CSCI 5444: Introduction to Theory of Computation

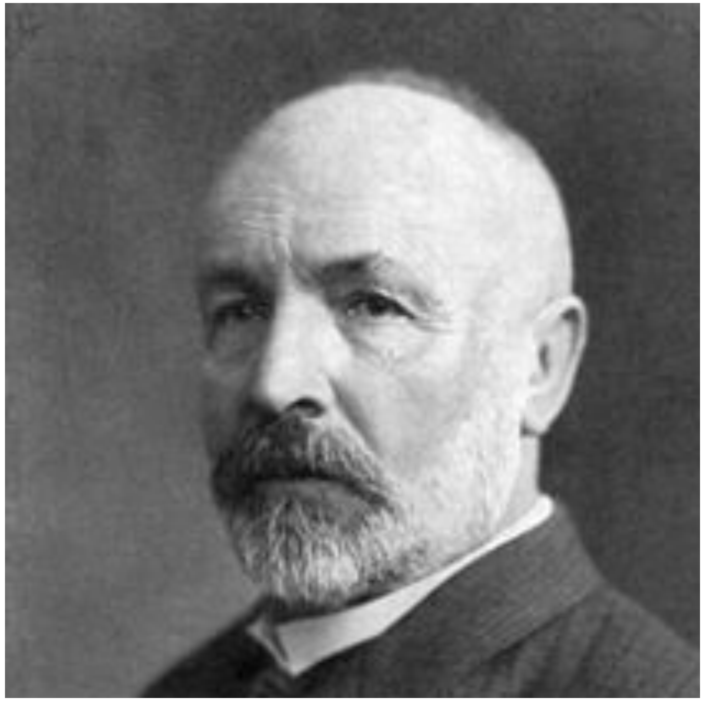
Lecture 02: Entscheidungsproblem and Turing Machines

Alexandra Kolla (Alexandra.Kolla@colorado.edu)
Department of Computer Science, University of Colorado Boulder

Discrete Mathematics: Review

Discrete Mathematics: Review

- A set is a collection of objects, e.g.
  - $A = \{a, b, c, d\}$ and $B = \{b, d\}$
  - Empty set $\emptyset = \{\}$ (why it is not same as $\{\emptyset\}$)
  - $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ and $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$
  - $\mathbb{Q}$ is the set of rational numbers.
  - $\mathbb{R}$ is the set of real numbers.
- $a \in A$: element of a set, belongs to, or contains
- Subset of $A \subseteq \mathbb{N}$, or proper subset of $A \subset \mathbb{N}$
- Notions of set union, intersection, difference, and disjoint
- Power set 2^A of a set A (example)
- Partition of a set



Georg Cantor
March 3, 1845 – January 6, 1918

Discrete Mathematics: Review (Contd.)

- A ordered pair is a pair (a, b) of elements with natural order
- Similarly we define triplet, quadruplet, n-tuples, and so on
- Cartesian product $\mathrm{A} \times B$ of two sets is the set of orderd pairs

$$\mathrm{A} \times B = \{(a, b) : a \in A \; and \; b \in B\}$$

- Binary relation R on two sets A and B is a subset of $\mathrm{A} \times B$
- Recall definitions of
  - Reflexive, Symmetric, and Transitive relations,
  - and Equivalence relation.

Discrete Mathematics: Review (Contd.)

- A function f from set A to B, formally $f: A \rightarrow B$, is a binary relation such that for all $a \in A$ we have $(a, b) \in f$ and $(a, b') \in f$ implies that $b = b'$.
- Unless specified otherwise, we assume that the function $f: A \rightarrow B$ is a total function, i.e. for all $a \in A$ there is a $b \in \mathrm{B}$ such that $(a, b) \in f$.
- We often write $f(a)$ for the unique element b such that $(a, b) \in f$.
- Function $f: A \rightarrow B$ is one-to-one if for any two distinct elements $a, b \in A$ we have that $f(a) \neq f(b)$.
- Function $f: A \rightarrow B$ is onto if for every element $b \in B$ there is an element $a \in A$ such that $f(a) = b$.
- Function $f: A \rightarrow B$ is called bijection if it is both one-to-one and onto.

Cardinality of a Set

- Cardinality $|S|$ of a set S is a measure of "number of elements" in S
  - For the set $\mathrm{A} = \{\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}\}$ we have $|A| \;=\; 4$
  - For the set $\mathbb{N}$, its cardinality $|\mathbb{N}|$ is an infinite number $\aleph_0$ (aleph-null).
- Two sets have same cardinality if there is a bijection between them.
- A set is countably infinite (or denumerable) if it has same cardinality as $\mathbb{N}$.
- A set is countable if it is either finite or countably infinite.
- A transfinite number is a cardinality of some infinite set.

Theorem: Cardinality

Theorem

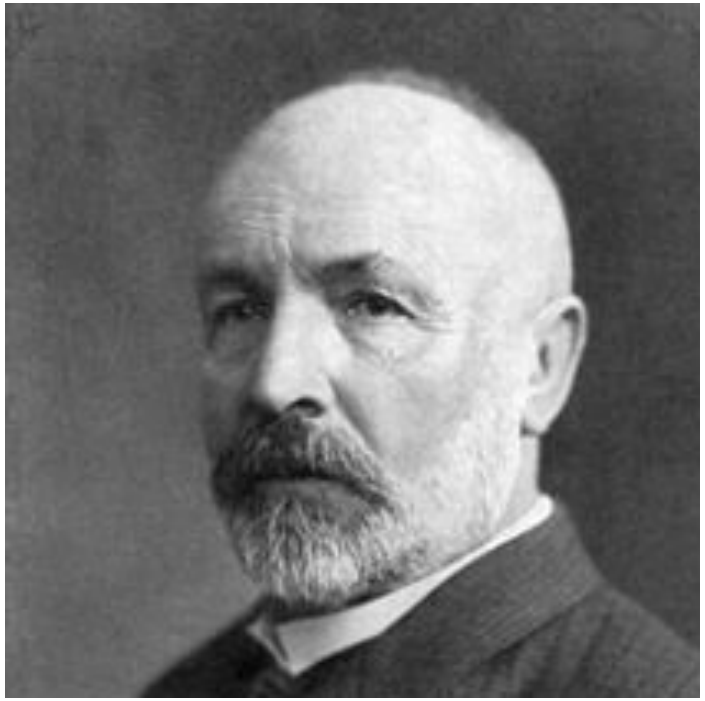
1. *The set of integers is countably infinite. (idea: interlacing)*
2. *The union of a finite number of countably infinite sets is countably infinite as well. (idea: dove-tailing)*
3. *The union of a countably infinite number of countably infinite sets is countably infinite.*
4. *The set of rational numbers is countably infinite.*
5. *The power set of the set of natural numbers has a greater cardinality than itself. (idea: contradiction, diagonalization)*

Theorem. *The power set of the set of natural numbers has a greater cardinality than itself. (idea: contradiction, diagonalization)*

Proof. The proof is by contradiction.

1. Assume that the power set of $\mathbb{N}$ has the same cardinality as $\mathbb{N}$.
2. It follows that there is a one-to-one correspondence between $2^{\mathbb{N}}$ and $\mathbb{N}$. Consider an arbitrary such mapping f.
3. Consider the table T s.t. $T[i,j] = true$ if the subset $S \subseteq \mathbb{N}$ mapped to the index i, i.e. $f(S) = i$, contains j, i.e. $j \in S$.
4. Consider the set $S_* = \{ i \in \mathbb{N} : T[i,i] = false \}$.
5. Notice that the set S_* is not mapped to any element of $\mathbb{N}$. Why?
6. A contradiction. □

Cantor’s Theorem

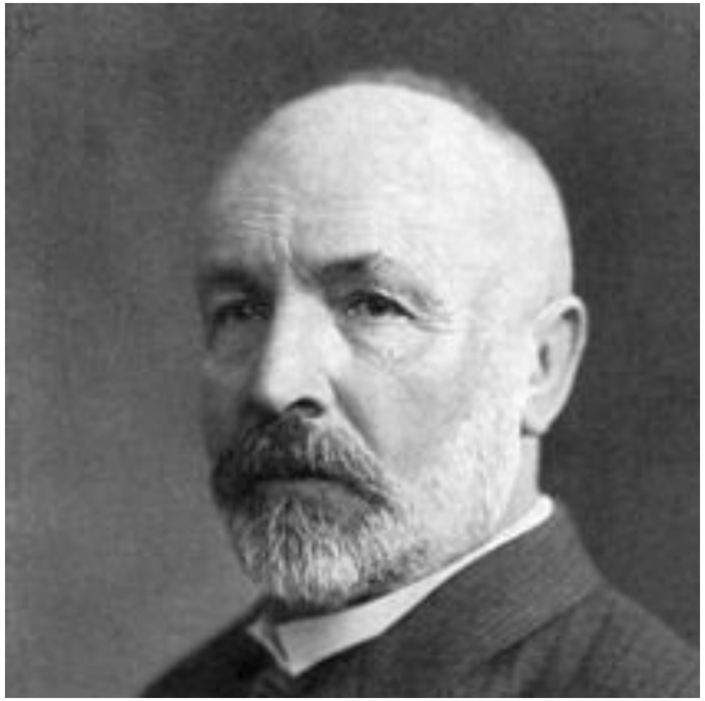


Theorem. *If a set S is of any infinite cardinality then its power set 2^S has a greater cardinality, i.e.* $|2^S| > |S|$.

(hint: happy, sad sets).

Corollary. *There is an infinite series of infinite cardinals.*

Cantor’s Theorem



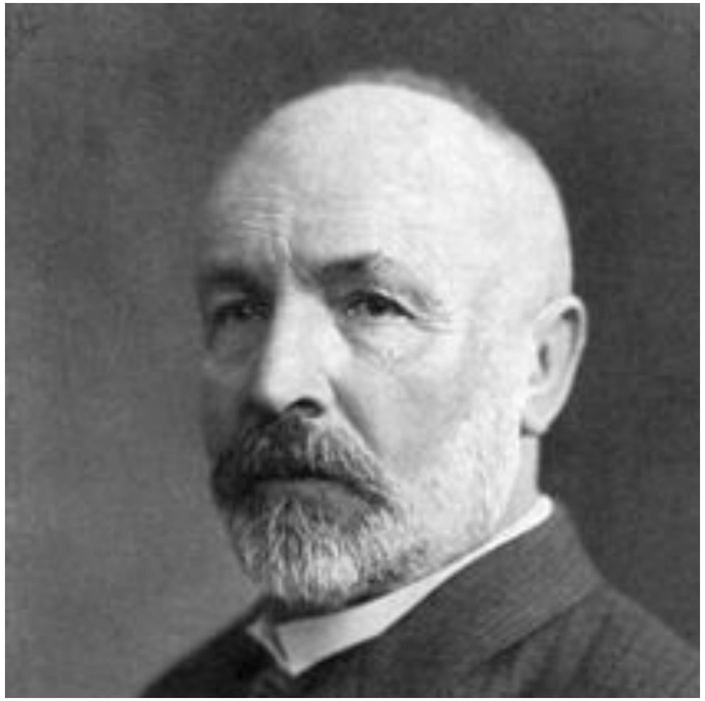
Theorem. *There is an infinite series of infinite cardinals.*

“ a "grave disease" infecting the discipline of mathematics*” ─Henri Poincaré*

“ I don't know what predominates in Cantor's theory – philosophy or theology, but I am sure that there is no mathematics there*”*─ Leopold Kronecker

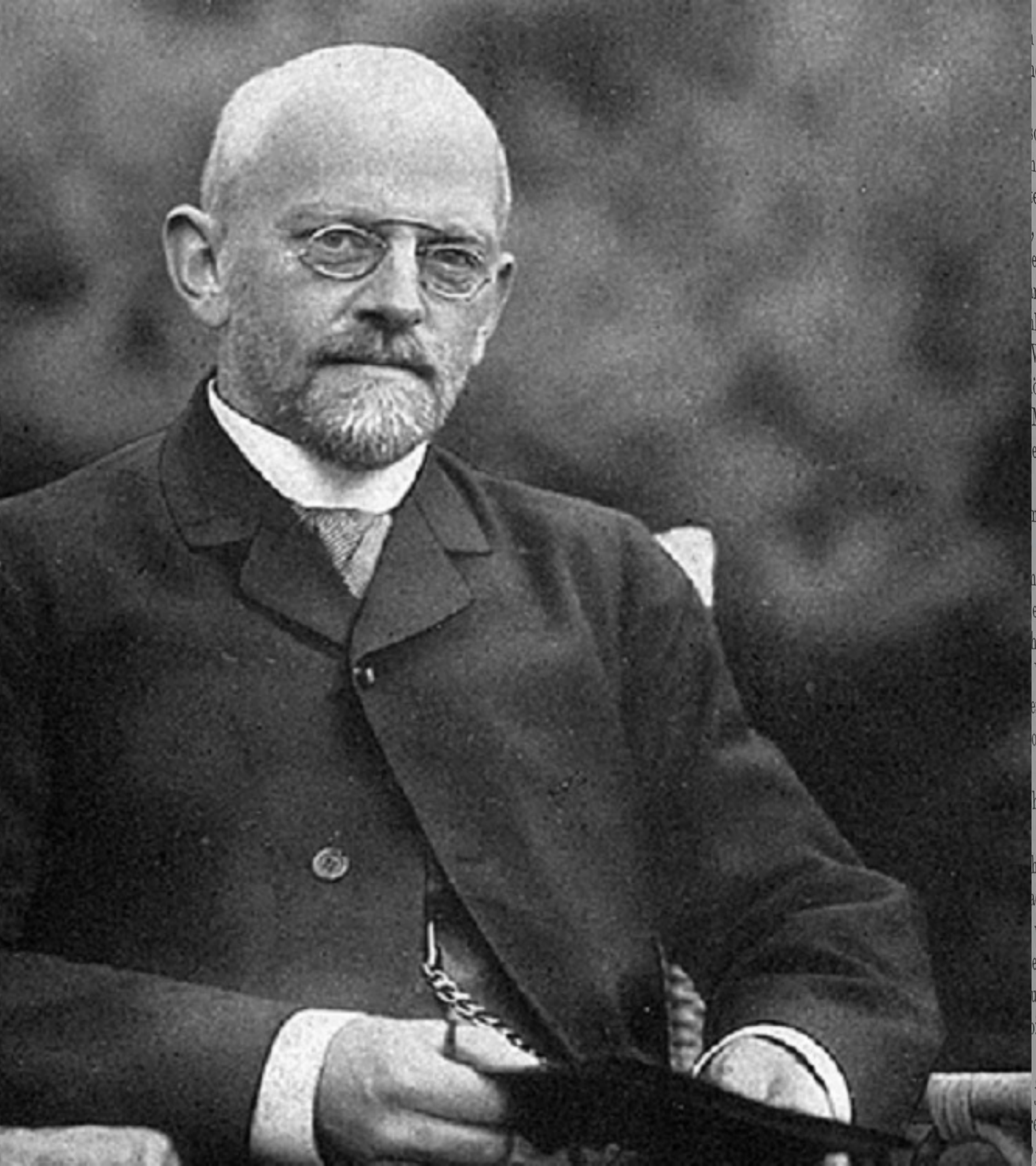
“… about one hundred years too soon .. ” ─ Gosta Mittag-Leffler

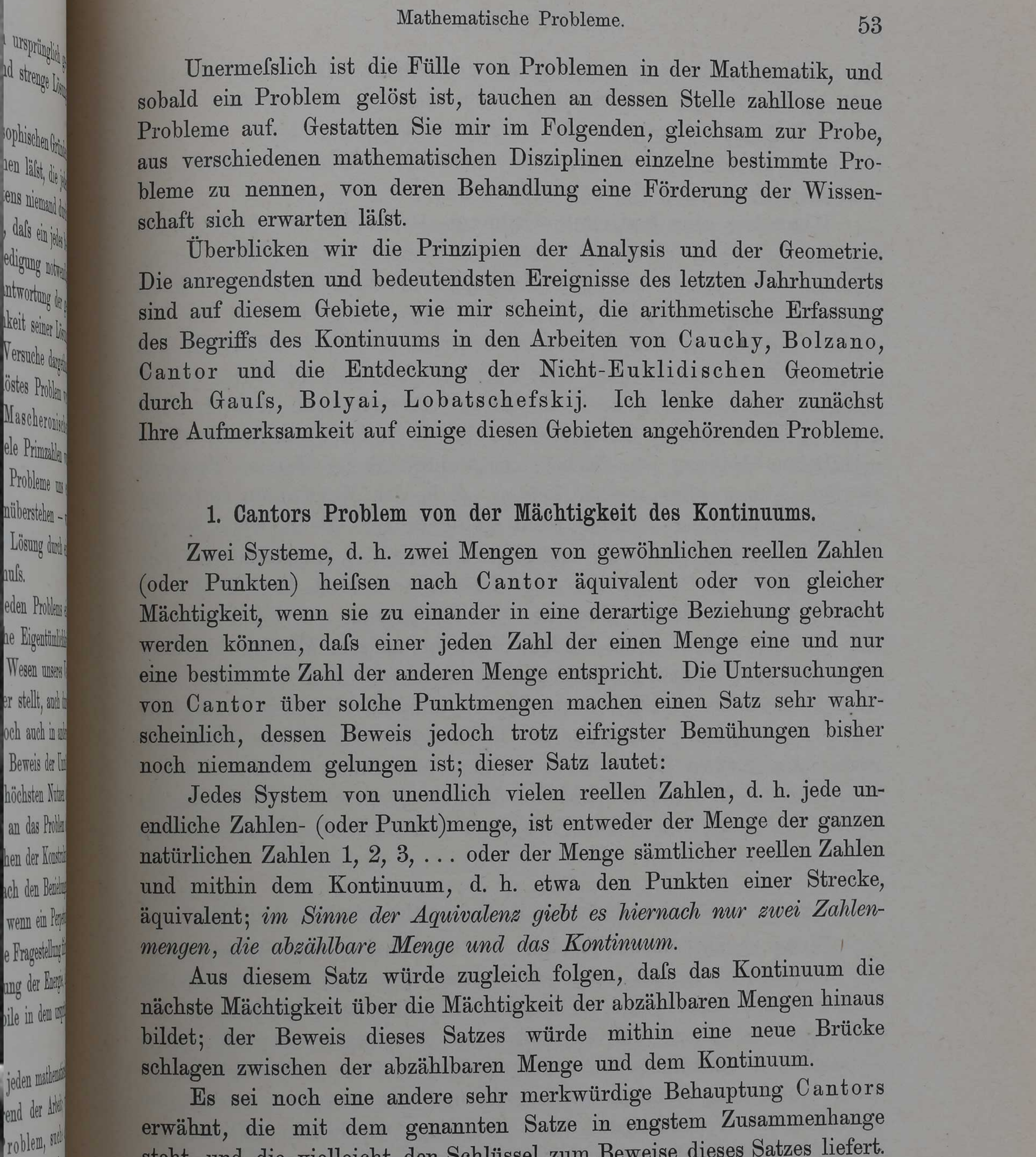
Theorem. *There is an infinite series of infinite cardinals.*



“Most admirable flower of mathematical intellect”
David Hilbert

David Hilbert
23 January 1862 – 14 February 1943





Unermeſslich ist die Fülle von Problemen in der Mathematik, und sobald ein Problem gelöst ist, tauchen an dessen Stelle zahllose neue Probleme auf. Gestatten Sie mir im Folgenden, gleichsam zur Probe, aus verschiedenen mathematischen Disziplinen einzelne bestimmte Probleme zu nennen, von deren Behandlung eine Förderung der Wissenschaft sich erwarten läſst.

Überblicken wir die Prinzipien der Analysis und der Geometrie. Die anregendsten und bedeutendsten Ereignisse des letzten Jahrhunderts sind auf diesem Gebiete, wie mir scheint, die arithmetische Erfassung des Begriffs des Kontinuums in den Arbeiten von Cauchy, Bolzano, Cantor und die Entdeckung der Nicht-Euklidischen Geometrie durch Gauſs, Bolyai, Lobatschefskij. Ich lenke daher zunächst Ihre Aufmerksamkeit auf einige diesen Gebieten angehörenden Probleme.

1. Cantors Problem von der Mächtigkeit des Kontinuums.

Zwei Systeme, d. h. zwei Mengen von gewöhnlichen reellen Zahlen (oder Punkten) heiſsen nach Cantor äquivalent oder von gleicher Mächtigkeit, wenn sie zu einander in eine derartige Beziehung gebracht werden können, daſs einer jeden Zahl der einen Menge eine und nur eine bestimmte Zahl der anderen Menge entspricht. Die Untersuchungen von Cantor über solche Punktmengen machen einen Satz sehr wahrscheinlich, dessen Beweis jedoch trotz eifrigster Bemühungen bisher noch niemandem gelungen ist; dieser Satz lautet:

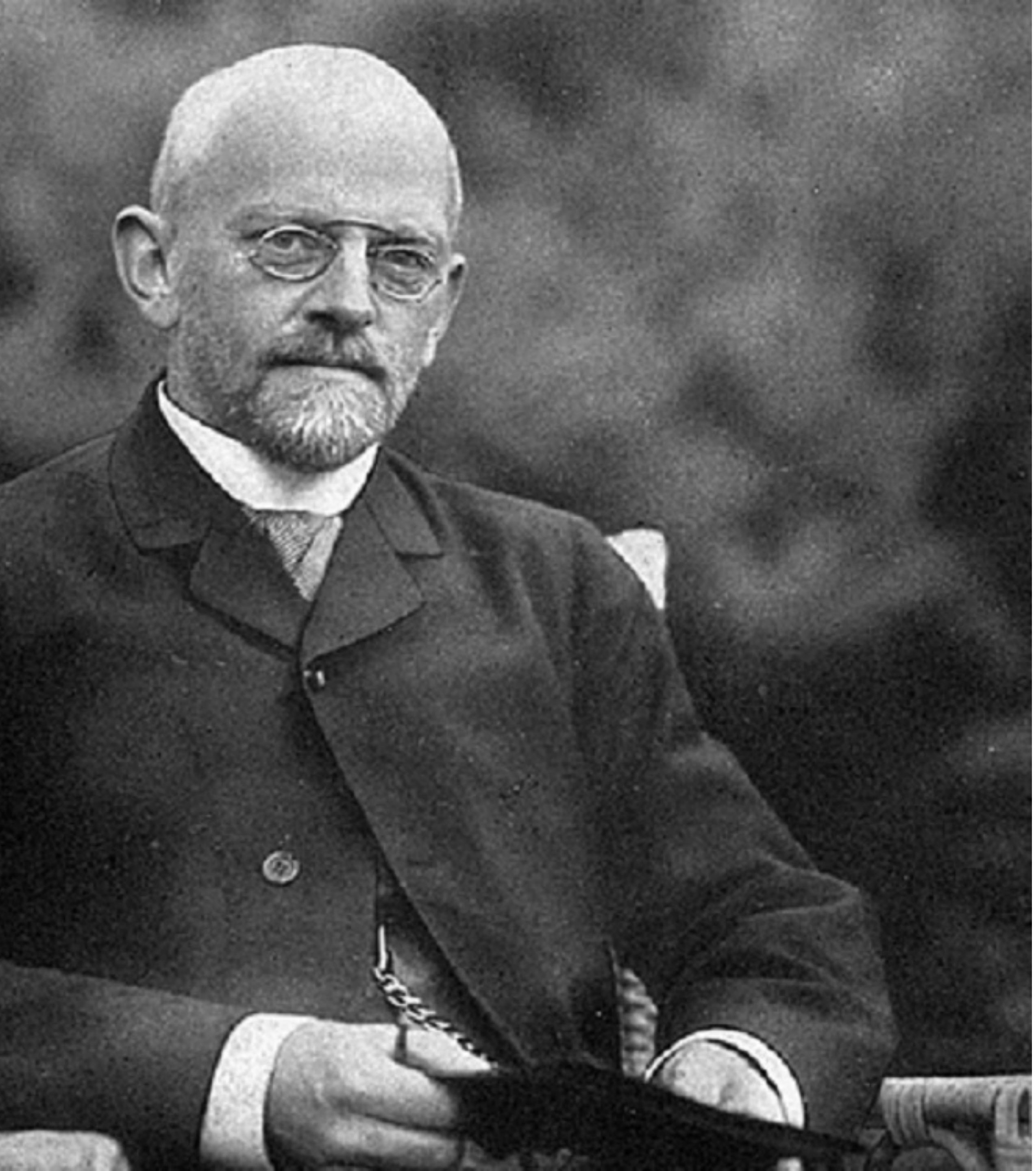
Jedes System von unendlich vielen reellen Zahlen, d. h. jede unendliche Zahlen- (oder Punkt)menge, ist entweder der Menge der ganzen natürlichen Zahlen 1, 2, 3, . . . oder der Menge sämtlicher reellen Zahlen und mithin dem Kontinuum, d. h. etwa den Punkten einer Strecke, äquivalent; *im Sinne der Aquivalenz giebt es hiernach nur zwei Zahlenmengen, die abzählbare Menge und das Kontinuum.*

Aus diesem Satz würde zugleich folgen, daſs das Kontinuum die nächste Mächtigkeit über die Mächtigkeit der abzählbaren Mengen hinaus bildet; der Beweis dieses Satzes würde mithin eine neue Brücke schlagen zwischen der abzählbaren Menge und dem Kontinuum.

Es sei noch eine andere sehr merkwürdige Behauptung Cantors erwähnt, die mit dem genannten Satze in engstem Zusammenhange steht, und die vielleicht den Schlüssel zum Beweise dieses Satzes liefert.

The 23 Mathematical Problems of Hilbert

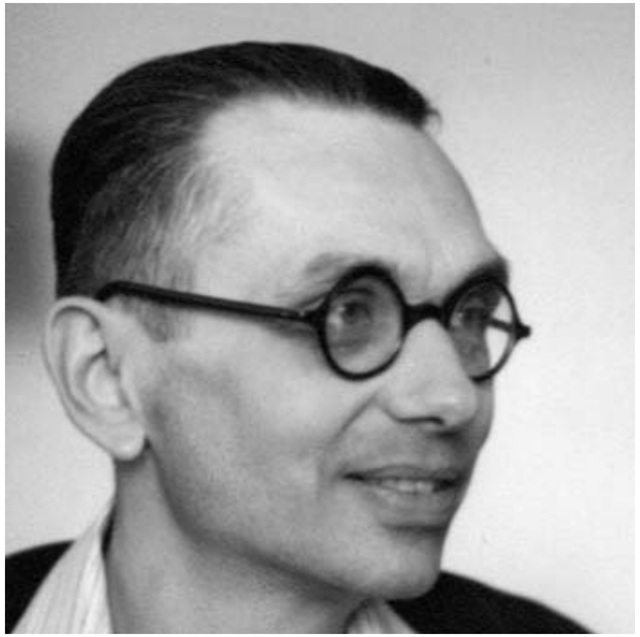
1. The continuum hypothesis
2. Prove that the axioms of arithmetic are consistent.
3. Given any two polyhedra of equal volume, is it always possible to cut the first into finitely many polyhedral pieces which can be reassembled to yield the second?
4. Construct all metrics where lines are geodesics.
5. Are continuous groups automatically differential groups?
6. Mathematical treatment of the axioms of physics
7. Is a^b transcendental, for algebraic $a \neq 0,1$ and irrational algebraic b ?
8. The Riemann hypothesis ("the real part of any non-trivial zero of the Riemann zeta function is ½") and other prime number problems, among them Goldbach's conjecture and the twin prime conjecture
9. Find the most general law of the reciprocity theorem in any algebraic number field.
10. Find an algorithm to determine whether a given polynomial Diophantine equation with integer coefficients has an integer solution.
11. Solving quadratic forms with algebraic numerical coefficients.
12. Extend the Kronecker–Weber theorem on abelian extensions of the rational numbers to any base number field.
13. Solve 7-th degree equation using continuous functions of two parameters.
14. Is the ring of invariants of an algebraic group acting on a polynomial ring always finitely generated?
15. Rigorous foundation of Schubert's enumerative calculus.
16. Describe relative positions of ovals originating from a real algebraic curve and as limit cycles of a polynomial vector field on the plane.
17. Express a nonnegative rational function as quotient of sums of squares.
18. (a) Is there a polyhedron which admits only an anisohedral tiling in three dimensions?
    (b) What is the densest sphere packing?
19. Are the solutions of regular problems in the calculus of variations always necessarily analytic?
20. Do all variational problems with certain boundary conditions have solutions?
21. Proof of the existence of linear differential equations having a prescribed monodromic group
22. Uniformization of analytic relations by means of automorphic functions
23. Further development of the calculus of variations



Hilbert’s Programs:

1. **Axiomatization for mathematics**, beginning with arithmetic, and a finitary consistency proof of that system.

2. **Entscheidungsproblem (decision problem)**. statements about mathematics be regarded as formal sequences of symbols, and Entscheidungsproblem was to find an algorithm to decide whether a statement was valid or not.



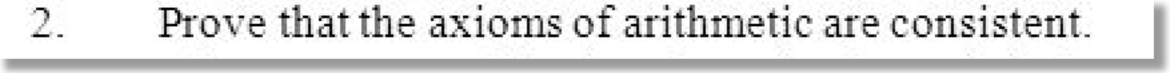
Kurt Gödel
April 28, 1906 – January 14, 1978

Program 1. Axiomatization for mathematics with finite consistency proofs.

Gödel's Incompleteness Theorems.

(Diagonalization)

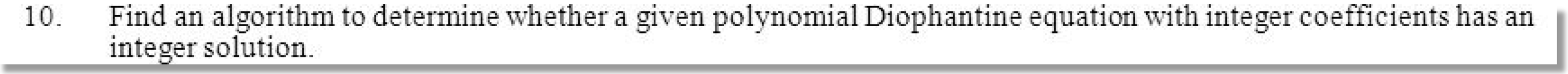
1. *Any consistent formal system is incomplete.*
2. *Any consistent formal system containing elementary arithmetic can not prove its own consistency.*



2. Prove that the axioms of arithmetic are consistent.

Program 2. **Entscheidungsproblem (decision problem)**.

“ Statements about mathematics be regarded as formal sequences of symbols, and Entscheidungsproblem was to find an algorithm to decide whether a statement was valid or not. “



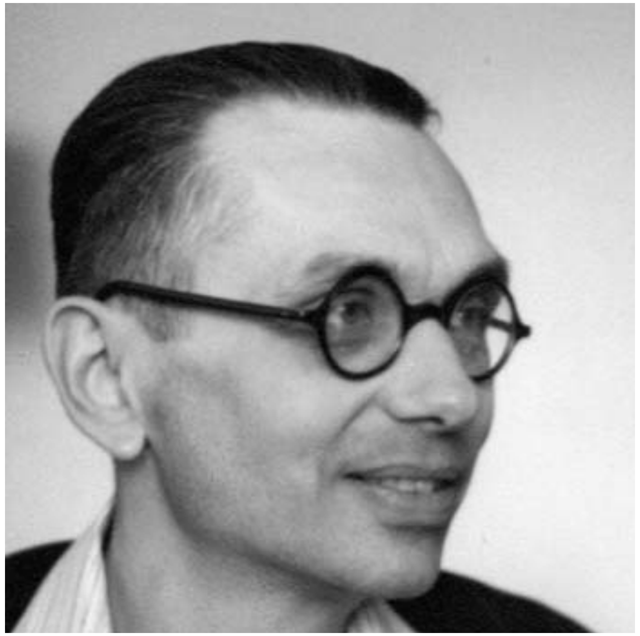
10. Find an algorithm to determine whether a given polynomial Diophantine equation with integer coefficients has an integer solution.

Program 2. Entscheidungsproblem (decision problem).

" Statements about mathematics be regarded as formal sequences of symbols, and Entscheidungsproblem was to find an algorithm to decide whether a statement was valid or not. "

Challenges:

1. *Find a precise mathematical definition for the intuitive idea of algorithm;*
2. *Demonstrate beyond doubt that every algorithm has been captured; and*
3. *Prove that no algorithm on the list can be the solution of the Diophantine equation problem*.



Kurt Gödel
April 28, 1906 – January 14, 1978

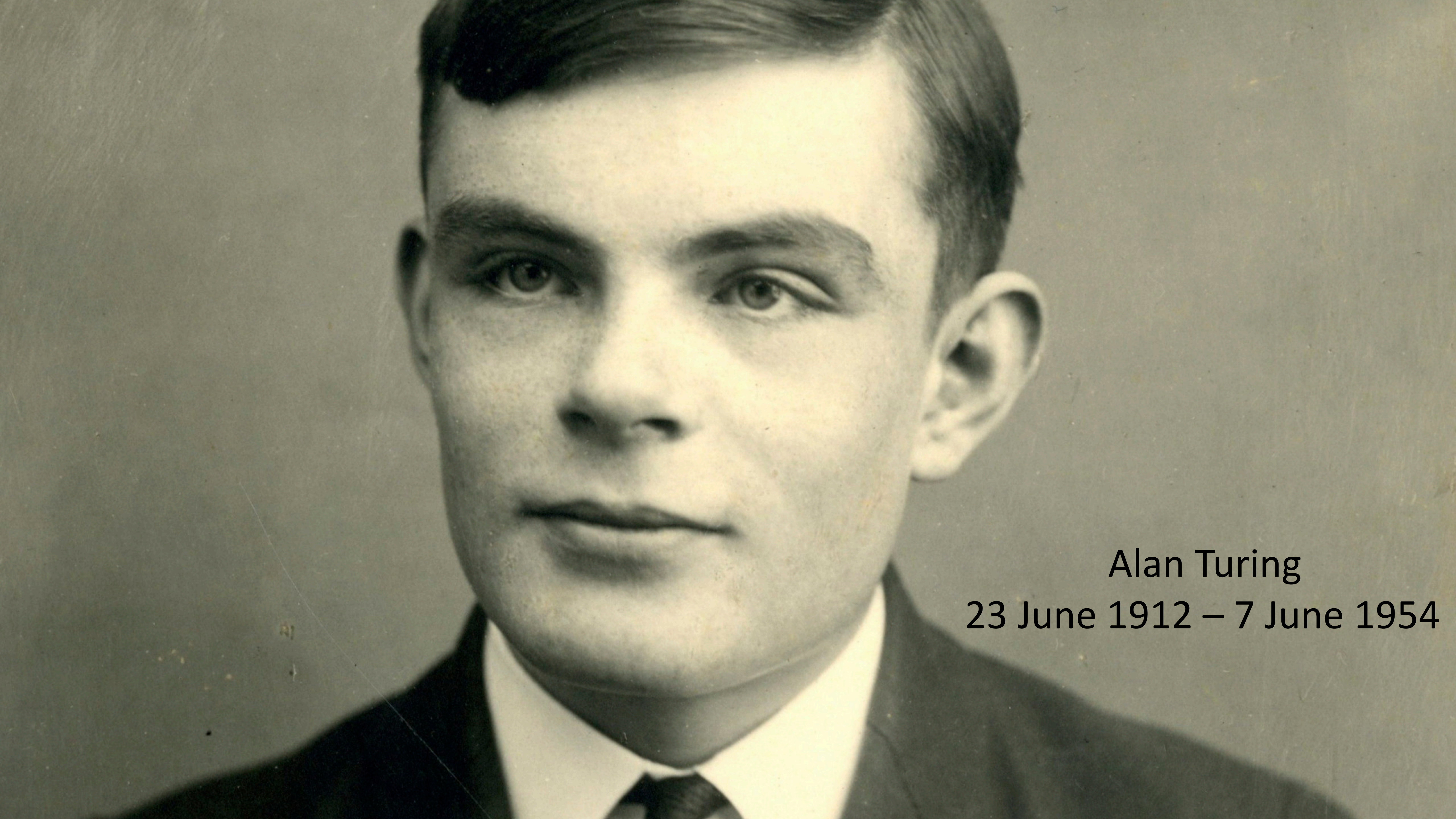
$\lambda - definable\ functions$

$\mu - recursive\ functions$

Alonzo Church
June 14, 1903 – August 11, 1995

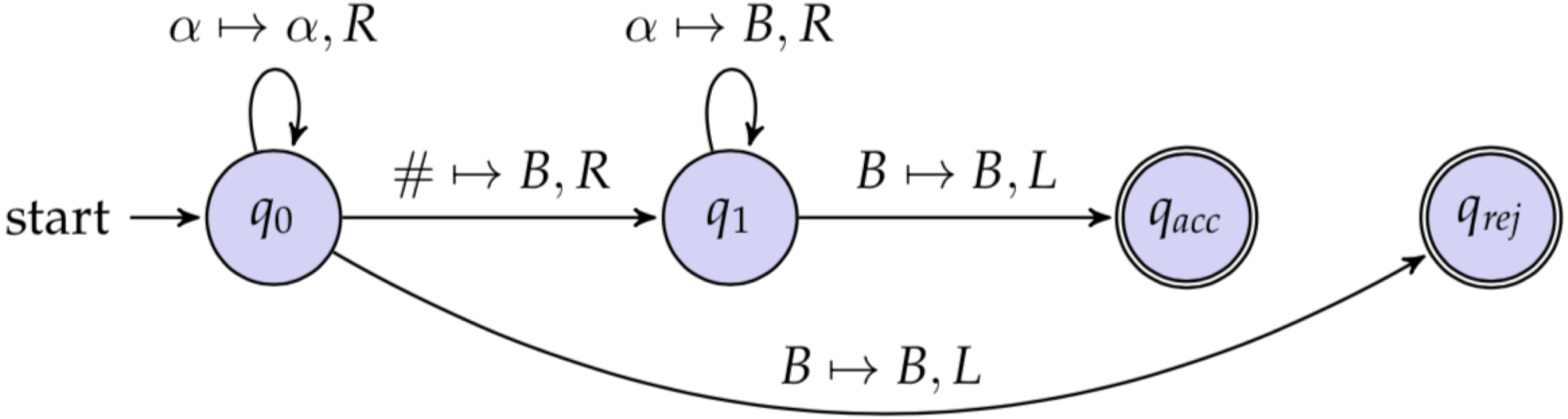
Challenges:

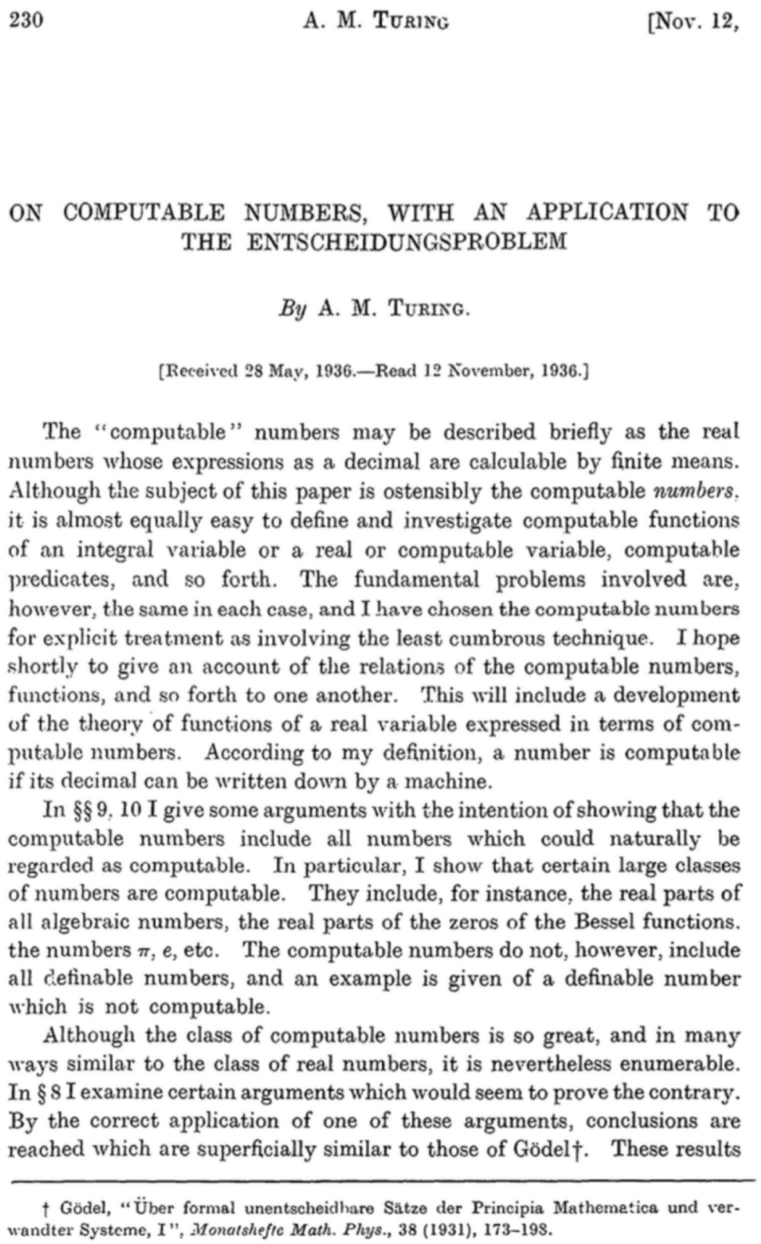
1. *Find a precise mathematical definition for the intuitive idea of algorithm;*
2. *Demonstrate beyond doubt that every algorithm has been captured; and*
3. *Prove that no algorithm on the list can be the solution of the Diophantine equation problem.*



Alan Turing
23 June 1912 – 7 June 1954

tape →	1	1	1	0	#	1	1	1	0	B	$\cdots$





ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

By A. M. TURING.

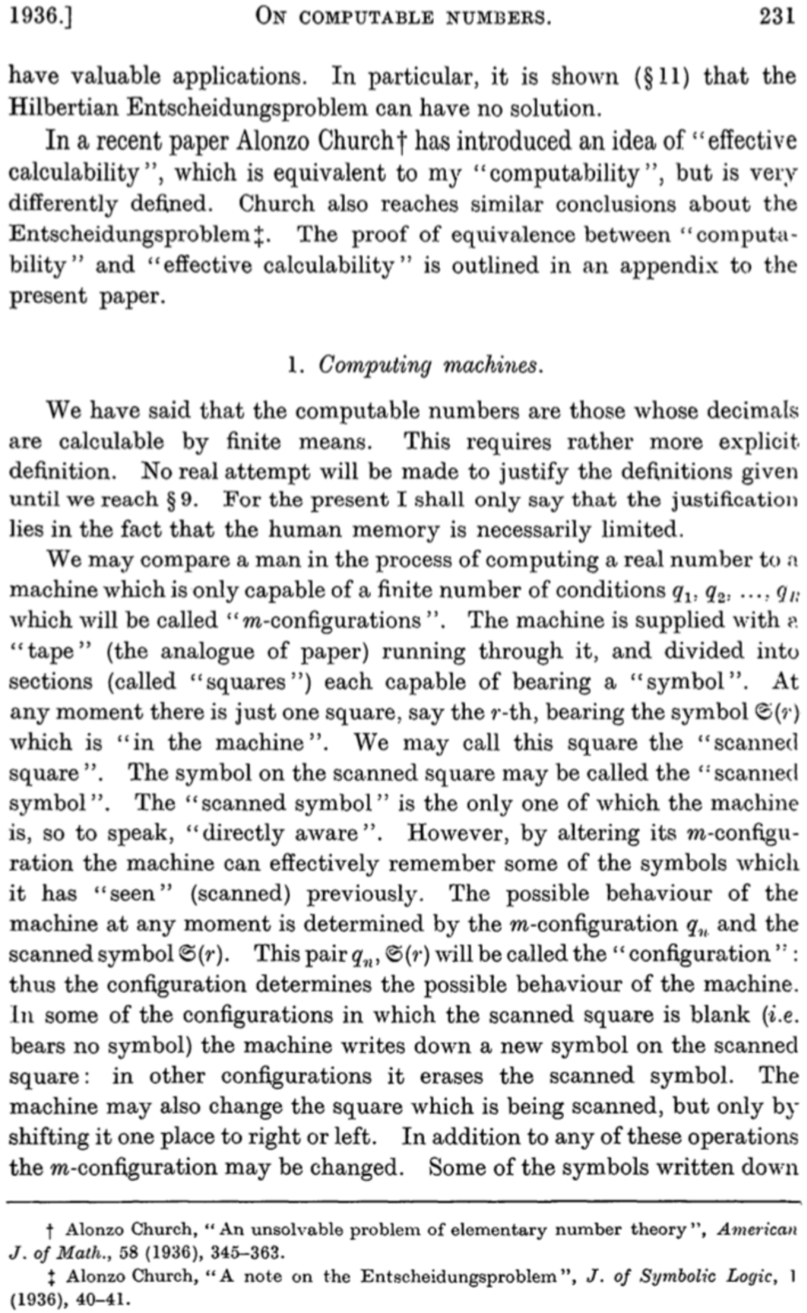
[Received 28 May, 1936.—Read 12 November, 1936.]

The "computable" numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable *numbers*, it is almost equally easy to define and investigate computable functions of an integral variable or a real or computable variable, computable predicates, and so forth. The fundamental problems involved are, however, the same in each case, and I have chosen the computable numbers for explicit treatment as involving the least cumbrous technique. I hope shortly to give an account of the relations of the computable numbers, functions, and so forth to one another. This will include a development of the theory of functions of a real variable expressed in terms of computable numbers. According to my definition, a number is computable if its decimal can be written down by a machine.

In §§ 9, 10 I give some arguments with the intention of showing that the computable numbers include all numbers which could naturally be regarded as computable. In particular, I show that certain large classes of numbers are computable. They include, for instance, the real parts of all algebraic numbers, the real parts of the zeros of the Bessel functions, the numbers π, e, etc. The computable numbers do not, however, include all definable numbers, and an example is given of a definable number which is not computable.

Although the class of computable numbers is so great, and in many ways similar to the class of real numbers, it is nevertheless enumerable. In § 8 I examine certain arguments which would seem to prove the contrary. By the correct application of one of these arguments, conclusions are reached which are superficially similar to those of Gödel[†]. These results

[†] Gödel, "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I", *Monatshefte Math. Phys.*, 38 (1931), 173–198.



have valuable applications. In particular, it is shown (§ 11) that the Hilbertian Entscheidungsproblem can have no solution.

In a recent paper Alonzo Church[†] has introduced an idea of "effective calculability", which is equivalent to my "computability", but is very differently defined. Church also reaches similar conclusions about the Entscheidungsproblem[‡]. The proof of equivalence between "computability" and "effective calculability" is outlined in an appendix to the present paper.

1. *Computing machines.*

We have said that the computable numbers are those whose decimals are calculable by finite means. This requires rather more explicit definition. No real attempt will be made to justify the definitions given until we reach § 9. For the 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 $q_1, q_2, \ldots, q_R$ 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

[†] Alonzo Church, "An unsolvable problem of elementary number theory", *American J. of Math.*, 58 (1936), 345–363.

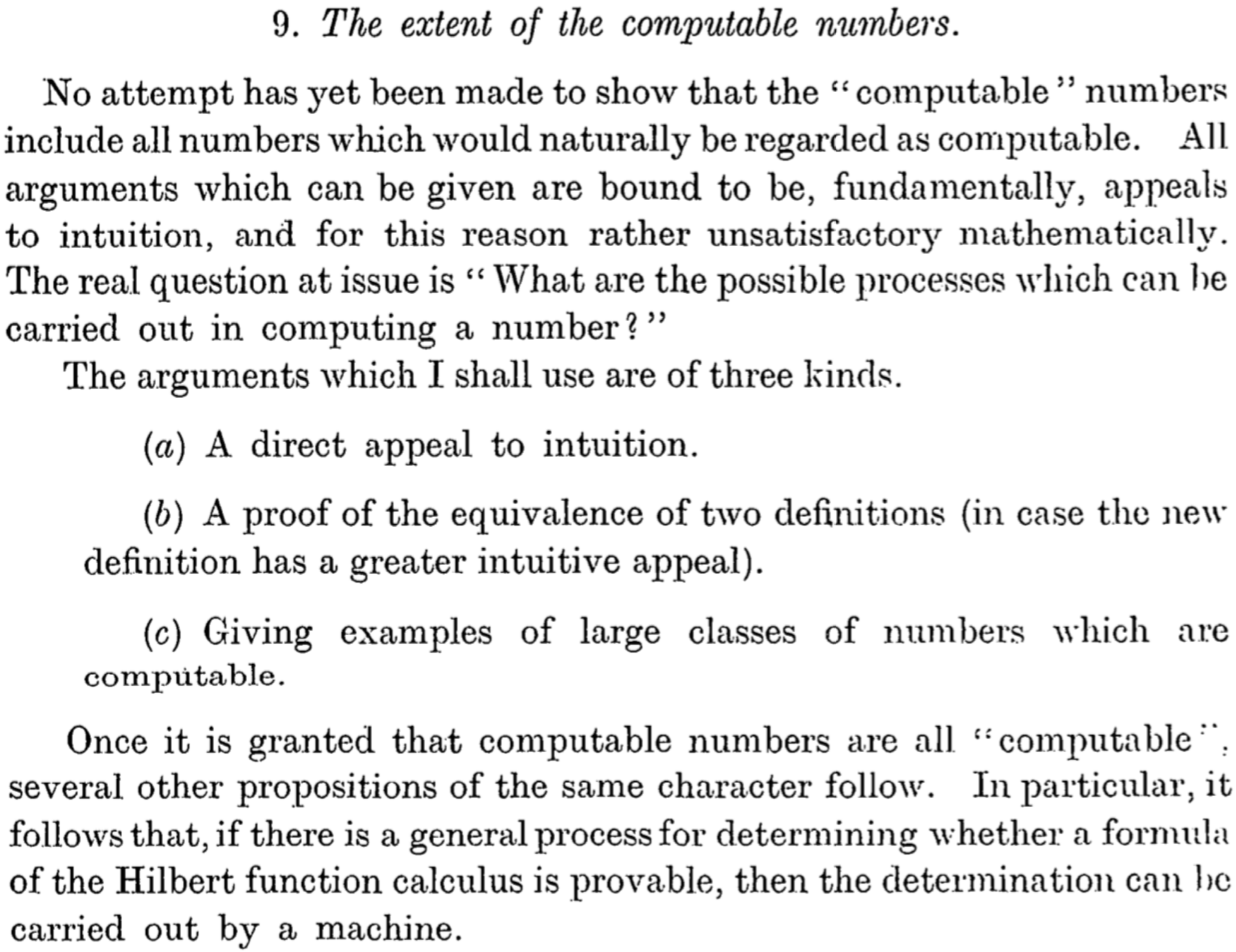
[‡] Alonzo Church, "A note on the Entscheidungsproblem", *J. of Symbolic Logic*, 1 (1936), 40–41.

11. *Application to the Entscheidungsproblem.*

The results of § 8 have some important applications. In particular, they can be used to show that the Hilbert Entscheidungsproblem can have no solution. For the present I shall confine myself to proving this particular theorem. For the formulation of this problem I must refer the reader to Hilbert and Ackermann's *Grundzüge der Theoretischen Logik* (Berlin, 1931), chapter 3.

I propose, therefore, to show that there can be no general process for determining whether a given formula $\mathfrak{A}$ of the functional calculus K is provable, *i.e.* that there can be no machine which, supplied with any one $\mathfrak{A}$ of these formulae, will eventually say whether $\mathfrak{A}$ is provable.

It should perhaps be remarked that what I shall prove is quite different from the well-known results of Gödel[†]. Gödel has shown that (in the formalism of Principia Mathematica) there are propositions $\mathfrak{A}$ such that neither $\mathfrak{A}$ nor $-\mathfrak{A}$ is provable. As a consequence of this, it is shown that no proof of consistency of Principia Mathematica (or of K) can be given within that formalism. On the other hand, I shall show that there is no general method which tells whether a given formula $\mathfrak{A}$ is provable in K, or, what comes to the same, whether the system consisting of K with $-\mathfrak{A}$ adjoined as an extra axiom is consistent.



9. *The extent of the computable numbers.*

No attempt has yet been made to show that the "computable" numbers include all numbers which would naturally be regarded as computable. All arguments which can be given are bound to be, fundamentally, appeals to intuition, and for this reason rather unsatisfactory mathematically. The real question at issue is "What are the possible processes which can be carried out in computing a number?"

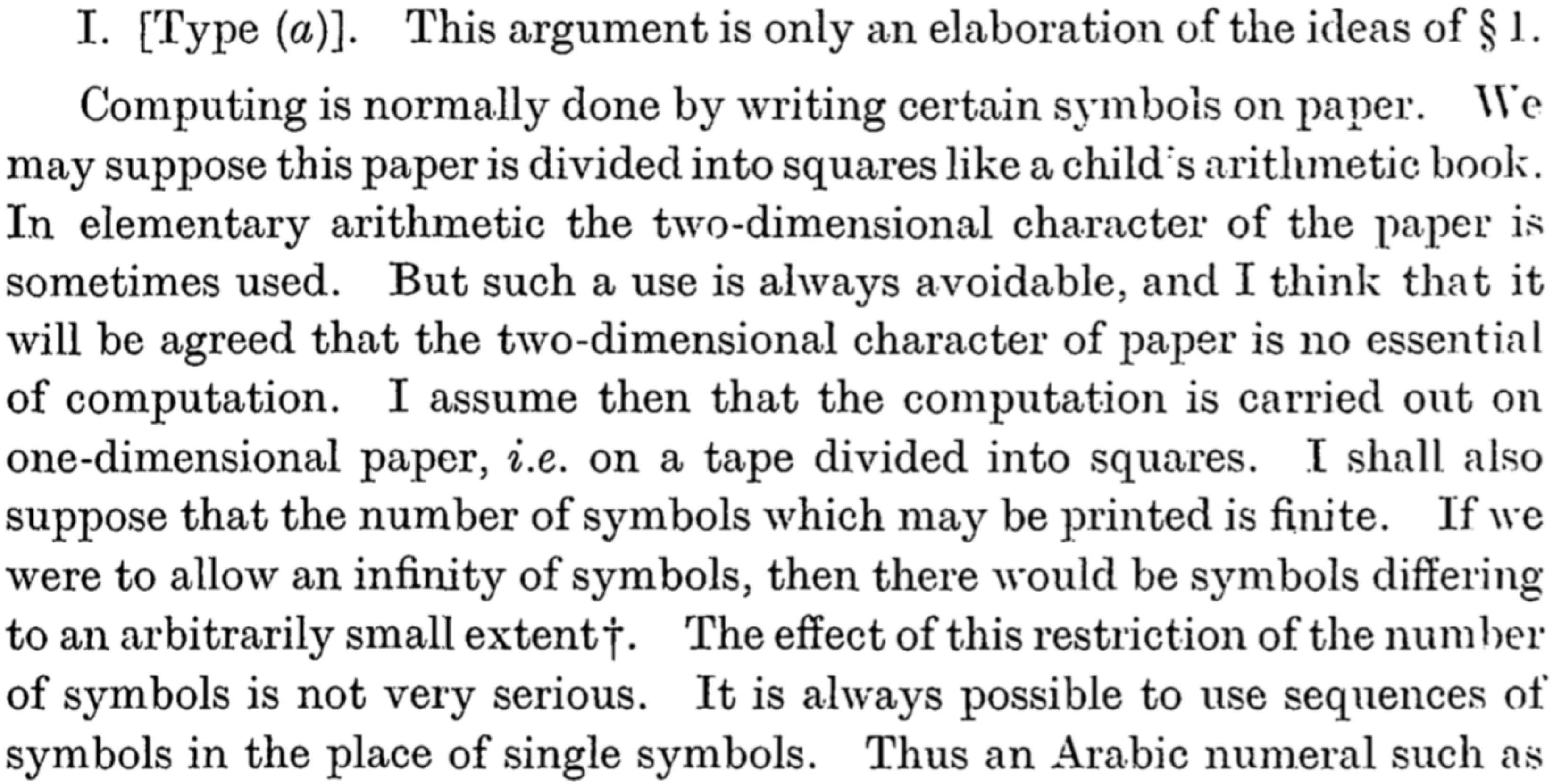
The arguments which I shall use are of three kinds.

(*a*) A direct appeal to intuition.

(*b*) A proof of the equivalence of two definitions (in case the new definition has a greater intuitive appeal).

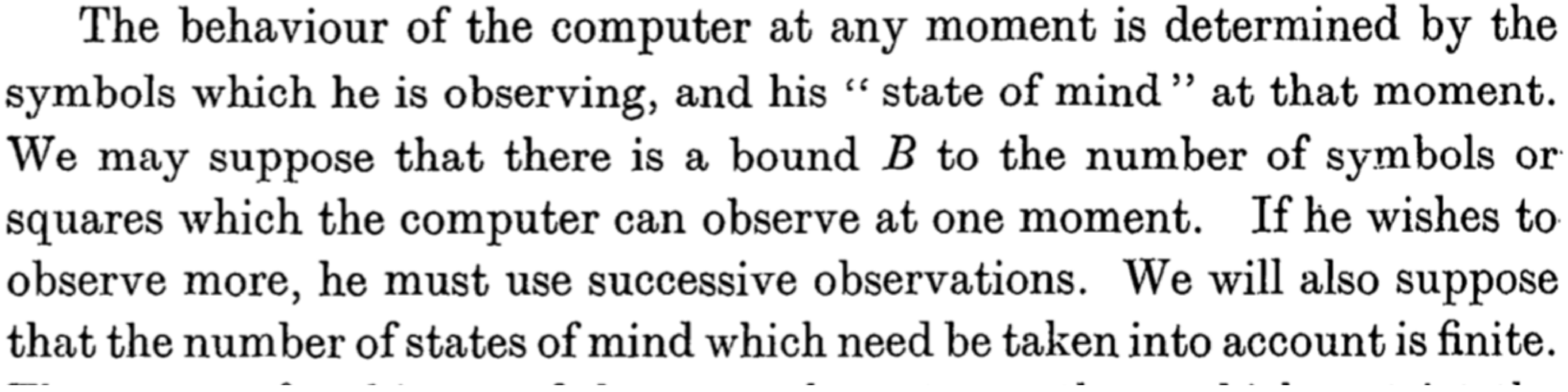
(*c*) Giving examples of large classes of numbers which are computable.

Once it is granted that computable numbers are all "computable", several other propositions of the same character follow. In particular, it follows that, if there is a general process for determining whether a formula of the Hilbert function calculus is provable, then the determination can be carried out by a machine.

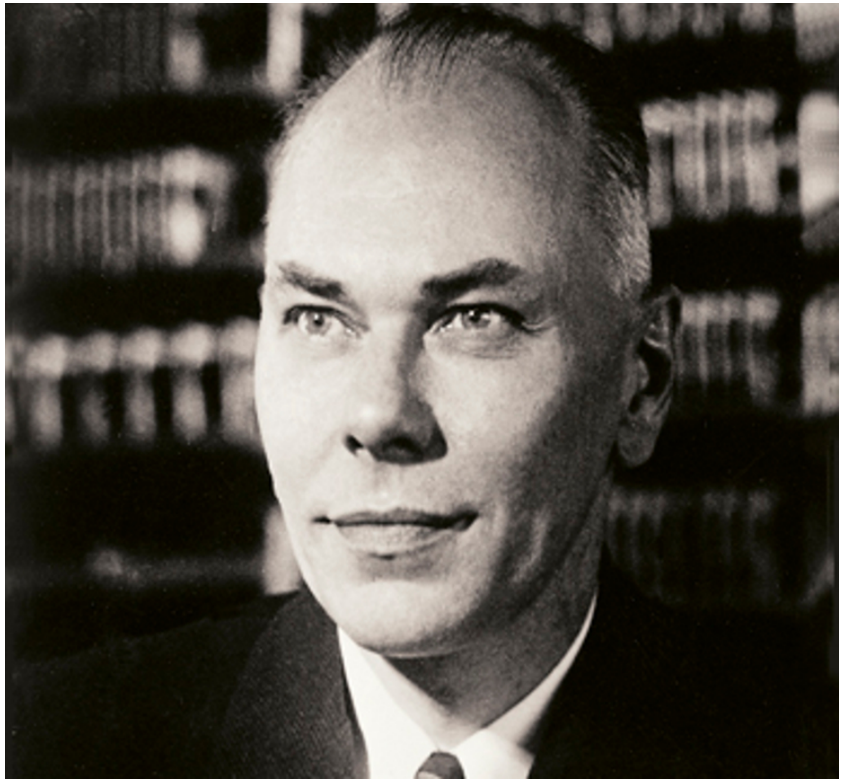


I. [Type (*a*)]. This argument is only an elaboration of the ideas of § 1.

Computing is normally done by writing certain symbols on paper. We may suppose this paper is divided into squares like a child's arithmetic book. In elementary arithmetic the two-dimensional character of the paper is sometimes used. But such a use is always avoidable, and I think that it will be agreed that the two-dimensional character of paper is no essential of computation. I assume then that the computation is carried out on one-dimensional paper, *i.e.* on a tape divided into squares. I shall also suppose that the number of symbols which may be printed is finite. If we were to allow an infinity of symbols, then there would be symbols differing to an arbitrarily small extent†. The effect of this restriction of the number of symbols is not very serious. It is always possible to use sequences of symbols in the place of single symbols. Thus an Arabic numeral such as



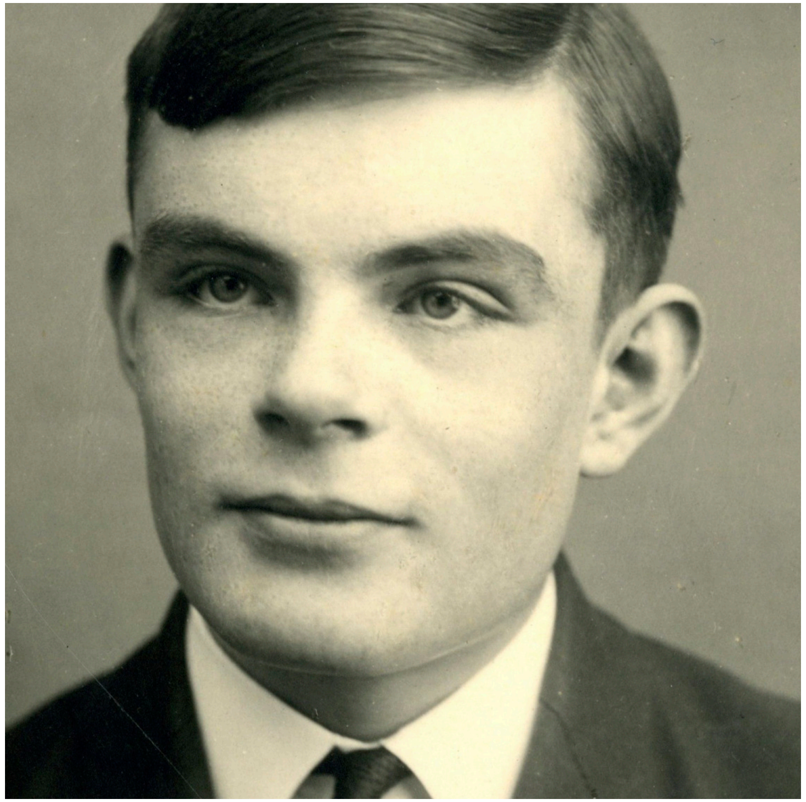
The behaviour of the computer at any moment is determined by the symbols which he is observing, and his "state of mind" at that moment. We may suppose that there is a bound B to the number of symbols or squares which the computer can observe at one moment. If he wishes to observe more, he must use successive observations. We will also suppose that the number of states of mind which need be taken into account is finite.



Howard Hathaway Aiken (March 8, 1900 – March 14, 1973) was an American physicist and a pioneer in computing, being the original conceptual designer behind IBM's Harvard Mark I computer.

"If it should turn out that basic logics of machine designed for the numerical solutions of differential equations coincide with the logics of a machine intended to make bills for a department store, I would regard this as the most amazing coincidence I have ever encountered."

Howard Aiken, 1956



"Let us now return to analogy of the theoretical computing machines ... It can be shown that a single special machine of that type can be made to do the work of all. It could in fact be made to work as a model of any other machine. This special machine may be called the universal machine."

Alan Turing, 1947

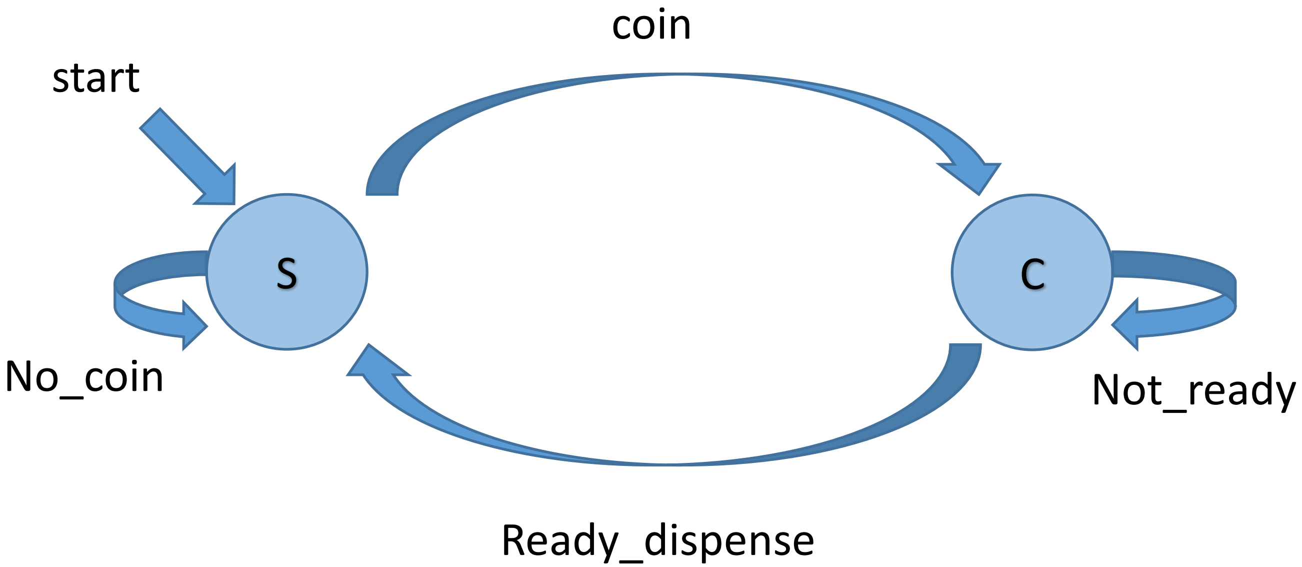
Part I: Automata Theory

What’s an automaton?

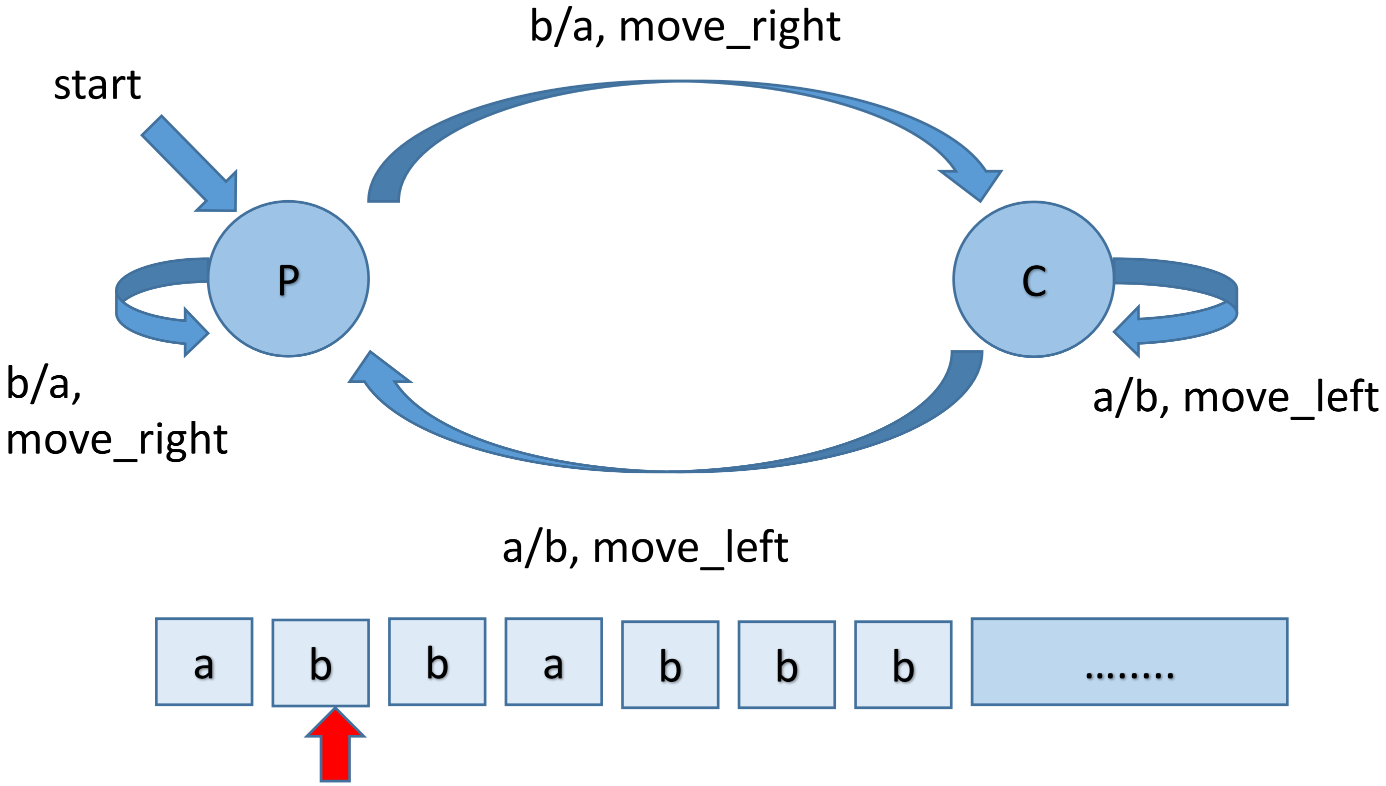
1. A moving mechanical device made in imitation of a human being.
2. A **machine** that performs a **function** according to a **predetermined set** of coded **instructions**.

What’s an automaton?

1. A moving mechanical device made in imitation of a human being.
2. A **machine** that performs a **function** according to a **predetermined set** of coded **instructions**.



Finite instruction machine with finite memory (*Finite State Automata*)



start

b/a, move_right

b/a,
move_right

a/b, move_left

a/b, move_left

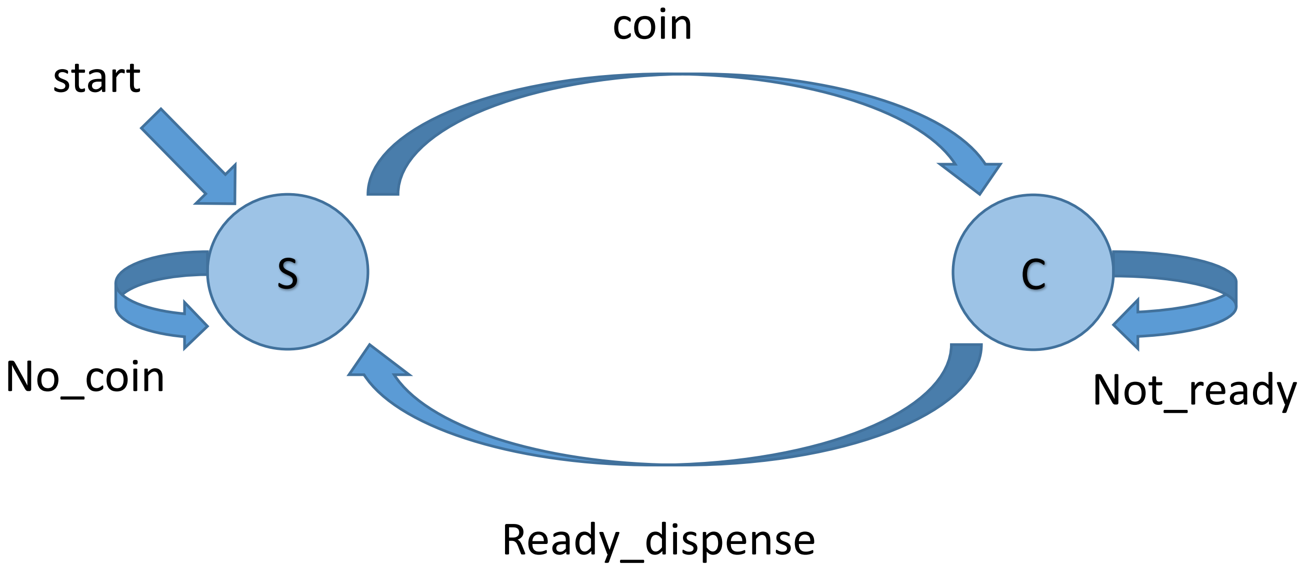
P

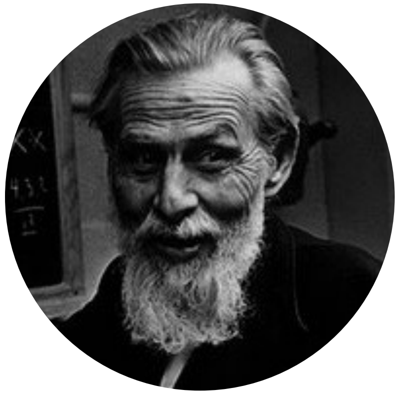
C

a	b	b	a	b	b	b	……..

Finite instruction machine with unbounded memory (Universal *Turing machine*)

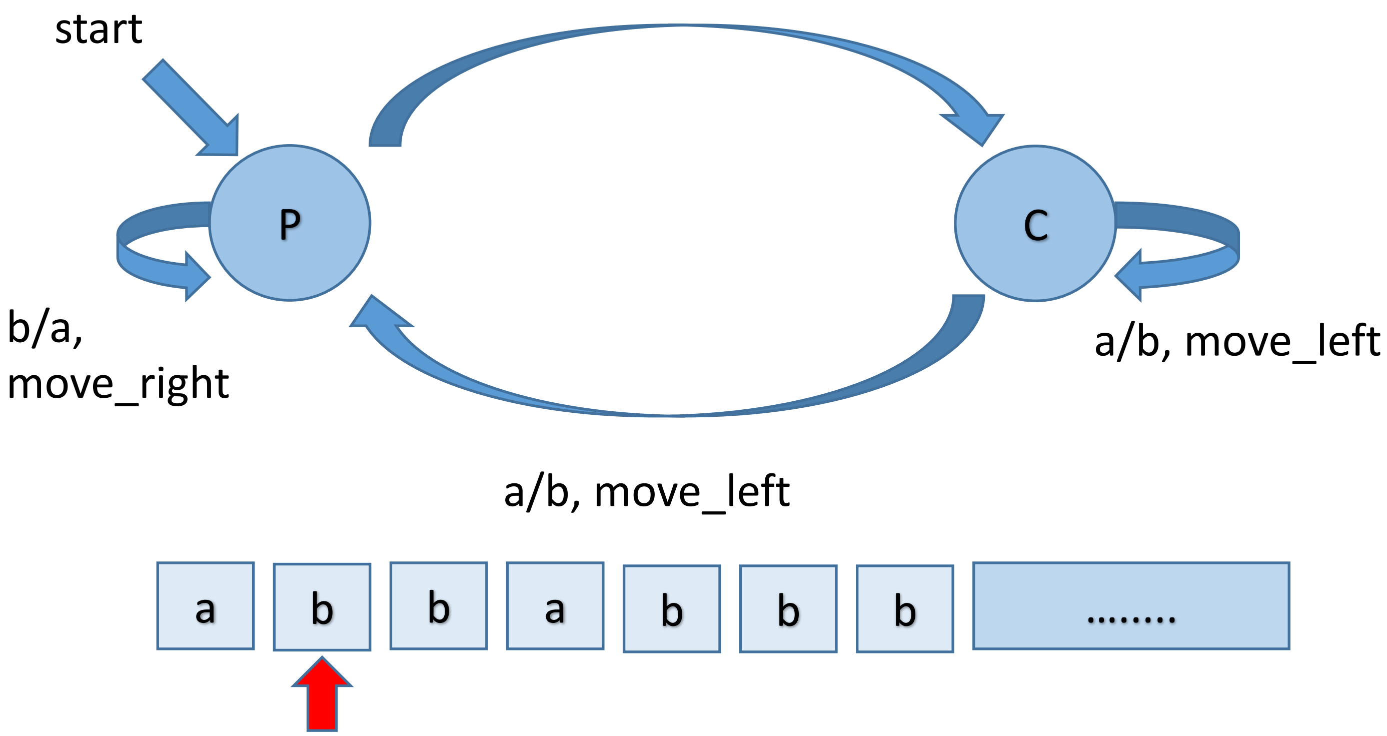
Finite State Automata





- Introduced first by two neuro-psychologists **Warren S. McCullough** and **Walter Pitts** in 1943 as a model for human brain!
- Finite automata can naturally model **microprocessors** and even **software programs** working on variables with bounded domain
- Capture so-called **regular sets** of sequences that occur in many different fields (logic, algebra, regular Expressions)
- Nice theoretical properties
- Applications in **digital circuit/protocol verification**, **compilers**, **pattern recognition**, and so on.

Turing Machines



- Introduced by **Alan Turing** as a simple model capable of expressing any imaginable computation
- Turing machines are widely accepted as a synonym for algorithmic computability (Church-Turing thesis)
- Using these conceptual machines Turing showed that *first-order logic validity problem* is **non-computable**.
- I.e. there exists some problems for which you can never write a program no matter how hard you try!