Logic and the Development of Computing Machines

Maria M. Dimarogkona

NTUA

May 13, 2022

Table of contents

History of Objects

Figure: Abacus

Figure: Antikythera mechanism (2nd century BCE)

History of Objects

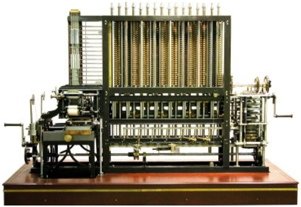


Figure: Babbage engine (designed in the 1820s)

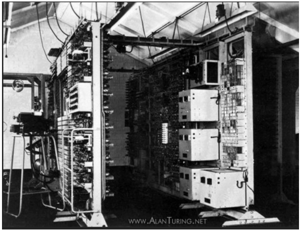


Figure: Colossus (built during WW2)

History of Ideas

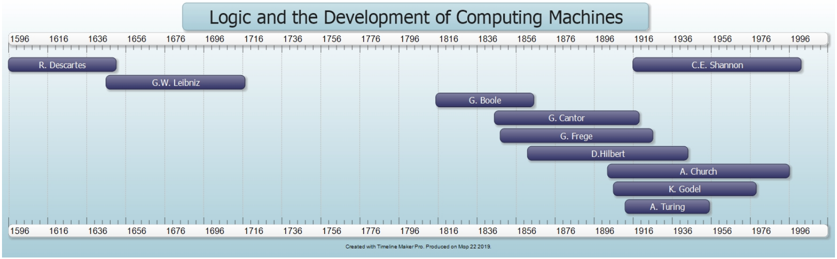
Computer science developed mainly out of ideas which emerged from mathematical logic.

- Aristotle → Leibniz → G.Boole → G.Frege

The evolution of computer science from mathematical logic culminated in the 1930s.

- C.Shannon (1937) *A symbolic analysis of Switching and Relay Circuits*
- A.Turing (1936) *On Computable Numbers, with an Application to the Entscheidungsproblem*

Timeline



Created with Timeline Maker Pro. Produced on Map 22 2019.

Gottfried Wilhelm Leibniz (1646-1716)

- 1673: Presented model of a computing machine (four basic arithmetic operations) - Leibniz wheel.
- 1674: Described a machine that could solve algebraic equations.
- Compared logical thought to a mechanism and focused on reducing logic to some kind of calculations and build a machine capable of performing such calculations.

Characteristica Universalis

Universal "Concept Language"

- Contrary to alphabetical symbols which have no meaning, symbols in arithmetic, algebra, chemistry, astronomy, differential and integral calculus represent clear ideas in a natural and appropriate way.
- **Characteristica Universalis**: characteristic system of symbols covering the whole range of human knowledge.
- Encyclopedia of human knowledge, inference rules: **calculus ratiocinator** (symbolic logic).

Calculus Ratiocinator

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, and say to each other: Calculemus -Let us calculate.”

— Leibniz - Dissertio di Arte Combinatoria (1666)

The Symbolic Calculus of Leibniz

A,B: concepts

- $A \infty B$: all the concepts making up concept A are also contained in concept B and vice versa $(A = B)$.
- $A \oplus B \infty C$: the concepts in A and those in B wholly constitute those in C $(A \cup B = C)$.
- $AB \infty C$, all concepts in both A and B wholly constitute the concept C $(A \cap B = C)$.
- $A \infty AB$: All As are Bs (universal affirmative judgement).
- $A \infty AB$; $B \infty BC$; therefore $A \infty AC$ (syllogism).

George Boole (1815 - 1864)

- English mathematician, philosopher and logician, largely self-taught.
- Worked on differential equations, and general methods in probability.
- Founder of the algebraic tradition in logic.
  - Created algebraic logic by applying symbolic algebra to logic.
  - (Aristotelian) logic: cataloging the valid syllogisms of various simple forms while
    Boole's method: general algorithms in an algebraic language which applied to an infinite variety of arguments of arbitrary complexity.
- Boolean algebra is basic to the design of digital computer circuits.

George Boole's *Laws of Thought*

> *"The design of the following treatise is to investigate the fundamental laws of those operations of the mind by which reasoning is performed; to give expression to them in the symbolical language of Calculus, and upon this foundation to establish the science of Logic ... and, finally, to collect .. some probable intimations concerning the nature and constitution of the human mind."*
>
> *—G.Boole - Laws of Thought (1854)*

Historical Retrospective

> *"In its ancient scholastic form, indeed, the subject of Logic stands almost exlusively associated with the great name of Aristotle. As it was presented to ancient Greece in the partly technical, partly metaphysical disquisitions of The Organon, such, with scarcely any essential change, it has continued to the present way."*
>
> —*G.Boole - Laws of Thought (1854)*

- Aristotelean logic: syllogistic.
  - *syllogismos* (deduction) - valid arguments.
- Euclid's *Elements*: rigorous deductive reasoning.
- Stoic logic: system of propositional logic
  - modus ponens
- Medieval Scholastic logic.

Boole's Work on Logic

"Since Aristotle, logic has been unable to take a single step forward, and therefore seems to all appearance to be finished and complete."

— I. Kant - Critique of Pure Reason (2^{nd} edition 1787)

- Boole's goal: Do for Aristotelean logic what Descartes had done for Euclidean geometry: free it from the limits of spacial intuition by giving it a precise algebraic notation.
- Boole's *Laws of Thought* created a new scholarly field: Mathematical logic.
- B.Russell (1872 - 1970) for *Laws of Thought*:

  "the work in which pure mathematics was discovered."

Boolean Algebra

$< B, +, \cdot,', 0, 1 >$ where B is closed under $+, \cdot,'$.

Axioms

- a + b = b + a
- (a + b) + c = a + (b + c)
- $a + (b \cdot c) = (a \cdot b) + (a \cdot c)$
- $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$
- a + 0 = a , $a \cdot 1 = a$
- $\forall b \in B(\exists! b' \in B : b + b' = 0, b \cdot b' = 1)$

Boole's System in *Laws of Thought*

1: a class denoting the universe.

0: the empty class.

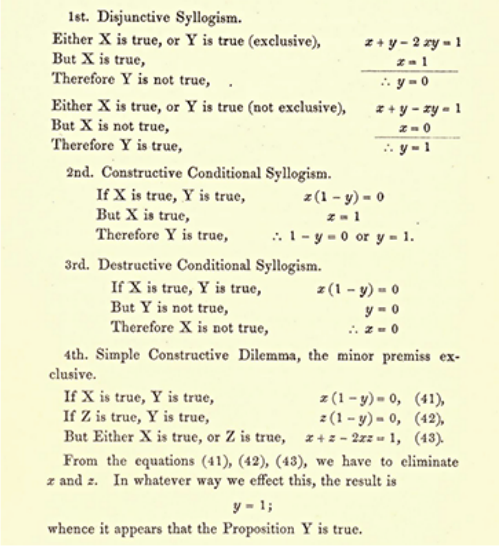
$\cdot$: intersection.

$+$: union of disjoint classes.

Laws

- xy $=$ yx
- $x^2 = x$
- x $+$ y $=$ y $+$ x
- z (x $+$ y) $=$ zx $+$ zy
- x - y $=$ -y $+$ x
- z (x $+$ y) $=$ zx - zy

Boole's Mathematical Analysis of Logic



1st. Disjunctive Syllogism.

Either X is true, or Y is true (exclusive), $x + y - 2xy = 1$
But X is true, $x = 1$
Therefore Y is not true, $\therefore y = 0$

Either X is true, or Y is true (not exclusive), $x + y - xy = 1$
But X is not true, $x = 0$
Therefore Y is true, $\therefore y = 1$

2nd. Constructive Conditional Syllogism.

If X is true, Y is true, $x(1 - y) = 0$
But X is true, $x = 1$
Therefore Y is true, $\therefore 1 - y = 0$ or $y = 1$.

3rd. Destructive Conditional Syllogism.

If X is true, Y is true, $x(1 - y) = 0$
But Y is not true, $y = 0$
Therefore X is not true, $\therefore x = 0$

4th. Simple Constructive Dilemma, the minor premiss exclusive.

If X is true, Y is true, $x(1 - y) = 0$, (41),
If Z is true, Y is true, $z(1 - y) = 0$, (42),
But Either X is true, or Z is true, $x + z - 2xz = 1$, (43).

From the equations (41), (42), (43), we have to eliminate x and z. In whatever way we effect this, the result is

$$y = 1;$$

whence it appears that the Proposition Y is true.

Gottlob Frege (1848 - 1925)

- **Begriffscchrift (1879)**
  - Goal: devise a formal system within which all mathematical proofs might be presented, offering a guarantee against incorrect argumentation.
  - previous systems could not express all mathematical statements (e.g. sentences involving multiple generality) and thus could not be used as formal languages for mathematical reasoning.
  - Frege's innovations: relational predicates, variables, quantifiers.
    - Introduced device of quantification:adequate account of statements involving multiple generality.
    - Introduced variables for relations and functions.

Frege's Reception

- B.Russell:considerably influenced.
- L.Wittgenstein: profoundly influenced.
- Zermelo and Dedekind respected him.
- Cantor was hostile.
- Hilbert dismissed his work as vitiated by the paradoxes.
- Peano did not take him very seriously.
- Lowenheim had long correspondence.

Grundlagen der Arithmetic (1884)

Actual formalization of a mathematical theory - arithmetic.

- Sketches his method of constructing arithmetic (without use of symbols).
- Original conception: introduce appropriate primitives,formulate axioms, derive theorems from axioms in accordance with rules of formal system.
- Conception of logicism: arithmetic could be so analysed as to employ no primitive notions or axioms peculiar to it.
- Notion of class ('extension of a concept'): completely unanalyzed in *Grundlagen*.

Grundgesetze der Arithmetic Vol 1 (1893)

- Formalization of the construction of arithmetic sketched in *Grundlagen*.
- Formal system resembles that of *Begriffscchrift* though axiomatization of logic is different.
    - Introduction of classes.
    - Their theory must be treated as part of logic, if the claim to have reduced arithmetic to logic was to be supported.
    - **Axiom V**
- Russell showed that Frege's system allowed the existence of self-contradicting sets - **Russell's Paradox**.

Russell's Paradox

- As volume II of *Grundgesetze* was in press, Frege received Russell's letter announcing his discovery of a contradiction in his theory of classes.
- Frege's modification:**Axiom V'**.
- (1930) logician Lesniewski proved that Axiom V' yields the conclusion that there are no two distinct objects.
- The proof of the basic theorem that every natural number has a successor breaks down when Axiom V is weakened to V'.

David Hilbert (1862 - 1943)

- German mathematician.
- Hilbert's 23 problems: enumeration of 23 most important open mathematical questions - 1900 Paris conference of the International Congress of Mathematicians.
- Hilbert's Program: find an complete and consistent set of axioms for all of mathematics(early 1920s).
- Godel's Incompleteness Theorems - early 1930s.

Hilbert's Program

Formalize mathematics following specific requirements like

- **Completeness**
    - There must be a proof that all mathematical statements can be proven in the formal system.
- **Decidability**
    - The Decision Problem (**Entsceidungsproblem**): Is there an algorithm that can determine whether an arbitrary mathematical statement is true or false?

The Untenability of Hilbert's program

- **Godel's Incompleteness Theorem (1931)**: Any consistent logical system powerful enough to encompass arithmetic must also contain statements that are true but cannot be proven to be true.
- **Turing, Church**: No algorithm can exist that determines whether an arbitrary mathematical statement is true or false.

Claude E. Shannon (1916 - 2001)

A symbolic analysis of Switching and Relay Circuits (1937)

- Master thesis at MIT - electrical engineering.
    - His adviser Vannerah Bush built a prototype computer called *Differential Analyzer*. This device was mostly mechanical, with subsystems controlled by electrical relays which were organized in an ad hoc manner (no systematic theory underlying circuit design).
- Primary reference: G. Boole -*Laws of Thought* (1854).

> *"It just happened that no one else was familiar with both fields at the same time."*
>
> — *Shannon*

Digital Circuits Design

- Shannon's insight: Boole's system could be mapped directly onto electrical circuits.
  - The right systematic theory for circuit design would be exactly analogous to the calculus of propositions used in Boolean logic.

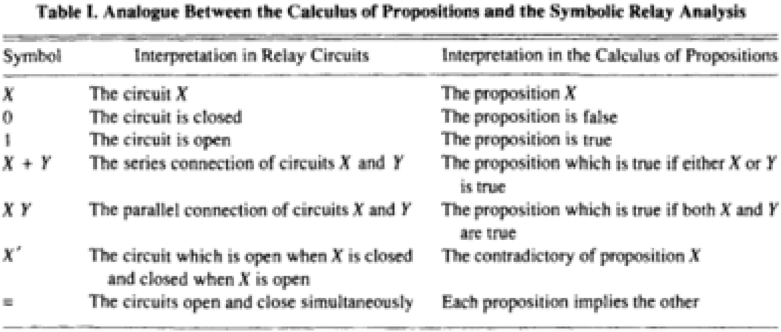


Table I. Analogue Between the Calculus of Propositions and the Symbolic Relay Analysis

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
$X\,Y$	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

Adder Circuit

- In the second part of his paper, Shannon showed how Boolean logic could be used to create a circuit for adding two binary digits.
- By stringing adder circuits together, arbitrarily complex arithmetical operations could be constructed.
    - Adder circuits: basic building blocks of *arithmetical logic units*.

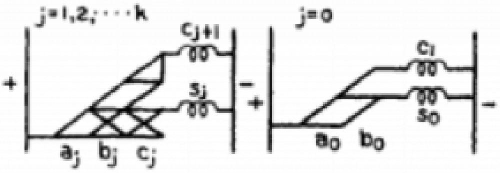


Figure 35. Circuits for electric adder

Logical and Physical Layer of Computing

Shannon was the first to distinguish between the *logical* and the *physical* layer of computers.

- Physical Layer
  - 1947: Invention of transistor - William Shockley and co., Bell Labs.
  - Transistors: dramatically improved versions of Shannon's electrical relays (transistors are the best known way to physically encode Boolean operations).
  - 2016: iphone has about 3.3 billion transistors, each one functioning as a "relay switch" like those pictured in Shannon's diagrams.

Allan Turing (1912 - 1954)

- Shannon showed how to map logic onto the physical world.
- Turing showed how to design computers in the language of mathematical logic.

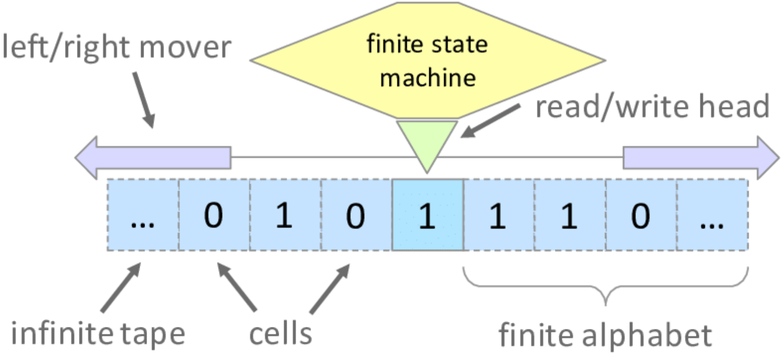
On Computable Numbers, with an Application to the Entscheidungsproblem (1936)

- Answers Hilbert's "decision problem" .
- Provides a template for computer design - mathematical model of an all-purpose computing machine.

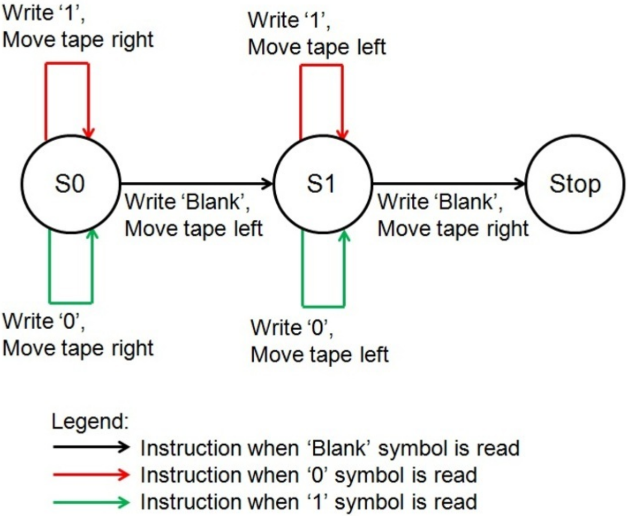
Turing's Answer to the Decision Problem

- Turing first developed a mathematical model of computation.
  - Turing machine: abstract machine which manipulates symbols on a strip of tape according to a table of rules. The input is given in binary form in the machine's tape, and the output consists in the contents of the tape when the machine halts.
  - A different Turing machine must be constructed for every new computation to be performed.
- He then proved that no such machine can determine whether a given proposed conclusion follows from given premises using Frege's rules.
- He concluded that no algorithm exists for the decision problem.

Turing Machine



Turing Machine



Allan Turing (1912 - 1954)

- Turing showed that a general solution to the decision problem is impossible, if the intuitive notion of "effectively calculable" is fully captured by the functions which are computable by a Turing machine.
- *computable function* (formal notion):
  a function calculable by a Turing machine.
- *effectively calculable* function (informal notion):
  its values can be found by some purely mechanical process.

Church-Turing thesis:
Every effectively calculable function is a computable function.

Formalization of the Notion of Computability

- 1933 Gödel, Herbrand: class of general recursive functions.
- 1936 Church: λ-calculus - λ-computable functions.
- 1936 Turing : Turing machines - Turing-computable functions.

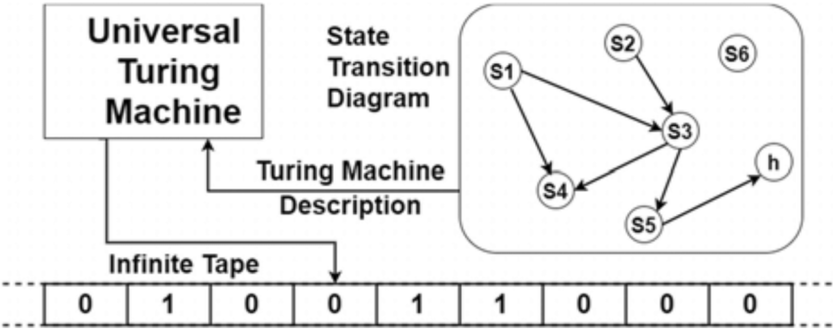
Church and Turing proved that these three formally defined classes of computable functions coincide:

> **A function is λ-computable iff it is Turing-computable and iff it is general recursive.**

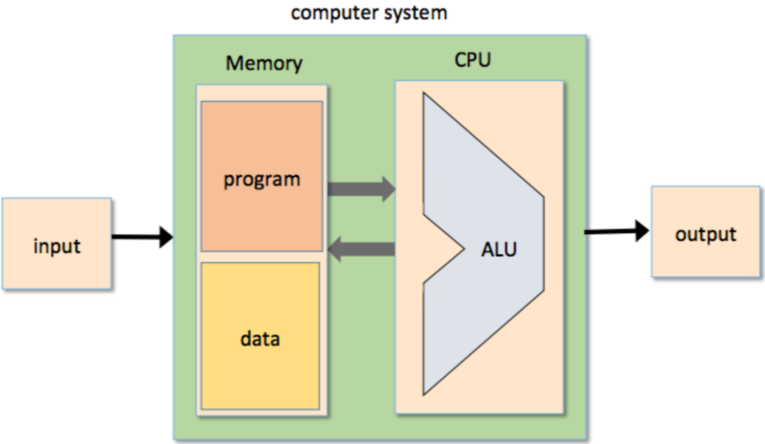
"Von Neumann Architecture"

- "Stored-program" architecture.
- Before Turing hardware (machine), software (program e.g. punch cards), and data were treated as separate entities
  - machine: physical object.
  - program: plan for doing a computation.
  - data: numerical input.
- Universal Turing Machine:
  A Turing machine which can simulate any other Turing machine. A UTM takes as input the description of a Turing machine T and an input for T, and simulates T on that input.

Universal Turing Machine



Stored-Program Architecture



First Computers

- EDVAC: Early stored-program electronic digital computer designed by Von Neumann, originally commissioned from the University of Pennsylvania's Moore School by the US Army in 1944. (similar to modern CISC processors)
- ACE: British early electronic serial stored-program computer designed by Alan Turing. (similar to modern RISC processors).
- The first computers to be based on Boolean logic and stored-program architectures.