

# Thinking Machines: The Foundations for Modern Computing (pre-1950)

Kyle Ryan, University of Rochester

Prof. Brown, CSC 242W

Spring 2017

## 1 Abstract

The early history of computing is mostly not taught today beyond mathematics and history classes. Although the equations of early computing are used today, less attention is given to the stories of the real people who were working to represent the world around them. We can take modern computing for granted, but computing had a beginning. Who were the major players? Who contributed to computing as it is now? Today, there is no sector of society that is not touched by computing. If we use it everyday, why are its origins not common knowledge? The purpose of this paper is to provide a clearer answer to these questions. Since computing is one of the greatest assets of modern society, it is important to ask the question: where did it come from? In order to provide this full picture, the story of computing will be investigated through *ideas* instead of objects, so that a clear line can be drawn between the past, present, and the future of computing.

## 2 Introduction

If you traveled to a history museum on computing, you would likely discover a museum of *objects*. You would find the early computing machine replicas, pictures of the ENIAC, or a series of transistors. However, the history of computing is the history of *ideas* about how

machines can think logically, solve problems, and accomplish tasks. “Thinking machines” was a term that Alan Turing coined in his paper: *Computing Machinery and Intelligence* in 1950. Thinking machines did not start with Alan Turing or the transistor. Thinking machines started with hundreds of mathematicians and logicians including Aristotle, Euclid, Descartes, Boole, and more before 1950. In those days, they were ideas about a system of representing the “fundamental laws of thought”. The declaration and evolution of systems for representing these fundamental laws of thought eventually allowed people like Claude Shannon to map these laws of thought onto relay circuits. Later, Alan Turing, based in Britain, could map out the limitations of these computing machines stating that “It is possible to invent a single machine which can be used to compute any computable sequence.” Without the contribution of early mathematicians, a system of representing logical statements would not have been possible, which would not have led to the creation of modern computing itself.

## 3 Early Computing

### 3.1 Aristotle’s Contributions to Logical Thinking

The foundations of modern computing were laid out long before Bell Labs and the invention of the transistor in 1947. The classic Greek philosopher Aristotle was one of the first to explore logical thinking; his work was the earliest known *formal* work in logic (Smith, 2017). It is therefore an important first step in understanding the foundations of modern computing. Aristotle’s work in logic during this period would later be known as *Aristotelian Logic* which is the approach to logic that began with him and ended with the development of predicate logic in the late 19<sup>th</sup> century. As mathematicians would later discover with modern developments in logic, Aristotle’s work is not a complete paradigm for representing the logic of science and mathematics. His work would eventually be expanded by others like Euclid, Descartes, Boole, Turing. His work did have limitations compared to modern formal logic such as predicate logic — a system that is much more adept at representing logical propositions through functions and conjunctions. Aristotle found that logic could be

represented through objects and predicates, such that the structure was the only thing that contributed to the logic. It meant that a one statement of objects and predicates could apply to any number of objects and predicates. The most common example taught in Discrete Mathematics and Artificial Intelligence classes is the syllogism:

**All men are mortal.**

**Socrates is a man.**

**Therefore, Socrates is mortal.**

The use of “all”, “is”, and “therefore” provide the structure to the logical statement. You could replace these objects and predicates with anything else: All students drink coffee. John drinks coffee. Therefore, John is a student.

### 3.2 *The Organon*

Aristotle’s work on logic was combined into a collection called *The Organon* (*the "Instrument"*) which contains a myriad of contributions to the field (Smith, 2017). He stated that deduction was the process of using *things supposed* (premises) result in the *results of necessity* (Smith, 2017). This definition can be thought of as logical consequences. A logical consequence in modern terms would be that  $P \rightarrow Q, P, \therefore Q$ . If all men are mortal, Socrates is a man, therefore Socrates is mortal. In propositional logic, this form of a valid argument is referred to as Modus Ponens. It is one of the most well known syllogisms. Here, *Socrates* can be replaced with any object, and *mortal* can be replaced with any other predicate. Besides deduction, Aristotle also proposed induction — the process of making generalizations out of specific information. He proposed axioms in this work setting up non-contradiction (no statement can be true and false) and excluded middle (every statement is either true or false) (Smith, 2017). Aristotle would leave inspiration for the work of Euclid and *Elements* which was a seminal area in deductive reasoning.

### 3.3 Further Developments in Logic

Aristotle’s work in the domain of logic inspired later philosophers, most notably the school of Stoicism, which developed a formal system of logic called propositional logic (propositional

calculus). The basis for this system is the proposition — a statement that is either true or false. Elements was written by Euclid around 300 BCE in the then Alexandria, Egyptian region. It covers propositions, mathematical proofs, what later became Euclidean geometry, and also Number Theory (Gray, 2013). Other mathematicians like Descartes in the 1630s made further contributions to symbolic mathematics. His work produced the algebraic notion we have today. We want able to represent geometry, which previously used diagrams, into the form of equations which could be operated on. Since Aristotelian logic was incomplete, a later mathematician, Gottfried Leibniz created propositional logic as it is today, which is effectively known as symbolic logic. He published the paper, *On Formally Undecidable Propositions of Principia Mathematica and Related Systems*. At the time, Leibniz's work was not widely praised, since most at the time did not think that such a system would be influential for mathematics. Leibniz's work was later influential for Boole and DeMorgan, as this paper will show. It is the manipulation of symbols that was the real breakthrough in this work. Mathematics was now much more focused on formulas instead of diagrams. That change toward symbol manipulation later set the stage for calculus which was developed by Leibniz himself as well as Issac Newton. Boole and DeMorgan mostly re-created Leibniz's system for their needs.

Propositional logic was a reaction to the limitations of Aristotle's syllogistic logic. Propositional logic would later have even more advancements through the work of Gottlob Frege. The document Frege is credited with in this area was the *Begriffsschrift*. He took inspiration from Boole's attempts at converting Aristotelian logic into a better system. Frege's developed a system called Predicate Logic which allowed logicians to express much more complex logical statements for inference. For example, the limitations of propositional logic did not allow for the manipulation of individual symbols that could map to an outcome. The Modus Ponens example from earlier is limited by the way that it can manipulate symbols:

$$P \rightarrow Q \tag{1}$$

$$P \tag{2}$$

$$\therefore Q \tag{3}$$

Frege’s system set the stage for a more complete system of formal logic. Predicate logic was much more accurate. He developed quantifiers (like  $\exists$  and  $\forall$ ) and other symbol manipulation techniques. He also developed syntax that is common today like functions which can map an input to an outcome. You could now do things like:

$$\exists x(Crown(x) \wedge OnHead(x, John)) \tag{4}$$

This above statement would represent a sentence similar to “There exists an object such that it is a crown and it is on John’s head”. You can see how much more useful this system can be for inference given a larger general knowledge base, a specific problem, and a query. (Lipson, 2011)

## 4 Pre-Modern Computing

### 4.1 George Boole

Synonymous with logic is the “boolean” type in most modern programming languages which allows for the representation of true or false in a computer’s memory. The source of this oddly sounding name is George Boole — one of the most prominent mathematicians and logicians of the 19th century and in the history of computing. He was the son of a librarian and was raised in the Lincolnshire region of England (Burris, 2010). Even though he did not go an elite English school, he had access to books and information to engage his interests. When his father could no longer provides for the family, he worked as a teacher and eventually opened his own school. 15 years later, he became a professor at Queen’s University of Ireland. Because he had access to the library as a child, he was able to learn French, German, and

Italian. This breadth of language fluency allowed him to read mathematical papers in these languages (Burris, 2010).

Boole’s work as a professor focused on symbolic algebra, which allowed differential equations to be represented as ordinary polynomials for use in analysis. During this same period of work, he befriended Augustus DeMorgan, later an important contributor to inference and propositional logic. Between 1847 and 1854, Boole published two books on mathematical logic. In his words, Boole worked towards, “The construction of a system or method of Logic upon the basis of an exact summary of the fundamental laws of thought. (Boole, 48)” The names of the two books were *Mathematical Analysis of Logic* and *Laws of Thought*. George Boole turned Aristotle’s logic into an algebraic form. He applied symbolic algebra to logic that could be applied to an infinite variety of logic problems.

## 4.2 *The Mathematical Analysis of Logic*

In 1847, Boole created took an algebraic approach to Aristotelian logic through an 82 paged paper called *The Mathematical Analysis of Logic* (Boole, 1948). In the chapters of this book, Boole put forward new algebraic notations for representing operations such as composition ( $xy$ ), the distributive law  $x(u+v) = xu + xv$ , and the commutative law ( $xy = yx$ ). This is seen on Page 20 when he states:

“First, it is evident, that according to the above combinations, the order in which two symbols are written is indifferent. The expressions  $xy$  and  $yx$  equally represent that class of things to the several members of which the names or descriptions  $x$  and  $y$  are together applicable.” (Boole, 48)

The text is filled with examples very similar to the above one that provided foundational value to inference rules algebra with logic. Boole’s focus on an algebraic approach to Aristotelian logic was unique in that it allowed for symbolic representation of any possible assertion. For instance, he used the proposition, “The stars are the suns and the planets” where stars,  $x$ , suns,  $y$ , and planets,  $z$ , could be represented as:

$$x = y + z \tag{5}$$

He could then make a simple algebraic operation to get:

$$x - z = y \tag{6}$$

which would now say, “The stars, except the planets, are suns.” which logically follows from “the stars are the suns and the planets”. Both equations represent the same world, but they make two logical assertions, all through a simple algebraic operation on the first equation! (Boole, 25).

For instance, he turned the Aristotelian statement that all X is Y into  $xy=x$  which naturally suggests that all items in the set x are also in the set xy. As we know from an earlier section in this paper, syllogisms were crucial to Aristotelian logic. They were a major step forward in applying logic inference to assertions. This logical inference through syllogisms was not met by Boole with excitement, since he felt that syllogisms were not suitable for representation in algebraic form. In one of the chapters of *Mathematical Analysis of Logic* called Properties of Elective Functions, Boole put forward theorems for working with his algebraic logic (Boole, 1948). With this, his intention was to express the whole of Aristotelian logic in algebraic form.

### 4.3 *The Laws of Thought*

The second of Boole’s books published during this period was *The Laws of Thought* was published in 1854 and was an update to his previous book *The Mathematical Analysis of Logic* (Boole, 1854). In this text, he included work on probability theory and its application to his algebraic logic. An important aspect of this text was the explanation of how every equation is *interpretable* such that it can be converted into simpler equations. Boole went on to discuss more conversions of Aristotelian syllogisms to algebraic logic. Boole did not feel that Aristotelian logic was a complete system, that it had flaws, and that improvements were important to have a logic system that was both complete and exhaustive. Boole clearly stated that Aristotilean logic was not a “general method” for logic, and that a more perfect logical system could be created. He did however, suggest that Aristotle’s work *did* lead to the elementary processes of reasoning. His quote:

“The Logic of Aristotle, in its rules of syllogism and conversion, sets forth the elementary processes of which all reasoning consists, and that beyond these there is neither scope nor occasion for a general method.” — *The Laws of Thought*

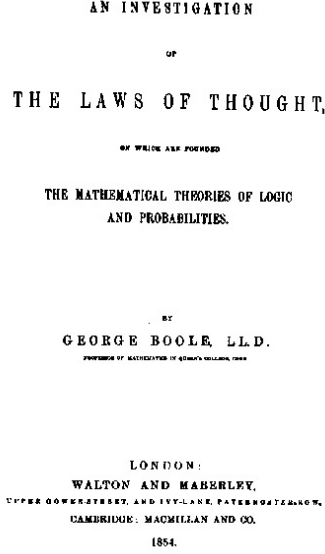


Figure 1: The cover of Boole’s Laws of Thought manuscript published in 1854.

#### 4.4 Influence of DeMorgan on Boole

Another mathematician during this period was Augustus DeMorgan who published a paper called *Formal Logic* published in 1847 (Morgan, 2014). The main stride with *Formal Logic* was the inference techniques and the addition of quantifiable terms in logic. He suggested that there could be a *calculus of inference* such that inference could be formed on assertions (Morgan, 2014). DeMorgan’s rules of inference allowed for the expression of conjunctions and disjunctions using negation (Guichard, 2011). He proposed a set of formal notation techniques for representing inference rules later come to be known as “DeMorgan’s Laws”. These laws showed how negations could manipulate syllogisms containing conjunctions and disjunctions. DeMorgan’s Laws can be seen in the following two propositional statements:

$$\neg(P \wedge Q) \leftrightarrow (\neg P \vee \neg Q) \tag{7}$$

$$\neg(P \vee Q) \leftrightarrow (\neg P \wedge \neg Q) \tag{8}$$



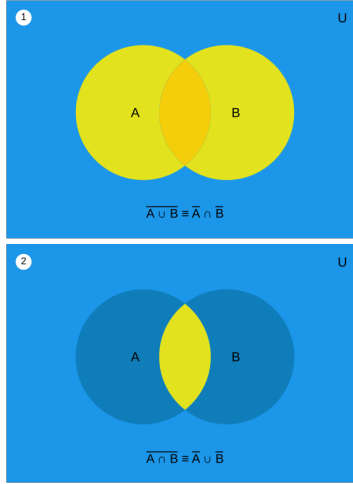


Figure 2: A visual representation of DeMorgan's Laws shown in boolean-style notation.

You can think of this first statement as saying “I don’t like coffee and tea” iff “I do not like coffee or I do not like tea.” Through inference — making implicit information explicit — you can see that these two statements express congruent assertions. The second propositional statement suggests that “I do not like tea or coffee” iff “I do not like tea and I do not like coffee.” DeMorgan’s representation of implication using negation is especially helpful in simplifying assertions like  $\neg P \rightarrow Q$ ). You can see how this type of operation could be useful to mathematicians at the time. Basically, you can convert  $\neg P \rightarrow Q$  into  $P \wedge \neg Q$  using Equation 1 and Equation 2.

$$\neg(P \rightarrow Q) \leftrightarrow \neg(\neg P \vee Q) \quad (9)$$

$$\leftrightarrow (\neg\neg P) \wedge (\neg Q) \quad (10)$$

$$\leftrightarrow P \wedge \neg Q \quad (11)$$

Figure 3: A visual representation of DeMorgan's Laws

Since Boole and DeMorgan worked during the same time period, they were competitors in a certain sense. They both looked to represent logic mathematically, and they went about it in different ways. Although DeMorgan’s work was helpful for logical inference,

Boole’s system of algebraic logic was very influential on the development of computer circuits later on. Boole’s system of algebraic logic became paramount in converting hundreds of years of work in logic to computing and circuits.

## 4.5 Impact of Boole’s Algebraic Logic System

The nature of Boole’s work in algebraic logic was critical for the later development of “thinking machines” such that mathematical logic could be mapped to the construction of electronic circuits. For the first time, George Boole opened up the world of logic to mathematics. Through an algebraic system, logic could now be represented purely mathematically, unlike a somewhat fragmented Aristotelian system of logic based in philosophy. In the next section of this paper covering Claude Shannon, we will see how he Boole’s algebraic logic was paramount to early electronic circuits. In one paper, Claude Shannon mapped Boolean operations to relay circuits. With this, mathematical logic could now be compatible with electronic circuits. For instance, functions like addition could then be mapped easily from Boolean algebraic logic to electronic circuit relays, opening up a world of possibilities for our “thinking machines”.

## 4.6 Alan Turing

Turing was born in 1912 London in the upper-middle class and family (Hodges, 2002). Before university, he grew up interested in science and was described as eccentric with an intense skepticism. He worked at Cambridge University and studied math and was eventually appointed as a fellow in 1935 (Hodges, 2002). As seen, Shannon laid the foundation for bringing logic from mathematics to the physical world. Alan Turing was able to design computing machines in the form on Shannon’s methods. Developed at Cambridge, Turing became well-known in logic for his paper, *On Computable Numbers, With an Application to the Entscheidungsproblem*, which laid a foundation for computing machines (Turing, 1937). This paper better defined the limitations of computing. He explained the workings of such a machine on Page 231 saying, “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” (Turing, 1937). The paper gets more technical and in-depth; he goes on to talk on page 232 about how one of these machines could be programmed to perform a calculation. He stated, “if a machine is supplied with a blank tape and set in motion, start from the correct initial m-configuration, the sub-sequence of the symbols printed by it which are of the first kind (symbols) will be called the *sequence computed by the machine*.” As stated, this paper was critical for establishing the limitations of computing machines in this form. These machines could produce large classes of numbers as he says on page 241, “It is possible to invent a single machine which can be used to compute any computable sequence (Turing, 1937).”

The Entscheidungsproblem proposes the question: can a computer solve any arbitrary mathematical statement? Is there a program that can determine whether a mathematical statement is true or false? Turing used this paper to come to a conclusion about the Entscheidungsproblem. On page 259, he states that “[his results] can be used to show that the Hilbert Entscheidungsproblem can have no solution.” (Turing, 1937) He goes on to say “there can be no machine which, supplied with any one of these formulae, will eventually say whether [the formula] is provable (Turing, 1937).”

## 4.7 Computing Machinery and Intelligence

Turing worked on a decoding system for Enigma — the German enciphering machine for the British government. His work in decryption enabled the British government to intercept U-Boat communications. After the war, Turing turned his attention to computing machines, and published the paper *Computing Machinery and Intelligence* in 1950 (Turing, 2012). He opened up the field of Artificial Intelligence by asking, “Can machines think? This should begin with the definitions of the terms ‘machine’ and ‘think’.” Within this paper, Turing developed the concept of a “Turing Test” which is now widely known in Artificial Intelligence practices. It concerns itself with the problem of distinguishing between a human and a machine. It involves using an interrogator to test a computer and a machine using questions. The goal of the interrogator is to determine which response is coming from the machine and which response is coming from the human. To the point at which the experimenter can no longer tell the difference between a human’s response and the computer’s responses, a Turing

Test can determine the degree to which a system is intelligent (Turing, 2012).

## 4.8 Claude Shannon

Claude Shannon, an American mathematician during the time of Turing, was impactful for turning Boole’s logic system into representations of electronic circuits. His paper, *A Symbolic Analysis of Relay and Switching Circuits* took Boole’s work and mapped it onto the logic of physical circuitry (Shannon, 2010). He stated that “any circuit [can be] represented by a set of equations, the terms of the equations corresponding to the various relays and switches in the circuit (Shannon, 2010).”

Symbol	Interpretation in Relay Circuits	Interpretation in the Calculus of Propositions
$X$	The circuit $X$	The proposition $X$
0	The circuit is closed	The proposition is false
1	The circuit is open	The proposition is true
$X + Y$	The series connection of circuits $X$ and $Y$	The proposition which is true if either $X$ or $Y$ is true
$XY$	The parallel connection of circuits $X$ and $Y$	The proposition which is true if both $X$ and $Y$ are true
$X'$	The circuit which is open when $X$ is closed and closed when $X$ is open	The contradictory of proposition $X$
$=$	The circuits open and close simultaneously	Each proposition implies the other

Figure 4: Conversion of Propositional statements to relays by Claude Shannon

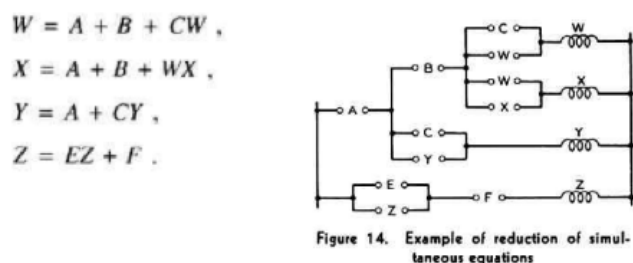


Figure 5: Diagrams from Shannon’s paper that clearly show the mappings of algebraic logic to circuits.

One of Claude Shannon’s advisers on this paper was Vannevar Bush, who developed a machine called the *Differential Analyzer*. With a basis in mechanical parts and electric

relays, calculations could be performed using the machine based on Shannon’s paper. Once complete at MIT, the machine had two thousand vacuum tubes, a few thousand relays, and a hundred motors. At the end of the second world war, this machine considered one of the most important in the world, for its use during the war.

## 5 Modern Day and Beyond

The investment during the Second World War into computing infrastructure paved the wave for the computing progress of the 50 years after it. The decade of the 1950s in particular set the stage for the modern computer. Within these ten years, computing moved from a realm of mathematics to tangible circuits. People could now use real “thinking machines” for computation, unbeknown of the rapid progress that would occur forty years later with desktop computers and the Internet in the 1990s. The architecture that Turing has described in his paper became the foundation for modern computer — now known as Von Neumann architecture (Hayle, 2006). It was an architecture that recognized that the distinction between machine, program, and data were non-existent. The logic could be encoded in a machine and also in software. The man himself, John Von Neumann, worked in Los Alamos on the atomic bomb project. In 1945, he developed the EDVAC, a storage-capable logic based computer.

It is impossible to have a conversation about the history of computing without tying it into the thread of the *future* of computing. Three modern advancements in computing have given rise to new capabilities for computers, and they will likely open up a new era in the story of computer. This is the recent advancements in artificial intelligence with machine learning and neural networks. In the 1950s, Warren McCulloch and Walter Pitts developed a way to form of math that could apply logic to neurons in the brain. Just like George Boole’s work to develop algebraic logic, McCulloch and Pitt’s work developed a sort of calculus for neurons. Despite a lull period where artificial neural networks (ANNs) were thought to be fully explored, the recent work in applying “big data” and machine learning, statistical techniques to ANNs allows them to learn new information. This learning process means that computer programs can take in new information, adapt, and perform tasks without being “coded” by a programmer.

When Aristotle first worked on logic, he wanted to develop laws of thought. With the addition of several other logicians and mathematicians, people could create physical machines that followed these laws of thought — machines that could reason using logic. Computing’s future may not only lie in creating machines that can reason, but in creating machines that can create new minds. When a computer can learn better than a human, they may one day surpass our abilities at certain tasks. Indeed, they already have.

## 6 Conclusion

Alan Turing in his 1950 paper *Computing Machinery and Intelligence* asked the question, “Can machines think?” That question has not been answered yet, but the journey towards that answer has been nonetheless exciting. This history of computing is not the history of a single person, machine, or invention. This history of computing is *rather* the history of ideas. It is the pursuit of the fundamental laws of thought. Computing in modern times has created unparalleled advancements in society. There is almost no sector of that is not touched by computing. It has been shown how the very early developments of Aristotelian logic, though flawed, inspired later questioning about logic. Aristotle established a working foundation on which other mathematicians could use to develop better, more complete systems. This system was later refined by Euclid, Leibniz, DeMorgan, and Boole. The work of these individuals led to Turing, Shannon, and Bush to develop physical machines to perform these logical calculations. The infrastructure investments of World War Two allowed computing to gain usefulness beyond war, into a machine that could be infinitely helpful addition to the human condition.