropolitanque* **7** (1758-59), 203–209.
- [60] Shannon, C., “A Symbolic Analysis of Relay and Switching Circuits,” *Transactions American Institute of Electrical Engineers*, **57** (1938), 713–723.
- [61] Shannon, C.E., “A Mathematical Theory of Communication,” *Bell System Technical Journal*, **27** (1948), 379–423 and 623–656.
- [62] Shepherdson, J., “The Reduction of Two-Way Automata to One-Way Automata,” *IBM Journal of Research and Development*, **3** (1959), 198–200.
- [63] Sloane, N.J.A., Wyner, A.D. (editors), *Claude Elwood Shannon: Collected Papers*, The Institute of Electrical and Electronics Engineers, Inc., New York, 1993.
- [64] Stern, N., *From ENIAC to UNIVAC: An Appraisal of the Eckert-Mauchly Computers*, Digital Press, Bedford, Massachusetts, 1981.
- [65] Swetz, F. J., “Leibniz, the *Yijing*, and the Religious Conversion of the Chinese,” *Mathematics Magazine*, **76**, No. 4 (2003), 276–291.
- [66] Turing, A. M., “On Computable Numbers with an Application to the Entscheidungsproblem,” *Proceedings of the London Mathematical Society* **42** (1936), 230–265. A correction, **43** (1937), 544–546. This paper with a short foreword by Davis was reprinted on pages 115–154 of [13].
- [67] Turing, A. M., “Computability and λ -Definability,” *Journal of Symbolic Logic*, **2** (1937), 153–163.
- [68] Veblen, O., “An Application of Modular Equations in Analysis Situs,” *Ann. of Math.*, **14** (1912-13), 42–46.
- [69] von Neumann, J., “First Draft of a Report on the EDVAC,” in *From ENIAC to UNIVAC: An Appraisal of the Eckert-Mauchly Computers*, N. Stern, Digital Press, Bedford, Massachusetts, 1981, 177–246.
- [70] Young, R. M., *Excursions in Calculus: An Interplay of the Continuous and the Discrete*, Mathematical Association of America, Washington, D.C., 1992.