Logic, Algorithms, and Automata
A Historical Journey

Wolfgang Thomas

RWTH AACHEN

Francqui Lecture, Mons, April 2013

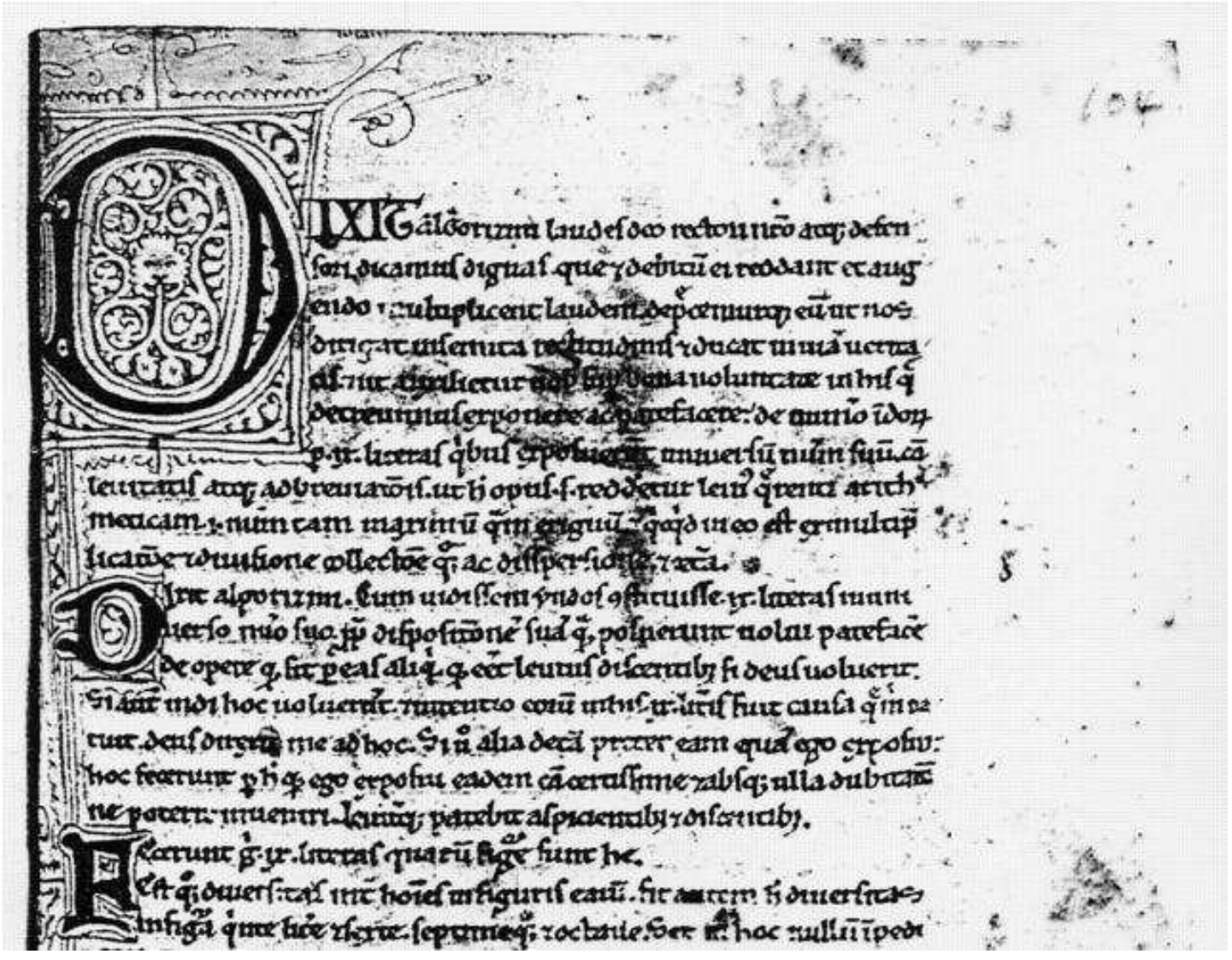
Prelude

Some “Prehistory”: Al-Khwarizmi and Leibniz

Bagdad around the year 800

Muhammad Abu-Abdullah Abu-Jafar ibn Musa
Al-Khwarizmi Al-Madjusti Al-Qutrubulli
(ca. 780 - 850)

Al-Khwarizmi: On the Indian Numbers



Gottfried Wilhelm Leibniz (1646 - 1716)

From a Letter to Duke Johann Friedrich

“In Philosophia habe ich ein Mittel funden, dasjenige was Cartesius und andere per Algebram et Analysin in Arithmetica et Geometria gethan, in allen scientien zuwege zu bringen per Artem Combinatoriam [. . .]. Dadurch alle Notiones compositae der ganzen welt in wenig simplices als deren Alphabet reduciret, und aus solches alphabets combination wiederumb alle dinge, samt ihren theorematibus, und was nur von ihnen zu inventiren müglich, ordinata methodo, mit der zeit zu finden, ein weg gebahnet wird.”

Arithmetization of Logic I (1685-87)

Non inelegans specimen demonstrandi in abstractis

(A not inelegant example of abstract proof method)

Theorem XIII.

Si coincidentibus addendo alia fiant coincidentia, addita sunt inter se communicantia.

If from coincidents one obtains other coincidents by addition, the added entities have something in common.

If $A + B = A + N$ **and** $A \neq A + B$**, then** $B \cap N \neq \emptyset$

This prepares Boolean Algebra as a calculus, using notation of arithmetic.

Arithmetization of Logic II (1679)

Elementa calculi

(Elements of a calculus)

Verbi gratia quia Homo est Animal rationale (et quia Aurum est metallum ponderosissimum) hinc si sit Animalis (metalii) numerus a ut 2 (m ut 3) Rationalis (ponderosissimi) vero numerus r ut 3 (p ut 5) erit numerus hominis seu h idem quot ar id est in hoc exemplo $2, 3$ seu 6 (et numerus auri solis s idem quot mp id est in hoc exemplo $3, 5$ seu 15.

This prepares the idea of Gödel numbering: Coding concepts by prime numbers and their conjunction by multiplication.

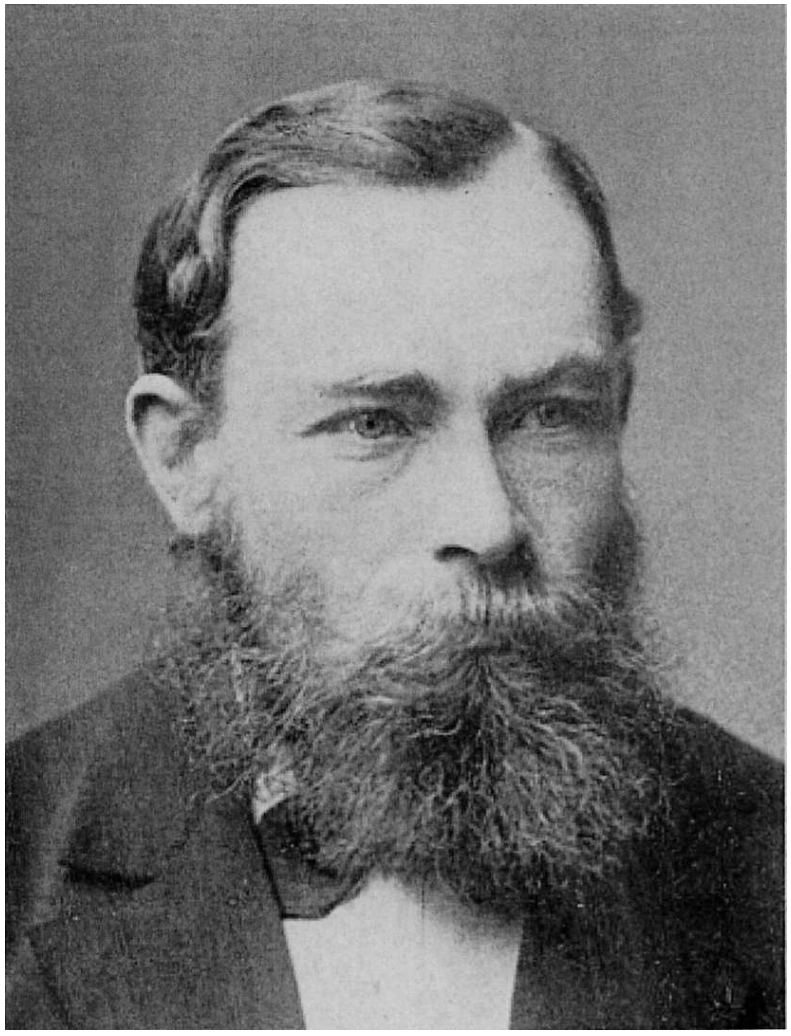
Optimism

from the untitled manuscript “Fundamentals of a universal characteristic”:

When this language is introduced sometime by the missionaries, then the true religion, which is unified to the best with rationality, will be founded firmly, and one does not need to fear a renunciation of man from it in the future, just as one does not need to fear a reunciation from algebra and geometry.

I think that some selected people can do the job in five years, and that already after two years they will reach a stage where the theories needed most urgently for life, i.e., moral and metaphysics, are managable by an unfallible calculus.

The Rise of Mathematical Logic



Gottlob Frege (1848 - 1926)

BEGRIFFSSCHRIFT,

EINE DER ARITHMETISCHEN NACHGEBILDETE

FORMELSPRACHE

DES REINEN DENKENS.

VON

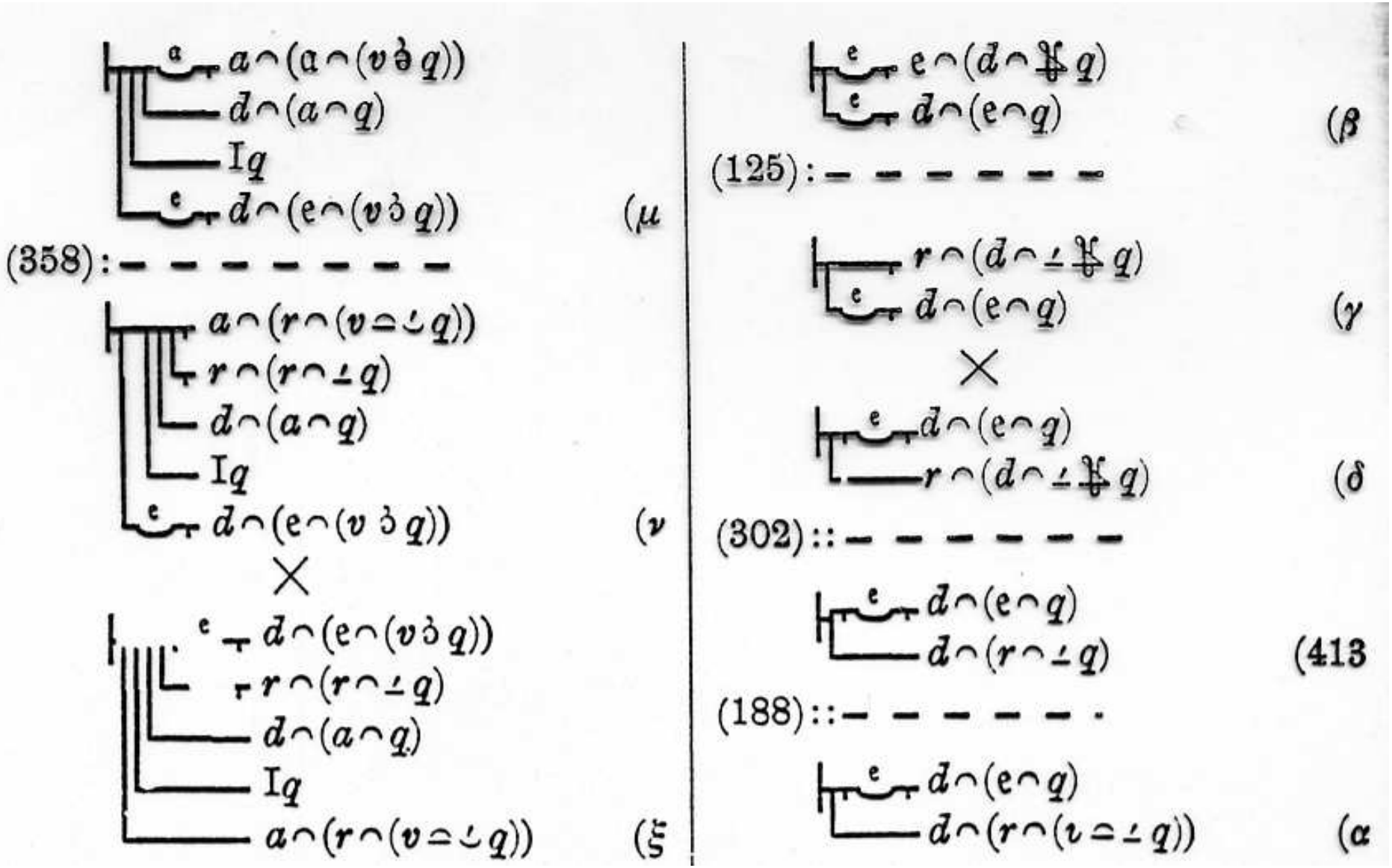
DR. GOTTLOB FREGE,

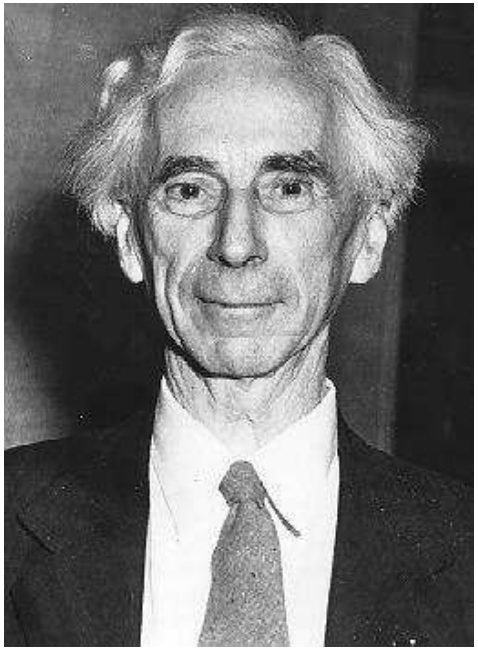
PRIVATDOCENTEN DER MATHEMATIK AN DER UNIVERSITÄT JENA.

HALLE A/S.

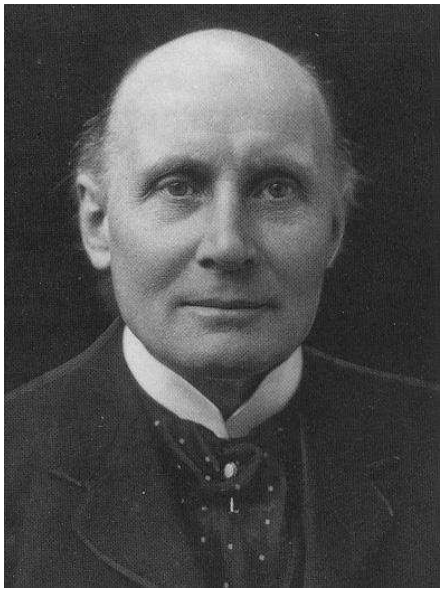
VERLAG VON LOUIS NEBERT.

1879.


a⌢(a⌢(v∂q))
d⌢(a⌢q)
Iq
e d⌢(e⌢(v∂q)) (μ
(358):
a⌢(r⌢(v⊒⊥q))
r⌢(r⌢⊥q)
d⌢(a⌢q)
Iq
e d⌢(e⌢(v∂q)) (ν
×
e d⌢(e⌢(v∂q))
r⌢(r⌢⊥q)
d⌢(a⌢q)
Iq
a⌢(r⌢(v⊒⊥q)) (ξ
e e⌢(d⌢⊬q)
e d⌢(e⌢q) (β
(125):
r⌢(d⌢⊥⊬q)
e d⌢(e⌢q) (γ
×
e d⌢(e⌢q)
r⌢(d⌢⊥⊬q) (δ
(302)::
e d⌢(e⌢q)
d⌢(r⌢⊥q) (413
(188)::
e d⌢(e⌢q)
d⌢(r⌢(ι⊒⊥q)) (α



B. Russell



A. N. Whitehead

$*54{\cdot}443.\quad \vdash :: x \neq y : \phi(x,y) . \equiv . \phi(y,x) : \supset :.$
$z, w \in \iota`x \cup \iota`y . z \neq w . \supset_{z,w} . \phi(z,w) : \equiv . \phi(x,y) \quad [*54{\cdot}442]$

$*54{\cdot}45.\quad \vdash :. (\exists z, w) . z, w \in \iota`x \cup \iota`y . \phi(z,w) .$
$\equiv : \phi(x,x) . \vee . \phi(x,y) . \vee . \phi(y,x) . \vee . \phi(y,y) \quad [*51{\cdot}235]$

$*54{\cdot}451.\quad \vdash :: \sim\phi(x,x) . \sim\phi(y,y) . \supset :. (\exists z, w) . z, w \in \iota`x \cup \iota`y . \phi(z,w) .$
$\equiv : \phi(x,y) . \vee . \phi(y,x) \quad [*54{\cdot}45]$

$*54{\cdot}452.\quad \vdash :: \sim\phi(x,x) . \sim\phi(y,y) : \phi(x,y) . \equiv . \phi(y,x) : \supset :$
$(\exists z, w) . z, w \in \iota`x \cup \iota`y . \phi(z,w) . \equiv . \phi(x,y) \quad [*54{\cdot}451]$

$*54{\cdot}46.\quad \vdash : (\exists z, w) . z, w \in \iota`x \cup \iota`y . z \neq w . \equiv . x \neq y \quad [*54{\cdot}452 . *13{\cdot}15{\cdot}16]$

$*54{\cdot}5.\quad \vdash :. \alpha \in 2 . \supset : \alpha \subset \iota`z \cup \iota`w . \equiv . \alpha = \iota`z \cup \iota`w$

Dem.

$\vdash . *54{\cdot}4 . \supset$

$\vdash :. \alpha \subset \iota`z \cup \iota`w . \supset : \alpha = \Lambda . \vee . \alpha = \iota`z . \vee . \alpha = \iota`w . \vee . \alpha = \iota`z \cup \iota`w \qquad (1)$

$\vdash . *54{\cdot}3 . *24{\cdot}54 . \qquad \supset \vdash : \mathrm{Hp} . \supset . \alpha \neq \Lambda \qquad (2)$

$\vdash . *54{\cdot}26 \frac{z, z}{x, y} . *13{\cdot}15 . \qquad \supset \vdash : \mathrm{Hp} . \supset . \alpha \neq \iota`z \qquad (3)$

$\vdash . (3) \frac{w}{z} . \qquad \supset \vdash : \mathrm{Hp} . \supset . \alpha \neq \iota`w \qquad (4)$

$\vdash . (1) . (2) . (3) . (4) . *3{\cdot}53 . \supset \vdash : \mathrm{Hp} . \supset : \alpha \subset \iota`z \cup \iota`w . \supset . \alpha = \iota`z \cup \iota`w \qquad (5)$

David Hilbert (1862 - 1943)

Hilbert's Program

Coding mathematics to enable tight consistency proofs:

- **Development of a proof calculus**
- **Development of axiomatizations of mathematical theories**
- **Finitary analysis of formal proofs to exclude the derivation of "0 = 1"**

Fundamental problems:

- **Soundness and completeness: Are precisely the universally valid formulas formally derivable?**
- **Complete axiomatizations of concrete theories**

Kurt Gödel (1906 - 1978)

Hilbert's Entscheidungsproblem (1928)

Das Entscheidungsproblem ist gelöst, wenn man ein Verfahren kennt, das bei einem vorgelegten logischen Ausdruck durch endlich viele Operationen die Entscheidung über die Allgemeingültigkeit bzw. Erfüllbarkeit erlaubt.

Universally vaild: $\exists x \forall y Rxy \rightarrow \forall y \exists x Rxy$

Alan Turing

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

Seven Innovations

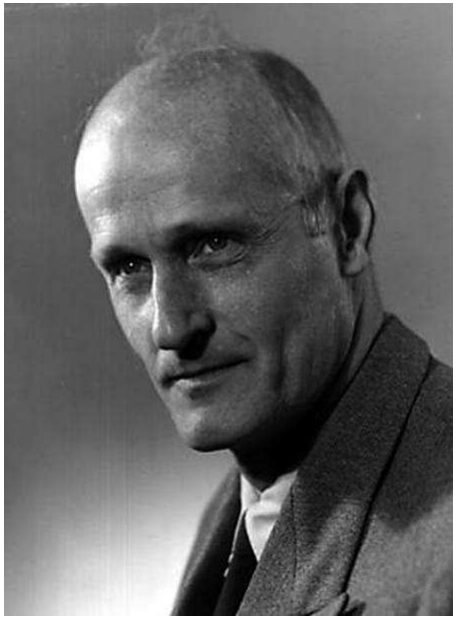
1. A machine model capturing computability
2. Its justification
3. Conception and implementation of a universal program
4. Establishment of a non-solvable problem
5. Proof that Hilbert's Entscheidungsproblem is undecidable
6. Equivalence between Turing machines and λ-calculus
7. Initial steps to computable analysis

Towards the Turing Machine

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

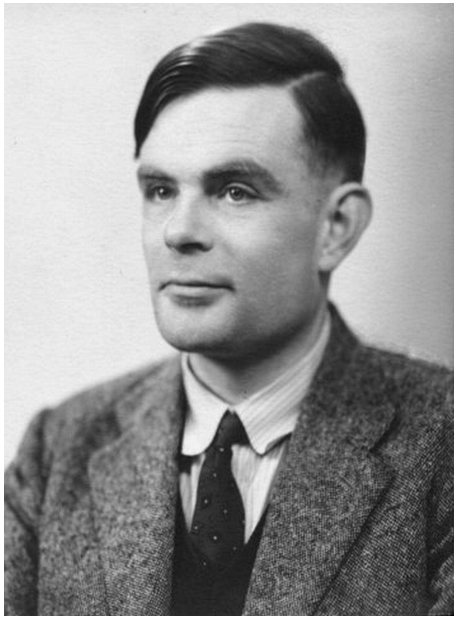
Pioneers of 1936

A. Church



S. Kleene

E. Post



A. Turing

Gödel in 1946

Tarski has stressed [. . .] the great importance of the concept of general recursiveness (or Turing's computability). It seems to me that this importance is largely due to the fact that with this concept one has for the first time succeeded in giving an absolute definition of an interesting epistemological notion, i.e., one not depending on the formalism chosen.

By a kind of miracle it is not necessary to distinguish orders.

Algorithms: Further Dimensions

- Algorithms over discrete structures
  (Lists, trees, graphs, etc.)
- Algorithms of analysis and geometry
- Non-terminating reactive systems
- Etablishing equilibria in distributed systems
- Procedures for cognition and classification
  (in data mining, image and speech processing)
- Hierarchical system architectures

Turing's work had a double influence:

- **as the final step in attempts over centuries to obtain a complete understanding of "algorithm"**
  — in the context of symbolic computation, unifying arithmetic and logic,

- **as a starting point**

  giving rise to a new science – informatics – that has enormously widened the range of algorithmic methods.

Moves towards Computer Science

1. Turing's work on computer architecture and verification

2. Post's establishment of undecidable purely combinatorial problems (e.g. word problem for Semi-Thue systems or Post's Correspondence Problem)

3. Kleene's nerve nets, automata, and equivalence to regular expressions

4. Church's Problem of circuit synthesis

Maurice Boffa (1939-2001)

Automata

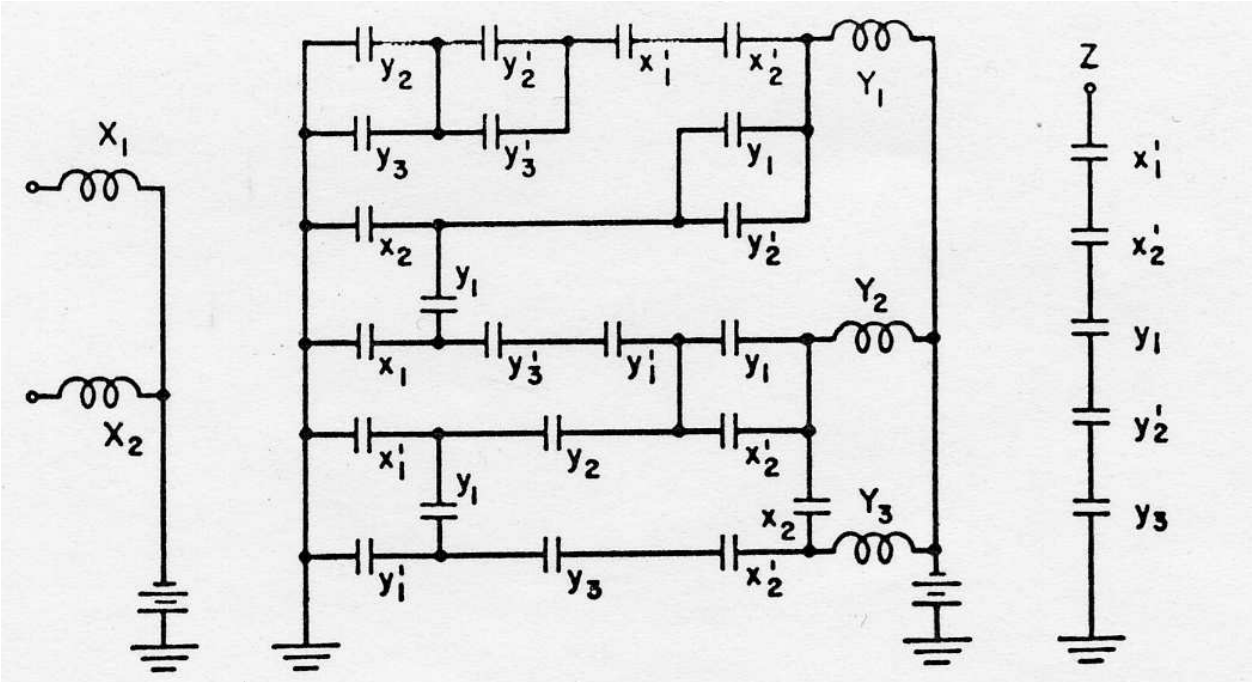


FIG. 3. A sequential switching circuit.

aus: D.A. Huffman, The synthesis of sequential switching circuits, J. Franklin Inst. 1954

S. Ulam, R. Feynman, J. von Neumann

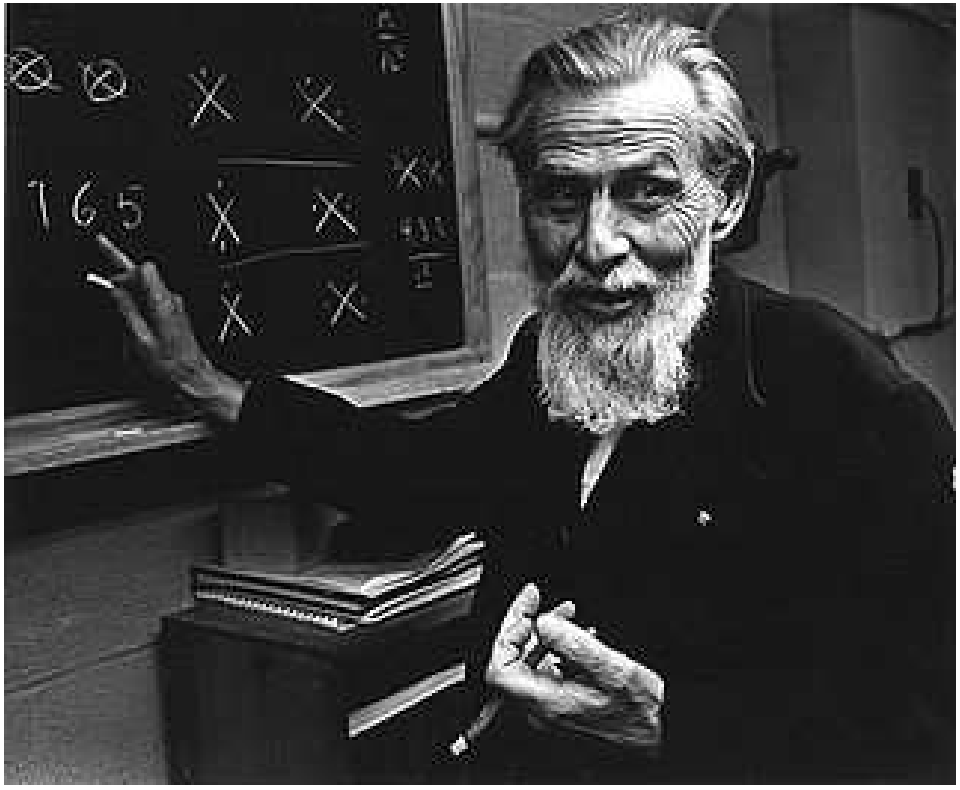
BULLETIN OF
MATHEMATICAL BIOPHYSICS
VOLUME 5, 1943

A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY

WARREN S. MCCULLOCH AND WALTER PITTS

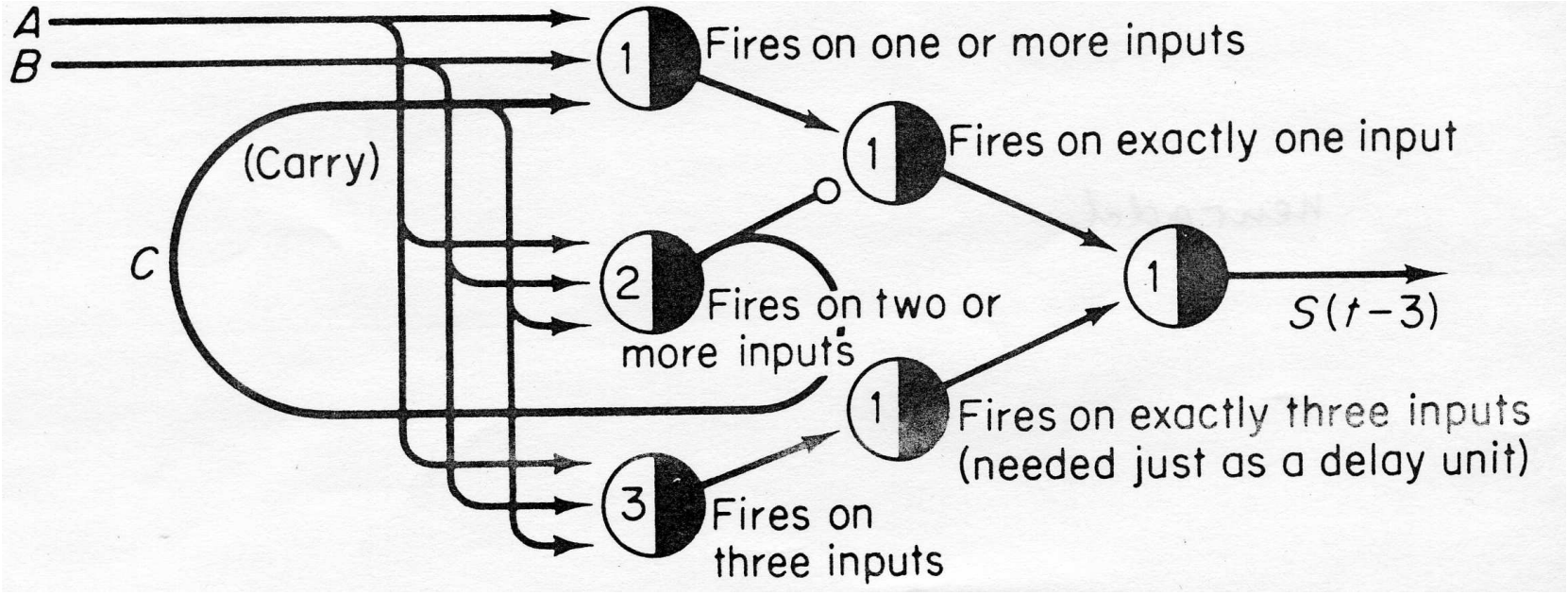
FROM THE UNIVERSITY OF ILLINOIS, COLLEGE OF MEDICINE,
DEPARTMENT OF PSYCHIATRY AT THE ILLINOIS NEUROPSYCHIATRIC INSTITUTE,
AND THE UNIVERSITY OF CHICAGO

Because of the "all-or-none" character of nervous activity, neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms, with the addition of more complicated logical means for nets containing circles; and that for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes. It is shown that many particular choices among possible neurophysiological assumptions are equivalent, in the sense that for every net behaving under one assumption, there exists another net which behaves under the other and gives the same results, although perhaps not in the same time. Various applications of the calculus are discussed.

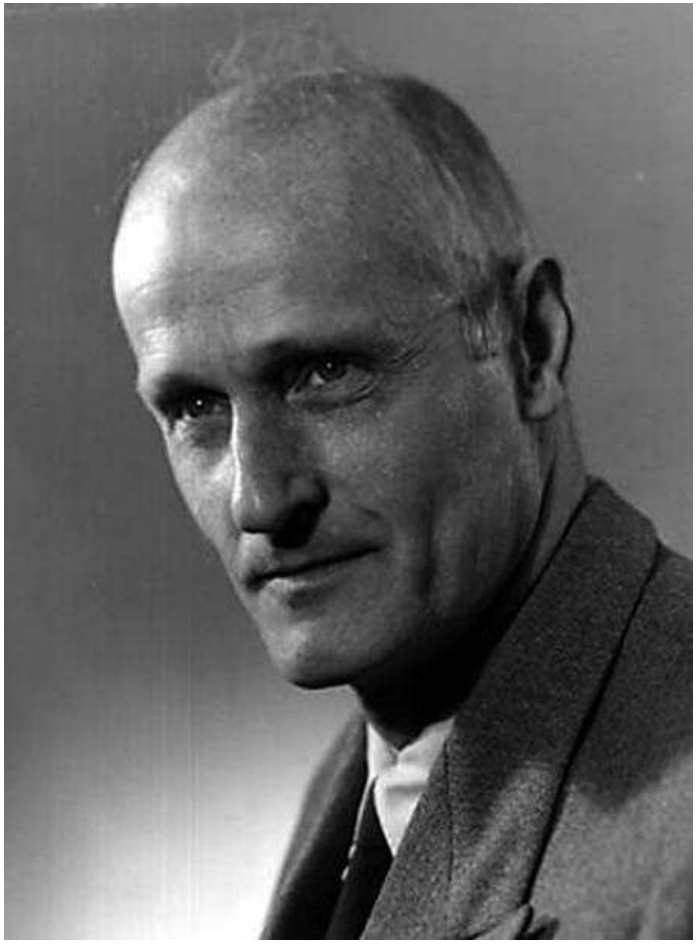


W. McCulloch

W. Pitts



from M. Minsky: *Computation – Finite and Infinite Machines*
1967



S.C. Kleene

REPRESENTATION OF EVENTS IN NERVE NETS AND FINITE AUTOMATA[1]

S. C. Kleene

INTRODUCTION

1. Stimuli and Response

An organism or an automaton receives stimuli via its sensory receptor organs, and performs actions via its effector organs. To say that certain actions are a response to certain stimuli means, in the simplest case, that the actions are performed when and only when those stimuli occur.

In the general case both the stimuli and the actions may be very complicated.

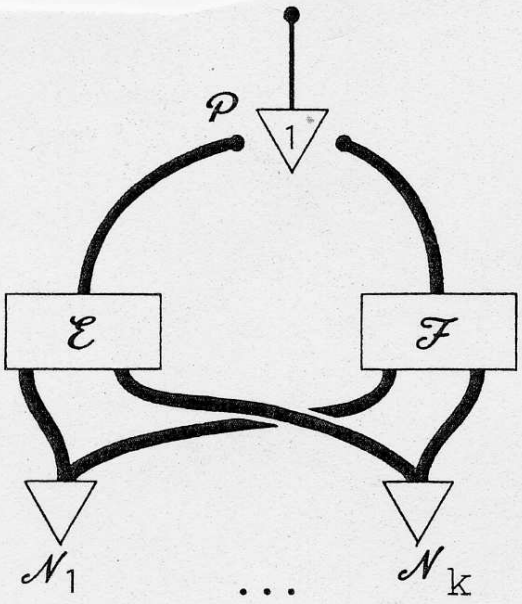


FIGURE 22 E V F

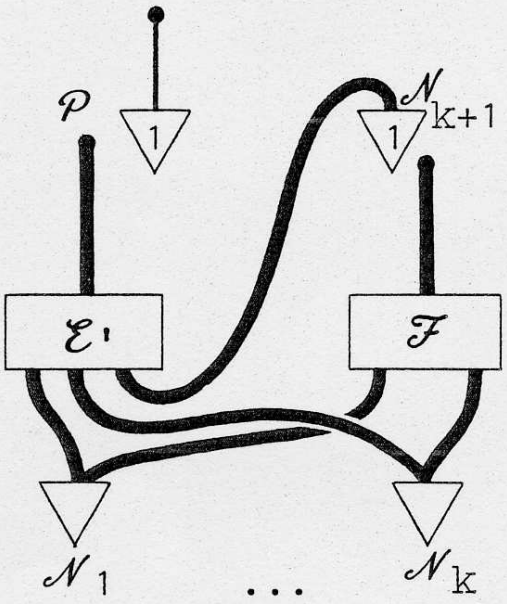


FIGURE 23 EF

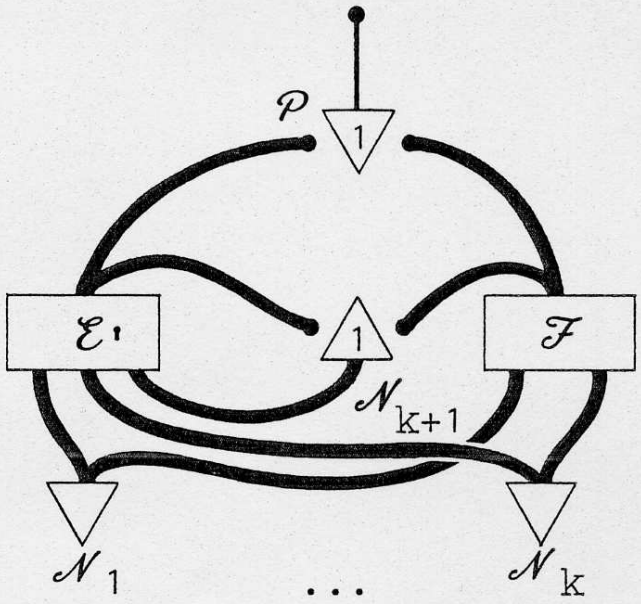
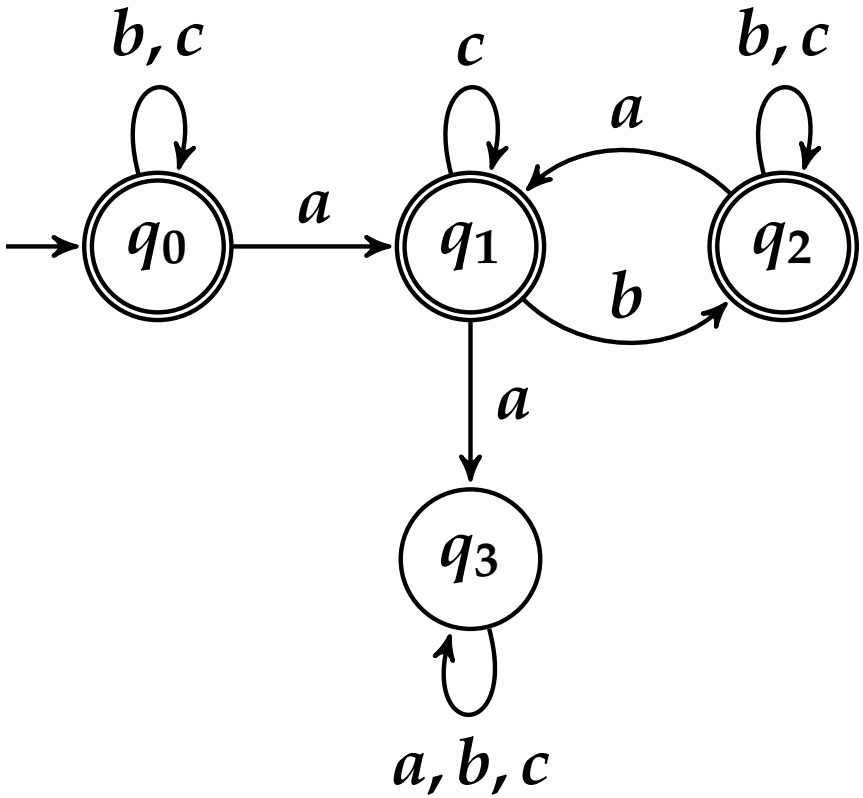


FIGURE 24 E*F

Abstract Automata

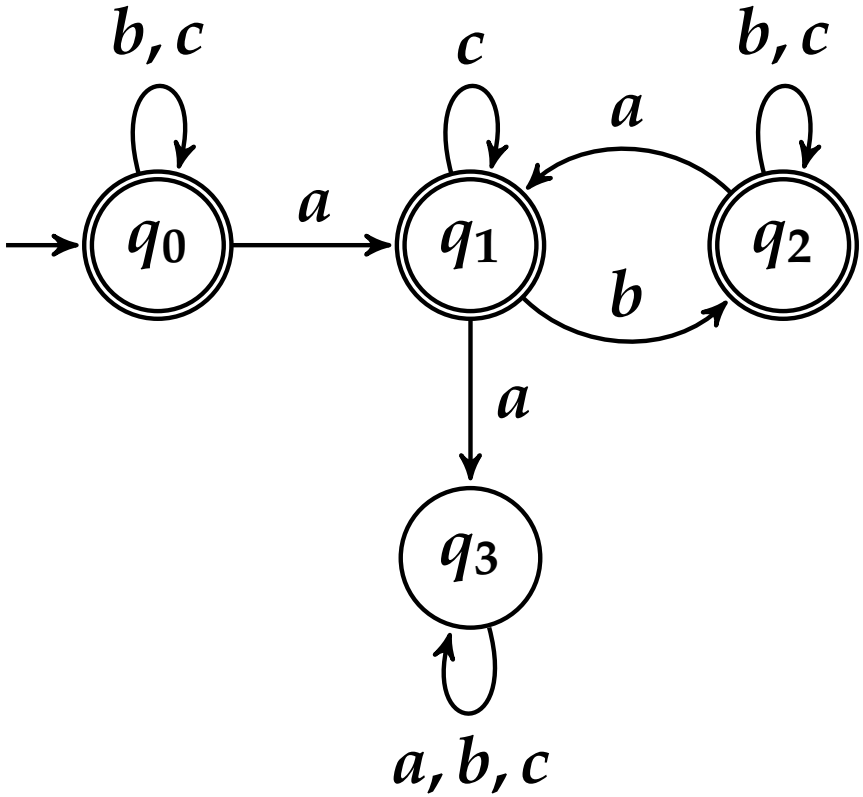


M.O. Rabin, D.S. Scott, Finite Automata and Their Decision Problems, IBM J. of Res. and Dev. 1959

M.O. Rabin, D.S. Scott, Mrs. Rabin

(Wroclaw 2007)

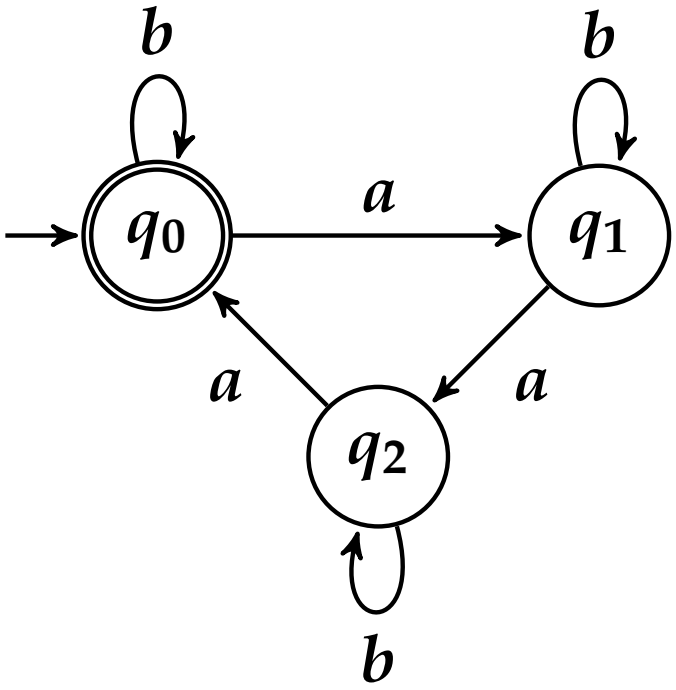
Honorary Doctorate for M.O. Rabin, Wroclaw 2007



“between any two letters a there is somewhere a b”

$$\forall x \forall y (x < y \wedge P_a(x) \wedge P_a(y) \rightarrow \exists z (x < z < y \wedge P_b(z)))$$

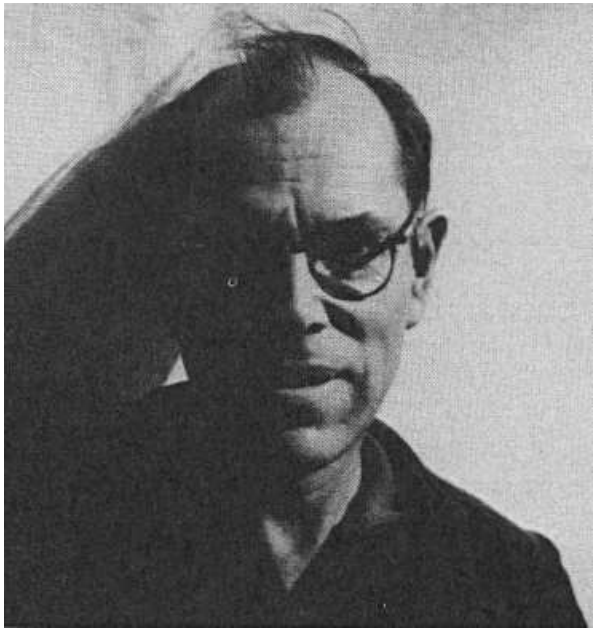
First-order formula



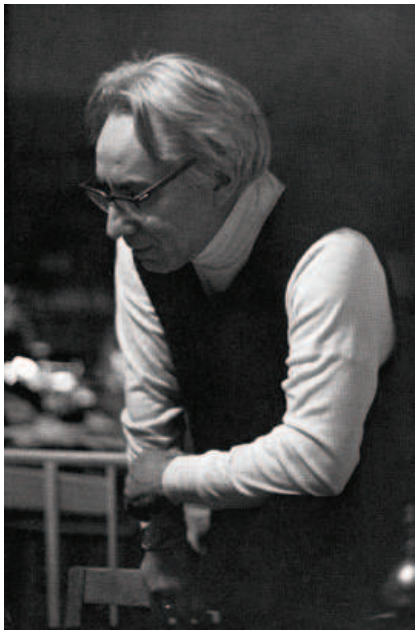
b	a	a	b	a	a	a	a	b	a	a	b	b	a	b
1	2	3	4	**5**	6	7	**8**	9	10	11	12	13	**15**	16

$\neg\exists x\, P_a(x)$ $\vee$ **"$\exists$ set X of positions, containing each third position with a and also the last position with a)"**

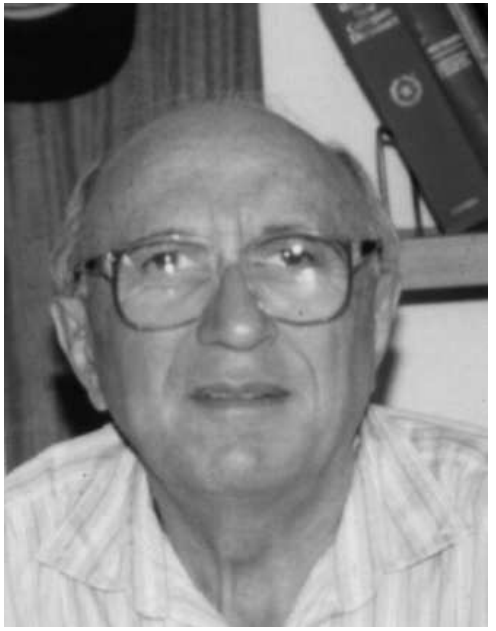
Formula of monadic second-order logic



J.R. Büchi



C.C. Elgot



B.A. Trakhtenbrot

Theorem of Büchi-Elgot-Trakhtenbrot (1960):

Finite automata and monadic second-order fomulas can express the same word properties.

Automata versus Logic

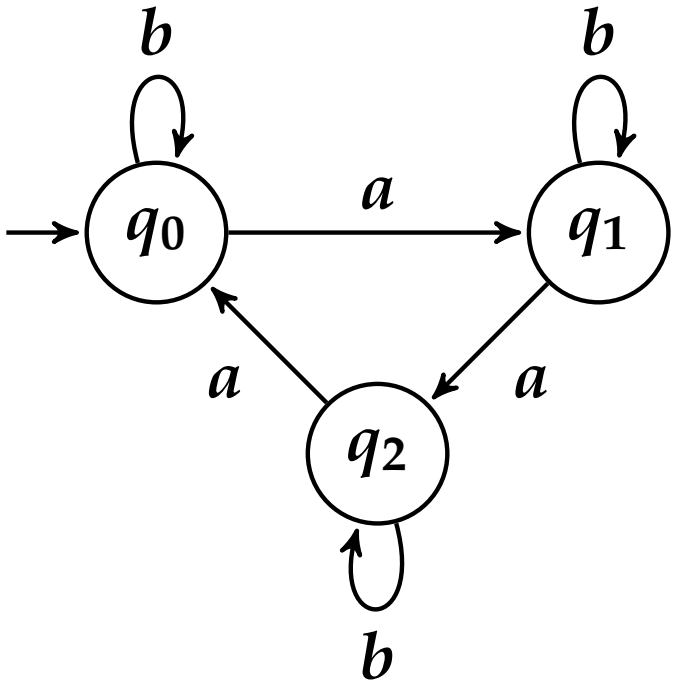
Automata are “state-based implementations”:

- **presentable in graphical form**
- **in principle easy to analyze (using graph algorithms)**
- **but unstructured, not modular, not decomposable**

Formulas of logic are “specifications”:

- **textual objects**
- **structured, modular, compositional**
- **but hard to analyze**

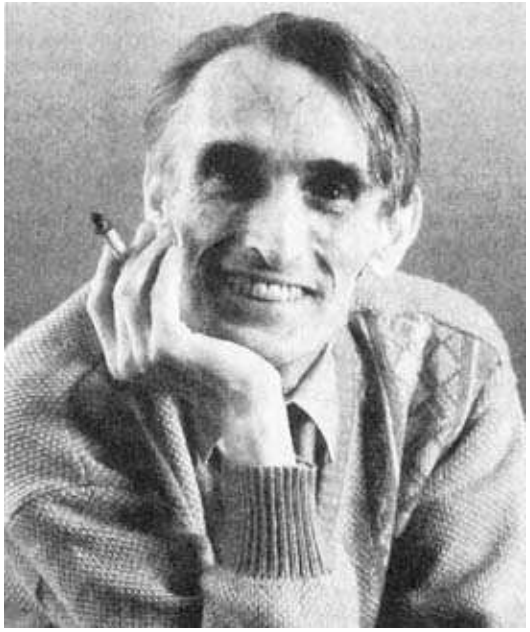
The effective equivalence between automata and logical formulas is the basis of a new calculus for understanding and designing systems – this is for computer science what standard calculus (differential equations) is for physics and classical engineering.

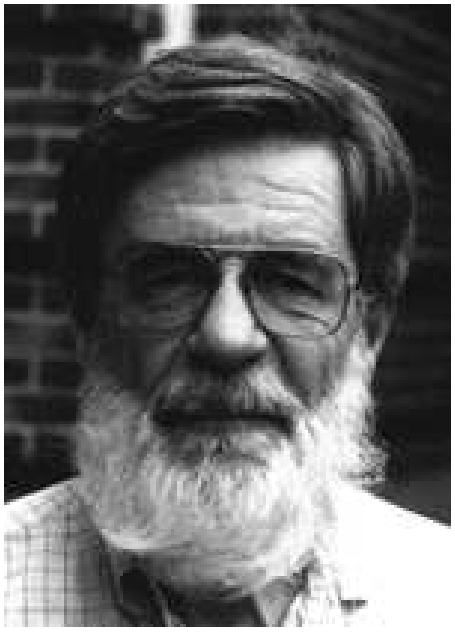


Letter b induces induces the identity, words a, aa two shifts.

Each word induces one of these functions. These functions form a monoid.

Syntactic Monoid of a regular language
= monoid of state transformations of corresponding minimal automaton





M.P. Schützenberger **R. McNaughton**

Theorem of Schützenberger / McNaughton (1965/1972)

A regular language is definable in first-order logic iff its syntactic monoids is “group-free” (does not contain a nontrivial group).

MOD-Quantifiers

First-order formulas with MOD-quantifiers:

$$\exists^{\equiv 0(3)} x \;\; P_a(x)$$

Can one define all regular languages already with this logic?

H. Straubing, D. Thérien, W. Ths. (1988):

A regular language is definable in first-order logic with MOD-quantifiers iff its syntactic monoid only contains solvable groups.

The non-solvable group A_5 shows that first-order formulas with MOD-quantifiers cannot define all regular languages.

Golden 1960's

saw the introduction of automata over

- **finite trees rather than finite words**
- **infinite words rather than finite words**
- **infinite trees**

Verification

Scenario

Application domain: Reactive non-terminating systems

Simplest setting:

System consists of Environment E und Control C

Finite state space

E und C choose actions a_i resp. b_i in alternation.

System run: $a_0 b_0\ a_1 b_1\ a_2 b_2 \ldots$

Control C works with program P:

It responds to environment actions a_i by own actions b_i.

Correctness

Specification: Requirement on system runs

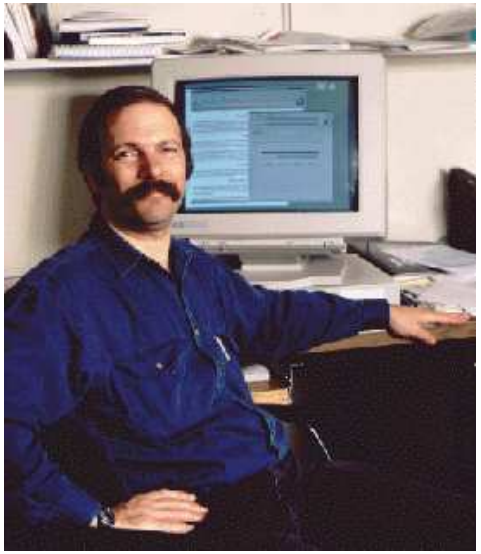
Example: "After each event c the event d will continue to hold until eventually e happens."

Temporal logic LTL (A. Pnueli): $\varphi = \mathrm{G}(c \rightarrow \mathrm{X}(d\mathrm{U}e))$

Model-Checking-Problem:

Given system S with finite state space
and an LTL-specification φ
(more generally a "regular specification"),

does each system run of S satisfy φ?

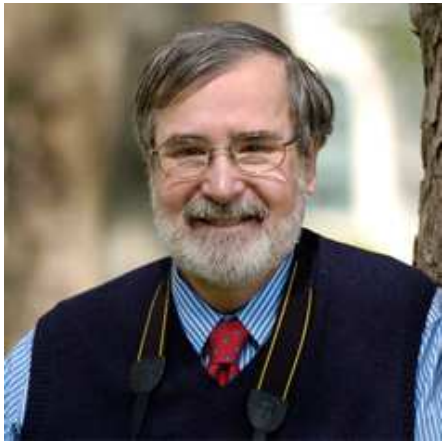


A. Pnueli **M. Vardi** **P. Wolper**

Vardi, Wolper 1994:

The Model-Checking-Problem for LTL-formulas φ is solvable in exponential time in $|\varphi|$.

Promoters of Computation Tree Logic



E. Clarke

A. Emerson

J. Sifakis

Lectures “Monadic Theories”

We explore the range and limitations of the logic-automata connection, over infinite structures:

1. **The basic results**
2. **Prefix rewriting and the pushdown hierarchy**
3. **Composition method**
4. **Undecidability results**

Synthesis

Alonzo Church

A Pre-LaTeX-Time Paper

APPLICATION OF RECURSIVE ARITHMETIC TO THE PROBLEM OF CIRCUIT SYNTHESIS

Alonzo Church

RESTRICTED RECURSIVE ARITHMETIC

Primitive symbols are individual (i.e., numerical) variables x, y, z, t,···, singulary functional constants i_1, i_2,···,i_μ, the individual constant 0, the accent ' as a notation for <u>successor</u> (of a number), the notation ( ) for application of a singulary function to its argument, connectives of the propositional calculus, and brackets [ ].

Axioms are all tautologous wffs. Rules are modus ponens; substitution for individual variables; mathematical induction,

from $P \supset S^a_a, P|$ and $S^a_0 P|$ to infer P;

and any one of several alternative recursion schemata or sets of recursion schemata.

Church's Problem (1957)

"Given a requirement which a circuit is to satisfy, we may suppose the requirement expressed in some suitable logistic system which is an extension of restricted recursive arithmetic. The *synthesis problem* is then to find recursion equivalences representing a circuit that satisfies the given requirement (or alternatively, to determine that there is no such circuit)."

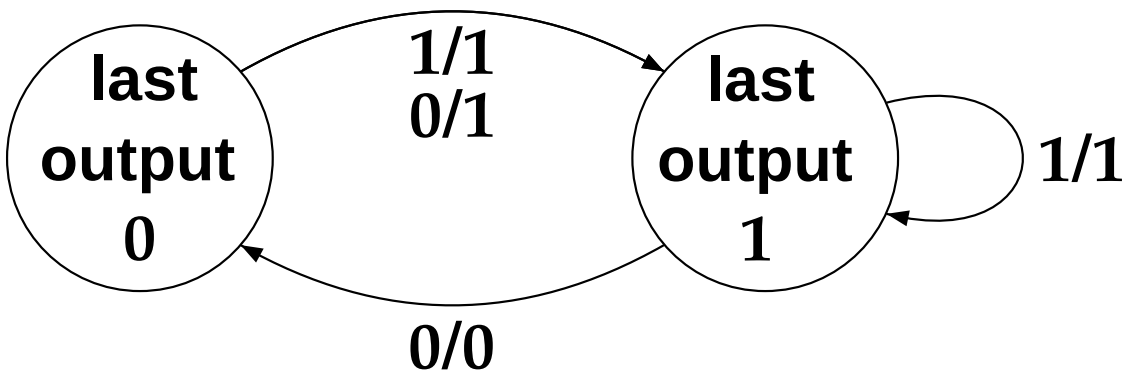
(By "circuits", Church means finite automata with output.)

Example

Specification for the production of an output bitstream given an input bitstream.

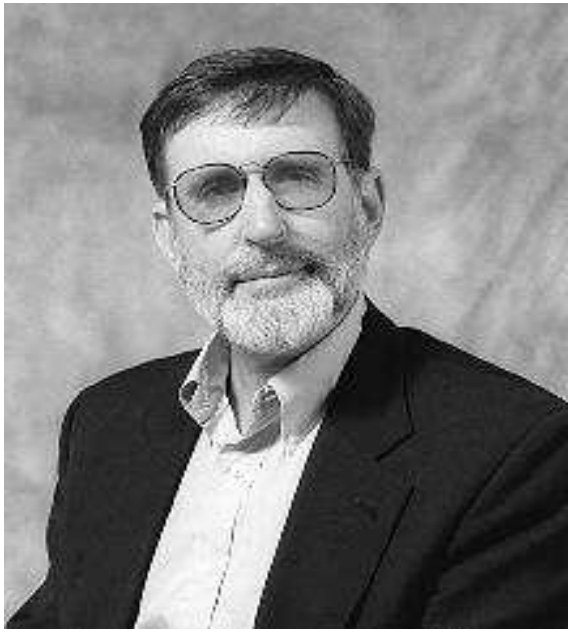
- **An input 1 requires the corresponding output bit to be 1.**
- **Never there are two successive output bits 0.**
- **If again and again the input bit is 0, then the output bit should also be 0 again and again.**

Solution (transition system in format of Mealy automaton)



Büchi-Landweber-Theorem (1969)

J.R. Büchi



L.H. Landweber

Given a specification in monadic second-order logic, one can decide whether this specification is realizable, and in this case one can construct a finite state-machine with output, meeting the specification.

Second Set of Lectures

- Infinite two-player games: The fundamental constructions
- Definability of winning strategies
- Generalization of “strategy”
- Analysis of infinite games using finite plays

Conclusion

Summary and Some Perspectives

- **Automata theory enables us to solve problems of logic in an algorithmic way.**
- **Ingredients: Graphs (transition systems),**
  words rather than numbers,
  semantical equivalences rather than isomorphism
- **It is an essential part of a new kind of applied mathematics, which one can call “system calculus” in computer science.**

Three problem areas:

- **Understand better the logic-automata connection.**
- **Understand better the complexity issues.**
- **Join the discrete methods with continuous ones**

“Only in small minds is there a contrast beween engineering, science, and mathematics. It is a historical fact that the same personality has often created the fundamental theoretical and practical ideas of a given subject matter — for example, Archimedes, Galilei, Newton, Euler, Gauss.”

J.R. Büchi, *Finite Automata, Their Algebras and Grammars*, Springer, New York 1989, S. XVI.