Friedrich L. Bauer

Origins and Foundations of Computing

In Cooperation with
Heinz Nixdorf MuseumsForum

Springer

Origins and Foundations of Computing

Friedrich L. Bauer

Origins and Foundations of Computing

In Cooperation with
Heinz Nixdorf MuseumsForum

With Editorial Assistance from Norbert Ryska

Springer

Prof. Dr. Friedrich L. Bauer
Professor Emeritus für Mathematik und Informatik
Technische Universität München
Boltzmannstr. 3
85748 Garching, Germany

Translation from the German language edition: "Kurze Geschichte der Informatik" authored by Friedrich L. Bauer. Copyright © Wilhelm Fink Verlag, Paderborn, Germany 2007

ISBN 978-3-642-02991-2 e-ISBN 978-3-642-02992-9
DOI 10.1007/978-3-642-02992-9
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2009940334

ACM Computing Classification (1998): K.2, A.0

© Springer-Verlag Berlin Heidelberg 2010
This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.
The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Cover design: KünkelLopka GmbH, Heidelberg

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

The Heinz Nixdorf Museum Forum (HNF) is the world's largest computer museum and is dedicated to portraying the past, present and future of information technology.

In the "Year of Informatics 2006" the HNF was particularly keen to examine the history of this still quite young discipline. The short-lived nature of information technologies means that individuals, inventions, devices, institutes and companies "age" more rapidly than in many other specialties. And in the nature of things the group of computer pioneers from the early days is growing smaller all the time.

To supplement a planned new exhibit on "Software and Informatics" at the HNF, the idea arose of recording the history of informatics in an accompanying publication.

My search for suitable sources and authors very quickly came up with the right answer, the very first name in Germany: Friedrich L. Bauer, Professor Emeritus of Mathematics at the TU in Munich, one of the fathers of informatics in Germany and for decades the indefatigable author of the "Historical Notes" column of the journal Informatik Spektrum.

Friedrich L. Bauer was already the author of two works on the history of informatics, published in different decades and in different books. Both of them are notable for their knowledgeable, extremely comprehensive and yet compact style. My obvious course was to motivate this author to amalgamate, supplement and illustrate his previous work.

Only one thing occasionally hindered the otherwise fruitful, almost daily exchange of information between Friedrich L. Bauer and myself – the incompatibility of our two computer systems: a situation that remains irritating even 60 years after the invention of the computer and which will never be entirely resolved.

I heartily thank the publishing houses Springer and Vieweg for permission to use their published material.

For a number of years now the HNF has been editing its publications jointly with the publishing house Schöningh in Paderborn, and my inquiry as to whether it was interested in this publication very promptly received a positive response, for which I sincerely thank Dr. Raimar Zons.

Many thanks to Alfred Wegener for procuring and archiving numerous text and image files. Dr. Jochen Viehoff dealt with the optimization

of numerous images for printing. And many thanks to Marcel Jaspaert for his substantial contribution to laying out the manuscript, for which task he – unwillingly but with his usual alacrity and competence – had to acquaint himself with Donald Knuth's computer typesetting program TEX, to which Professor Bauer remains faithful.

LS Language Services GmbH translated the text with great technical expertise and sensitivity.

I very much hope that this little book – "pauca, sed matura", as Carl Friedrich Gauss would have said – will in future be found on many computer scientists' bookshelves, and particularly on those of computer science and informatics students. No one can claim to have fully mastered his or her own technical discipline without knowledge of its history. Even the originators of the HNF had to begin their design process by directing their attention "back to the roots" in order to attain a vantage point from which they could observe the inception of later developments and evaluate their significance.

A scientific discipline such as informatics possesses numerous roots and is well grounded. But it is only natural to have doubts, to wonder whether, perhaps, one is not merely a sideshoot from some more significant rootstock. This feeling of uncertainty has accompanied informatics since it was established as an academic subject towards the end of the 1960s. Perhaps this "Brief History of Informatics" can also serve to give it more confidence and security.

Like the "Year of Informatics 2006", we hope this book will arouse both curiosity about the field of information technology and interest in current digital developments in our society.

Paderborn
August 2009

Norbert Ryska
Director
Heinz Nixdorf MuseumsForum

Contents

Preamble **1**
The Roots of Informatics 3
Informatics and Mathematics 3

Numerical Calculations **7**
Arithmetic Computation and Its Mechanization 7
Computation in the Binary Number System 14
Floating-Point Arithmetic 18

Calculations Using Symbols **23**
Cryptology 23
Logical Calculation 27
Codes Used in Data Transmission Technology 30
The Principle of Binary Coding 35
Mathematical Roots of the Burgeoning Science of Informatics 36

After 1890: In Thrall to Mechanical and Electromechanical Devices **41**
Mechanization and Automation of Digital Calculations 41
Mechanical and Electrical Cryptological Instruments 46
Message Coding Technology and Information Theory 53
Instruments and Machines for Propositional Calculus, Signaling and Safety Technology 55
Digression into Analog Computers 57
Musical Boxes and Other Precursors of Automata 62
Automata, the Principle of Automatic Machines 63
Process Control as a Precursor of Program Control 64
Universal Program Control 67
Feedback and Regulation 70

After 1935: Formal Languages and Algorithms, 'Universal Machines' and Electronic Solutions **73**
Basic Data Structures 76
Binary Arithmetic 77
Floating-Point Arithmetic 80

Unlimited Integer Arithmetic 81
The Binary Principle 83
Recursive Data Structures and Syntax Diagrams 84
Basic Operative Structures 86
Algorithms 86
Algorithmic Languages 88
Recursion 90
The von Neumann Machine 91
Sequential Formula Translation and Pushdown Mechanisms 93
High-Level Programming Languages and Algorithms . . 95
Machine-Oriented Programming 98
The End of Analog Computers 100
Systems 100
Recognition Systems 101
Cryptanalysis 104
Effects of Cryptanalysis 106
Networked Systems 107
Digression into Electronic Circuits, Devices and Machines . . 111

After 1960: Informatics Begins to Take Shape 113
Applications 114
Constructive Logic – 'Proofs as Programs', 'CIP' 115
Programming Languages 116
Software Engineering 118
The Transition to Minicomputers and Highest-Performance Computers 119

Conclusion: Informatics and Microelectronics Are Mutually Dependent 123

Index 129

Index of Figures 139

Selected Readings 141

Preamble

> Si daretur vel lingua quaedam exacta (qualem quidam Adamicam vocant) vel saltem genus scripturae vere philosophicae, qua notiones revocarentur ad Alphabetum quoddam cogitatio num humanarum, omnia, quae ex datis ratione assequi, inveniri possent quodam genere calculi, perinde ac resolvuntur problemata arithmetica aut geometrica.
>
> LEIBNIZ, De scientia universali seu calculo philosophico[1]

Informatics is a young science with a very young name. In Germany, the word *Informatik* was first used in this context in 1968 by GERHARD STOLTENBERG (1928–2001), at that time a federal government minister, at the opening of a conference in Berlin[2], not long after *informatique* had been coined[3] in France on 19th January 1968 and subsequently used[4] by the Académie Française. Since then corresponding terms have become common elsewhere in Europe: Dutch *informatika*, Italian *informatica*, Spanish *informatica*, Polish *informatyka*, and Russian информатика. Meanwhile, the English-speaking world generally prefers to avoid it altogether in favor of *computer science*[5], which has a more strongly theoretical connotation. For the purpose of this book we will use the word 'informatics' in its 'European' sense.

Modern informatics is the result of a stormy development over the last 40 years, but many of its roots extend much further back into history. We can say that informatics began when the first attempt was made to mechanize what we call 'intellectual activities'; this was undoubtedly not the work of a single individual. If we really wished to choose a single name, that name would have to be LEIBNIZ (1646–1716), who

[1] Quoted from *Gottfried Wilhelm Leibniz*, by ERICH HOCHSTETTER. In: *Herrn von Leibniz Rechnung mit Null und Eins*, printed privately in 1966 by Siemens AG.

[2] Joint conference of the Technical University of Berlin and the Massachusetts Institute of Technology, opened on 29th July 1968.

[3] *L'informatique: Science du traitement rationnel, notamment par machines automatiques, de l'information considérée comme le support des connaissances humaines et des communications, dans les domaines techniques, économiques et socials* (Académie Française, 1966).

[4] For example by the *Délegation à l' informatique*.

[5] As for instance in the ACM Curriculum 1968.

F.L. Bauer, *Origins and Foundations of Computing*,
DOI 10.1007/978-3-642-02992-9_1,
© Springer-Verlag Berlin Heidelberg 2010



Fig. 1: Ramon Llull

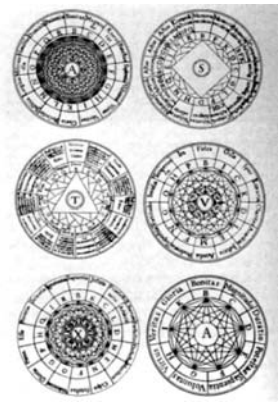


Fig. 2: Logic diagrams



Fig. 3: Athanasius Kircher

qualifies as one of the founders of informatics in a number of ways: his four-species adding machine, his dual system, and, following RAMON LLULL (1235–1315) and ATHANASIUS KIRCHER (1602–1680), the 'concept notation'. In particular, it was Leibniz who introduced the concept of "freeing humanity from the wearisome burden of monotonous mental activity".

The Roots of Informatics

> " ... eine allgemeine Methode, in der alle Wahrheiten der Vernunft auf eine Art Berechnung zurückgeführt sind[6]".
>
> LEIBNIZ, *De arte combinatoria, 1666*

> *"If I were to choose a patron saint for cybernetics out of the history of science, I should have to choose Leibniz."*
>
> NORBERT WIENER, *Cybernetics or Control and Communication in the Animal and the Machine*

Historically, the 'mechanization of allegedly mental activities' is first encountered in a number of extremely special cases: for numerical calculations – using the Arabic decimal digits that have been widely accepted throughout Europe since the beginning of the 16th century – and a number of other algorithmic processes that are nowadays also termed 'calculation', for instance the manipulation of terms formulated using symbols, and simple or compound logical propositions. The essence of informatics is characterised by the comprehensive elaboration of this program, parts of which were even outlined by LEIBNIZ. Questions of encoding belong here, particularly binary code, with cryptology as an interesting fringe area. The elaboration includes complete process automation, which Leibniz did not yet have, which manifests itself as algorithmic thinking and culminates in questions of the syntax and semantics of algorithmic languages.

As an applied science, the historical development of informatics is highly dependent on its technical and engineering realization poten tial, and runs parallel to the development of signaling, (mechanical, electrical and electronic) control engineering, and data storage tech nology – including devices for reading and writing.

Informatics and Mathematics

Informatics is somewhat out of place in the illustrious canon of mathematical disciplines, for two reasons. Firstly, it is extremely young.

[6] ... a generalized method in which all the truths offered by reason are deduced to a kind of calculation.

In 1890 it was unheard-of, and not only by name. The greater part of its development falls into the 20th century, between 1940 and 1990. But this is a good opportunity to give a more detailed account of its relatively unknown early history, which began centuries ago and consolidated itself in the second half of the 19th century. If not entirely, this development is to a considerable extent associated with mathematicians.

And this brings us to the second peculiarity: at its best informatics is a mathematical science, if we may be permitted to use the term. It belongs to mathematics just as much or as little as does theoretical physics or geodesics. Where it differs from mathematics is that mathematics is a purely intellectual science, and this really also ap plies to what is called applied mathematics. Informatics, by contrast, is an engineering science, and this should be taken to include even theoretical informatics.

It is characteristic of mathematicians that they are able to create a purely intellectual, abstract structure. For them, true reality lies only in the mind: images and (plaster) models are permitted 'only' as illustrations, as didactic crutches: one is immediately warned not to let them get the upper hand; they really ought to be unnecessary and are therefore considered not quite respectable.

Typical for computer scientists is the ability to come up with unexpected ideas for the resolution of difficult problems. They are creatively target-oriented, they possess ingenuity, astuteness and inventiveness; their objective is to see a piece of machinery doing useful work; one is immediately warned against purely intellectual speculation such as transfinite induction; bald existence theorems are useless, because they are unproductive, and are therefore frowned upon among computer scientists.

To exaggerate somewhat, and regardless of the derisive laughter that is to be expected from the Boeotians, we can say: 'Mathematics serves to edify man with the fruits of his intellect'. CARL GUSTAV JACOB JACOBI (1804–1851) went so far as to say: "*Die Mathematik dient einzig und vor allem der Ehre des menschlichen Geistes*"[7].

By contrast, the purpose of informatics is to relieve humanity from the burden of monotonous mental activity.

Just because something possesses a number of objectives it does not follow that they are irreconcilable. Naturally there are (more or less) pure mathematicians in this sense, and (more or less) pure computer scientists in that sense, but there is also a *Homo faber* in almost every

[7] The principal and only purpose of mathematics is to honor the human spirit.

mathematician, just as there is a *Homo cogitans* in almost every computer scientist.

Mathematics is the only one of the humanities that can be described as an 'exact' science; it therefore undoubtedly comes closest to that branch of engineering science which deals with intangible, non-physical *ingenium*.

Mathematics and computer science simply are both abstract, are both intangible. And this binds them together, making informatics the sister of mathematics, if not its daughter.

The following discussion will principally consider the development of the core areas of informatics:

- problem-oriented practical informatics ("algorithmic programming"),
- machine-oriented practical informatics ("systems programming") without neglecting the branches,
- theoretical informatics, and
- technical informatics.

This dissertation does not include the field of numerical mathematics, which, in the USA, is considered a branch of computer science. Noted figures that could be mentioned in this context include VERA NIKOLAEVNA FADDEEVA (1906–1981), ALSTON SCOTT HOUSEHOLDER (1904–1993) and WALLACE GIVENS (1910–1993), JAMES HARDY WILKINSON (1918–1986) and GEORGE FORSYTHE (1917–1972), PE-



Fig. 4: Eduard Stiefel



Fig. 5: Olga Taussky-Todd



Fig. 6: From left to right: James Hardy Wilkinson, Wallace Givens, George Forsythe, Alston Scott Householder, Peter Henrici, Friedrich L. Bauer (Gatlinburg Committee)

TER HENRICI (1923–1987), and both EDUARD STIEFEL (1909–1978) and OLGA TAUSSKY-TODD (1906–1995).

Before LEIBNIZ, and for a number of decades afterwards, there were only a few rivulets contributing grains of sand and occasionally gravel to the as yet unstructured informatics. These were quite specific, specialized and still completely isolated tasks aimed at relieving mankind from the burden of monotonous mental activity:

- the mechanization, automation and algorithmization of numeri cal calculations,
- the mechanization and automation of symbolic operations, with the origins of cryptology, logic, security and communications technology,
- process control and the regulation of automatic systems.

Numerical Calculations

> "Die ganzen Zahlen hat der liebe Gott geschaffen, alles andere ist Menschenwerk[8]."
>
> LEOPOLD KRONECKER

Counting is one of humanity's achievements, for *Homo sapiens* a relatively late one. Independent cultures developed it independently, as we know from the existence of different number systems. Man very soon began to use counting and calculating instruments to facilitate working with numbers.

Arithmetic Computation and Its Mechanization

> *"La machine arithmetique fait des effets qui approchent plus de la pensée que tout ce que font des animaux."*
>
> BLAISE PASCAL

The language root that we find in 'calculate' comes from the counting pebbles that were used by the ancient Romans, in Latin *calculi*. The Indo-Germanic root **reg* that gave us the word 'reckon' means 'make orderly', originally as applied to a collection of such pebbles. Calculating instruments soon emerged. The Roman abacus, the Chinese *suanpan* or *chu pan* and Russian schoty that are still in use to this day, the Turkish *coulba*, and the Armenian *choreb* were all used to 'arrange' counters on a counting board, French *comptoir*. But neither this 'reckoning on the lines' nor the use of notches on a tally (late Latin *computare*) marks the beginning of the mechanization of mental activities.

It was the symbols for finger numbers, the digits, which originated in India and were brought to Europe by the Arabs, used in combination with a zero to facilitate the place value notation, which first led to a technical solution in which the teeth of a wheel can take the place of the counters. This was the adding machine with digit carry invented by WILHELM SCHICKARD[9] (1592–1635) in 1623, which multiplied according to the *Gelosia*-method using Napier's bones and allows multi-

[8] God created the integers, all else is the work of man.

[9] WILHELM SCHICKARD was Professor of Biblical Languages and later of Astronomy in Tübingen. He was born on 22.4.1592 in Herrenberg and died of the plague on 23.10.1635.

F.L. Bauer, *Origins and Foundations of Computing*,
DOI 10.1007/978-3-642-02992-9_2,
© Springer-Verlag Berlin Heidelberg 2010



Fig. 7: Wilhelm Schickard

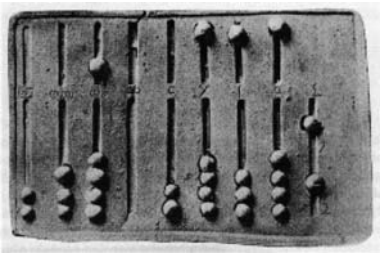


Fig. 8: Roman abacus

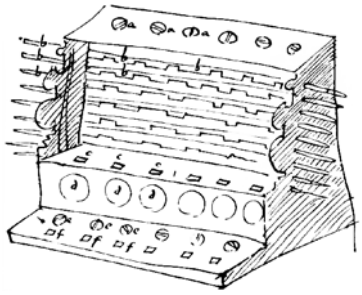


Fig. 9: Drawing of adding machine

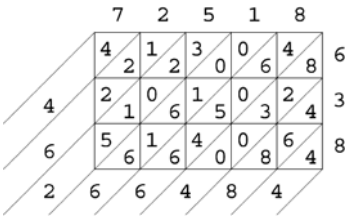


Fig. 10: *Gelosia* method of multiplication

ples of the multiplier to be read out. In s'Hertogenbosch in 1640 the Jesuit father JOHAN CIERMANS (1602–1648) suggested calculating with the help of 'iron wheels'. From 1641, BLAISE PASCAL (1623–1662) also developed an adding machine, which was completed in 1645[10]; its last two positions already possessed a non-decimal carry appropriate to the monetary system of the time[11].

[10] Example in *Conservatoire des Arts et des Métiers, Paris*, described in Diderot's Encyclopédie.

[11] The arrangement invented by RENÉ GRILLET in France in 1678, which had three rows of seven disks, does not possess any carry facility and was used only to record numbers, not for addition. In 1666 the Englishman S. SAMUEL MORLAND (ca. 1625–1685) constructed a machine in which the carry was not propagated automatically but was merely registered and had to be added on by hand.



Fig. 11: Blaise Pascal



Fig. 12: Pascaline



Fig. 13: Gottfried W. Leibniz

In 1671–1674, GOTTFRIED WILHELM LEIBNIZ finally managed to extend the mechanization to produce a genuine four-species adding machine, by introducing a sliding carriage as counter and using a stepped-drum to store the multiplicand setting. Subsequently, ANTONIUS BRAUN (1686–1728) in about 1726 in Vienna, and PHILIPP MATHÄUS HAHN (1739–1790) from 1770 in Echterdingen, built machines with counting wheels arranged concentrically around the stepped-drum, probably the first really useable devices. The same construction was also seized upon by CHRISTEL HAMANN (1870–1948) in 1905 for the machine 'Gauss' and was finally used again in the 'Curta' constructed in 1948 by the Viennese KURT HERZSTARK (1902–1988), which weighed only 230



Fig. 14: Braun's calculating machine

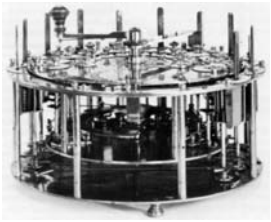


Fig. 15: Calculating machine by Hahn

grams and had the appearance of a prayer-wheel. In 1722 CHRISTIAN LUDWIG GERSTEN (1701–1762), a professor of mathematics in Gießen, built a machine for addition and subtraction. Further improvements were devised in Darmstadt in 1783 by JOHANN HELFRICH MÜLLER (1746–1830), and were incorporated into the machines built by JOHANN CHRISTIAN SCHUSTER (1759–1823). In Paris in 1818, CHARLES XAVIER THOMAS (1785–1870) from Colmar began series production of an 'arithmomètre'[12].

A different technical solution, the pin-wheel, was invented in Padua in 1709 by GIOVANNI POLENI (1683–1761).

The pin-wheel is encountered again in the machine developed by DIDIER ROTH (1807–1885) in 1841, in the patent awarded to DAVID ISAAC WERTHEIMBER in 1843, and eventually in the work of BÜTTNER (1889) and HEINRICH ESSER (1892). The pin-wheel achieved its greatest practical significance after it was used in 1872 in the USA by FRANK STEPHEN BALDWIN (1838–1925) and in St. Petersburg in 1871 by WILLGODT THEOPHIL ODHNER (1845–1905). It makes it possible for subtractions to be performed simply by turning the crank the other way: in Thomas's stepped-drum machine this had called for adjustments to the gears.

In 1890 the current technology as regards mechanised four-species numerical calculations was still characterised by the "arithmomètres", hand-crafted stepped-drum machines based on LEIBNIZ's construction. Roughly 1500 such machines were constructed between 1820 and 1878 at a price of a few hundred French francs (production of these machines continued until 1930, reaching an overall total of about 4000 machines). But the call was for cheaper yet faster machines. After DUBOIS D. PARMELEE was awarded a patent in 1850 for a convenient key-operated

[12] In 1858 a rotation counter was added to this.



Fig. 16: Charles X. Thomas de Colmar



Fig. 17: Philipp M. Hahn



Fig. 18: Thomas Arithmometer

column adding machine such machines did then become available – simple devices for summing a single column of figures[13]. (Neither PASCAL's machine nor the beautifully handcrafted 18th century specimens of HAHN and SCHUSTER had actually managed to be of any practical use.) At this point the development is lost in a fog of technical details and numerous patents and productions, right down to toothed-rack adders made of sheet metal.

The experiments carried out in 1886 by EDUARD SELLING (1834–1920) and in 1887 by LÉON BOLLÉE (1870–1913) attempted to replace LEIBNIZ's method of multiplication by repeated addition by the use of 'multiplication bones' (as suggested by NAPIER)[14] for each position in turn, but they were technically not a success[15].

[13] AS MAURICE D'OCAGNE recorded, V. SCHILT in 1851 and F. ARZBERGER in 1866 constructed the first column adding machines with keys, although the latter possessed only two keys – for 1 and for 3.

[14] JOHN NAPIER (1550–1617), Multiplication Bones, 1617.

[15] In 1850, once VIKTOR MAYER AMÉDÉE MANNHEIM (1831–1906) had



Fig. 19: Frank S. Baldwin

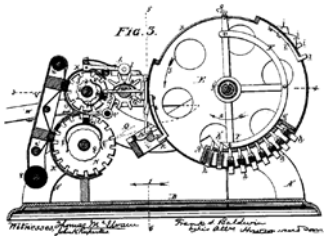


Fig. 20: Baldwin's pin-wheel



Fig. 21: Willgodt T. Odhner

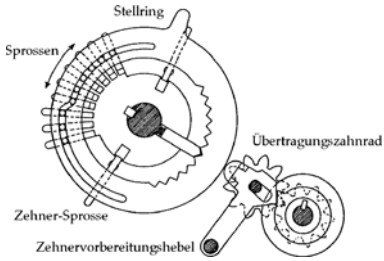


Fig. 22: Pin-wheel by Odhner

The breakthrough did not come until 1892 when the Swiss OTTO STEIGER (1858–1923) was awarded a patent for a mechanical multiplication block cast in metal. Such direct multipliers were then used in the MILLIONAIRE built by HANS W. EGLI (1862–1923).

The multiplication process by repeated addition was automated in 1905 by CHRISTEL HAMANN in the 'Mercedes' on the basis of an experiment carried out by CHEBYSHEV in Paris in 1882, and division followed in 1908 in the 'Madas', which used a principle invented in 1902 by ALEXANDER RECHNITZER (1883–1922): these are the first recorded instances of a loop-like process structure.

finalized their design, logarithmic slide-rules soon became available more cheaply.



Fig. 23: John Napier

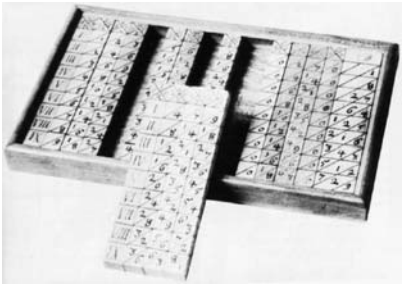


Fig. 24: A set of Napier's bones



Fig. 25: Charles Babbage

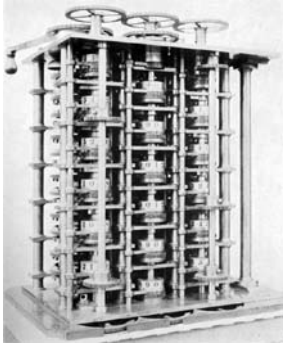


Fig. 26: Babbage's Difference Engine

A print station was first attached to a mechanical desk calculator in 1889 by the company Burroughs, although the automatic creation of print matrices had been planned by CHARLES BABBAGE (1791–1871) in 1823 when he started work on his 'difference engine', which was intended to perform production of tables by means of interpolation. The difference engine was further advanced in Sweden in 1835 by PEHR GEORG SCHEUTZ (1785–1873), and EDVARD RAPHAEL SCHEUTZ (1821–1881) subsequently developed it sufficiently for it to be of practical use, but the machine then stood unused in the Dudley Observatory in Albany, USA, from around 1855 to 1924. In 1851 BABBAGE was also already familiar with the principle of single-step carry-



Fig. 27: Pehr G. Scheutz

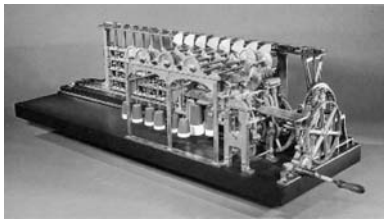


Fig. 28: Difference Engine by Scheutz

ing[16], but nothing further is known about any implementation of this idea.

Computation in the Binary Number System

In the early years of what is considered the 'modern age' number systems other than decimal were considered less peculiar than they are today[17]. For instance the last two positions of PASCAL's machine[18] had wheels with 20 and 12 teeth corresponding to the mixed radix of the currency in use at the time, in which the livre was subdivided into 20 sous each equivalent to 12 deniers. The use of various number systems, from base two to base twelve, was investigated in 1670 by Bishop JUAN CARAMUEL Y LOBKOWITZ (1606–1682) in his *Mathesis biceps, vetus et nova*, Campania 1670. GEORGE LOUIS LECLERC DE BUFFON (1707–1788) was also a proponent of a purely duodecimal system. LEIBNIZ, however, uncompromisingly changed over to using base two. The first of his manuscripts that deals with this was written on 15th March 1679, but he had been considering it since his time in Paris.

[16] "This new and rapid system of carrying the tens when two numbers are added together reduced the actual time of the addition of any number of digits, however large, to nine units of time for the addition, and one unit for the carriage. Thus in ten's units of time, any two numbers, however large, might be added together." CHARLES BABBAGE, *Passages from the Life of a Philosopher*, p. 116ff.

[17] This doesn't apply to the British, who waited until 1971 to decimalize their monetary system.

[18] Pascal was the son of a tax collector and developed the machine for fiscal calculations.

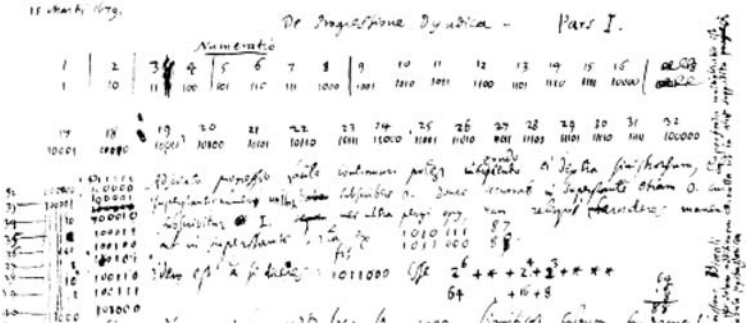


Fig. 29: Beginning of Leibniz' handwriting of 15th March 1679

"The dyadic number system, part I. The sequence shown here can easily be continued, by working from right to left, writing a 0 underneath each 1 in the number in the row above until you come to a 0 and then writing a 1 underneath that; there is no need to go any further because all the following digits remain the same as in the number in the row above. …" (translated from the Latin by P. Franz Xaver Wernz SJ, Munich).

A fact that more people should be aware of is that in this work, *De Progressione Dyadica*, LEIBNIZ also sketches the design of a machine operating on the binary principle:

> "This type of calculation could also be carried out using a machine. The following method would certainly be very easy and without effort: a container should be provided with holes in such a way that they can be opened and closed. They are to be open at those positions that correspond to a 1 and closed at those positions that correspond to a 0. The open gates permit small cubes or marbles to fall through into a channel; the closed gates permit nothing to fall through. They are moved and displaced from column to column as called for by the multiplication. The channels should represent the columns, and no ball should be able to get from one channel to another except when the machine is put into motion. Then all the marbles run into the next channel, and whenever one falls into an open hole it is removed. Because it can be arranged that two always come out together, and otherwise they should not come out."

LEIBNIZ harks back to his *binaria arithmetica* in numerous letters, and in a paper written in 1703 "*Explication de l'Arithmétique Binaire*".

This idea then remained buried for 250 years, apart from a remark



Fig. 30: Louis Couffignal

made by the above-mentioned Hesse-Darmstadt engineering pioneer JOHANN HELFRICH MÜLLER in 1786 that his machine was capable of being configured to use LEIBNIZ's *arithmetica dyadica*. MAURICE D'OCAGNE (1862–1938) considers this MÜLLER machine to have been the precursor of BABBAGE's process control.

In a French patent granted priority on 12th September 1931, RAYMOND LOUIS ANDRÉ VALTAT used toothed wheels with $2^3 = 8$ positions. In 1936 he pointed out the advantages of binary arithmetic for the design of (mechanical) arithmetic units, which idea was followed that same year by LOUIS COUFFIGNAL (1902–1966) in France and E. WILLIAM PHILLIPS in Britain.

The latter demonstrated a mechanical model of a multiplier that used the binary system and recommended the compatible octal system for use in tabulators. Even before this, in 1934, KONRAD ZUSE (1910–1995) had decided[19] to use the binary system in his purely mechanical pilot model, finished 1937/38. He then moved, in a natural technical consequence, in 1938 to the use of electromagnetic relays, which are capable of two states (open and closed). JOHN VON NEUMANN (1903–1957) and HERMAN HEINE GOLDSTINE (1913–2004) then took up the binary system again in their design of the 'Princeton Machine', a report of which was widely distributed in 1946–1948. In Britain the development of the pilot ACE (JAMES HARDY WILKINSON) influenced by ALAN MATHISON TURING (1912–1954) was also designed to use binary in-

[19] KONRAD ZUSE, Der Computer, mein Lebenswerk. First ed. 1970, Verlag moderne industrie, p. 48.



Fig. 31: Konrad Zuse

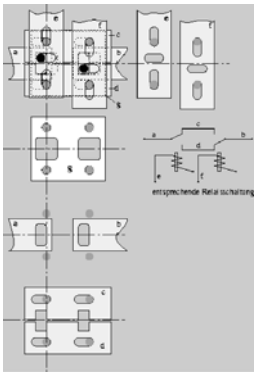


Fig. 32: Zuse's switching element



Fig. 33: John von Neumann

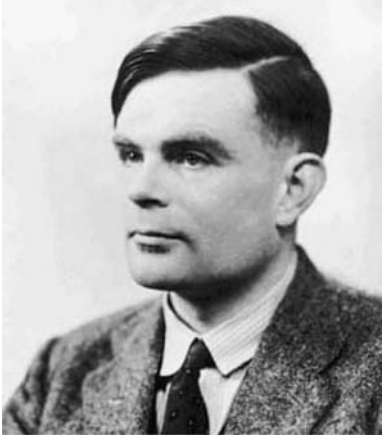


Fig. 34: Alan M. Turing

ternally and octal externally, as proposed by PHILLIPS, and all other British developments were correspondingly binary-oriented: MAURICE VINCENT WILKES (*1913), FREDERIC CALLAND WILLIAMS (1911–1977) and TOM KILBURN (1921–2001). Meanwhile developments that had been started in the USA by HOWARD HATHAWAY AIKEN (1900–1973), GEORGE R. STIBITZ (1903–1995), JOHN PRESPER ECKERT (1919–1995), JOHN WILLIAM MAUCHLY (1907–1980) and WALLACE J. ECKERT (1902–1971) stuck to the decimal system. CHARLES ERYL WYNN-WILLIAMS (1903–1979) had already started using electronic counters that operated on the binary system back in 1931.



Fig. 35: James H. Wilkinson

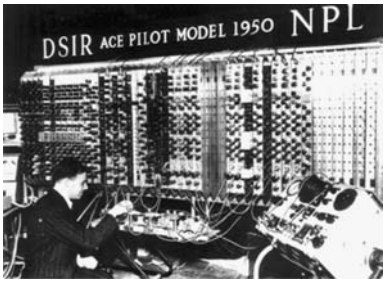


Fig. 36: Wilkinson at Pilot ACE



Fig. 37: Maurice V. Wilkes



Fig. 38: Wilkes at EDSAC computer

Floating-Point Arithmetic

KONRAD ZUSE, however, made a further contribution to the development. He recognized that the magnitude of numbers can be better represented by their logarithms and introduced the 'semi-logarithmic form' that is now known as 'floating-point representation'. His first machine, Z1, possessed seven binary positions for the exponent and sixteen for the mantissa.

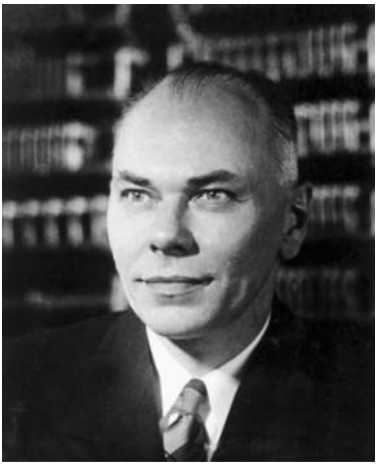


Fig. 39: Howard H. Aiken



Fig. 40: George R. Stibitz



Fig. 41: J. Presper Eckert at UNIVAC



Fig. 42: J. Presper Eckert

However, apart from the relay computer Model V created by STIBITZ in 1947 this idea was also not taken up in the USA or in Britain until much later; wired-in floating-point calculations did not reappear until the Swedish development BARK, a number of Soviet developments, and the PERM in Munich.



Fig. 43: John W. Mauchly



Fig. 44: Wallace J. Eckert

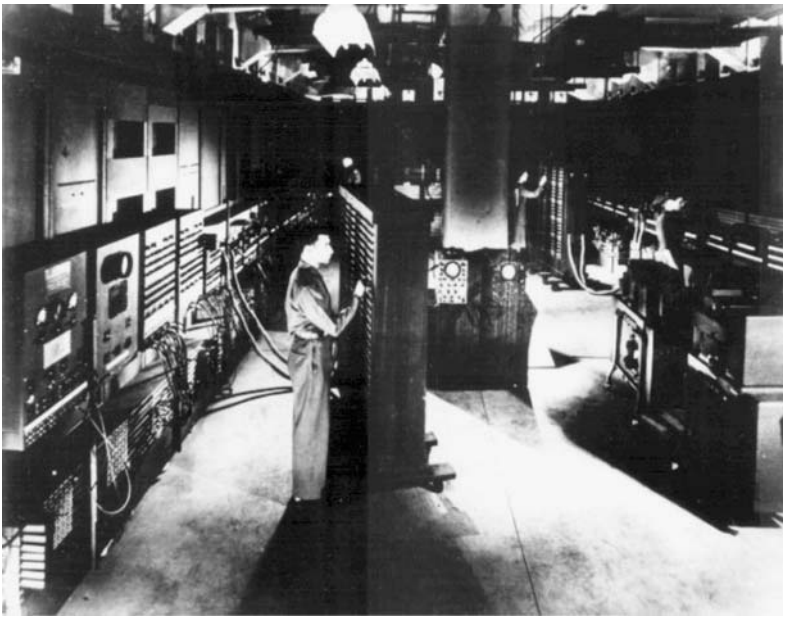


Fig. 45: 1946 ENIAC

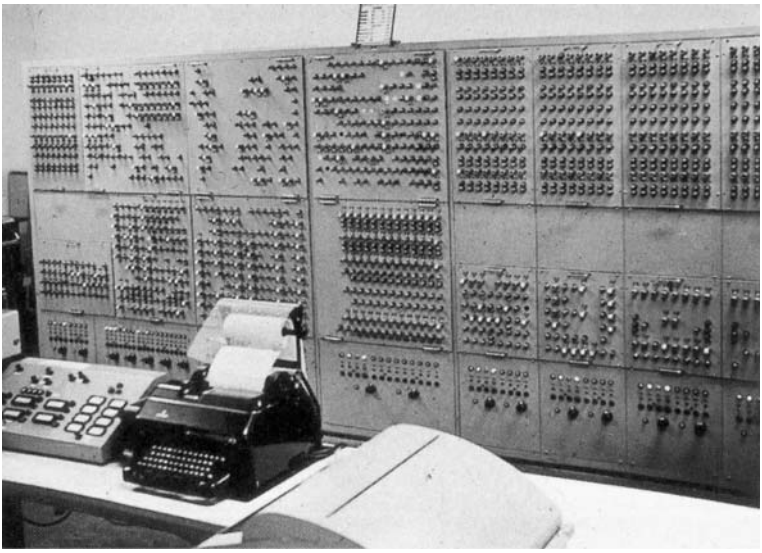


Fig. 46: Munich Technical University computer PERM

Calculations Using Symbols

> "Es wird dann beim Auftreten von Streitfragen zwischen zwei Philosophen nicht mehr Aufwand an wissenschaftlichem Gespräch erforderlich sein als für zwei Rechnerfachleute. Es wird genügen, Schreibzeug zur Hand zu nehmen, sich vor das Rechengerät zu setzen und zueinander (wenn es gefällt, in freundschaftlichem Ton) zu sagen: Laßt uns rechnen.[20]"
>
> LEIBNIZ, around 1680: *De scientia universali seu calculo philosophico*

The words 'arithmetic' and 'calculation' have to do with more than numbers. An old German word for algebra is 'Buchstabenrechnen', which can be translated as 'alphabet arithmetic'. The most significant contribution of Indian mathematics in the early Middle Ages was the introduction of letters to designate arbitrary mathematical objects.

And the idea is not restricted to mathematics. As early as 1663, ATHANASIUS KIRCHER compiled a universal multilingual concept notation that included 1048 concepts. LEIBNIZ spent decades pursuing the idea of representing connotation symbolically so as to make them manageable. In around 1680, in "*De scientia universali seu calculo philosophico*", he expressed the hope that it would be possible for academic disputes to be resolved by 'calculation'. LEIBNIZ made tentative attempts to assign numbers to concepts in such a way that conceptual subsumption was expressed by divisibility. In modern parlance we would say that he introduced a lattice of concepts. The *ars magna* that LEIBNIZ frequently mentions in this context goes back to RAMON LLULL[21].

Cryptology

The earliest manipulations of symbols are to be found in the use of secret writing and secret languages. Not for nothing do we find ATHANASIUS KIRCHER's 'universal concept notation' in his great work on cryptog-

[20] The occurrence of disputes between two philosophers will then not call for any more time spent in scientific discussions than for two expert arithmeticians. It will be sufficient to take your pen in your hand, sit in front of the computer, and say to one another (in a friendly tone of voice, if you like): Let us calculate.
[21] RAMON LLULL (Latinized to Raimundus Lullus), born around 1235, died in 1315.

F.L. Bauer, *Origins and Foundations of Computing*,
DOI 10.1007/978-3-642-02992-9_3,
© Springer-Verlag Berlin Heidelberg 2010

raphy, "*Polygraphia nova et universalis*". In Kircher's time, scientific cryptography had not been in existence for very long. Simple substitutions, which JULIUS CAESAR (100–44 BC) is known to have used, and simple transpositions, achieved by means of the Greek $\sigma\kappa\nu\tau\acute{\alpha}\lambda\eta$ (scytale), a rod around which a strip of parchment is wound, were not methods that could be taken seriously. The father of cryptography is considered to be the great architect LEON BATTISTA ALBERTI (1404–1472), who introduced both polyalphabetic substitutions and superencrypted codes. The book he wrote in 1466 or 1467 is the earliest preserved work on cryptology.

In 1474, CICCO SIMONETTA, secretary to the powerful FRANCESCO SFORZA, taught how a simple substitution encryption can be broken, a method that the Arabian AHMAD AL QALQASHANDI (1355–1418) had also described in around 1400. In 1518 the learned Abbot of Würzburg, JOHANNES TRITHEMIUS (1462–1518), published the first printed book about cryptology. It contained the first mention of a masked secret code and the first polyalphabetic encryption using a progressive fixed key as a periodic series of Caesar substitutions (erroneously named after VIGENÈRE). In 1553 GIOVAN BATISTA BELASO (1505–?) introduced arbitrary, periodically repeating keywords. GIOVANNI BATTISTA PORTA (1535–1615) was already systematically distinguishing between substitution and transposition. He was the first to solve a polyalphabetic encryption.

In 1660, the ARGENTIS at the papal court, particularly MATTEO ARGENTI (1561–1610?), already knew how to use numerous tricks in order



Fig. 47: Leon Battista Alberti

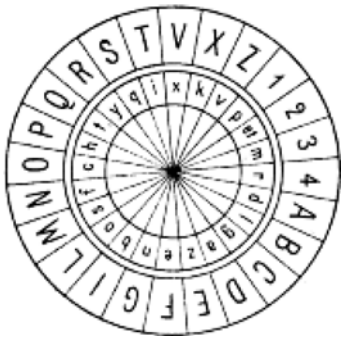


Fig. 48: Alberti's cipher disk

to make decipherers' work more difficult, for instance by using code words of different lengths – single digits or two-digit numbers – and homophones, i.e. different arbitrarily chosen code words for the same plain word or symbol. FRANÇOIS VIÈTE, Seigneur de la Bigotière (1540–1603), who worked for Henri IV of France, is only one of the many mathematicians who dealt with decryption[22].

GERONIMO (GIROLAMO) CARDANO (1501–1576) also wrote a number of books about the subject. He was responsible for the first grid method for a concealed secret code. And in 1649, at the age of 32, JOHN WALLIS (1616–1703) was awarded an Oxford professorship in recognition of his services as decipherer.

The 19th century brought fresh impetus. In 1854 CHARLES WHEATSTONE (1802–1875) introduced a method of encryption (misleadingly named after Baron PLAYFAIR) which used diagrams and was so straightforward that "even diplomats could be expected to use it". BABBAGE also ventured into this field, both practically and theoretically. But in 1863 the really great success went to a retired Prussian Infantry Major, FRIEDRICH W. KASISKI (1805–1881). He taught how to break a polyalphabetic substitution encryption with a periodically repeated keyword. Since identical causes have identical effects, a frequently occurring syllable or a frequently occurring word will occasionally lead to the same code, namely when it coincides with the same part of the key. The distance between such 'parallel points' is then a multiple of the length of the key. Once you have located a number of parallel points then the highest common factor of the distances between them gives you the length of the key. And if you then divide up the message using the suspected key length you get a number of separate monoalphabetic substitutions that can be solved by analyzing frequencies, possibly even as Caesar substitutions. In 1883 AUGUSTE KERCKHOFFS[23] (1835–1903) achieved a second breakthrough in his brilliant monograph '*La cryptographie militaire*'. He showed how even polyalphabetic encryptions with long keys can be broken if the same key is used for a number of messages.

In the cryptology of 1890, sliding rules and disks were still the only available polyalphabetic encryption aids. Since FRIEDRICH W. KA-

[22] After VIÈTE had decrypted a Spanish message, PHILIP V complained to the Pope that it could only have been done by black magic. But the accusation failed: the Pope had GIOVANNI BATTISTA ARGENTI as adviser.

[23] KERCKHOFFS was also an ardent supporter of the artificial language VOLAPÜK.



Fig. 49: Gilbert Vernam



Fig. 50: Claude E. Shannon

SISKI had showed how to systematically decipher periodic polyalphabetic encryptions, mechanical encryption was not merely a matter of convenience but also a problem of security. At the beginning of the 20th century, when machines made their entrance into cryptology, most of them used polyalphabetic substitutions with a long periodic key. Some of these systems were broken. GILBERT S. VERNAM (1890–1960), however, who in 1917 thought up a machine for the encryption of telegraph messages, i.e. using the BAUDOT (1845–1903) alphabet, helped to find the only way of making a system absolutely secure: by using a key that is entirely random and is never repeated ('individual key'). That this is the case was eventually proved by CLAUDE E. SHANNON (1916–2001), who, during the Second World War, used mathematical methods to investigate the security of encryption methods, and, in pursuance of these studies, founded information theory.

Logical Calculation

> "*We must beware of needless innovations, especially when guided by logic.*"
>
> SIR WINSTON CHURCHILL

A proposition is a linguistic expression that is capable of being both asserted and denied and for which the truth or falsehood can be decided. The very simplest of logical theories, propositional calculus, is not concerned with the contents of propositions, only with their truth value. Propositions can be combined and the rules of logical conclusion can be used to derive further propositions from them.

The above-mentioned operations of propositional calculus, combination and conclusion, are entirely rigid, even when they are performed mentally. We can therefore perform them using mechanical, electromechanical or electronic instruments and machines and save ourselves intellectual work. In view of the fallibility of thought processes, using such a machine in suitable simple cases may even be advantageous. For really extensive and complex relationships it is impossible to use propositional calculus without the help of a machine.

In the narrower sense, as we have seen, man has been attempting to mechanize logical calculations since time immemorial. RAMON LLULL had already described a graphical method of performing logical combinations, the value of which was stressed by ATHANASIUS KIRCHER and the idea of which fascinated the young LEIBNIZ. In the years 1672–1676 LEIBNIZ sketched the rough design of a logical calculator; he wanted to create a *calculus ratiocinator*, a *machina ratiocinatrix*.

LEONHARD EULER (1707–1783) and JOHANN HEINRICH LAMBERT (1728–1777) also worked on diagrams for explaining and carrying out logical operations, and in 1880 the Englishman JOHN VENN (1834–1923) used their work as the inspiration for his Venn diagrams. The equivalent rectangle diagrams suggested in 1881 by ALLAN MARQUAND show that these are what we would in modern parlance call 'value tables'.

A decisive step was taken in 1847 by GEORGE BOOLE (1815–1864) with the algebraization of logic, followed by AUGUSTUS DE MORGAN (1806–1871). CHARLES MAHON, Viscount, later Earl STANHOPE (1753–1816) in 1777, WILLIAM STANLEY JEVONS (1835–1882) in 1870 and ALLAN MARQUAND (1853–1924) in 1883 were the first to



Fig. 51: John Venn

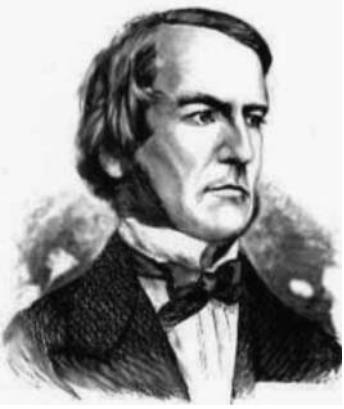


Fig. 52: George Boole



Fig. 53: Augustus De Morgan

build mechanical logical machines. In 1885 CHARLES SANDERS PEIRCE suggested an electrical solution.

Where GEORGE BOOLE had clarified the principle of working with truth values, in around 1870 GOTTLOB FREGE (1848–1925) and PEIRCE (1839–1914) laid the foundation for a mathematical logic. The processing of syllogisms was still interesting as a practical exercise; from 1880 a variety of mechanical aids, such as window cards and the like, were promoted by VENN and by CHARLES LUTWIDGE DODGSON (aka LEWIS CARROLL, 1832–1898), who placed tokens on the card fields. Only JEVONS stands out from the rest: in 1869 he constructed a ‘mechanical



Fig. 54: Charles S. Peirce

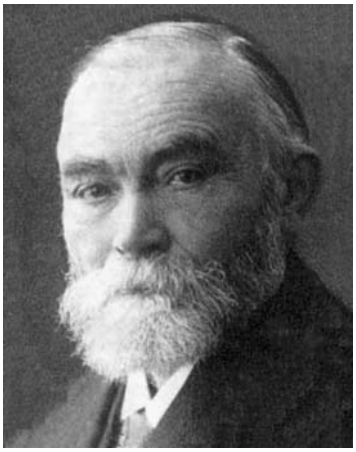


Fig. 55: Gottlob Frege



Fig. 56: Charles L. Dodgson

piano' that made it possible to deal with any combinations of propositions involving up to four variables.

By 1890, the relationship between binary code and two-valued logic was just beginning to become clear, for which we particularly have to thank ALEXANDER MACFARLANE (1851–1913).

With the increase in railway traffic, the logical problems associated with safety and signal technology were only just beginning. For a long time railroad signaling used mechanical binary circuits and registers for selecting and securing routes and for setting signals. And railroads still



Fig. 57: Jevon's 'mechanical piano'

use optical signals (semaphores and lights); with the introduction of running controls operated by track conductors we have the beginning of another new epoch.

Codes Used in Data Transmission Technology

Ancient history records that POLYBIUS taught how to transmit messages using an arrangement of from one to five torches in two hands. This is obviously related to gesture languages that are spoken with the fingers. Fire signals have always been used as alarms and the candle in the window has shown many a traveler the way. Signaling began to acquire modern features with the optical telegraph, in 1787 that of JOSEPH CHUDY (ca. 1752–1813) and in 1789 that of CLAUDE CHAPPE (1763–1805). These were semaphore arms that were able to adopt various positions. The first commercial telegraph route between Paris and Lille was set up in 1794. In the Napoleonic era Europe had a well-functioning message system based on optical telegraphs; the individual relay stations were about 10 km apart, generally located on towers. A character that was displayed for about 20 seconds could travel the 130 miles from Lille to Paris in about 8 minutes.

The earliest methods to use electrical phenomena had a separate line for each symbol, i.e. a one-out-of n code. But binary and ternary codes, which had already been used for flashing signals, were soon also introduced here, for instance in around 1832 in the needle tele-



Fig. 58: Claude Chappe

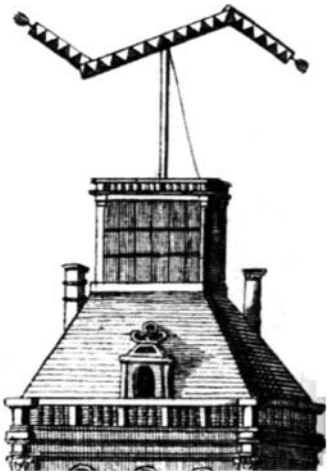


Fig. 59: Optical telegraph

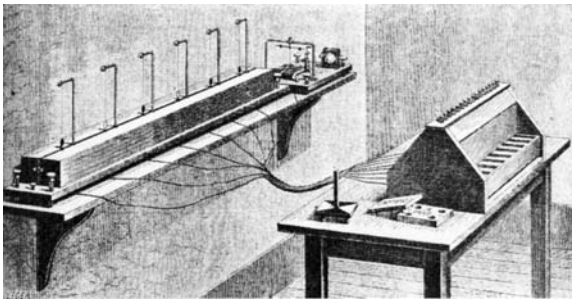


Fig. 60: Needle telegraph by Schilling

graph invented by PAUL (PAVEL LVOVICH) SCHILLING (1786–1837) and in 1833 in one invented by two professors from Göttingen university, CARL FRIEDRICH GAUSS (1777–1855) and WILHELM WEBER (1804–1891). This electrical needle telegraph was soon also used for the emergent railroads; in 1843 a five-needle telegraph that had been invented in 1837 by two Englishmen, WILLIAM FOTHERGILL COOKE (1806–1879) and CHARLES WHEATSTONE, was introduced on the Aachen-Ronheide line. More fundamental, however, was the influence of WILHELM F. COOKE on the invention of the Morse telegraph and the introduction of the Morse alphabet. Incidentally, in 1837, SAMUEL MORSE (1791–1872) initially used only digits to transmit messages, representing words by groups of numbers from a code book. In around 1840, in



Fig. 61: Gauss and Weber

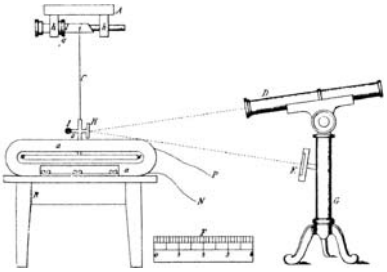


Fig. 62: Telegraph by Gauss and Weber



Fig. 63: Charles Wheatstone

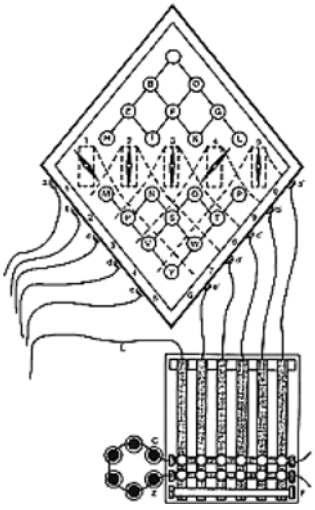


Fig. 64: The five-needle telegraph

order to conserve battery power, he introduced a variable-length word code created by ALEXANDER BAIN (1818–1877). GAUSS and WEBER had previously used a similar code, as had CARL AUGUST VON STEINHEIL (1801–1870) in 1837. The modern Morse alphabet was based on a suggestion made by FRIEDRICH CLEMENS GERKE (1801–1888) in 1848. It was first accepted by the German–Austrian Telegraph Association in 1852 and received worldwide acceptance when, with a number of additions, it was adopted by the International Telegraph Association in 1865.



Fig. 65: Samuel Morse

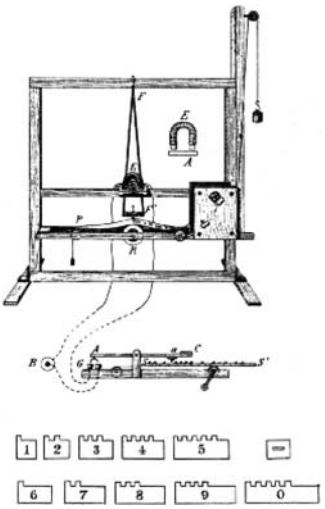


Fig. 66: First 'writing telegraph'

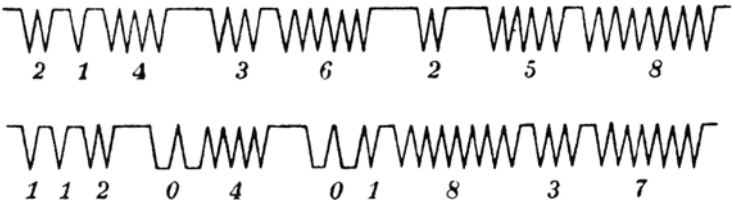


Fig. 67: First telegraph sign code

In 1841 WHEATSTONE built a telegraph transmitter controlled by punched tape, which used a genuine binary representation of the Morse signals. The 5-bit code, which had already been used by FRANCIS BACON (1561–1626), was not revived until 1874 in the printing telegraph invented by EMILE BAUDOT. In 1887, on the basis of this, CARPENTIER introduced five-track punched tape. The CCITT-2 code was adopted as an international standard in 1929.

Commercial telegraph codebooks, principally to reduce the cost of sending telegrams, were first introduced in 1845 by MORSE's partner FRANCIS O.J. SMITH (1806–1876) and were soon very much in use. The first transatlantic cables were laid in 1866.

The telephone invented in 1861 by JOHANN PHILIPP REIS (1834–1874) then made the transfer of messages independent of signaling skills,



Fig. 68: Francis Bacon



Fig. 69: Emile Baudot



Fig. 70: CCITT-2 telegraph code

it became simply a question of patience. But where electrical lines were not practicable, such as on the open sea, there was still the need for optical signals. In 1817 the English sea captain FREDERICK MARRYAT (1792–1848) introduced the International Flag Code, based on a maritime signal book that had been used by Admiral ADAM RICHARD KEMPENFELDT (1718–1782) in the American War of Independence. In 1897 GUGLIELMO MARCHESE MARCONI (1874–1937) liberated signaling from its landlines by introducing wireless telegraphy (radio transmission). Wireless telephony became possible somewhat later. And, for the moment, wireless television (in around 1935) completes the development of message communications equipment[24].

The association with binary code was already evident in 1841 in the telegraph transmitter invented by CHARLES WHEATSTONE and then again in 1874 by EMILE BAUDOT in his printing telegraph.

[24] PALMSTRÖM, a character in a number of poems by CHRISTIAN MORGENSTERN, once suggested building a smell organ. But whether it was to be operated by wireless was not revealed.

The Principle of Binary Coding

The coding principle developed by POLYBIUS when he represented twenty-five single letters by twenty-five pairs of five digits was character-by-word encryption (as opposed to the more straightforward character-wise encryption). It was much used in the Middle Ages for cryptographic purposes. FRANCIS BACON was the first to realize that two digits were sufficient, and thereby introduced the principle of binary coding for a masked secret code.

LEIBNIZ made use of this not only in his *binaria arithmetica* but also for philosophical purposes. We have already mentioned its subsequent use in calculation, telegraphs and telexes (CHUDY 1787, BAUDOT 1874).

The binary principle was first used to store large quantities of numerical data in 1889 by the American HERMAN HOLLERITH (1860–1929), who used punched cards for the national census. The Norwegian FREDRIK BULL (1882–1925), the Americans JAMES POWERS (1870–1915), JOHN KINSEY GORE (1864–1943) and JOHN ROYDEN PEIRCE (died 1933), and HENDRICUS STUIVENBERG in The Netherlands also worked on the technical problems of punched card machines. Alphanumeric punched card machines were introduced in 1931.

CHARLES BABBAGE suffered a setback in his plans for his analytical engine, after which, until 1890, process control for calculating machines made no further progress beyond the difference engine of PEHR G. SCHEUTZ. But 1890 marked the first successful use of punched card counting machines (tabulators) in national censuses in the USA and in Austria by HERMAN HOLLERITH, who had invented them in 1884, and by THEODOR HEINRICH OTTO SCHÄFFLER (1838–1928) who constructed them under license. In 1891, SCHÄFFLER invented the "general commutator" (Generalumschalter), a device that permits rapid switching between samples, thus paving the way for the programmed use of the tabulator. This work brought SCHÄFFLER the first patent ever awarded for programming.

In 1887 HOLLERITH had already equipped his counting machine with an adder, but it was some time before it was actually used for numerical tasks.

The punched paper tape that made its first appearance for telegraph machines in 1841 (WHEATSTONE) had already been used at the beginning of the 19th century for mechanical pianos and musical boxes and goes right back to 1801 when it was used by JOSEPH-MARIE JACQUARD

(1752–1832) for controlling looms; even that was based on earlier equipment introduced by JACQUES DE VAUCANSON (1709–1782) in 1741, JEAN BAPTISTE FALCON in around 1728, and BASILE BOUCHON in around 1725. Amazing though it seems today, BABBAGE was still using a one-out-of n code to control his store. KONRAD ZUSE was probably the first to employ a selection pyramid, in resolute adherence to the binary principle.

Mathematical Roots of the Burgeoning Science of Informatics

On the purely mathematical side, the origins of modern informatics were only vaguely discernable to those in the profession in around 1890, and engineers entirely overlooked them for many years to come. The gradually emerging axiomization of algebra – of which early precursors were BERNARD BOLZANO (1781–1848) with his "Größenlehre" and HERMANN GRASSMANN (1809–1877) with his "Ausdehnungslehre" – provided the initial impetus.

In 1882 WALTHER VON DYCK (1856–1934) had characterized geometric transformation groups by their generators and relations. In 1889 GIUSEPPE PEANO (1858–1932), with his definition of the natural numbers, provided a concrete definable algebraic structure. These advances marked the beginning of the formal thinking that is so typical of informatics.



Fig. 71: Bernhard Bolzano



Fig. 72: Hermann Grassmann

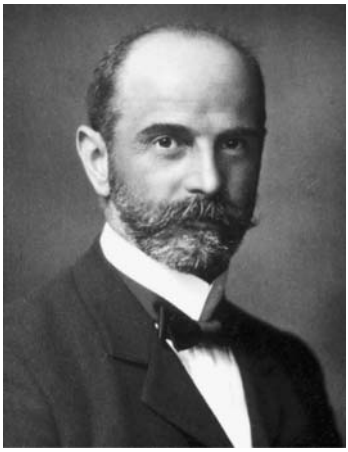


Fig. 73: Walther von Dyck

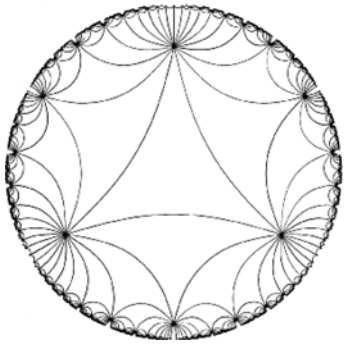


Fig. 74: Hyperbolic tesselation



Fig. 75: Giuseppe Peano



Fig. 76: Ernst Schröder

In 1877, ERNST SCHRÖDER (1841–1902) considerably advanced the formalization of mathematics with his first work, "*Operationskreis des Logikkalküls*"; in 1890 his "*Algebra der Logik*" had just been published. At the same time, SCHRÖDER had already set out the basis for a general theory of relations.

But discreteness and discretization were still in the shadows. Analog calculation was still widely acclaimed. In around 1890 the triumph of the real numbers, under GEORG CANTOR (1845–1918), RICHARD DEDEKIND (1831–1916) and KARL WEIERSTRASS (1815–1897), made



Fig. 77: Elie J. Cartan

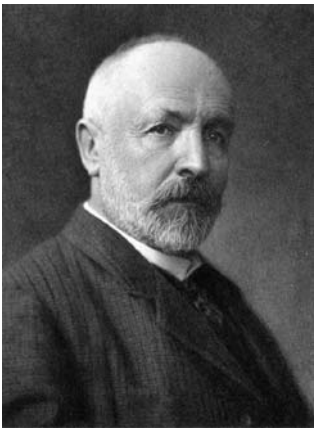


Fig. 78: Georg Cantor



Fig. 79: Sophus Lie

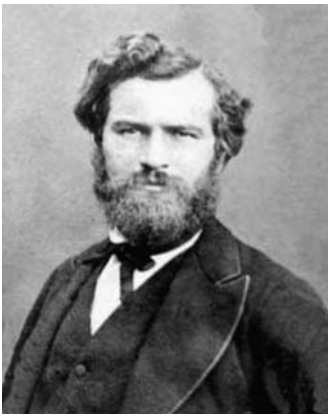


Fig. 80: Camille Jordan

the continuum the preferred venue of mathematics. Topological groups were more interesting than discrete groups; in 1888 WILHELM KILLING (1847–1923) had enumerated the semi-simple representations of "classical groups", followed in 1894 by the young ELIE JOSEPH CARTAN (1869–1921) in his thesis. As far back as 1875, MARIUS SOPHUS LIE (1842–1899) had studied the infinitesimal transformation groups, after CAMILLE JORDAN (1838–1922), in 1868, had determined the groups of motions in three-dimensional Euclidean space and their closed subgroups. However, JORDAN also advanced a theory of finite groups. In

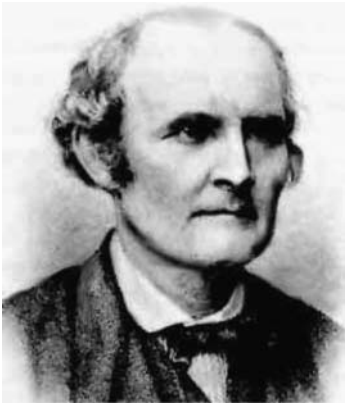


Fig. 81: Arthur Cayley



Fig. 82: Eugène C. Catalan

1894 the "*Lehrbuch der Algebra*" by HEINRICH WEBER (1842–1913) recorded current developments in the theory of groups, rings and fields, the favorite children of algebra. Lattice theory, such as DEDEKIND's dual groups in 1897, was still only just beginning.

In 1890 combinatorial analysis was making progress. The diversions of RAMON LLULL (1232–1315) mentioned in LEIBNIZ's "*ars magna*" and "*ars combinatoria*" had become insignificant, but combinatorics received a fillip from a number of practical successes, such as the work of EUGÈNE CHARLES CATALAN (1814–1894) and that of ARTHUR CAYLEY (1821–1895) on enumerating trees, which was of significance for structural formulae in chemistry.

In 1890 the idea of algorithms and associated questions such as complexity had not yet been addressed. There would have been occasional mention of algorithms for multiplication, division and taking roots using digits in a place value system, just as LEIBNIZ had mentioned such things. Not only was the term "Gaussian algorithm" for the solution of a system of linear equations not yet in use, the general idea of algorithms was still dormant. But in his "*Die Grundlagen der Arithmetik*" in 1884, FREGE took an essentially constructive approach to providing mathematics with a foundation of pure logic – which was later to be found to be particularly appropriate for informatics.

Hopefully, these few keywords will prove sufficient to sketch the position of mathematics at the turn of the century as it relates to the germinating field of informatics. Many fundamental issues had not yet been raised at all. Analog computing – which belongs not under informatics

but in the technical area of modeling – was highly thought of, planimeters were to remain in use for a long time, and integration mechanisms were still being built during the Second World War.

The progression to non-countable sets, however, which acquired increasing significance for mathematics following Cantor's theory of infinity, had no direct relevance to informatics.

After 1890: In Thrall to Mechanical and Electromechanical Devices

In the years 1890–1935 the budding science of informatics was still completely overshadowed by the devices and machines that were intended to facilitate working with numbers and other data. The classic demands made of engineers predominated: building reliable implementations at sensible prices.

The problems encountered were those of information processing on the one hand and those of program control on the other, but one of these generally outweighed the other and often there was no attempt at reconciling them.

At this time formal aspects were still of secondary importance; only towards the end of the epoch did languages and algorithms adopt consideration of abstract objects (KONRAD ZUSE's abstract circuit notation is a simple example).

Mechanization and Automation of Digital Calculations

> "Das numerische Rechnen wird in alter Zeit als eine Kunst, nämlich als die Rechenkunst, bezeichnet. Gegenwärtig ist es jedoch angemessener, von einer Technik des Rechnens zu reden.[25]"
>
> HEINRICH BRUNS, 1907

Digital arithmetic is a means of processing either notches on a tally, as used by primitive peoples, or numbers written using digits in a place value system. For theoretical purposes, tally arithmetic, as used by logicians, does have certain advantages; for the purposes of mechanization the latter, more efficient method, is more appropriate. And here, as BLAISE PASCAL already showed, the actual place value is not so very important: PASCAL's machine was basically decimal but had twelve as

[25] In antiquity, numerical arithmetic was described as an art, namely the art of reckoning. But at present it is more appropriate to speak of a technology of reckoning.

F.L. Bauer, *Origins and Foundations of Computing*,
DOI 10.1007/978-3-642-02992-9_4,
© Springer-Verlag Berlin Heidelberg 2010

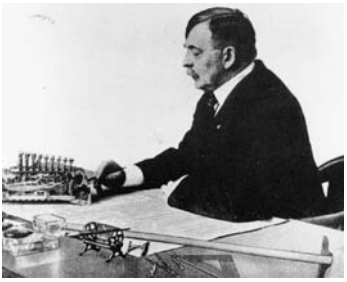


Fig. 83: Dorr E. Felt



Fig. 84: Comptometer by Felt

the base value of its last position and twenty for the last but one, corresponding to the Carolingian currency: the underlying number system did not have a simple radix. The decisive feature of PASCAL's machines, and presumably also those of WILHELM SCHICKARD, was an adder with an automatically propagated carry, which permitted addition to be carried out in each position (one after the other) using a stylus. The construction of dedicated adding machines was improved from 1890, in particular they became lighter and cheaper. In around 1890, LOUIS TRONCET and PETER GUTHRIE introduced a series of small machines, often made from tinplate, which possessed toothed wheels, racks and chains, of which the simplest offered only semiautomatic carrying. In 1913 MICHAEL BAUM constructed what was probably the flattest stylus adding machine of the century.

Further improvements came about from 1884 with the introduction of multiple-order keyboards by DORR EUGENE FELT (1862–1930) and WILLIAM SEWARD BURROUGHS (1857–1898).

These are particularly associated with the construction of an appara tus for the repeated addition of a stored summand ('two-stage addition'), which BURROUGHS introduced in 1888, to make it possible to print the individual items in a total. The latched key was the 'switching organ' corresponding to the stepped-drum of LEIBNIZ' machine. However the commercial machines built by FELT, BURROUGHS and their successors did not have the movable carriage that made it possible to shift the position of the stored multiplicand. This constructional difficulty was not overcome until after 1945, when time had almost run out for mechanical calculators.



Fig. 85: William S. Burroughs



Fig. 86: Burroughs' calculator



Fig. 87: Oskar Sundstrand



Fig. 88: Arthur Burkhardt

From 1901, WILLIAM HOPKINS (1850–1916), HUBERT HOPKINS (1859–?) and OSKAR SUNDSTRAND (1889–?) introduced a convenient numeric keypad for adding machines, which was in some ways a backward step in that two-handed use, which was quicker, was no longer possible.

The first multiplication machines to be industrially manufactured efficiently were the stepped-drum machines made by ARTHUR BURKHARDT (1857–1918) in Glashütte in around 1890; from 1895 they were followed by a number of manufacturers who, one by one, also installed full keyboards and later even numeric keypads for the multiplier.

Another switching organ, the previously mentioned pin-wheel, introduced into commercial series calculating machines in 1872 by FRANK



Fig. 89: Friden's calculator



Fig. 90: Marchant calculator

S. BALDWIN and in 1874 by WILLGODT T. ODHNER, made it possible to build smaller and cheaper machines for multiplication and division, which dominated the market throughout the entire epoch.

In 1921 the company Marchant brought out constructional improvement on the pin-wheel, the set segment ("Stellsegment"), which makes it possible to avoid the rotating set lever, and in 1933 HAROLD T. AVERY (1892–1974) achieved a significant increase in calculation speed with a proportional gear transmission. This innovation made the MARCHANT the star of mechanical calculators. In 1943, when extensive numerical calculations had to be carried out by hand in Los Alamos, the preferred machines were MARCHANTs and FRIDENs – the last of the purely stepped-drum machines.

Other switching organs had been invented by CHRISTEL HAMANN: proportional levers in 1913, and switch handles ("Schaltklinken") in 1925.

Genuine multiplication machines don't operate with multiple addition of the multiplicand, they build up their partial products directly. After a failed attempt by EDUARD SELLING in 1866, LÉON BOLLÉE successfully created a multiplication device that was able to represent the multiplication table up to ten, from which OTTO STEIGER then constructed a well-functioning machine, the MILLIONAIRE. This was cumbersome and expensive, and also lacked any direct provision for divisions, but by 1935 almost 5000 of them had been built. By comparison, small pin-wheel machines were selling in their hundreds of thousands. Multiplication devices were occasionally introduced for accounting machines, for instance in 1892 in the 'Moon-Hopkins' of WILLIAM S. BURROUGHS, in 1927 in the 'United', and in 1934 in the 'Rheinmetall'.

As you see, the essential ideas for mastering the mechanization of digital calculation were already to be found at the beginning of the period

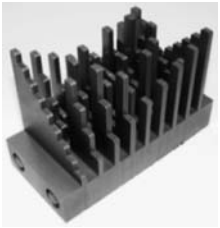


Fig. 91: Multiplication device



Fig. 92: Steiger's Millionaire

1890–1935, and this continued to be the case for automation. In 1902, with the introduction of motorized propulsion, ALEXANDER RECHNITZER had already begun automating multiplication and division; this was perfected in 1913 in the proportional lever machine MERCEDES EUKLID built by CHRISTEL HAMANN and the MADAS, a stepped-drum machine built by HANS W. EGLI and ERWIN JAHNZ.

These mechanical calculating machines typically used wheels or toothed racks to represent their (generally decimal) numbers. Gear wheels with two teeth do not constitute a viable solution for working in the binary system, for which reason RAYMOND L.A. VALTAT, in 1931, had already suggested working with eight teeth, i.e. to use octal in order to get nearer to the binary system – although this did not offer any particular advantages. From 1935 KONRAD ZUSE then abandoned toothed wheels and went over to more technically expedient binary principles. LOUIS COUTURAT (1868–1914) energetically propagated LEIBNIZ's ideas regarding the binary system.

Using electromagnetic devices to execute the fundamental arithmeti cal operations of addition, subtraction, multiplication and division was an obvious step only in areas where electromechanics was already being employed, particularly in association with punched-card tabula tors. Electromechanical incremental counters were used to represent and store the numbers. Addition and subtraction using incremental counters was tried in around 1910 by JAMES POWERS and HERMAN HOLLERITH; from 1931 there were fully automatic tabulators that could also handle multiplication and division. Although printing out results was of primary importance there were no facilities for reading in (written) initial data. In 1931 GUSTAV TAUSCHEK (1899–1945) developed a machine that could "see": an apparatus for reading printed numbers.

All the calculating machines mentioned here are restricted to fixed numbers of positions.



Fig. 93: Gustav Tauschek

Mechanical and Electrical Cryptological Instruments

> *"De tous les cryptographes qui ont eté inventés dans ces dernières années, celui de Wheatstone me parait être, sinon le plus sùr, du moins le plus simple.*[26]*"*
>
> AUGUSTE KERCKHOFFS, 1883

Mechanization came to cryptography later than to numerical calculation. Initially the emphasis was on substitution using an alphabet of $Z = 10$ digits or (more recently) $Z = 26$ letters. It was not until 1925 that ALEXANDER VON KRYHA (1891–?) equipped the classic cipher disk with a device that advanced it irregularly and on a relatively long cycle to provide for polyalphabetic encryption. Somewhat before this (for instance as introduced by KARL HAAS and HEINRICH STUDT in 1908, and later by ARVID GERHARD DAMM (?–1928)) there had been simple devices with something like ten to thirty code alphabets arranged on a cylinder in such a way that, when the entire cylinder was rotated, exactly one alphabet appeared in a window. The devices patented by SERGE KANSCHINE and EMIL JELLINEK-MERCEDES (1853–1918) in 1911 and EDWARD HUGH HEBERN (1869–1952) and FRED HOFFMANN in

[26] Of all the cryptographs that have been invented in the last few years, WHEATSTONE's seems to me to be, if not the most reliable, at least the most straightforward.



Fig. 94: Alexander von Kryha

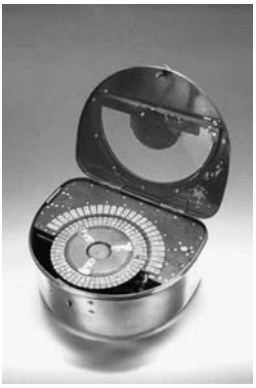


Fig. 95: Kryha's cipher machine



Fig. 96: Edward Hebern

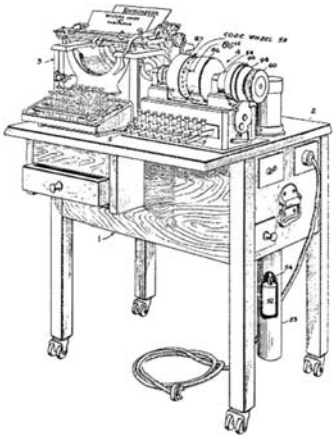


Fig. 97: Rotor machine by Hebern

1914 were equally banal: neat little covers that could be attached to the keys of a typewriter to permutate the alphabet.

An electrical solution had obvious appeal. In 1912 HEBERN suggested using a bundle of wires to connect two electric typewriters; substitution would be effected by permutating the connections. In 1917 he invented a device which he called a 'rotor' for generating varying permutations $\varrho^{-i}\pi\varrho^{i}$, where π is a fixed permutation and ϱ is cyclic of order Z. The same idea (originally for ten digits, later also for 26 characters) occurred to ARTHUR SCHERBIUS (1878–1929) in 1918 and to HUGO ALEXANDER KOCH (1870–1928) and ARVID G. DAMM in



Fig. 98: Arthur Scherbius

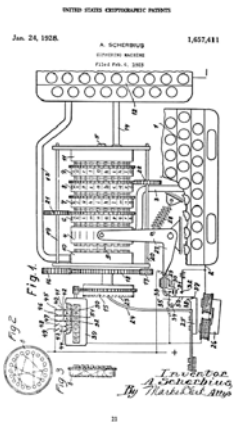


Fig. 99: Enigma Patent



Fig. 100: Arvid G. Damm



Fig. 101: Arvid G. Damm's A-21

1919 (their patent applications, in The Netherlands and in Sweden, respectively, were only three days apart).

Soon a number of rotors were being connected in series and advanced irregularly. SCHERBIUS had already applied for a patent on such a device in 1920 (five years earlier than VON KRYHA). In around 1930 WILLI KORN made the rotors exchangeable. A commercial version with the trade name ENIGMA went on the open market in 1923 but was not a success. This changed after 1933: a later development, the ENIGMA C had already been introduced for the German Navy in 1926 and reached the German Army in 1928.



Fig. 102: Boris Hagelin



Fig. 103: B-21 cipher machine

The ENIGMA I, introduced in 1930, had an additional plug-in substi tution. The '*Wehrmacht* ENIGMA' of 1936 was introduced in even greater numbers: at a rough estimate 50,000.

In 1926, BORIS HAGELIN (1892–1983) departed from DAMM's electric rotors. His first machine, the B-21 was already a success.

Eventually the USA armed forces built 140,000 M-209s under license from Hagelin.

In 1917, GILBERT S. VERNAM set about polyalphabetic encryption in a fundamentally different way. Starting from the 5-bit code used by telegraph machines (International Telegraph Alphabet No. 2) he simply needed to perform a bitwise addition modulo 2 (exclusive OR), i.e. identity wherever the key has a *0* and with the binary digits O and L exchanged wherever the key has a *1* (Vernam cipher). VERNAM combined this with the idea of using, as key, an individual and entirely random series of *0*s and *1*s, provided to both the sender and receiver on identical 5-bit punched tapes that were to be destroyed after a single use. Practical difficulties made VERNAM's original method unviable, although it is unbreakable if properly used – a fact which was subsequently formally proved in the USA by JOSEPH O. MAUBORGNE (1881–1971) and in Germany by WERNER KUNZE and RUDOLF SCHAUFFLER.

VERNAM ciphers were, however, used in 1931 combined with irregular high-period advancing in Siemens' encrypting telegraph machine T 52 and in 1942 in Lorenz AG's '*Schlüsselzusatz 42*'. The Siemens T 52 used an additional transposition of the bits of a 5-bit group, but this was effectively equivalent to a substitution using an alphabet with $Z = 32$ characters.



Fig. 104: Joseph O. Mauborgne

Polyalphabetic encryption using addition over the ring of integers modulo a number p, which was already the basis of the systems developed by LEONE BATTISTA ALBERTI and BLAISE DE VIGENÈRE (1523–1596), is particularly inviting for digital codes (modulo 10). An algebraic version was first used effectively in 1885 by GAËTAN HENRI LÈON DE VIARIS (1847–1901). As early as 1868 F.J. SITTLER had, in principle, propagated '*superencrypted*' codes, specifically in groups of four modulo 10,000. Towards the end of the 19th century '*additives*' were a standard method of code superencryption. It is all the more astonishing that nobody came up with the idea of modifying an adding machine by removing the digit carry at suitable positions in order to mechanize a polyalphabetic substitution over a dozen or so digits. Incidentally, 'one-time pads' – individual and completely random series of 'additives' – had also come into use for superencryption over exceptional communications routes.

In 1929, LESTER S. HILL (1891–1961) went a considerable step further in this direction. He suggested using n-dimensional linear transformations over the residue class ring modulo Z, typically with $n = 4$, and in around 1930 he also developed a wheel-driven cipher machine that used this very principle. Thus, in the 1920s, the field of cryptography was already being invaded by professional mathematics, particularly algebra and stochastics – the latter being contributed by WILLIAM FREDERICK (WOLF) FRIEDMAN (1891–1969), the most significant cryptolo-



Fig. 105: Blaise de Vigenère



Fig. 106: William F. Friedman



Fig. 107: A. Adrian Albert

gist of the century. His cryptoanalytic methods ('index of coincidence', 1920) plus the chi-test and phi-test, which were developed under FRIEDMAN's influence in 1935 by SOLOMON KULLBACK (1907–1994) and ABRAHAM SINKOV (1907–1998), required extensive statistical evaluations and thereby led to the massive use of punched card machines for cryptanalysis. Optical aids were also much used for character pattern matching; an idea that had already been used prior to 1933 by DERRICK HENRY LEHMER (1905–1991) in his number sieve device for factorizing large numbers. In 1941, after the USA's entry into Second World War, the algebraist A. ADRIAN ALBERT (1905–1972) underlined the significance of mathematical thinking and methods in the field of cryptology.



Fig. 108: Derrick H. Lehmer

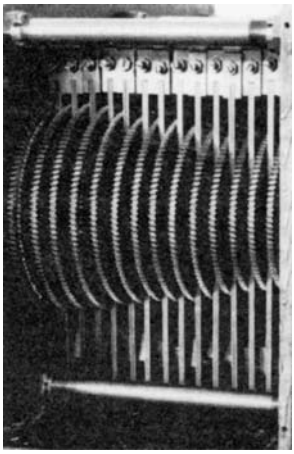


Fig. 109: Mechanical number sieve



Fig. 110: William T. Tutte

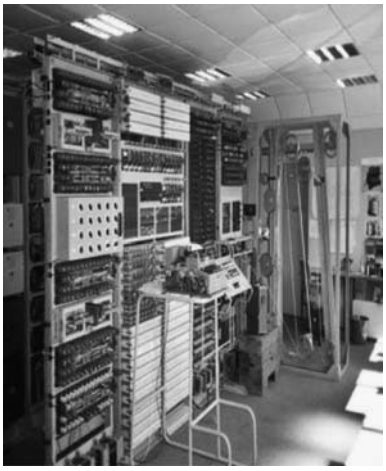


Fig. 111: Colossus reconstruction

In general, the cryptographic devices that were used, provided they exploited their technology to the full, were quite capable of with standing attacks from devices based on the same technology. The British authorities defeated the German electromechanical cipher telegraph machines, which used automatically generated keys, only by using fast electronic comparators. The breakthrough achieved by the ma thematician WILLIAM THOMAS TUTTE (1917–2002) in February of 1942 led to the construction of the COLOSSUS machines.

The fundamental impossibility of complete disorder, a theoretical result proved by F.P. RAMSEY (1903–1930) in 1925, should have been a warning to cryptologists to be careful in their use of machine-generated ciphering.

Message Coding Technology and Information Theory

Unlike cryptography, message coding technology does not aim to conceal the message; this type of code only serves to increase technical efficiency, and most of the codes used are fixed.

In 1874 EMILE BAUDOT introduced a 5-bit code that was superior to Morse code, and in 1887 CARPENTIER put this into the form of the International Telegraph Alphabet that is still in use today. For decades there were no great advances – encoding language did not become widespread until after 1935.

A theory of coding was not developed until quite late. Initially the emphasis was on the construction of (binary) codes with specific technically relevant properties: BAUDOT in 1874 and subsequently GRAY in 1953 had introduced single-step codes, in 1889 BAUDOT introduced chain codes, and in 1959 GEOFF TOOTILL introduced single-step chain codes. BAUDOT's introduction of both single-step codes (in 1874) and chain codes (in 1889) was a breakthrough: for the first time codes exhibited internal properties that went beyond their lexicographical arrangement.

The task of making telegraph codes resistant to noise was first solved (in 1937) by HENDRIK C.A. VAN DUUREN (1903–1981) by means of a special 7-bit code, a 3-out-of-7 code, after a simple parity check had been found to be inadequate for radio transmissions. In 1950, RICHARD WESLEY HAMMING (1915–1998) was then the first to specify a code that permitted certain types of errors to be corrected. In around 1880 a 'two-character distance' had already become usual for commercial codes and by around 1920 there was also a safeguard against inadvertently exchanging two neighboring letters.

CLAUDE E. SHANNON introduced an important new approach. In 1948, working on the basis of investigations into cryptographic security that had been carried out during the Second World War (and were published in 1949) he introduced (decision)-information theory based on purely statistical assumptions about the source of a message.



Fig. 112: Richard W. Hamming

0	0000000	1	000LLLL
3	00LL00L	2	00L0LL0
5	0L0L0L0	4	0L00L0L
6	0LL00LL	7	0LLLL00
9	L00LL00	8	L0000LL
10	L0L0L0L	11	L0LL0L0
12	LL00LL0	13	LL0L00L
15	LLLLLLL	14	LLL0000

Fig. 113: Hamming code

We define a decision to be a choice of one character from a set of n characters, where $n \geq 2$; it can be carried out by means of a cascade of binary decisions. If we require k_i binary decisions in order to isolate the ith character, then its probability is $p_i = (\frac{1}{2})^{k_i}$. Conversely, to choose a character that occurs with probability p_i we require $k_i = \mathrm{ld}(\frac{1}{p_i})$ binary decisions[27]. We call k_i the decision content of the character, measured in bits[28].

The average decision content of an arbitrarily selected character is then

$$H = \Sigma_i p_i \,\mathrm{ld}\left(\frac{1}{p_i}\right) \quad [\text{bit}]$$

H is also referred to as the information per character, or the entropy of the message source.

The association with cryptology is that the laws of statistics can be applied to a message or a key in order to break the cipher, and that an 'individual key' contains no information. In a series of epoch-making papers SHANNON showed how much information can be conveyed even in the presence of noise. It this he almost echoed the work of NORBERT WIENER (1894–1964), who, in 1948, investigated how to eliminate noise that conforms to specific statistical laws ('smoothing') and even how to predict signal behavior. The Russian mathemati cian AN-

[27] Where ld (*logarithmus dualis*) denotes a logarithm to base 2.

[28] A bit is the smallest unit of information, the choice between two binary digits.

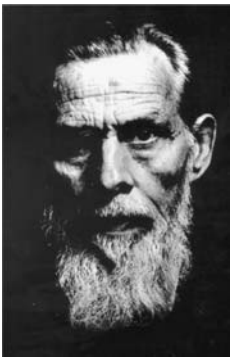


Fig. 114: Warren McCulloch



Fig. 115: Walter Pitts

DREI NIKOLAJEWIČ KOLMOGOROV (1903–1987) independently recorded a number of results along similar lines. In the context of the rapid development of electronic computer systems, and ornamented with superficial interpretations of the neurophysiological work of WARREN MCCULLOCH and WALTER PITTS (1943), WIENER amalgamated his own amazing results and those of SHANNON to constitute a superscience, which he called 'cybernetics' (*Control and Communication in the Animal and the Machine*, 1949).

Instruments and Machines for Propositional Calculus, Signaling and Safety Technology

> *"A logic machine is a device, electrical or mechanical, designed specifically for solving problems in formal logic. Let no reader imagine that logic machines are merely the playthings of engineers who happen to have a recreational interest in symbolic logics."*
>
> MARTIN GARDNER, 1958

Propositional calculus machines also made only one significant ad vance: in 1890, following the initial mechanical attempts by JEVONS and the improvements made to them in 1883 by ALLAN MARQUAND, MARQUAND discussed the use of relays and switching circuits for sentential combinations with the logician CHARLES S. PEIRCE, and HERMAN



Fig. 116: Paul Ehrenfest

HOLLERITH also used such connectors in the selector circuits of his punched-card machines.

The first practical use of electrical contacts to verify syllogisms was made in 1936 by BENJAMIN BURACK (*1914) – right at the end of the period we are considering and only one year prior to SHANNON's fundamental investigation that demonstrated the isomorphism, between propositional calculus and switching circuits.

ROBERT W. EDLER (1874–1958) in 1905 and N. LISCHKE in 1911 had already analyzed relay circuits without, however, drawing an explicit correlation to propositional calculus, and the analogy between propositional calculus and relays had been established already in 1910 by PAUL EHRENFEST (1880–1933).

Finally, in the mid-1930s we find papers about relay circuits by VITALI IWANOWICH SHESTAKOV (*1907) in the Soviet Union, by AKIRA NAKASIMA (1908–1970) and MASAO HANZAWA (*1913) in Japan, and by JOHANNA PIESCH (1898–1992) in 1939.

Railway signaling and safety technology soon developed ways of com bining conditions that amounted to formulas in propositional calcu lus. In 1897 M. BODA had already recognized their correlation with electrical relay circuits, but for the time being mechanical solutions for such problems as 'route locking' were to remain predominant. Through his father the young KONRAD ZUSE became familiar with rail traffic regulations, particularly signaling and safety technology; this may have paved the way for his early experiments with mechanical switching, and he was also influenced by PIESCH's work.

Digression into Analog Computers

Numerical arithmetic, ciphers and logical calculation are typically 'discrete' and digital in nature, but there is also a 'continuous' realization of reckoning. This is predominately concerned with 'analog'representations of real numbers and rationals using displacements and rotations, representing quantities physically by means of rods and sliders on the one hand or by wheels and disks on the other. Where such machines employ toothed racks and gear wheels these are used only to give form-fit connections: in purely analog instruments they do not have any discretizing significance.

Simple analog instruments, for instance slotted straight edges and articulated linkages for drawing curves, attained a certain refinement in 1864 with the Inversor invented by CHARLES PEAUCELLIER (1832–1913). This was simply a linkage for drawing a semi-ellipse, it was mentioned in 1891 by JOHANN KLEIBER (1865–1941), a student of VON DYCK, but really had no practical significance. Yet there are surprises here, too: in 1926 SEMEN ARONOVICH GERSHGORIN (1901–1933) constructed a wheel-driven device for drawing ellipses that was particularly smooth-running.

Analog calculations make use of scales. Addition is implemented as the conjunction of sliders or disks. If ϕ is a scale representation, then slider or disk addition gives us $\phi^{-1}\big(\phi(a) + \phi(b)\big)$ just as WILLIAM



Fig. 117: Semen A. Gershgorin



Fig. 118: William Oughtred

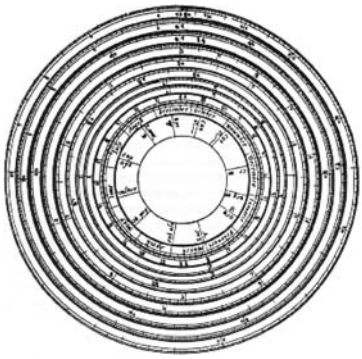
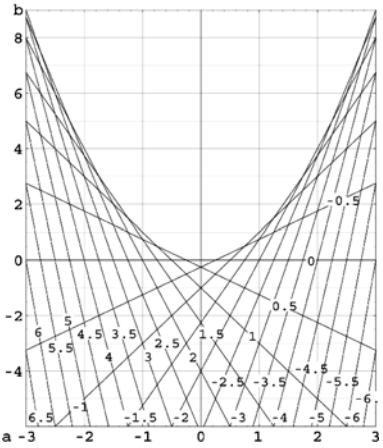


Fig. 119: Oughtred's circles of proportion



Horizontal scale: a
Vertical scale: b
Along the straight lines: solutions x

Fig. 120: Nomogram of the quadratic equation $x^2 + 2ax + b = 0$

OUGHTRED (1575–1660) found in 1622. Not only multiplication using logarithmic scales but many other commutative monotonic two-argument functions can be handled in this way.

The next step is nomograms. The two common forms, concurrency charts and alignment charts, are geometrically dual and therefore theoretically equivalent and are used for two-argument functions which, in this case, do not necessarily need to be commutative.

But they are not really instruments, merely sheets of paper. For an alignment chart the user only needs to use a ruler to connect values mea-

sured off on two adjustable scales and use a result scale to read off the result at the intersection. (Example: alignment chart for the formula $\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2}$.) For concurrency charts one does not even need a ruler: the set points a and b are understood as the coordinates of a certain point $P(a, b)$; from a family of curves, each curve associated with a coefficient x, the user has to find that curve on which point $P(a, b)$ is located. (Example: concurrency charts for the quadratic equation $x^2 + 2ax + b = 0$.)

This lends practical significance to HILBERT's problem: in what circumstances can an n-argument function in the form of a tree be realized using two-argument functions? In the nature of things, nomograms do not lead to any more far-reaching mechanized achievements. From the 1960s they shared the fate of analog computers and fell out of use.

We cannot really speak of analog computation until at least addition and multiplication by a constant factor are completely mechanized. In around 1890, planetary gears for the former task and gear ratios for the latter had long been known. Pulleys were often used as well as gearwheels. In 1879, WILLIAM THOMSON, LORD KELVIN (1824–1907), had already built such a machine for solving systems of linear equations; JOSEF NOWAK (1879–1916) used gearwheels. The technical problem is implementing analog multiplication. Single argument functions are comparatively easy to manage by means of curve followers; we frequently find multiplication implemented as $\mathrm{sq}(a + b) - \mathrm{sq}(a - b)$, where sq is defined by $\mathrm{sq}(x) = x^2/4$ – a method of numerical calculation that may already have been used in ancient times. Such solutions typically exhibit a relatively restricted argument space.



Fig. 121: Lord Kelvin

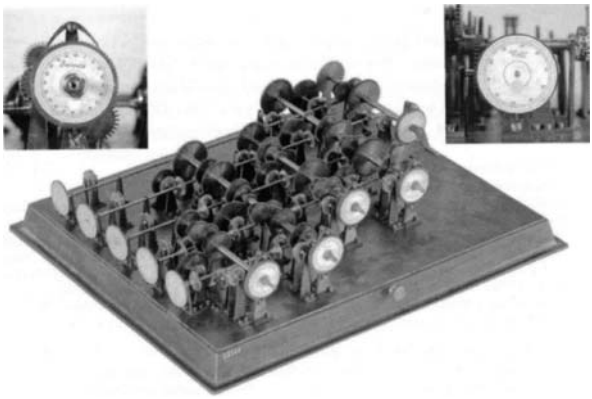


Fig. 122: Machine by Josef Nowak



Fig. 123: Vannevar E. Bush



Fig. 124: Bush's Differential Analyser

What was remarkable for analog computations, at least for a while, was the obvious and simple method of performing integration using a friction gear such as those familiar from planimeters. Since this does not provide sufficient adherence it is necessary to use an additional torque amplifier. In 1930, VANNEVAR E. BUSH (1890–1974) was the first to build a successful analog system for integrating differential equations. Other differential analyzers were built in 1933 by DOUGLAS RAYNER HARTREE (1897–1958) and in 1938 by SVEIN ROSSELAND (1894–1985). The torque amplifier was invented in 1927 by C.W. NIEMANN (1886–1963) based on the capstans that are much used on ships and in dockyards. In around 1940 there was a general move to electrical torque amplifiers; in 1943 ROBERT SAUER (1898–1970) and H. PÖSCH made use of the amplification provided by a servomechanism. These were also



Fig. 125: Douglas R. Hartree



Fig. 126: Alwin Walther



Fig. 127: Robert Sauer

used in 'curve followers' that not only recorded curves but, with cams, could also be used to represent single-argument functions. Finally, the multiplication $u \cdot v$ was frequently implemented as $\int u\,dv + \int v\,du$.

Occasionally a knife-edged friction wheel (Schneidenrad) was used instead of a friction gear, for example in 1939 by ALWIN WALTHER (1898–1967) and WILFRIED DE BEAUCLAIR (*1912).

In the period 1935–60 analog computation was still in command of practical electronic solutions, and this took its technology closer to the electronic implementation of discrete calculations using numbers and symbols, against which it was then no longer able to compete. But, in any case, in the taxonomy of sciences, analog computation, relying as it does on the concept of the continuum, has no place in informatics: an implementation that makes use of physical analogies does not permit arbitrary increases in precision.

Musical Boxes and Other Precursors of Automata

Marionettes, mechanical toys, moveable masks, counters and measuring instruments are the precursors of automata: 'self movers'. Early examples of automata include clepsydras and even mechanisms resembling clockwork astronomical calculators (one such example from 82 BC was found off the island of Antikythera).

In the first century BC, PHILON OF BYZANTIUM constructed mechanically moving theaters. Heron constructed numerous automata in around 100 AD. During the Renaissance, the development of clockwork – the '*Zytglogge*' in Bern was put into operation in around 1500 – gave the construction of automata a sizeable fillip, and the figures were continually refined right up to the Rococo period, by PIERRE GAUTIER; JAMES UPJOHN (1722–1795); JACQUES DE VAUCANSON, who, in 1738, demonstrated a flute player and a mechanical duck; LUDWIG KNAUS (1724–1789) in Vienna, who, in 1760, constructed a writing automaton; and HENRI-LOUIS JAQUET-DROZ (1752–1791), who, in 1774, demonstrated a drawing device, an automatic writer and a piano-playing machine. The 'androids' (*L'homme machine*) introduced in 1748 by JULIEN OFFROY DE LAMETTRIE (1709–1751) also have a role to play in the literature of the 19th century: *Maschinenmann* by JEAN PAUL (properly JOHANN PAUL FRIEDRICH RICHTER (1763–1825)), *Sandmann* by ERNST THEODOR AMADEUS HOFFMANN (1776–1822), *Kater Murr* and *Olympia* in the operetta by JACQUES OFFENBACH, *Coppelia* in the ballet of that name by LEO DÉLIBES, and finally a couple of late arrivals from the beginning of the 20th century, the robot



Fig. 128: Antikythera mechanism

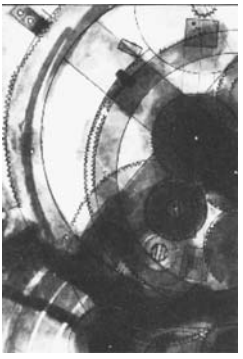


Fig. 129: X-ray photograph

of KAREL CAPEK (1890–1938) and the golem of GUSTAV MEYRINK (1888–1932).

Automata, the Principle of Automatic Machines

> "Denn es ist ausgezeichneter Menschen unwürdig, gleich Sklaven Stunden zu verlieren mit Berechnungen.[29]"
>
> LEIBNIZ

As LEIBNIZ foresaw: freeing mankind from the burden of repetitive, tedious mental activity has been a powerful driving force throughout the development of informatics.

If we compare the process of multiplication using Leibniz's stepped-drum machine with that of the multiplication device in STEIGER's MILLIONAIRE machine we see that the latter involves only one step, whereas the former requires a whole series of steps. Accordingly, automating a stepped-drum machine calls for an additional process control; in Steiger's multiplier the extra effort goes into the processor, the switching organ.

For multiplications by factors with more than a single digit even the MILLIONAIRE requires multiple steps, summing the partial products. It would be conceivable, if there were not too many positions, to construct all the partial products simultaneously and let them accumulate. This approach would enable even multiposition multiplications to be carried out in a single step.

Thus we see that process control is a means of performing a given processing task using a less complex processor but in a way that requires a number of steps (and therefore more time). This is subject to a duality: the effort invested in constructing the processor can be replaced by effort expended on process control, and vice versa.

By around 1890, process control had reached a considerable degree of precision, for instance in musical boxes, but it was essentially simply repetitive, occasionally involving nested loops. At this time, regulators, such as those invented by JAMES WATT (1736–1819) for the safe operation of steam engines were not really a part of process control: in musical boxes they were important only for ensuring an extremely constant speed.

[29] It is unworthy of excellent men to lose hours like slaves in the labor of calcula tion.

It is characteristic of the period up to 1935 that process control was used only to minimize the mechanical effort of the processor. This was dictated both by cost considerations and technological factors: mechanical and electrical solutions quite soon reached limits of complexity that could no longer guarantee reliable operation.

Process Control as a Precursor of Program Control

For a time, the automation of information processing got along very well using the rigid sequences with at most nested loops that were familiar from clocks and musical boxes. In 1902, this approach ena bled ALEXANDER RECHNITZER to completely automate multiplication and division. Even the extraction of square roots was subsequently automated mechanically for the FRIDEN. From 1931, punched-card tabulator machines were able to offer fully automatic multiplication and division. Complex composite processes made up of many individual processing steps were initially automated in tabulator machines by means of punched-card technology. Control panels with cords and plugs were used to determine the sequence that was to be initiated. This was an extremely laborious undertaking, but replaceable panels did make it possible to keep reusing a previously plugged task. If we discount external influences such as manual intervention or punching control positions on the cards, it was these panels that housed the programs that controlled the process. This type of program control, which had been anticipated

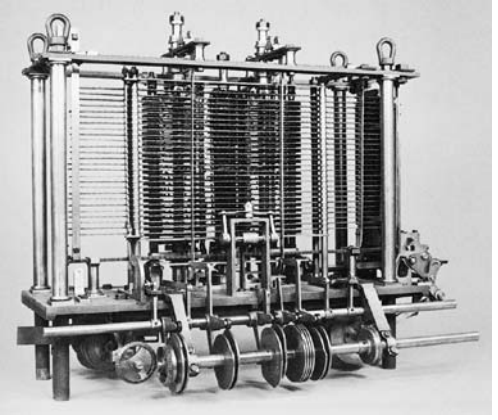


Fig. 130: Analytical Engine

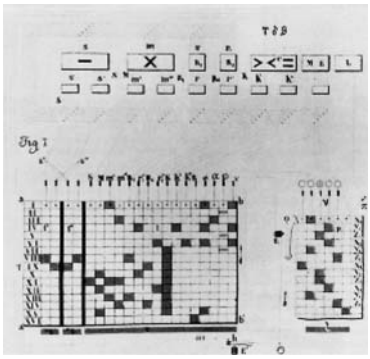


Fig. 131: Sketch of electromech. machine



Fig. 132: L. Torres y Quevedo

in the exchangeable programs for the automated figurines of the 18th century and the replaceable punched boards for musical boxes, came into fashion during the 1920s, culminating in the D 11 tabulator from DEHOMAG in 1936.

On a smaller scale, accounting machines were also provided with process control facilities. In 1931 there was a particularly remarkable example in the LOGABAX of FRANCISCO P. CAMPOS (1885–1955), which possessed an impressive number of accumulators.

Alongside these practical and successful constructions a number of broader approaches were merely incidental: in 1903, PERCY EDWIN LUDGATE (1883–1922), following BABBAGE's Analytical Engine, suggested incorporating jumps in the process, an innovation that would have made the manyfold available (although still entirely rigid) loop structures easier to handle.

Between 1910 and 1920 LEONARDO TORRES Y QUEVEDO (1852–1936) constructed an electromechanical machine with automatic multiplication and division, in the course of which, in 1913, he performed the very first theoretical investigation into possible process structures. GUSTAV TAUSCHEK developed an electromechanical accounting machine that was intended to be able to handle 10,000 accounts.

In 1929 LESLIE JOHN COMRIE (1893–1950) organized ordinary commercial accounting machines as difference machines to carry out interpolating tabulator calculations.

In 1909 HAMANN had already built a printing dual calculating machine for second-order differences, and this was followed by FRANZ TRINKS (1852–1931) in 1928 with the Brunsviga Dupla.



Fig. 133: Lesley J. Comrie

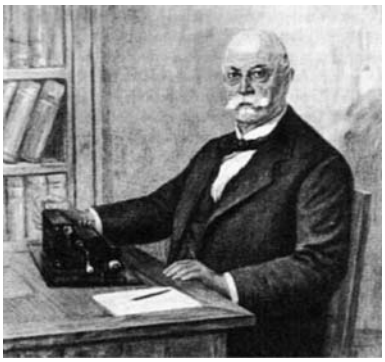


Fig. 134: Franz Trinks

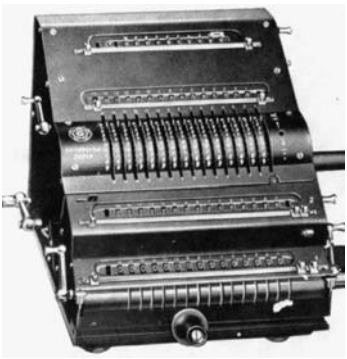


Fig. 135: Brunsviga Dupla

In 1920, A.J. THOMPSON connected four Triumphator desk calculators to form a fourth-order difference machine. During the 1920s, complex systems of punched card machines were also constructed, for instance in 1927 by LESLIE J. COMRIE, in 1928 by A.J. THOMPSON, in 1933 by WALLACE J. ECKERT, and in 1936 by VANNEVAR E. BUSH. Many such attempts, such as that of JOHN TODD (1911–2007) in 1943, made use of the NCR 3000 model accounting machine, which had six result units and permitted the semiautomatic processing of interpolation tasks. JOHN VON NEUMANN also gathered programming experience on this machine. In 1926 EMIL SCHILLING (1887–1963) applied for a patent for a pneumatic control unit for desk calculators.

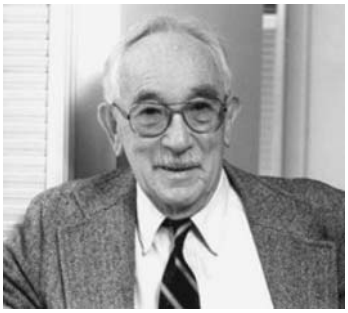


Fig. 136: John Todd

Universal Program Control

"The Analytical Engine weaves algebraical patterns."
ADA AUGUSTA COUNTESS OF LOVELACE, 1843

The use of punched tape to control production processes was already mentioned in our section on 'The Principle of Binary Code'. The possibility of programming anything more than a simple cyclic repetition was first recognized by CHARLES BABBAGE with his Analytical Engine, the design of which, dated 1833, provides for it to be guided by the results of its calculations, i.e. to possess a control unit[30]. This was to be supplemented by a 'mill' or arithmetic unit, a 'store', an input unit and a printer. Babbage's son, Major-General HENRY PREVOST BABBAGE, constructed a number of individual components, sufficient to be able to compile a table of the multiples of π. ADA AUGUSTA (1815–1852), Countess of Lovelace, was a pupil of BABBAGE, and we know of her deliberations about the question of programming from the officer LUIGI FEDERICO MENABREA (1809–1896) from Turin. Lady LOVELACE clearly recognized that BABBAGE's machine "does not occupy common ground with mere 'calculating machines'. It holds a position wholly its own ... in enabling mechanisms to combine together general symbols".

In 1909, PERCY E. LUDGATE went beyond BABBAGE's ideas. He not only recognized the significance of conditional jumps, he already introduced three-address commands. Compared to this, the punched-card machines, tabulators, invoicing machines, and even the typewriter

[30] BABBAGE said he would make the machine "bite its own tail".



Fig. 137: Augusta Ada Lovelace



Fig. 138: Joseph Petzval



Fig. 139: Konrad Zuse

with attached calculator invented by L. TORRES Y QUEVEDO in 1910 were just rudimentary ideas that came to nothing.

It is, however, interesting that as early as 1840 JOSEPH PETZVAL (1807–1891) had employed a group of gunners to carry out extensive calculations for the development of a system of optical lenses, and had had them work according to a program ('calculation plan'). Some such calculation forms were published. For example, ALWIN WALTHER in Darmstadt encouraged PAUL TEREBESI to publish a form for calculating Fourier coefficients. KONRAD ZUSE records that it was a form he had used as a student, for calculating moments of inertia for structural engineering, that opened up the direct route to program control that eventually led to his Plankalkül in 1945.

In modern parlance, these forms were programs written for human computers, who either performed the calculations "in their head" or using simple devices, often merely slide rules or adding machines. Forms were the precursors of software.

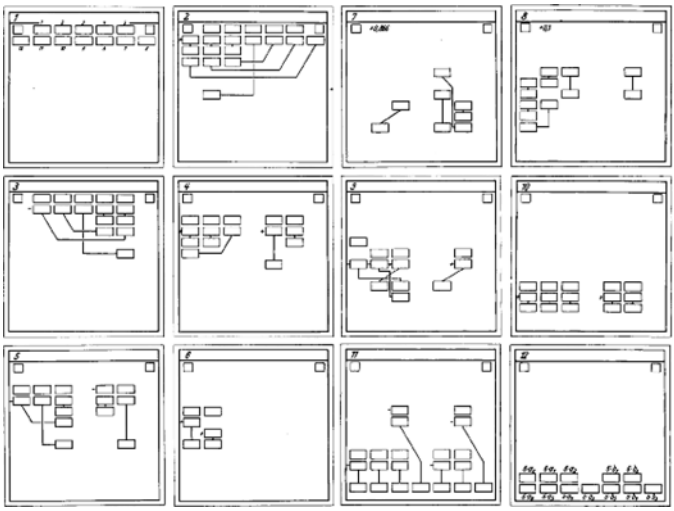


Fig. 140: Terebesi's form for calculating Fourier coefficients

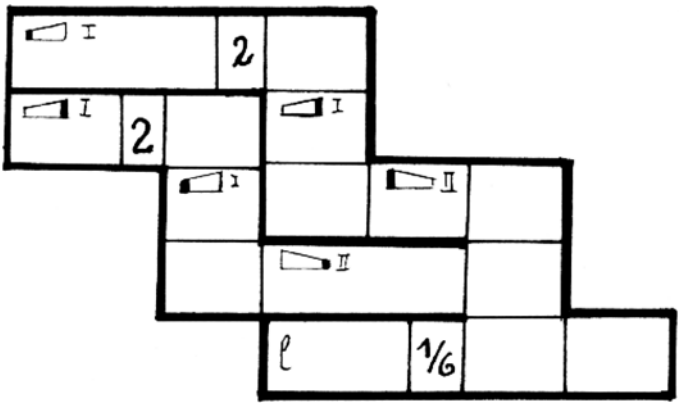


Fig. 141: Form for the calculation of static momentums by Zuse

In 1938, LOUÏS COUFFIGNAL did not progress beyond very embryonic program control, but the developments he had been working on since 1934 did, in 1941, enable KONRAD ZUSE to complete the first functioning program-controlled computer system, the Z 3, costing 25,000 Reichsmarks. In 1944 the US government put the Mark I Computer, built by HOWARD H. AIKEN, into operation. As early as 1942, Bell Telephone Laboratories had begun to use a relay computer developed by GEORGE R. STIBITZ. The first computer system to use



Fig. 142: Maurice V. Wilkes

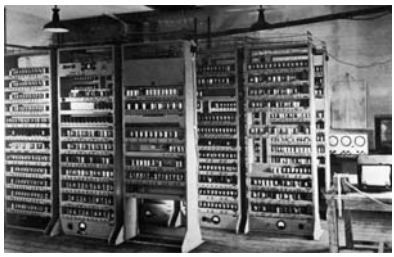


Fig. 143: EDSAC computer

valves, the ENIAC, which was begun in 1943 by J. PRESPER ECKERT and JOHN W. MAUCHLY, was first used in 1946, but these systems did not achieve the universality of Zuse's design.

JOHN VON NEUMANN, with his design of the 'Princeton Machine' during the period 1946–1948, was the first to achieve a complete breakthrough culminating in a universal computing machine. The basic idea of a stored-program computer system was initially put on paper on 30th June 1945 by JOHN VON NEUMANN in collaboration with JOHN W. MAUCHLY and J. PRESPER ECKERT. This was then taken over by MAURICE V. WILKES, who put the first such system, the EDSAC, into operation in May 1949.

In 1944 MAUCHLY, ECKERT and GOLDSTINE also introduced the principle of serial operation, using a single digit adder to add all positions one by one. HEINZ BILLING (*1914) and ANDREW DONALD BOOTH (*1918) recognized the advantages of this when used with a rotating storage unit.

Feedback and Regulation

Using signals to control a mechanism rather than to convey messages to other people is one of the earliest achievements of any technology. Feedback provided by signals reporting on the achieved effect of other signals was initially an obvious monitoring step. But when the returned signal itself mechanically gives rise to a further control signal, for instance to correct a departure from some desired situation, then we have a new kind of feedback: regulation by means of a closed signal cycle. Initially

used by JAMES WATT to regulate the rotational speed of steam engines, this idea was soon extended to cover pneumatic, hydraulic and electrical control procedures. Investigations into the stability of control circuits were first carried out in 1877 by the mathematician EDWARD JOHN ROUTH (1831–1907) and in 1895 by the mathematician ADOLF HURWITZ (1859–1919), and their work was continued in 1921 by the mathematician ISSAI SCHUR (1875–1941) and the communications engineer HARRY NYQUIST (1891–1970). WILHELM CAUER (1900–1945), the most significant theoretician in the field of linear circuits, recognized the connection between stability and passivity in electrical networks and the role physical causality has to play in the theory of signal transference. NORBERT WIENER crowned these considerations by contributing a theoretical synthesis of probability theory, Fourier transformation and passivity conditions. With his sampling theorem CLAUDE E. SHANNON made a contribution linking coding theory with the theory of continuous signals. In 1925 the physiologist R. WAGNER first established the occurrence of feedback in biological systems, and in 1933 RAGNAR FRISCH (1895–1973) introduced economic cycles into the discussion.

Feedback is a special feature of analog calculating instruments. Unlike tasks involving only formula evaluation and/or integration, an equation or differential equation inherently produces feedback, particularly when the differential equation is resolved at the highest level of derivative. In mechanical integrators the independent variable is also represented by a rotary quantity (and not by the real time). In this case the real time has purely an auxiliary function, and the feedback is the only element of a classic integrator system that exhibits a loop structure – not the sequence of events but the effect ('effective cycle'). There is, however, no direct way to transmit this effect. Analog computing on the one hand, and digital program-controlled computing on the other, cannot help but move further apart.

Negative feedback is a particularly useful factor in classic control technology. How effective the above-mentioned servo mechanisms are as regulators depends quite critically on how well we can handle their inherent feedback.

After 1935: Formal Languages and Algorithms, 'Universal Machines' and Electronic Solutions

As late as 1930, mathematicians still did not perceive much call for formal syntactical descriptions of their notation. On the contrary, it was determined by tradition, with the result that separate schools of usage arose associated with different countries or cities, and sometimes with individual printers.

At the beginning of the 1930s, logicians, with their inborn need to question everything and influenced by explicit studies of derivation rules and systems, began to take notice of this; one such was PAUL HERTZ (1881–1940) in 1929. In 1930, JAN LUKASIEWICZ (1878–1956) informally introduced his bracket-free 'Polish notation'.

Then, in 1934, RUDOLF CARNAP (1891–1970) took up the term 'syntax', which had already been used in 1826 by E. COLLINS, and specified an entirely formal system: a 'formal language', incorporating a theoretical treatment plus rules for constructing and deriving terms. A fundamental paper published in 1943 goes back to work completed in the period 1920–1922 on 'normal systems' by EMIL LEON POST (1897–1954). But as yet there is no mention of Post algorithms nor of Markov algorithms. The word 'algorithm' is itself not yet much heard, but the idea is spreading.

The Proof Theory of the 1930s incidentally prepared the ground for both theoretical informatics and the theory of computability, which was based on the work of EMILE BOREL (1871–1956) and initially introduced generally recursive functions: this began in 1931 with some rudimentary work by JACQUES HERBRAND (1908–1931) and was continued in 1934 by KURT GÖDEL (1906–1978). In 1935, on the basis of this, STEPHEN COLE KLEENE (1909–1994) published the first formulation of a theory of recursion using text replacement and simplification, which later became central to informatics; he also showed the equivalence to μ-recursive (partially recursive) functions. Then, in 1936, there appeared the pioneering paper *On Computable Numbers* by ALAN M. TURING. This was the birth of the Turing machine as a conceptual construct. It really ought to be called the Turing–Post Machine, because in 1936 E. POST independently described a startlingly simi-

F.L. Bauer, *Origins and Foundations of Computing*,
DOI 10.1007/978-3-642-02992-9_5,
© Springer-Verlag Berlin Heidelberg 2010



Fig. 144: Jan Lukasiewicz



Fig. 145: Rudolf Carnap



Fig. 146: Emil L. Post



Fig. 147: Emile Borel

lar construction. In 1937 it had already been established that Turing-computable functions were equivalent to the 'λ-definable functions', which had been investigated from 1932 by ALONZO CHURCH (1903–1995) and KLEENE, and also to the related combinatorially definable functions introduced from 1924 by MOSES SCHÖNFINKEL (1889–1942) and from 1930 by HASKELL B. CURRY (1900–1982), and to the generally recursive functions. Thanks to this theory there was no longer any



Fig. 148: Jacques Herbrand



Fig. 149: Kurt Gödel



Fig. 150: Alonzo Church

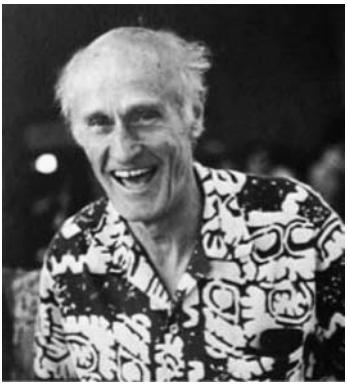


Fig. 151: Stephen C. Kleene

need to actually build these universal computing machines for theoretical reasons – neither TURING's nor KLEENE's.

Recursive thinking was, however, not exclusively the province of logicians: the Hilbert curves described in 1890, Sierpinski curves, and later examples had all already been defined by means of recursive algorithms.

By 1935 the seeds of almost all the elements of informatics were already present. In particular, the interaction of the constructivist orientation of HASKELL B. CURRY with the intuitionist orientation of LUITZEN EGBERTUS JAN BROUWER (1881–1966) was to have an as yet unsuspected significance for programming theory.



Fig. 152: Haskell B. Curry



Fig. 153: Luitzen E.J. Brouwer

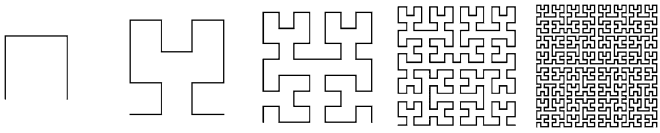


Fig. 154: Recursive construction of the space-filling curves of David Hilbert

The period 1935–1960 now introduced numerous new ideas and techniques that began to flesh out the bare bones of informatics. These years also revolutionized the ways of thinking and of working that now characterize the face of informatics. These were pioneer years, in which – compared to today – there were only a few people at the cutting edge of research. The field was a fruitful one, and many results were almost inevitable. Successes were also frequent, and this led to an often burgeoning and sometimes naïve optimism. People's mental attitudes were determined by what was possible. And given enough resources very much appeared possible that could previously only have been dreamt of. Computer systems became prestige objects. The Hilbert curve could be used as a logogram for Informatics.

Basic Data Structures

After 1935 abstraction was gaining more and more significance, but it was still the hardware that determined what actually happened. Not until the end of this period, around 1960, could one begin to suspect that software would one day acquire spectacular economic significance.

Binary Arithmetic

> "Der Übergang zum konsequenten Denken in Ja-Nein-Werten war um 1934 keineswegs selbstverständlich.[31]"
>
> KONRAD ZUSE, 1970

The most noticeable external feature for this revolutionary period is the adoption of the binary system. This took place in two separate ways, both resulting from one and the same realization: when we change over from a mechanical implementation to an electrical one the 'on–off' of a switch is a determinant. One typical approach was to continue to calculate in decimal, but with the individual decimal digits encoded in binary, while others straightaway began using binary arithmetic. Since it is not feasible to convert the quantities in everyday use into any of the radixes that are trivially equivalent to the binary system, such as octal or hexadecimal, this group was then faced with the ongoing problem of converting input and output data between the binary and the decimal systems.

The above development, however, did not begin with electrical solutions. Unlike RAYMOND L.A. VALTAT, who adhered to the use of wheels, KONRAD ZUSE, in 1935 set out on a new path: he used genuinely binary mechanical switching elements. This technology – bundles of slotted plates floating one above the other with orthogonally positioned pilot pins – could already have been used in 1890 or even earlier. These switch elements were cheap and could be manufactured using the simplest of tools: for the pilot model, nowadays referred to as the Z 1, on which work began in 1936, ZUSE used a fretsaw to make the slotted plates. Nevertheless, this technology was no longer easy to manage for large systems. ZUSE wrote

> "I had spent two years struggling with mechanical constructions but eventually I had to give up"

and that he

> "often had to spend two weeks taking the system apart when something had become jammed."

[31] In 1934 the transition to logical thinking in terms of yes and no was by no means obvious.

In this context we should point out that ZUSE's ideas initially centered on the problems of storing large volumes of data (a question he began addressing in 1934) and that, as he records, he originally considered using relays but shied away from that idea because of the space involved. His mechanical flip-flop worked extremely well, and he therefore initially planned to construct the arithmetic unit using the same technology...

> "but I was forced to recognise that this mechanism is not flexible enough for such tasks"

ZUSE never revealed what made him think of using metal plates, perhaps it was a combination of his youthful hobby of metalwork and his impressions of railway safety systems gleaned from a book in his father's library. We also do not know whether it was ZUSE's original ideas about recording data by punching holes in forms (we shall come to this later) that led him to binary numbers.

In 1936 he completed the mechanical memory of the Z 1 and in 1938 also its arithmetic and control units.

Following a further setback with the 'experimental relay model' of an arithmetic and control unit, nowadays referred to as the Z 2, the relay machine that is now called the Z 3, on which work began in 1940, was a complete success. It was demonstrated as operational on 12th May 1941, the first freely programmable program-automated binary computer system. The arithmetic unit contained 600 relays, and the memory a further 1400.

His consistent use of not only binary switch elements but also the binary number system put ZUSE years ahead of his time. Counting technology was still clinging to the decimal system when thermionic valves came into use; there were even a number of decimal counting tubes developed, in 1938 by M. MATHIAS and in 1939 by W.W. HONDORF.

In 1936, independently of ZUSE, E. WILLIAM PHILLIPS's knowledge of VALTAT's patent application led him to suggest one particular use of the binary system, namely for a fast multiplication unit, specifically in combination with optical switching elements. He recorded that he was also thinking of an implementation using thermionic valves, but expunged all reference to this from his patent application on instructions from the British government.

In 1936, LOUIS COUFFIGNAL also promoted binary arithmetic. In 1932, CHARLES E. WYNN-WILLIAMS had already constructed binary counters (binary circuits) using thyratrons for physics experiments on

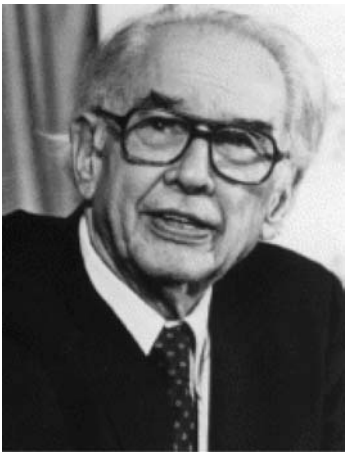


Fig. 155: John V. Atanasoff



Fig. 156: Clifford E. Berry



Fig. 157: Atanasoff–Berry computer

cosmic radiation, and in 1935 JOHN VINCENT ATANASOFF (1903–1995), also a knowledgeable physicist, had direct access to the binary system as a consequence of his use of tubes. His design, completed in 1938 in collaboration with CLIFFORD EDWARD BERRY (1918–1963), led to a pilot model, a dedicated computer for solving systems of linear equations, which was demonstrated in the autumn of 1939. Further work on the machine was discontinued in 1942 because of the war.

Binary arithmetic was encountered afresh in 1944 in the EDVAC, the successor to the ENIAC, which was designed by JOHN W. MAUCHLY

and J. PRESPER ECKERT and then elaborated by JOHN VON NEUMANN assisted by HERMAN H. GOLDSTINE and ARTHUR WALTER BURKS (1915–2008). The binary system became an integral part of the many machines that were influenced by this design.

But, for the time being, GEORGE R. STIBITZ and HOWARD H. AIKEN were clinging to binary-coded decimal arithmetic. They both began their work in 1937: STIBITZ used partly biquinary code and partly excess-three code, and AIKEN used a peculiar code with place values 2 4 2 1.

Large-scale systems, if they were intended for commercial purposes, continued to use binary-coded decimal arithmetic internally for quite some time in order to avoid having to make the complicated transition from decimal to the binary system. Pocket calculators and small-scale portable computers continue to use binary-coded decimal arithmetic to this day.

Floating-Point Arithmetic

Whereas binary arithmetic is merely a question of the representation of natural numbers, floating-point arithmetic involves numbers of a quite different kind with quite different operations performed on them, and initially it makes no difference whether they are represented in the binary or decimal systems.

By their very nature, the numbers that are encountered in calculations relating to physical and technical problems are only rarely integers. And calculations (or rather approximations) involving irrational numbers make it necessary to abandon natural numbers and use fractions. One way of doing this is to use pairs of numbers (numerator, denominator); we shall discuss this later. The other way is to use decimal fractions. Here one is only rarely in the fortunate situation that the initial data and all the intermediate results fall within a single order of magnitude $a - 10a$ or $a - 2a$, i.e. the location of the decimal or binary point remains fixed – as for instance when approximating to a number monotonically from above or from below, as with alternating series or with Gauss's arithmetic-geometric mean. In general, however, when we perform numerical calculations with decimal fractions we cannot simply let them run automatically without knowing the orders of magnitude of each of the intermediate results – our numbers would either get out of range, forcing us to abort the calculation, or we would lose precision, which is generally unacceptable. It is difficult to determine an appropriate scaling.

From 1920 it became increasingly usual for physicists to state measurements using scaling factors, e.g. to speak of $2.99776 \cdot 10^8$ m/sec instead of 299,977,600 m/sec. Multiplying quantities in this 'semi-logarithmic' representation, $a \cdot 10^b$ with integer values for a and b, involves multiplying their 'pseudomantissas' and adding their 'exponents'. The addition of two such 'floating-point numbers' is comparatively complicated because the exponents need to be adjusted.

As early as 1914, L. TORRES Y QUEVEDO had mentioned mechanizing working with numbers in this form, which was naturally already familiar to anyone skilled in performing calculations manually. KONRAD ZUSE proposed using binary floating-point arithmetic for the Z 1, as evidenced by a patent application from 1936. This was then implemented in the Z 3, completed in 1941: the mantissa contained 14 bits plus sign, and the exponent 7 bits plus sign. But as VON NEUMANN, GOLDSTINE and BURKS wrote in 1946, floating-point arithmetic was initially not considered worth the necessary outlay on additional hardware.

By the end of the 1940s, the relay machine BARK of CONNY PALM (1907–1961) and (in 1955) the Munich PERM of HANS PILOTY (1894–1969) and ROBERT SAUER had hardwired binary floating-point arithmetic. But in most cases floating-point arithmetic was implemented by means of internal programming with acceptance of the very great loss of speed this entailed.

Floating-point arithmetic in the decimal system could be found in 1944 in STIBITZ's relay machine model V (seven decimal digits plus sign for the mantissa, one decimal digit plus sign for the exponent, biquinary code), in 1945 in AIKEN's Mark II (Aiken code), and in 1946 in the card-programmed calculator of JOHN W. SHELDON and LISTON TATUM. Today, even programmable pocket calculators have floating-point arithmetic.

Unlimited Integer Arithmetic

The parallel addition used in most mechanical and electromechanical arithmetic units meant that the processing width was fixed. The transition to electronic circuits did not change this. Certainly early machines, such as the EDSAC (1949) of MAURICE V. WILKES, that performed serial binary or binary coded addition could have added numbers of arbitrary length, but the storage systems used strongly suggested a limit on the length.



Fig. 158: Harry D. Huskey



Fig. 159: Jay W. Forrester



Fig. 160: Whirlwind computer

Machines with parallel addition, such as the SWAC (1950) of HARRY DOUGLAS HUSKEY (*1916), the WHIRLWIND (1951) of PERRY O. CRAWFORD (*1917), JAY W. FORRESTER (*1918) and ROBERT R. EVERETT (*1921), and the PERM then continued to use the fixed number of positions determined by their construction.

With appropriate programming it was generally possible to calculate to twice or three times that number of positions, but unlimited integer arithmetic raised the problem of storage allocation. In 1957, FRIEDRICH LUDWIG BAUER (*1924) and KLAUS SAMELSON (1918–1980) suggested using pushdown stores ('stacks') to carry out integer arithmetic.



Fig. 161: Robert R. Everett



Fig. 162: Thomas Harriott

Unlimited integer arithmetic is available today in a number of programming and formula manipulation systems.

The Binary Principle

Binary code, which had been used in 1623 by FRANCIS BACON for cryptographic purposes, and which turned up again in the five-bit codes of THEODOR H.O. SCHÄFFLER and BAUDOT (where it formed the basis for the telegraph alphabet CCITT-2) and yet again on the punched cards of HERMAN HOLLERITH (initially with a one-out-of-10 code) is more elementary than the binary arithmetic originally discussed by THOMAS HARRIOTT (1560–1621) and promoted by GOTTFRIED W. LEIBNIZ. Binary codes with code words of equal length are the basis of most technical codes. Using code words of unequal length introduces the problem

of reconstructing the interstices, for which ROBERT M. FANO (*1917) came up with a solution. CLAUDE E. SHANNON showed that, for a stochastic message source it is possible to choose the lengths of the individual code words optimally and that one thereby strips the redundancy out of the message. Conversely, RICHARD W. HAMMING made use of this redundancy to create error-correcting codes.

The fact that any code can quite simply be reduced to a binary code had already been stated by KONRAD ZUSE in his journal in 1937. In 1945, Zuse's Plankalkül [plan calculus] then extended this binary principle to provide a mechanism for defining 'compound statements'. And for the first time this permitted the unrestricted definition of basic data structures.

Recursive Data Structures and Syntax Diagrams

The ALGOL Reports published in 1958 and 1960 introduced the construction of numerically indexed tuples of objects for the stepwise definition of complex data structures. The recursive definition of data structures had already been considered in 1961 by JOHN MCCARTHY (*1927) and was studied systematically in 1970 by CHARLES ANTONY RICHARD HOARE (*1937) and in 1971 by JACKSON CLARK EARLEY.

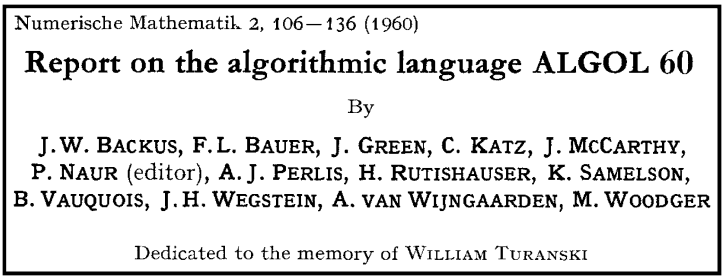
The ALGOL Reports of 1958 (edited by ALAN J. PERLIS (1922–1990) and KLAUS SAMELSON) and 1960 (edited by PETER NAUR) also included the first formal description of the syntax of programming languages; the 1960 report described the syntax in Backus–Normal Form, suggested by JOHN WARNER BACKUS (1924–2007). Early ALGOL



Fig. 163: John McCarthy



Fig. 164: C. Antony R. Hoare


Numerische Mathematik. 2, 106—136 (1960)

Report on the algorithmic language ALGOL 60

By

J. W. BACKUS, F. L. BAUER, J. GREEN, C. KATZ, J. MCCARTHY, P. NAUR (editor), A. J. PERLIS, H. RUTISHAUSER, K. SAMELSON, B. VAUQUOIS, J. H. WEGSTEIN, A. VAN WIJNGAARDEN, M. WOODGER

Dedicated to the memory of WILLIAM TURANSKI

Fig. 165: ALGOL 60 – Report



Fig. 166: John W. Backus

'translators' (compilers) had been written in machine-oriented assembler language, but in 1969 ALBERT A. GRAU understood the ALGOL syntax as the definition of a recursive data structure, and was therefore able to apply the technique known as 'bootstrapping' to write an ALGOL compiler in its own language.

KURT GÖDEL would have had no reason to frown: the ALGOL 60 language was sufficiently simple that there was no need to fear an antinomy. Syntax diagrams described these data structures as graphs; graph theory thus became one of the mathematical tools of informatics. Many results from discrete mathematics subsequently became useful aids for computer scientists. Significantly, the editors of the German Mathematical Society's 1990 centennial report '*Ein Jahrhundert Mathematik*' (GERD FISCHER, FRIEDRICH HIRZEBRUCH, WINFRIED SCHARLAU, WILLI TÖRNIG) placed the section on informatics immediately after the section on discrete mathematics.

Basic Operative Structures

Algorithms

The word 'algorithm' goes back to AL CHWARIZMI (in around 820), who helped to popularize the Indian way of writing numbers using 'Arabic' digits and thus to spread arithmetic throughout the Occident. The contrast between 'abacists' and 'algorithmicists' still dominated the 16th century. LEIBNIZ still used the word 'algorithm'in this sense: 'the multiplication algorithm'. Later the term became generally used for any "manipulation of symbols according to fixed rules", for instance in algebra, for Euclid's method of determining highest common factors, or for the 'q-d algorithm' of HEINZ RUTISHAUSER (1918–1970), but more recently, principally in the field of logic.

The history of algorithms is practically a repeat of arithmetic with symbols. With his idea of the *ars magna* RAMON LLULL was already striving for a universal method for discovering "all truth". LEIBNIZ endeavored, with no tangible success, to create a complete system of rules. He also thought about determinability and calculability; his work on these topics remained unpublished until 1900. The turning point came with formalization, the abandonment of (inherently imprecise) natural language for describing algorithms and their objects in favor of symbolic languages. GOTTLOB FREGE's 'concept notation' (1879), BERTRAND RUSSELL (1872–1970) and the book *Principia Mathemat-*



Fig. 167: Al Chwarizmi



Fig. 168: Alfred North Whitehead

ica (1910–1913) he wrote in collaboration with ALFRED NORTH WHITEHEAD (1861–1947), and GIUSEPPE PEANO were prominent.

The major questions, such as completeness, were then answered in 1930 by KURT GÖDEL. The problem of determinability, which, in individual cases, resisted an algorithmic solution (e.g. the word identity problem for finite generated groups, dealt with in 1882 by WALTHER VON DYCK, and again in 1914 by AXEL THUE (1863–1922)), then demanded a precise definition of the term 'algorithm'. The various attempts by means of equation calculus (HERBRAND 1931, GÖDEL 1934, KLEENE 1936), the Turing machine (TURING 1937), Post algorithms (POST 1936), Markov algorithms (MARKOV (1903–1979) 1947, 1954), lambda calculus (CHURCH 1936), and combinatory definability (J.B. ROSSER 1935), all proved to be mutually compatible and equivalent to a class of functions in number theory, the partially recursive functions (CHURCH's theorem).

The Turing machine[32] provided an algorithmic concept that permitted machine interpretation. The computer system ACE, which was built in Britain under TURING's influence (and completed in 1950), and the systems built in the USA and elsewhere from the plans of JOHN VON NEUMANN and put into operation from 1951, were the first universal computers that completely fulfilled the requirements of computability. Certain features of the Turing machine are reminiscent of BABBAGE.

[32] For TURING himself and for his intended purpose there was no need to actually construct such a machine, but models were subsequently built, for instance by GISBERT HASENJÄGER (1919–2006).

Algorithmic Languages

Formalized means of describing algorithms were not only of theoretical importance. The development of 'algorithmic languages' (a term that was introduced by HERMANN BOTTENBRUCH in 1958) as programming languages for computer systems was initiated in 1951 by RUTISHAUSER. Questions of notation were initially significant (ALEKSEI A. LJAPUNOV (1911–1973), 1957) but gradually the problem was reduced to that of finding a conceptual basis for programming. Essential contributions to this were found in the ALGOL 60 Report (1960), which was based on a previous version from 1958. This development was interrelated with the continuing advances in the functional capabilities of computer systems (symbolic addresses by WILKES 1953, indirect addressing by HEINZ SCHECHER (1922–1984) 1955) and influenced by the problem of machine translation from algorithmic language into machine language (RUTISHAUSER 1951, BAUER and SAMELSON 1957).

RUTISHAUSER made particular use of the *Plankalkül* of KONRAD ZUSE (1945), an earlier form of an algorithmic language, which was undoubtedly extremely machine-oriented but in certain features of its object structure went beyond the numerically flavored ALGOL 60. In this line of thought, the lists described in 1959 by JOHN MCCARTHY and the lattices described in 1966 by C. ANTONY R. HOARE reached the level of the present day. Other essential contributions to a uniform and fully conceptual system of programming, or, as it was also called, 'the creation of a semantic base', came in 1959 from JULIEN GREEN (*1924) (name generation) and KLAUS SAMELSON (block structure), and in 1965 from NIKLAUS EMIL WIRTH (*1934) (reference concept).

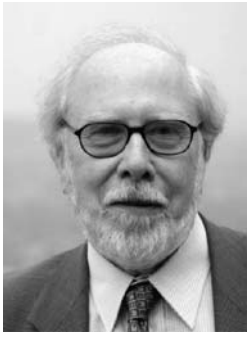


Fig. 169: Niklaus E. Wirth



Fig. 170: Dana S. Scott



Fig. 171: Edsger W. Dijkstra

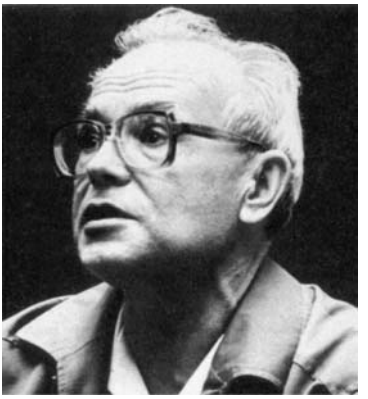


Fig. 172: Andrei P. Ershov

General questions of semantics were dealt with in 1962 by JOHN MCCARTHY, in 1964 by PETER LANDIN, in 1964 by CHRISTOPHER S. STRACHEY (1917–1975), in 1967 by ROBERT W. FLOYD, in 1969 by C. ANTONY R. HOARE, and, beginning in 1962, by ADRIAAN VAN WIJNGAARDEN (1916–1987), whose investigations into operative semantics led to ALGOL 68. Progress towards a fundamental mathematical treatment of semantics was initiated in 1970 by DANA S. SCOTT (*1932) with his fixed point theory, and continued in 1973 by EDSGER WYBE DIJKSTRA (1930–2002) with the theory of predicate transformations and JOHN W. BACKUS with investigations of functional programming. Formal questions of the equivalence of program procedures were first dealt with by YURI J. JANOV in 1956, soon followed by work on optimization by ANDREI PETROVICH ERSHOV (1931–1988). D.C. COOPER (1966), J. DARLINGTON and R.M. BURSTALL (1973), and BAUER and colleagues (1974) all investigated formal methods of program construction. A paper by CARL ADAM PETRI (*1926) (1961) on asynchronous sequential circuits proved to be highly significant for programming: it led to the semantic definition of simultaneous procedures using Petri nets.

Recursion

"*So, naturalists observe, a flea Hath smaller fleas that on him prey, And these have smaller still to bite 'em And so proceed ad infinitum*".

JONATHAN SWIFT, Jonathan

Anybody who has ever propagated their reflection in a Hall of Mirrors knows intuitively what recursion is. The term entered mathematics as a definition principle. In 1890, DAVID HILBERT (1862–1943) stated a recursive definition of a curve that fills a square. The use of recursively defined number functions in the field of mathematical logic led to recursion being explicitly adopted into programming languages (LISP 1959, MCCARTHY), although in 1958 HEINZ RUTISHAUSER's programming instructions for the ERMETH had already made provision for the recursive use of arithmetic instructions. The question of memory distribution was comprehensively dealt with by EDSGER W. DIJKSTRA in 1960. In 1961, ALBERT A. GRAU, and P. LUCAS and HARRY D. HUSKEY all



Fig. 173: Example of a recursion



Fig. 174: Heinz Rutishauser



Fig. 175: Heinz Rutishauser

recognized that the problem of sequentially decomposing a formula can be resolved quite naturally with a recursive treatment. From 1960 the door was open for recursion to become a core principle of informatics.

The revolution that took place between 1935 and 1960 is particularly characterized by the freeing of basic operative structures from rigid process structures, the introduction of recursion into practical programming as a prototypical feature of the universal machine. However, this does have to be seen in the context of the conclusions of ALONZO CHURCH in 1936 and ALAN M. TURING in 1937, namely that some problems are so general that they cannot be solved algorithmically.

The von Neumann Machine

In 1934, KONRAD ZUSE set out from a quite natural idea: a calculating machine for arithmetical operations with a read/write head that was moved about on an extensive arithmetic form by an overhead crane. In this it had a similarity to TURING's and EMIL L. POST's approach – to mechanize each of the steps that a human computer works through. But for all practical purposes ZUSE's vision of the form ended with rewritable memory and his calculation procedures with loop structures.

The simplest fundamental structure, the processing loop, is quite clearly discernable in the very first programs for the EDVAC, which were written in 1945 and 1946 in the form of flowcharts. But practical programming had already acquired more sophisticated features from NICHOLAS CONSTANTINE METROPOLIS (1915–1999) in Los Alamos and STANLEY GILL (1926–1975) and DAVID JOHN WHEELER (1927–2004) on the EDSAC.

The reason that J. PRESPER ECKERT and JOHN W. MAUCHLY's EDVAC design nonetheless constituted a universal machine was due to the fact that the program was able to modify itself, i.e. the flowchart could be transformed into a different one at run time. It was explicitly realized by JOHN VON NEUMANN and HERMAN H. GOLDSTINE in 1946, and also by JOHN W. MAUCHLY in 1947, that the EDVAC was a universal machine; rigorous proof of this was subsequently given by a number of logicians, in the first instance probably by HANS HERMES (1912–2003) in 1954. Incidentally, these program modifications generally took the form of changes to the address part of one or more instructions. Modifications to data addresses was an obvious possibility for indexed quantities, and had already been provided for in 1945 by ROBERT E. SEEBER in IBM's SSEC, but the crucial step was the modifi-

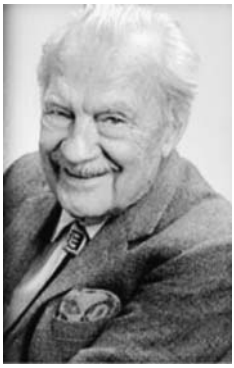


Fig. 176: Nicholas C. Metropolis

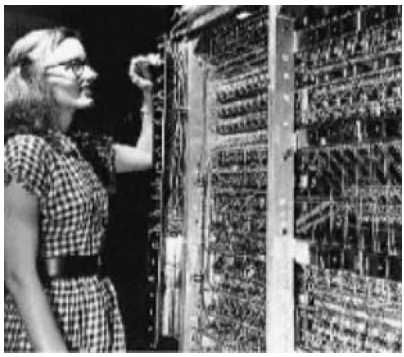


Fig. 177: Von Neumann's MANIAC



Fig. 178: David J. Wheeler



Fig. 179: Hans Hermes

cation of branch destinations and return addresses. This was recognized in particular by HEINZ RUTISHAUSER and was taken into account in the construction of the ERMETH. Back in 1948, by the way, RICHARD F. CLIPPINGER (1913–1997) had already used the adjustable function panels of the ENIAC to store versatile programs.

What had begun merely as a theoretical investigation acquired practical significance when recursion (the general recursive functions mentioned by HERBRAND, GÖDEL and KLEENE) arose almost naturally as a specification technique in the emerging programming languages.

"CHURCH's theorem has sometimes been interpreted pessimistically as a proof that there are absolutely unsolvable problems. But optimistically speaking, it is a rigorous proof that brains are indispensable, and that should be comforting to anyone who hopes that he can solve problems which a moron cannot."

PAUL C. ROSENBLOOM (*1907), 1950

Sequential Formula Translation and Pushdown Mechanisms

One obvious continuation of the computer modifying its own program was for the computer to calculate its own program by translating arithmetical formulae into machine language. This was discussed in 1951 by HEINZ RUTISHAUSER and shortly afterwards also by ANDREI P. ERSHOV. RUTISHAUSER's process has been described as a 'dancing procession over the bracket mountains'. In 1956, ALAN J. PERLIS restricted himself to languages for which the translation process could be controlled by comparing two immediately consecutive characters.

In 1955, FRIEDRICH L. BAUER and KLAUS SAMELSON introduced pushdown storage systems, known as 'stacks', for the sequential transla-



Fig. 180: Friedrich L. Bauer



Fig. 181: Klaus Samelson

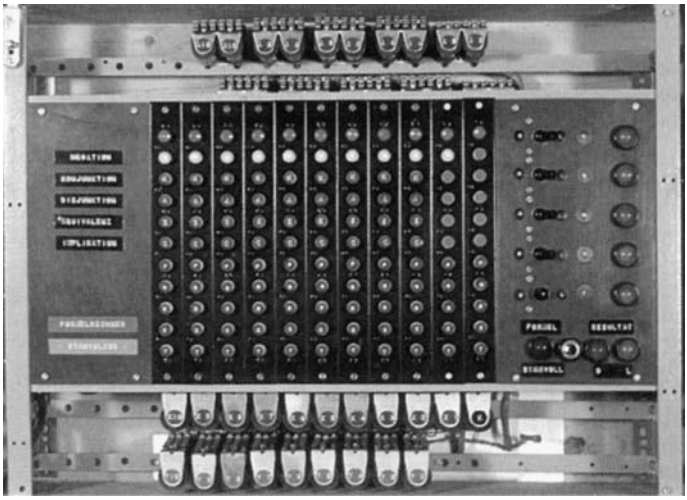


Fig. 182: Formula-driven relay device STANISLAUS by F.L. Bauer



Fig. 183: Noam Chomsky

tion of formulae, based on STANISLAUS, a formula-driven relay device constructed in 1950 for calculations in propositional calculus.

Subsequently, this pushdown storage, originally conceived in 1957 for a patent application as hardware, was used systematically as a programmed data structure for translating algorithmic language to machine language. This idea then influenced the design of the algorithmic languages ALGOL 58 and ALGOL 60. In collaboration with RUTISHAUSER they showed how general recursion can be managed by stacking the calls.

Patent considerations delayed publication, enabling EDSGER W. DIJKSTRA in 1962 to independently demonstrate the practical handling of recursion in programming languages by means of programmed stacks.

In 1956, ERSHOV also used programmed stacks for his 'programming program', and since 1957 they have also been used by ALLEN NEWELL (1927–2002) and JOHN CLIFFORD SHAW (*1922) to prove mathematical theorems. In 1962, a cooperation between the algebraist MARCEL PAUL SCHÜTZENBERGER and the linguist NOAM CHOMSKY (*1928) introduced the term 'pushdown automaton', which proved suitable for characterizing context-free languages.

High-Level Programming Languages and Algorithms

The face of informatics towards the end of the period 1935–1960 is characterized by the use of high-level programming languages. The first such high-level language was ZUSE's *Plankalkül* in 1945, but this was not designed to be compilable. The notation used by ALEKSEI A. LJAPUNOV and YURI J. JANOV in 1957 exhibited the same disadvantage. On the other hand the first compilers, such as the one written by GRACE MURRAY HOPPER (1906–1992), did not look like mathematical formulae. The changeover to problem-oriented programming, which had begun in 1955 with JOHN W. BACKUS's still extremely machine-oriented FORTRAN, was successfully achieved with ALGOL 58 and ALGOL 60,



Fig. 184: Grace Hopper



Fig. 185: Heinz Zemanek

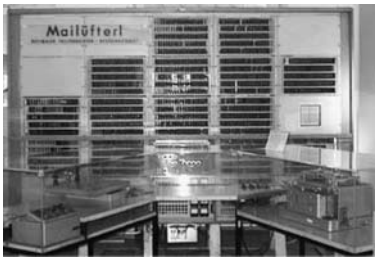


Fig. 186: Zemanek's Mailüfterl

which were created with some West European involvement. Initially the focus was on syntactical aspects, motivated by the problems of automatic translation into machine language; semantic problems were still generally rather neglected – the notable exception being JOHN MCCARTHY's LISP (from 1958). In their 'Vienna Definition Language' HEINZ ZEMANEK (*1920) and his colleagues stressed the practical necessity of having formal descriptions of programming languages.

But the first successful attempt to define the semantics of a formal language had already been made in 1931 by ALFRED (TAJTELBAUM) TARSKI (1901–1983).

The use of the term 'algorithm' began to flourish after 1950: it was no coincidence that ALGOL claimed to be an 'algorithmic language'. In 1951, ANDREI ANDREEVICH MARKOV wrote his book *Theory of Algorithms*.

In 1954, RUTISHAUSER spoke of a 'quotient difference algorithm'. After 1960, DONALD ERVIN KNUTH (*1938) began the systematic investigation of fundamental non-numerical algorithms. And SHMUEL WINOGRAD (*1936) in particular addressed problems of efficiency.

In 1965, PETER J. LANDIN suggested using lambda calculus as a basis for the semantics of programming languages; since 1964, PETER J. LANDIN, CHRISTOPHER S. STRACHEY and DANA S. SCOTT had been explaining recursive programs according to the classic theory. 1963 then saw the counterpart to the Turing machine, namely a true von Neumann machine in the form of the register machine of J.C. SHEPHERDSON and HOWARD EWING STURGIS (*1936), following a preliminary version created in 1959 by HEINZ KAPHENGST (*1932).



Fig. 187: Alfred Tarski



Fig. 188: Andrei A. Markov

In 1964 ABRAHAM ROBINSON (1900–1974) and CALVIN C. ELGOT (1922–1980) studied the register machine from the point of view of programming-language semantics. Other attempts to define universal machines that were close to the prevailing methods of programming were made in 1958 by ANDREI P. ERSHOV (operator algorithms) and ROSZA PÉTER (graph schemata).

The complexity theory of algorithms, however, did not really get off the ground until 1971 with the result of STEPHEN A. COOK (*1939),

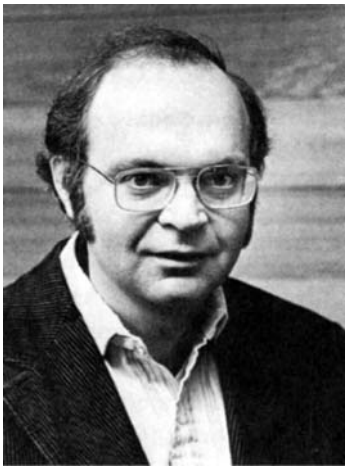


Fig. 189: Donald E. Knuth



Fig. 190: Abraham Robinson

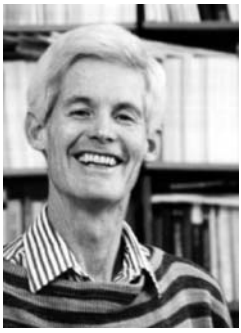


Fig. 191: Stephen A. Cook

namely that the feasibility problem for statement forms – for which the run time of the classic, exhaustive solution algorithm increases exponentially with the number of variables – is non-deterministic polynomial (NP) complete.

Machine-Oriented Programming

From around 1960, machine-dependent programming retreated further and further into the background, with just one exception: questions relating to storage allocation and addressing. Indirect addressing, invented in 1954 by HEINZ SCHECHER, offered a very convenient way of mapping



Fig. 192: Allen Newell



Fig. 193: Herbert A. Simon

complex ‘compound statements’, as used in ZUSE’s *Plankalkül*, onto a homogenous, serially numbered memory. We find other early uses of references (links and pointers) in 1953 by HANS PETER LUHN and in 1956 by ALLEN NEWELL, JOHN C. SHAW and HERBERT ALEXANDER SIMON (1916–2001) in IPL II. Attempts to express these features of machine-oriented address handling using suitable higher-level concepts that were better adapted to machine-independent programming led to the abstract programming-language idea of ‘references’, which were studied in 1959 by JULIEN GREEN and others, including in 1965 NIKLAUS E. WIRTH. The associated programming style of ‘organized memory’ (to which was applied the catchword ‘procedural’) that was made necessary by the technological reality of memory that could be overwritten and which, because of its limited availability at that time, had to be overwritten, remained characteristic for programming throughout this period and for quite some time thereafter.

Particular contributions to the use of references in implementing lists and trees were made by JOHN MCCARTHY in 1960 with ‘garbage collection’, ALAN J. PERLIS and ANATOL WOLF HOLT VON HOPFENBERG in 1960 with threaded trees, DOUGLAS TAYLOR ROSS in 1961 with multiple references, and JOSEPH WEIZENBAUM (1923–2008) in 1963 with reference counters.



Fig. 194: Joseph Weizenbaum

The End of Analog Computers

There were also attempts to make the rigid circuits of analog computers more flexible; there were even some configurations created with digital arithmetic units (hybrid computers). But digital and analog get along like fire and water, and fire has a very dehydrating effect: by 1960 it was becoming evident that analog computers were too expensive, error-prone and imprecise to be able to genuinely compete with digital computers. Analog technology was pushed aside into just a few application niches. This meant a complete change of course for control and regulation engineering, which had previously been almost exclusively geared to analog technology. Dual mode control methods, which had been investigated theoretically since 1950, were already moving in the right direction, and the arrival of discrete sampling techniques then consummated the changeover to digital technology.

Systems

There is an Austrian saying that defines a 'system' as an entity which is difficult to understand if you begin by looking at it from only one side, and that this is true whichever side you choose. Right up to the end

of this period, technical limitations made it possible to perform complicated operations only on simple objects, and to perform only the very simplest of operations on complex objects such as a whole page of text. But 1960 or thereabouts saw the first successful 'applications', such as packages for placing orders or making reservations, which needed large systems performing complex calculations on complicated objects, and they were soon appearing from all directions. Here we find two clearly identifiable fields of endeavor in the internal relationship between informatics and automation: the beginnings of recognition systems, and the beginnings of computer networks.

Recognition Systems

> "Sinnlos herumzuraten ist für Automaten ebenso wertlos wie für Menschen"[33].
>
> KARL STEINBUCH, 1959

Recognition systems have the task of locating specific substructures within an entity with a highly complex overall structure. The simplest examples of recognition tasks from the field of classical mathematics are: finding common subexpressions within an arithmetical expression (more generally, linking nodes representing identical subexpressions), determining an antiderivative for a complex function of one variable (more generally, inverting a complicated mapping) and factorizing polynomials in several indeterminates. Although an algorithm for this last task was stated as long ago as 1792, by FRIEDRICH THEODOR VON SCHUBERT (1758–1825).

In 1952 HARRY GEORGE KAHRIMANIAN had already written a program for mechanical differentiation; in 1961 JAMES R. SLAGLE (*1934) then had remarkable success with formulaic integration.

Formula translation was also a recognition problem of the same kind, namely the reconstruction of an intentionally discarded tree structure. The location of common subexpressions was a task that translator writers also addressed, for instance ANDREI P. ERSHOV in 1958.

Another general recognition problem that is significant for any algebraic structure is the unification problem for two terms. This was initially investigated in relation to logic by JACQUES HERBRAND in 1930, but it first achieved special significance in 1963 in association with the

[33] Futile speculation is just as worthless for machines as it is for people.

resolvent method of JOHN ALAN ROBINSON (*1930), which was itself based on a method described by DAG PRAWITZ in 1960.

Other recognition problems are found on the edge of mathematics, for instance for chess positions. In 1947, ALAN M. TURING and DAVID G. CHAMPERNOWNE (1912–2000) were already working on mechanized chess. In 1949, SHANNON introduced a rational approach using the von Neumann–Morgenstern minimax procedure. The first methodological restriction of the search space ('forward pruning') was achieved in 1956 by ALEX BERNSTEIN. In 1958, ALLEN NEWELL, JOHN C. SHAW and HERBERT A. SIMON produced a simplification of this method using a preliminary form of α–β pruning, which was further improved by T.P. HART and D.J. EDWARDS in 1961.[34]

KONRAD ZUSE had already made use of the rules governing permissible chess moves as an example for his *Plankalkül* in 1945. One of his predecessors was L. TORRES Y QUEVEDO who, in around 1900, built a machine to play the endgame 'king against king and rook'. Other endgame problems were addressed by D.G. PRINZ in 1952.

Similar problems presented themselves for other games, such as checkers, programmed in 1952 by CHRISTOPHER S. STRACHEY, with an improved version in 1959 by ARTHUR LEE SAMUEL (1901–1990), or Go, programmed by HORST REMUS (1928–2007) in 1961.

Such tasks, which, when considered from a purely combinatorial point of view, involve exponentially expanding cost, had the effect of introducing the idea of 'complexity'. ALLEN NEWELL studied this in 1958 using chess problems as his example. There are now numerous non-deterministic-polynomial complete problems – coloring problems, Hamiltonians, rucksack problems, traveling salesman problems – for which no method is known that does not call for exponentially increasing cost. One example from pure mathematics is reducing a natural number to its prime factors, a task that was typically tested using Mersenne primes. In 1952, with the help of SWAC, RAPHAEL M. ROBINSON (1911–1995) was able to state the 13^{th} to 17^{th} Mersenne primes. In 1957 HANS RIESEL proved that $2^{3217} - 1$ is prime, in 1961 ALEXANDER HURWITZ and JOHN L. SELFRIDGE did the same for $2^{4253} - 1$ and $2^{4423} - 1$. In 2006, $2^{39402457} - 1$ was the largest known Mersenne prime. It is often the case that, as here, inverting a task is considerably more

[34] SIMON was excessively optimistic in 1958 when he predicted that, by 1968, a program would be world chess champion. ZUSE, who, in 1938, prophesied that a program would be world chess champion in 1988, was rather nearer the mark.



Fig. 195: W. Grey Walter



Fig. 196: Walter's Elsie



Fig. 197: W. Ross Ashby

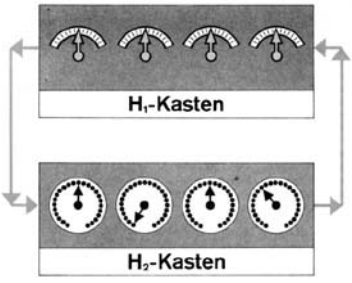


Fig. 198: Homeostat by Ashby

complex than the task itself. Since 1974 such 'trapdoors' have been significant in the field of cryptology.

In the late 1950s, systems using heuristic methods were frequently said to be capable of 'learning' and there were also 'cybernetic learning models': in 1948 W. GREY WALTER (1910–1977) introduced the electromechanical turtle, in 1950 SHANNON introduced a mazesolving mouse, and in 1952 there was the 'homeostat' of W. ROSS ASHBY (1903–1972). The conglomeration of theory and speculation that became 'cybernetics' can be attributed to the bizarre genius NORBERT WIENER.

The recognition problem is even more difficult in the context of machine translation of natural languages. The first attempts at this, which



Fig. 199: Norbert Wiener



Fig. 200: Yoshua Bar-Hillel

did not, however, pave the way for a machine, were carried out by P.P. SMIRNOV-TROYANSKIY in 1933. In 1946 WARREN WEAVER (1894–1978) and ANDREW D. BOOTH were already discussing the possibility of carrying out word-for-word translations by machine. YOSHUA BAR-HILLEL (1915–1975) was particularly concerned with the connections between logic and linguistics. But, after 1960, disproportionate optimism led to setbacks. No one had considered that translation is more than a recognition problem dealing only with predefined structures ('pre-existing images'), it is namely also a question of understanding, which goes beyond adapting a system to the reactions of its environment and calls for pre-existing images to be created from nowhere. We have the same difficulty when we are asked to translate the spoken word into written text or into action. This problem is not restricted to language in the linguistic sense: understanding images and situations calls for more than mere pattern matching, but in 1960 there was barely any hint of this.

Cryptanalysis

In cryptanalysis the problem is to reconstruct the original structure of a verbal message from whatever fragments survive the encryption process, no matter how minute. The British scientists MAXWELL HERMAN ALEXANDER NEWMAN (1897–1984), ALAN M. TURING and WILLIAM GORDON WELCHMAN (1906–1985) made crucial progress in this area when they broke the ciphers used by the German armed



Fig. 201: Maxwell H.A. Newman



Fig. 202: Thomas H. Flowers

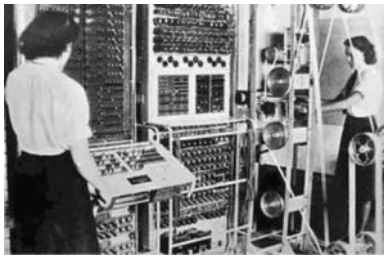


Fig. 203: Colossus Mark II

forces. These cryptanalytic tasks not only led to the construction of the first loop-driven electronic evaluation devices, the COLOSSUS in 1944 by THOMAS H. FLOWERS (1905–1998) and ALLEN W.M. COOMBS (1911–1995), and the American machines GOLDBERG in 1947 and DEMON in 1948, but also to HARVEST, the cryptanalytic sister machine for IBM's gigantic (for the time) STRETCH in the late 1950s and BOGART, the counterpart of the CDC 1604 (1964) – finally culminating in the CRAY 1 of SEYMOUR R. CRAY (1925–1996) in 1976. But these machines are taking us outside the period we have been considering.

This also applies to the asymmetric encryption (public-key) method that was introduced in 1976 by WHITFIELD DIFFIE and MARTIN E. HELLMAN. This had been invented back in 1970 by JAMES H. ELLIS



Fig. 204: Seymour R. Cray



Fig. 205: CRAY 1 computer

(1924–1997), but its publication was prevented by the British Official Secrets Act, as was revealed in 1998 by the Communications-Electronics Security Group of the British Government Communications Headquarters (GCHQ). In 1973, CLIFFORD COCKS found the multiplication of sufficiently large prime numbers to be a practically non-reversible operation (trapdoor) that was suitable for the construction of public keys.

In 1977, DES ('Data Encryption Standard') was published. The RSA (RONALD R. RIVEST, ADI SHAMIR, LEONARD M. ADLEMAN) method (1978, US Patent 4,405,829; 20th Sept. 1983) is based on the widely accepted conjecture that 'raising to a fixed power modulo q'creates a sufficiently safe trapdoor. Meanwhile, DES and RSA can no longer be considered safe. Around 2000, after the weakness of DES and RSA became obvious, AES ('Advanced Encryption Standard') appeared. The top candidate is at present RIJNDAEL by JOAN DAEMEN (*1965) and VINCENT RIJMEN (*1970), valid since 26th May 2002.

Effects of Cryptanalysis

One side-effect of the Second World War was the considerable progress made by cryptanalysis, the deciphering of cryptograms, and this was also reflected in the construction of large-scale decryption machines and subsequently in the creation of cryptanalytic computer programs. With the introduction of asymmetric ciphers, cryptological considerations gave mathematical complexity theory a previously overlooked practical significance. Actually making use of it does, however, remain problematical, because the theory of this complex, and therefore difficult, complexity theory delivers almost exclusively upper bounds for the complexity

of a problem, hardly ever a reassuring lower bound. For the moment, therefore, asymmetric ciphers are still a *va banque* game. Nevertheless, public keys have been in wide use since 1990; but for highly sensitive communication channels, perfectly secure encryption using one-time random keys is still needed.

Cryptology, which had already begun to fascinate SHANNON before 1945 and which was the central topic of his celebrated book on information theory The 'Mathematical Theory of Communication' in 1949, now provided mathematics with a second impetus: in 1977, developing ideas published by ANDREI N. KOLMOGOROV (1963), RAY J. SOLOMONOFF (1964) and PER MARTIN-LÖF (1966), GREGORY J. CHAITIN (*1947) published an essay on 'Algorithmic Information Theory' which paid more thorough attention to the aspect of computability.

Whereas SHANNON's information theory is considerably more stochastic in nature, in the appendix *Perfekte Sicherheit und praktische Sicherheit* to his book *Entzifferte Geheimnisse* published in 1995, FRIEDRICH L. BAUER provides an axiomatic basis with the help of three axioms for $H_{\mathcal{Y}}(\mathcal{X})$, which is 'the uncertaintly of $\mathcal{X}$, if $\mathcal{Y}$ is known':

(1)	$0 \leq H_{\mathcal{Y}}(\mathcal{X})$	("Uncertainty is nonnegative.")
For	$0 = H_{\mathcal{Y}}(\mathcal{X})$	we say "$\mathcal{Y}$ uniquely determines $\mathcal{X}$."
(2)	$H_{\mathcal{Y}\cup\mathcal{Z}}(\mathcal{X}) \leq H_{\mathcal{Z}}(\mathcal{X})$	("Uncertainty decreases, if more is known.")
For	$H_{\mathcal{Y}\cup\mathcal{Z}}(\mathcal{X}) = H_{\mathcal{Z}}(\mathcal{X})$	we say "$\mathcal{Y}$ says nothing about $\mathcal{X}$."

The critical axoim dealing with additivity is

(3) $H_{\mathcal{Z}}(\mathcal{X} \cup \mathcal{Y}) = H_{\mathcal{Y}\cup\mathcal{Z}}(\mathcal{X}) + H_{\mathcal{Z}}(\mathcal{Y})$

Since, in particular, $H_{\emptyset}(\mathcal{X} \cup \mathcal{Y}) = H_{\mathcal{Y}}(\mathcal{X}) + H_{\emptyset}(\mathcal{Y})$, we can call $H_{\emptyset}$ 'entropy', by analogy with additive thermodynamic entropy.

Networked Systems

Program-controlled machine complexes were already being constructed in the 1920s. For scientific calculations in particular, a number of punched-card machines would be connected together and programed by means of controlpanels. From 1927, LESLIE J. COMRIE in the National Almanac Office, in London was already using punched-card machines for astronomical calculations; from 1928 BENJAMIN DEKALBE



Fig. 206: Fritz-Rudolf Güntsch



Fig. 207: Theodor Fromme

WOOD (1894–1986) at Columbia University was using punched-card machines that had been specially modified to permit register transfers ('Packard').

In 1928–1929, the company Rheinmetall-Eichenauer built a machine complex consisting of a mechanical computer and a card punch. In the mid-1930s banks of punched-card machines were introduced in the USA for cryptanalytic work. Finally, in 1933, WALLACE J. ECKERT at Columbia University combined a number of punched-card machines to form a complex ('mechanical programmer') which was able to read, calculate and write, and possessed a process control that could perform up to 20 steps. During the Second World War this machine was used to calculate numeric tables for fire control and for atomic fission. In 1936, the influential American electrical engineer VANNEVAR E. BUSH built a mechanical process controller for a bank of punched-card machines.

When, in 1944, KONRAD ZUSE began thinking about building a computer system on the basis of the vacuum tube circuits invented by HELMUT T. SCHREYER (1912–1984), he was hoping for a 'minimalist' solution: on the one hand, tubes were expensive and difficult to obtain, and, on the other hand, the speed of electronic circuits was so impressive that it seems reasonable to use just a few switch organ elements and perform as many of the processing steps as possible, one after the other – i.e. to turn his back on the parallelization of addition that had become customary for mechanical computers. ZUSE's 'dwarf machine' of 1958, the Z 22, designed by THEODOR FROMME (1908–



Fig. 208: Carl A. Petri

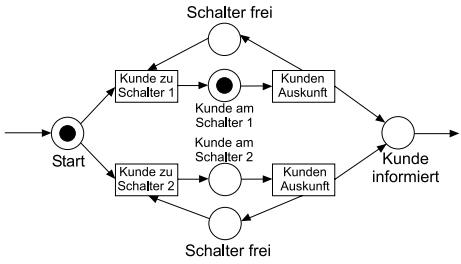


Fig. 209: Petri net

1959) and rationalized by FRITZ-RUDOLF GÜNTSCH (*1925), still exhibited traces of this idea. Subsequently, however, there was a general move to electronic computers with parallel addition and even partially parallel multiplication. This parallel activity and the transport between registers and working memory was synchronized by means of a clock pulse.

At the time there seemed little point in including the input and output devices in the clock synchronization. So the very first electronic computer systems were actually networks of computer systems (each with a central processing unit and working memory) plus peripheral devices, which communicated asynchronously with one another. The organizational problems that were initially encountered prompted CARL A. PETRI to carry out in-depth investigations that led to the descriptive theory of what are now called 'Petri nets'.

On the practical side, at the beginning of the 1960s we initially find operating systems for single-computer systems. These needed to pay particular attention to the problems of memory allocation. The practical necessity of making optimal use of a large computer system led to multiprogram operation and subscriber systems (time-sharing), for which FERNANDO J. CORBATO (*1926) achieved the breakthrough.

In 1965, FREDERICK P. BROOKS JR. (*1931) created a compatible family of computers (IBM System/360).

Machines that would later come to be called 'mid-range systems' opened up new possibilities for applications. In the USA the pioneer C. GORDON BELL (*1934) worked at DEC and developed the VAX;



Fig. 210: Fred Brooks



Fig. 211: IBM 360-65



Fig. 212: Heinz Nixdorf



Fig. 213: Konrad Zuse and Heinz Nixdorf

in Germany the market began with HEINZ NIXDORF (1925–1986). His concept for introducing computer technology to the workplace for the benefit of the users made him the pioneer of the decentralized data processing.

Computer networks introduced new problems. In 1966 LAWRENCE G. ROBERTS (*1937) introduced the method of packet switching for the ARPA network. Far more complicated problems arose with multiple-computer systems and there was a need for corresponding operative theories, which began to be provided in 1968 with the semaphores of EDSGER W. DIJKSTRA. But the single-minded further development of clustered and networked computer systems, right down to failsafe systems, belongs in a later period.

Finally, in 1960, telecommunication technology was just far enough advanced that it was possible to begin introducing digital switching systems that are able to recognize the level of utilization in a network and permit an appropriate reaction.

Digression into Electronic Circuits, Devices and Machines

Although the physicists and electrical engineers in the period 1945 to 1960 who developed basic electronic circuits and built computers with them could not generally be entirely won over to informatics, it is only right that we honor the achievements of those pioneering engineers who labored alongside WILKES, WILLIAMS, AIKEN, STIBITZ, J.P. ECKERT, MAUCHLY and have not all been mentioned:

SAMUEL N. ALEXANDER (1910–1967) ..... SEAC;
J. ALEXANDER ..... ORACLE[35] (Computer);
ROBERT S. BARTON ..... Burroughs B 5000;
C. GORDON BELL (*1934) ..... DEC, VAX;
JULIAN BIGELOW (1914–2003) ..... IAS;
HEINZ BILLING (*1914) ..... G 2;
ANDREW D. BOOTH (*1918) ..... SEC, APEXC;
WERNER BUCHHOLZ (*1922) ..... IBM 701;
JEFFREY CHUAN CHU (*1919) ..... Circuitry design, ORACLE (Computer);
STIG COMET ..... BESK;
ROBERT R. EVERETT (*1921) ..... WHIRLWIND;
THOMAS H. FLOWERS (1905–1998) ..... COLOSSUS;
JERRIER A. HADDAD (*1922) ..... IBM 701;
CUTHBERT C. HURD (1911–1996) ..... Early IBM machines;
DAVID A. HUFFMAN (1925–1999) ..... Circuitry design;
REYNOLD B. JOHNSON (1906–1998) ..... RAMAC 305;
TOM KILBURN (1921–2001) ..... ATLAS;
S.A. LEBEDEV (1902–1974) ..... MESM, BESM;
NIKOLAUS J. LEHMANN (1921–1998) .... Dresden Series: D1–D4;
HERMAN LUKOFF (1923–1979) ..... Circuitry design;
RICHARD P. MEAGHER ..... ORDVAC, ILLIAC;
HEINZ NIXDORF (1925–1986) ..... FAC 820;

[35] ORACLE: Oak Ridge Automatic Computer and Logical Engine.

ROBERT PILOTY (*1924) ..... PERM;
WILLEM L. VAN DER POEL (*1926) ..... ZEBRA;
JAMES H. POMERENE (*1920) ..... IAS, HARVEST;
NATHANIEL ROCHESTER (1919–2001) ..... IBM 701 and 702;
HAROLD D. ROSS (1922–2004) ..... ALU of IBM 701; AMBROS P. SPEISER (1922–2003) ..... ERMETH;
ANTONIN SVOBODA (1907–1980) ..... SAPO, EPOS;
KEITH W. UNCAPHER (1922–2002) ..... JOHNNIAC;
HEINZ ZEMANEK (*1920) ..... MAILÜFTERL.

After 1960: Informatics Begins to Take Shape

By the 1960s, informatics was already fully developed, but it was not yet referred to by that name. It was, however, acquiring more and more independent features and as it did so it became an academic discipline. For example, in 1963 the Technical University of Munich began appointing staff, and this was supported by the German Research Council and sanctioned by the Free State of Bavaria. By 1965 we can observe progress towards independence at a number of locations, such as in the USA at Stanford University and at MIT. As early as 1967, the ACM held a conference in Stony Brook to discuss a 'Computer Science Curriculum'. Encouraged by this, in the autumn of 1967, a number of American and European universities, including the Technical University of Munich began offering university courses: until 1968 the TUM's course was called *Informationsverarbeitung* [information processing].

The years following 1960 were marked by such rapid growth that the sheer volume of events makes it impossible to offer an overview of the principal figures involved, such as formed the basis of previous chapters. It is also inherently difficult to survey such an extremely wide and expanding landscape, and at this remove we lack the perspective that would permit a more dispassionate evaluation.

The following section therefore contains only a summary of the structure of the discipline 'informatics' that has emerged since 1960.

At the edge of this new field, where informatics borders on and sometimes overlaps its neighbors, mathematics on the one hand and electrical engineering on the other, we very soon encounter two distinguishing labels: 'theoretical informatics' and 'technical informatics'. By their very nature neither of these is typical for the content of informatics proper, but just as the arrival of theoretical physics in around 1910 forced the existing subject to adopt the name 'experimental physics' to avoid being misunderstood, so the field of informatics soon had to call itself 'core informatics' or even more narrowly 'practical informatics' in order to get its message across.

This core area of informatics can be subdivided, albeit fluidly, into 'problem-oriented informatics' and 'machine-oriented informatics'. The first typically includes areas such as programming languages, programming methodology, fundamental data structures and algorithms, specification and modeling. On the border we find the implementation of

F.L. Bauer, *Origins and Foundations of Computing*,
DOI 10.1007/978-3-642-02992-9_6,
© Springer-Verlag Berlin Heidelberg 2010

programming languages and of databases as well as graphical systems and man–machine interfaces.

Under machine-oriented informatics we find: system programming, operating systems, microprogramming, and real-time information processing.

Technical informatics is typified by memory structures, processor structures, computer architectures, chip development and also transaction systems and communications networks. This area comes into close contact with semiconductor physics and materials science.

Theoretical informatics includes topics such as automata theory, formal languages, theory of programming, codes, encryption, data compression, computability and complexity theory; it often needs to make considerable excursions into mathematics, particularly algebra, and logic. At one time it was quite normal for lectures on solid-body mechanics to be given alternately by a mathematician and a theoretical physicist: it would certainly be possible for lectures on some of the more profoundly mathematical topics central to the study of informatics to be given alternately by computer scientists and mathematicians.

Applications

Applications of informatics can nowadays be found in almost all scientific disciplines. In Germany, those areas that are not particularly appealing to the people engaged in 'core informatics' and do not directly require their creative contributions are often termed 'hyphen informatics' because of the way such compound words are written. So applications devoted to commerce, business, the law or medicine are classed as *Wirtschafts-Informatik*, *Betriebs-Informatik*, *Rechts-Informatik* or *Medizin-Informatik*, and, sometimes, it must be said, because the knowledge is considered, with some justification, to be rather second-hand, this has a derogatory undertone.

It is a different matter for those applications that fully extend the computer scientist, or are at present perhaps still beyond him or her. Here we can particularly cite applications in the fields of mathematics and logic: hopefully they will serve to strengthen our ties to those related disciplines.

A further area of application that makes the very highest demands is rather vaguely and often misleadingly called 'artificial intelligence'. This incorporates two subject areas that are mutually dependent and

mutually pervasive: the extension of recognition (a purely cognitive process) to understanding (understanding images, language, text, situations), and the extension of simply extracting information (expert systems, knowledge-based conclusion systems) to the formation of abstract contents (conceptual search and comparison processes, for example for legal purposes and for patent literature).

In both cases applications are asked to perform complex operations on complex objects.

Common to both areas is that we need to drill down through an external syntactically comprehensible layer to a lower-lying semantic core. Both are broadly anchored in the most diverse areas of core informatics, from theoretical informatics through practical informatics right down to technical informatics, and both make the most exacting demands. They will retain their fascination into the 21st century, and for a long time to come they will furnish a tie between mathematics and informatics.

'Artificial intelligence', in its historical sense as envisaged by ALAN M. TURING and JOHN VON NEUMANN (who merely hoped that investigating human thought processes would provide ideas and insights applicable to the construction of ideal computers), was turned on its head in the 1960s by epigones who made exaggerated prognoses of a 'general problem solver' – that simple methods would be sufficient to resolve any desired problem. At present, realistic views, for example about the quality of expert systems, seem justified.

Constructive Logic – 'Proofs as Programs', 'CIP'

In particular, we can expect the latest developments in programming techniques to affect the self-image of mathematical logic: "*... dass durch die Forschung auf dem Gebiet des maschinellen Beweisens ein weiteres Teilgebiet der Mathematik, nämlich die Logik, auf dem Weg ist, eine angewandte Wissenschaft zu werden*"[38] (HENNING BERGMANN, ANDREAS NOLL 1977).

Predicate logic goes somewhat further than pure propositional calculus by including statements such as (multiple-place) predicates over a given range[39], and typically also the quantifiers $\forall$ (*for all ... such*

[38] ... in that research into the area of machine-proof means that yet another part of mathematics, namely logic, is on its way to becoming an applied science.

[39] For example, over the integers $\mathbb{Z}$ the two-place predicate $x < y$ for variables $x, y \in \mathbb{Z}$.

that...) and $\exists$ (*there exists ... such that...*). One variety that is of significance for informatics is 'constructive logic' (Brouwer–Kolmogorov–Heyting logic, in older publications sometimes also called 'intuitionistic logic'), which does not recognize the classic rule *tertium non datur* and postulates that absurdity $\perp$ is unprovable.

By the 1970s, it was becoming time for mathematical logic, particularly constructive logic, to free itself from the constraints of a purely theoretical science. Following HASKELL B. CURRY, WILLIAM A. HOWARD and MICHAEL DUMMET worked on the generation of programs from proofs in constructive logic. Also during the 1970s, however, the Munich school under FRIEDRICH L. BAUER began with practical tests of the use of program transformations to construct algorithms from a constructive ('intuitionistic') definition of the task (the project was called 'Computer-Aided, Intuition-Guided Programming').

These two separate approaches proved to be closely related: a logical proof according to verifiable rules can be understood as a description of a series of program transformations according to the same rules. And, conversely, such a sequence of program transformations constitutes a proof.

Programming Languages

The mammoth enterprise of ADRIAAN VAN WIJNGAARDEN, namely ALGOL 68 (1965–1968), failed to achieve its objective, while the simpler attempt PASCAL (1971) by NIKLAUS E. WIRTH was successful. PASCAL was followed by C++ and by a host of dialects.

Meanwhile, off the beaten track, the expression-oriented APL (1972) of KENNETH EUGENE IVERSON (1920–2004) and ADIN FALKOFF (*1921) and the much less expressive BASIC of JOHN GEORGE KEMENY (1926–1992) both found their disciples. Special languages appeared for specific areas of application, such as VICTOR HUSE YNGVE's (*1920) COMIT in 1963 for processing character strings, and DAVID JACK FARBER's (*1935) SNOBOL in 1964.

SIMULA, developed in 1965–1967 by OLE-JOHAN DAHL (1931–2002), BJØRN MYHRHAUG and KRISTEN NYGAARD, initiated a progression from ALGOL towards operative-active object ('classes'), which was continued with WIRTH's MODULA in 1977 and ALAN KAY's SMALLTALK from 1972. Since 1975 the development of 'object-oriented programming languages' has coincided with the theoretical development of 'abstract data structures'.



Fig. 214: Kenneth E.Iverson

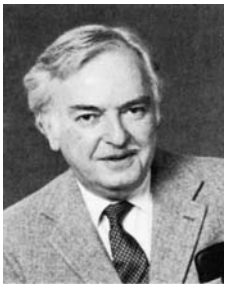


Fig. 215: John G. Kemeny



Fig. 216: Kristen Nygaard



Fig. 217: Ole-Johan Dahl

From 1971, JOHN W. BACKUS set off in a different direction, that of 'functional programming languages' (reduction languages).

Finally, in 1973, working on the basis of the resolvent method of JOHN ALAN ROBINSON and seizing on a notion of P.J. HAYES ("computation and deduction are very nearly the same"), ROBERT A. KOWALSKI (*1941) introduced ALAIN COLMERAUER's (*1941) 1972 version of 'the programming language for predicate logic specifications', namely PROLOG.

Programming languages are the linguistic instruments of informatics. ALGOL 60 influenced many of the subsequent developments in the area of programming languages, but was itself unable to succeed.

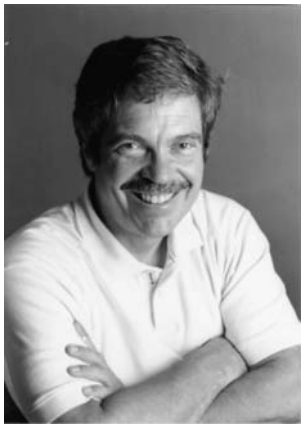


Fig. 218: Alan Kay

Software Engineering

In the early days, when doing arithmetic 'by hand' was still the rule and the opportunity to use a calculating machine was a real stroke of luck, there were already calculation forms (which had influenced KONRAD ZUSE while he was a civil engineering student) but at that time nobody found it necessary to speak of 'software'. By the time large-scale program-controlled machines had taken over, many things had altered: the inefficient use of computers, which would have been unthinkable earlier, was no longer so impossible, and inadequately trained, quickly hired programmers were frequently unable to cope with the complicated requirements of programming work. The transition from machine-oriented programming to the more convenient facilities offered by programming languages was urgently needed for reasons of economy, but this very convenience did sometimes lead to negligence and carelessness, and it was therefore hardly surprising that a number of large-scale projects were sloppily carried out. This became particularly critical in the USA when the affected projects were critical to national security, and therefore for good and sufficient reason – in the face of alarming delays to a particular project – the government asked the *NATO Science Committee* to think of a solution.

In the autumn of 1967 the *NATO Science Committee* therefore set up a study group on computer science, including noted scientists from the USA and a number of other countries. Their stated task was "assessing

the entire field of Computer Science, and in particular elaborating the suggestions of the Science Committee". This was the cautious official formulation; details of the actual task were conveyed only verbally. But the discussions that then took place towards the end of 1967 were held in plain English, and the recognized weaknesses were discussed openly. In the course of these discussions it was pointed out that members of established engineering disciplines would have known how to successfully counter such undesirable developments, whereupon one participant (F.L. BAUER) summarized the committee's opinion in the provocative form "What we need is Software Engineering" – the provocation being that, at the time, informatics was not considered to be one of the engineering sciences.

FRIEDRICH L. BAUER was then asked to organise an international conference. After meticulous preparation of the agenda, a conference of 60 participants was subsequently held from 7th to 11th October 1968 in Garmisch. The candid conference report did not fail to have a salutary effect on the development of informatics in the NATO countries and beyond. Nowadays it is taken for granted in the software industry that whatever tools are available should be used properly (with *ingenium*).

> *"The computer, one of the greatest inventions of engineers, has to go the complete way of engineering to its end" (Friedrich L. Bauer).*[40]

The Transition to Minicomputers and Highest-Performance Computers

Other application areas that make more than trivial demands on informatics include simulation, application-oriented program design, technical data protection and data security – so many that it is out of the question to list all the possible names.

We shall list just a few markers to illustrate the advances made in computer systems, the material instruments of informatics.

In 1960, KENNETH HARRY OLSEN (*1926) introduced a new class of powerful minicomputers, the PDP 1.

[40] Proc. IFIP Congress 71, p. 538.

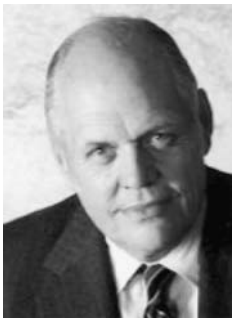


Fig. 219: Ken Olsen



Fig. 220: PDP8



Fig. 221: Gene M. Amdahl



Fig. 222: Ted Hoff

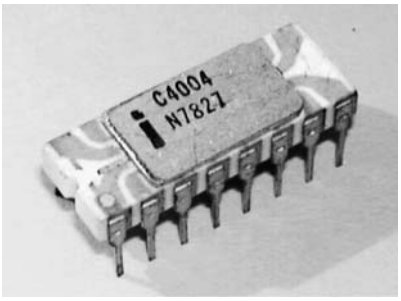


Fig. 223: Microprocessor Intel 4004

In the early 1970s, SEYMOUR R. CRAY and GENE M. AMDAHL (*1922) tried a new approach to mainframe computers, as WESLEY A. CLARK (*1927) did for workstations.

Finally, in 1969, MARCIAN EDWARD HOFF (*1937) contrived to accommodate an entire microprocessor on a single chip. In the 1970s, SEYMOUR R. CRAY designed the first highly parallelized supercomputer, CRAY-1, originally for cryptanalytic purposes, followed by CRAY Y-MD. One of the latest models, CRAY T3E-1200E (1998) achieved 2.4 teraflops. On the mathematical side, parallelization is done by 'domain-decomposition' (CHRISTOPH ZENGER).

Conclusion: Informatics and Microelectronics Are Mutually Dependent

After 1965 informatics experienced a previously undreamt-of upsurge. The numbers of researchers, books, journals and scientific papers increased tenfold about every ten years. The mushrooming software industry, drawing in informatics graduates like moths to the flame, began to assert itself alongside the previously dominating hardware manufacturers. The microminiaturization that began in 1960 and led to microelectronics also made processing power and memory capacity cheaper by a factor of ten every ten years. Large mainframe systems right up to supercomputers were still needed for large-scale research. While their price remained roughly the same they too grew in power every ten years by a factor of about ten, but numerically they were and still are becoming more and more insignificant: it is the prevailing 'computers-for-everyone' that are making the profit. They support a blossoming industry – and by 1990 they were already more widely disseminated than sliderules and log tables had ever been.

Microelectronics began in 1958 with the experiments of JACK ST. CLAIR KILBY (1923–2005) and ROBERT N. NOYCE (1927–1990), aimed at placing an entire circuit including its electrical connections and transistor functions on the surface of a single-crystal silicon wafer, which would then replace a number of individual interconnected silicon transistors. In 1960 it was possible to integrate ten transistor functions. The gradually developed manufacturing process, promoted particularly by GORDON E. MOORE[39] (*1929) and JEAN A. HOERNI (1924–1997), today incorporates alternating sequences of several hundred marking, doping, vapor deposition and etching processes; it calls for superclean rooms containing fewer than three dust particles per liter of air, and works in geometrical dimensions close to the wavelength of light. The necessary investment costs for such a process are correspondingly gargantuan, but the end product, the 'chips' – (rectangular) wafers with a surface of from several square millimeters to a few square centimeters – which are made a few hundred at a time on a (circular) slice of

[39] In 1965 MOORE also formulated the empirical law that the number of transistors on a chip doubles every two years. Experts are of the opinion that this will continue to hold true until 2015.

F.L. Bauer, *Origins and Foundations of Computing*,
DOI 10.1007/978-3-642-02992-9_7,
© Springer-Verlag Berlin Heidelberg 2010



Fig. 224: Jack Kilby

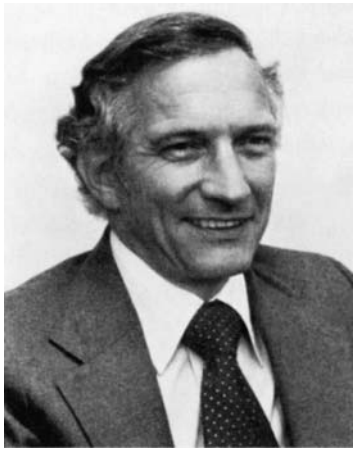


Fig. 225: Robert N. Noyce

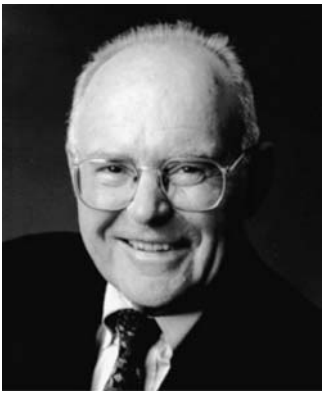


Fig. 226: Gordon E. Moore



Fig. 227: Jean A. Hoerni

silicon up to 20 cm across, are cheap because they are manufactured in such enormous quantities. Depending on the chips' size, complexity and marketability they are now sold for prices between a few cents (for use in cheap watches) and a few euros (for megabit memory chips.)

Naturally you need more than a processor chip and a memory chip to build a powerful computer. Simply mounting and bonding a chip costs money. A small computer such as was sold in 1990 needs at least a keyboard and a monitor, costing (at the time) 150 DM and 200 DM re-

spectively. But the (say) 16,000 processors required for a supercomputer will cost less than a complete midrange single-processor machine would have cost in 1960, with heat dissipation and cooling costs of the same order of magnitude.

With the cost of hardware exhibiting such a pattern, software is acquiring an increasingly dominating role. It is estimated that, since 1990, roughly 80% of the cost of purchasing and operating a computer system is down to the software. Unlike hardware, software is intangible and therefore has no material value, but it can be produced cheaply and its intellectual value is extremely high.

The wide-reaching consequences of the conflict with traditional ideas brought about by this situation are only gradually being revealed. In 1990 it was still an exaggeration to say that for the first time in human history software, the 'manifestation of the volatile spirit' (*formatio animi fugacis*), had become a commodity, a commercial asset: by 2010 this may well be apposite. Not only computer scientists should be pleased by this, the entire spectrum of mathematical science is experiencing an unparalleled upvaluation. The supercilious sneers of socialites who take pride in saying "I was never any good at math" can, in future, be countered with "if you had paid proper attention to math you might have made something of yourself". Sociologically, the dominance of rational thought in high-tech software is in sharp contrast to the all-pervading irrational, 'esoteric'mentality of the last quarter of the 20th century. But this is a wide field that is somewhat remote from our current considerations.

More immediate problems are arising from the new situation in patent law and in the relationship between classic (hardware) engineering and modern (software) engineering.

For a long time, purely intellectual achievements have had a hard time with patents. "A mere mathematical algorithm"[40] was a malicious expression that was considered an eminently suitable reason for rejecting a patent application. At present, intellectual achievements are protected in Germany only by copyright. That this is at least impractical is demonstrated by specific examples of software being considered patentable as soon as it is housed in circuits – an absurd rule requiring the compulsory particularization of an abstract, more rigorously comprehensible conceptual invention. ZEMANEK, in his *Geschichte der Schaltalgebra*, 1989, wrote: "*Bis heute haben die zuständigen Stellen nicht begriffen, daß die*

[40] US Federal Circuit Court criteria from 1990.

Ordnungskraft des Patentwesens, vor allem durch die Reduktion gefundener Neuerungen auf die klare Fassung von Patentansprüchen, für die Softwaretechnik nicht weniger nützlich wäre als für die Hardwaretechnik and daß daher – in Überwindung aller juridischen Schwierigkeiten – das Konstruktive am Software Engineering von Beginn an den gleichen Schutz hätte erhalten sollen wie jedes andere Feld der Ingenieur-Innovation.[41]"

The European Patent Office is more receptive, and it is foreseeable that software, even when described as such, could be patentable. In the USA this development has already progressed further. In 1988 two mathematical algorithms were patented for the first time, including one for NARENDRA KARMAKAR (*1957) (US Patent No. 4744028) for a linear-programming algorithm that is vastly more efficient than previously used algorithms. Although this development is most pleasing it does provoke 'pure' mathematicians and logicians to envy. "While mathematics sinks into poverty, a few mathematicians are making a fortune", warned JON BARWISE.

But perhaps it is true that more mathematicians yearn to partake of the anticipated benefits. There may be some justification in the concern that the traditionally wide-open field of mathematics might become secretive, but this will not be able to hold back the course of events. It will be interesting to see just where this poverty-stricken mathematics ends up.

***

The relationship between hardware engineers and software engineers is essentially a generation problem. Even the most conservative of engineers no longer questions the fact that informatics is an engineering science.[42] The classic areas of telecommunication technology, electrical engineering and control engineering are past their heyday, and research in their fields is being squeezed out by the materials sciences (charac-

[41] To this day the relevant authorities have not grasped that the regulatory power of patents would be no less valuable for software technology than it is for hardware, particularly as it would reduce the number of new discoveries and provide for clear statements of patent claims, and that for this reason – overcoming all juridical difficulties – the constructive element in software engineering should, from the very beginning, have received the same protection that is afforded every other kind of engineering innovation.

[42] This was demonstrated by readers' letters in response to an editorial I wrote for an informatics journal, asking whether Informatics and Information Technology represented a contradiction. ("*Informatik oder Informationstechnik – Ein Gegensatz?*" in *Informatik-Spektrum* (1988) 11:231-232.)

terized by physics) on the one hand and informatics (characterised by mathematics) on the other, visibly reducing their independent base. We must wait and see whether they will be able to maintain their positions as bridges. So far, newly founded technical faculties that do not have to carry around all the ballast of classical engineering disciplines have fared rather better.

But, to get back to informatics and microelectronics: their relationship is one of mutual interdependence. Without microelectronics, many of the things promised by informatics (leaving aside some of the exaggerated empty promises of 'artificial intelligence') would be impossible. But without informatics the most highly developed chip is entirely valueless. Chips with fixed inbuilt programs can be used only for the most rigid of tasks. It is the colossal flexibility afforded by the programmability of programmable chips that is the key to technological progress. And this can blossom only in the interplay between informatics and microelectronics.

The author would like to thank HEINZ ZEMANEK, HORST REMUS, CHRISTOPH ZENGER *and* MICHAEL LUNDGREEN *for their help with procuring biographical information.*

Index

Adleman, Leonard, 106
Aiken, Howard Hathaway, 17, 19, 69, 80, 81, 111
Al Chwarizmi, 86
Al Qalqashandi, Ahmad, 24
Albert, A. Adrian, 51
Alberti, Leon Battista, 24, 50
Alexander, J., 111
Alexander, Samuel N., 111
Amdahl, Gene M., 120
Argenti, Giovanni Battista, 25
Argenti, Matteo, 24
Arzberger, F., 11
Ashby, W. Ross, 103
Atanasoff, John Vincent, 79
Avery, Harold T., 44

Babbage, Charles, 13, 14, 16, 25, 35, 36, 65, 67, 87
Babbage, Henry Prevost, 67
Backus, John Warner, 84, 85, 89, 95, 117
Bacon, Francis, 33–35, 83
Bain, Alexander, 32
Baldwin, Frank Stephen, 10, 12, 44
Bar-Hillel, Yoshua, 104
Barton, Robert S., 111
Barwise, Jon, 126
Baudot, Emile, 26, 33–35, 53, 83
Bauer, Friedrich Ludwig, 6, 82, 88, 89, 93, 107, 116, 119
Baum, Michael, 42
Belaso, Giovan Batista, 24
Bell, Gordon, 109, 111
Bergmann, Henning, 115
Bernstein, Alex, 102
Berry, Clifford Edward, 79
Bigelow, Julian, 111
Billing, Heinz, 70, 111
Boda, M., 56
Bollée, Léon, 11, 44
Bolzano, Bernhard, 36
Boole, George, 27, 28
Booth, Andrew Donald, 70, 104, 111
Borel, Emile, 73, 74
Bottenbruch, Hermann, 88
Bouchon, Basile, 36
Braun, Antonius, 9
Brooks Jr., Frederick P., 109, 110
Brouwer, Luitzen Egbertus Jan, 75, 76
Bruns, Heinrich, 41
Buchholz, Werner, 111
Bull, Fredrik, 35
Burack, Benjamin, 56
Burkhardt, Arthur, 43
Burks, Arthur Walter, 80, 81
Burroughs, William Seward, 42–44
Burstall, R.M., 89
Bush, Vannevar E., 60, 66, 108
Büttner, 10

Caesar, Julius, 24
Campos, Francisco P., 65
Cantor, Georg, 37, 38
Capek, Karel, 63
Caramuel y Lobkowitz, Juan, 14
Cardano, Geronimo, 25

F.L. Bauer, *Origins and Foundations of Computing*,
DOI 10.1007/978-3-642-02992-9,
© Springer-Verlag Berlin Heidelberg 2010

Carnap, Rudolf, 73, 74
Carpentier, 33, 53
Carroll, Lewis, 28
Cartan, Elie Joseph, 38
Catalan, Eugène Charles, 39
Cauer, Wilhelm, 71
Cayley, Arthur, 39
Chaitin, Gregory J., 107
Champernowne, David G., 102
Chappe, Claude, 30, 31
Chebyshev, Pafnuty Lvovich, 12
Chomsky, Noam, 94, 95
Chuan Chu, Jeffrey, 111
Chudy, Joseph, 30, 35
Church, Alonzo, 74, 75, 87, 91
Churchill, Winston, 27
Ciermans, Johan, 8
Clark, Wesley A., 120
Clippinger, Richard F., 92
Cocks, Clifford, 106
Collins, E., 73
Colmerauer, Alain, 117
Comet, Stig, 111
Comrie, Leslie John, 65, 66, 107
Cook, Stephen A., 97, 98
Cooke, William Fothergill, 31
Coombs, Allen W.M., 105
Cooper, D.C., 89
Corbato, Fernando J., 109
Couffignal, Louis, 16, 69, 78
Couturat, Louis, 45
Crawford, Perry O., 82
Cray, Seymour R., 105, 106, 120, 121
Curry, Haskell B., 74–76, 116

Daemen, Joan, 106
Dahl, Ole-Johan, 116, 117
Damm, Arvid Gerhard, 46–49
Darlington, J., 89
de Beauclair, Wilfried, 61
de Colmar, Charles X. Thomas, 11
de Lamettrie, Julien Offroy, 62
De Morgan, Augustus, 27, 28
de Vaucanson, Jacques, 36, 62
de Viaris, Gaëtan Henri Lèon, 50
de Vigenère, Blaise, 24, 50, 51
Dedekind, Richard, 37, 39
Délibes, Leo, 62
Diffie, Whitfield, 105
Dijkstra, Edsger Wybe, 89, 90, 95, 110
d'Ocagne, Maurice, 11, 16
Dodgson, Charles Lutwidge, 28, 29
Dummet, Michael, 116
Dyck, Walther von, 36, 37, 57, 87

Earley, Jackson Clark, 84
Eckert, John Presper, 17, 19, 70, 80, 91, 111
Eckert, Wallace J., 17, 20, 66, 108
Edler, Robert W., 56
Edwards, D.J., 102
Egli, Hans W., 12, 45
Ehrenfest, Paul, 56
Elgot, Calvin C., 97
Ellis, James H., 105
Ershov, Andrei Petrovich, 89, 93, 95, 97, 101
Esser, Heinrich, 10
Euler, Leonhard, 27
Everett, Robert R., 82, 83, 111

Faddeeva, Vera Nikolaevna, 5

Falcon, Jean Baptiste, 36
Falkoff, Adin, 116
Fano, Robert M., 84
Farber, David Jack, 116
Felt, Dorr Eugene, 42
Fischer, Gerd, 85
Flowers, Thomas H., 105, 111
Floyd, Robert W., 89
Forrester, Jay W., 82
Forsythe, George, 5, 6
Frege, Gottlob, 28, 29, 39, 86
Friedman, William Frederick, 50, 51
Frisch, Ragnar, 71
Fromme, Theodor, 108

Gardner, Martin, 55
Gauss, Carl Friedrich, 31, 32
Gautier, Pierre, 62
Gerke, Friedrich Clemens, 32
Gershgorin, Semen Aronovich, 57
Gersten, Christian Ludwig, 10
Gill, Stanley, 91
Givens, Wallace, 5, 6
Gödel, Kurt, 73, 75, 85, 87, 92
Goldstine, Herman Heine, 16, 70, 80, 81, 91
Gore, John Kinsey, 35
Grassmann, Hermann, 36
Grau, Albert A., 85, 90
Gray, Frank, 53
Green, Julien, 88, 99
Grillet, René, 8
Güntsch, Fritz-Rudolf, 108, 109
Guthrie, Peter, 42

Haas, Karl, 46
Haddad, Jerrier A., 111
Hagelin, Boris, 49
Hahn, Philipp Mathäus, 9, 11
Hamann, Christel, 9, 12, 44, 45, 65
Hamming, Richard Wesley, 53, 54, 84
Hanzawa, Masao, 56
Harriott, Thomas, 83
Hart, T. P., 102
Hartree, Douglas Rayner, 60, 61
Hasenjäger, Gisbert, 87
Hayes, P.J., 117
Hebern, Edward Hugh, 46, 47
Hellman, Martin E., 105
Henrici, Peter, 6
Herbrand, Jacques, 73, 75, 87, 92, 101
Hermes, Hans, 91, 92
Hertz, Paul, 73
Herzstark, Kurt, 9
Hilbert, David, 59, 90
Hill, Lester S., 50
Hirzebruch, Friedrich, 85
Hoare, Charles Antony Richard, 84, 88, 89
Hochstetter, Erich, 1
Hoerni, Jean A., 123, 124
Hoff, Marcian Edward, 120, 121
Hoffmann, E.T.A., 62
Hoffmann, Fred, 46
Hollerith, Herman, 35, 45, 56, 83
Hondorf, W.W., 78
Hopkins, Hubert, 43
Hopkins, William, 43
Hopper, Grace Murray, 95
Householder, Alston Scott, 5, 6
Howard, William A., 116
Huffman, David A., 111
Hurd, Cuthbert C., 111
Hurwitz, Adolf, 71
Hurwitz, Alexander, 102

Huskey, Harry Douglas, 82, 90

Iverson, Kenneth Eugene, 116, 117

Jacobi, Carl Gustav Jacob, 4
Jacquard, Joseph-Marie, 35
Jahnz, Erwin, 45
Janov, Yuri J., 89, 95
Jaquet-Droz, Henri-Louis, 62
Jellinek-Mercedes, Emil, 46
Jevons, William Stanley, 27, 28, 55
Johnson, Reynold B., 111
Jordan, Camille, 38

Kahrimanian, Harry George, 101
Kanshine, Serge, 46
Kaphengst, Heinz, 96
Karmakar, Narendra, 126
Kasiski, Friedrich W., 25, 26
Kay, Alan, 116, 118
Kelvin (Lord Kelvin, William Thomson), 59
Kemeny, John George, 116, 117
Kempenfeldt, Adam Richard, 34
Kerckhoffs, Auguste, 25, 46
Kilburn, Tom, 17, 111
Kilby, Jack St. Clair, 123, 124
Killing, Wilhelm, 38
Kircher, Athanasius, 2, 23, 27
Kleene, Stephen Cole, 73–75, 87, 92
Kleiber, Johann, 57
Knaus, Ludwig, 62
Knuth, Donald Ervin, 96, 98
Koch, Hugo Alexander, 47
Kolmogorov, Andrei Nikolajewič, 55, 107
Korn, Willi, 48
Kowalski, Robert A., 117
Kronecker, Leopold, 7
Kryha, Alexander von, 46–48
Kullback, Solomon, 51
Kunze, Werner, 49

Lambert, Johann Heinrich, 27
Landin, Peter J., 89, 96
Lebedev, S.A., 111
Leclerc de Buffon, George Louis, 14
Lehmann, Nikolaus J., 111
Lehmer, Derrick Henry, 51, 52
Leibniz, Gottfried Wilhelm, 1, 3, 6, 9–11, 14–16, 23, 27, 35, 39, 42, 45, 63, 83, 86
Lie, Marius Sophus, 38
Lischke, N., 56
Ljapunov, Aleksei A., 88, 95
Llull, Ramon (Raimundus Lullus), 2, 23, 27, 39, 86
Lovelace (Countess of Lovelace, Ada Augusta King), 67, 68
Lucas, P., 90
Ludgate, Percy Edwin, 65, 67
Luhn, Hans Peter, 99
Lukasiewicz, Jan, 73, 74
Lukoff, Herman, 111
Lundgreen, Michael, 127

MacFarlane, Alexander, 29
Mahon, Charles, 27
Mannheim, Viktor Mayer Amédée, 11
Marconi, Guglielmo Marchese, 34
Markov, Andrei Andreevich, 87, 96, 97
Marquand, Allan, 27, 55

Marryat, Frederick, 34
Martin-Löf, Per, 107
Mathias, M., 78
Mauborgne, Joseph O., 49, 50
Mauchly, John William, 17, 20, 70, 79, 91, 111
McCarthy, John, 84, 88–90, 96, 99
McCulloch, Warren, 55
Meagher, Richard P., 111
Menabrea, Luigi Federico, 67
Metropolis, Nicholas Constantine, 91, 92
Meyrink, Gustav, 63
Moore, Gordon E., 123, 124
Morgenstern, Christian, 34
Morland, S. Samuel, 8
Morse, Samuel, 31, 33
Müller, Johann Helfrich, 10, 16
Myhrhaug, Bjørn, 116

Nakasima, Akira, 56
Napier, John, 11, 13
Naur, Peter, 84
Neumann, John von, 70
Newell, Allen, 95, 99, 102
Newman, Maxwell Herman Alexander, 104, 105
Niemann, C.W., 60
Nixdorf, Heinz, 110, 111
Noll, Andreas, 115
norm, 1
Nowak, Josef, 59
Noyce, Robert N., 123, 124
Nygaard, Kristen, 116, 117
Nyquist, Harry, 71

Odhner, Willgodt Theophil, 10, 12, 44
Offenbach, Jacques, 62
Olsen, Kenneth Harry, 119, 120
Oughtred, William, 58

Palm, Conny, 81
Palmström, 34
Parmelee, Dubois D., 10
Pascal, Blaise, 7–9, 11, 14, 41, 42
Paul, Jean, 62
Peano, Giuseppe, 36, 37, 87
Peaucellier, Charles, 57
Peirce, Charles Sanders, 28, 29, 55
Peirce, John Royden, 35
Perlis, Alan J., 84, 93, 99
Petri, Carl Adam, 89, 109
Petzval, Joseph, 68
Phillips, E. William, 16, 17, 78
Philon of Byzantium, 62
Piesch, Johanna, 56
Piloty, Hans, 81
Piloty, Robert, 112
Pitts, Walter, 55
Playfair, 25
Poleni, Giovanni, 10
Polybius, 30, 35
Pomerene, James H., 112
Porta, Giovanni Battista, 24
Pösch, H., 60
Post, Emil Leon, 73, 74, 87, 91
Powers, James, 35, 45
Prawitz, Dag, 102
Prinz, D.G., 102

Quevedo, Leonardo Torres y, 65, 68, 81, 102

Ramsey, F.P., 53
Rechnitzer, Alexander, 12, 45, 64
Reis, Johann Philipp, 33
Remus, Horst, 102, 127

Richter, Johann Paul Friedrich, 62
Riesel, Hans, 102
Rijmen, Vincent, 106
Rivest, Ronald, 106
Roberts, Lawrence G., 110
Robinson, Abraham, 97, 98
Robinson, John Alan, 102, 117
Robinson, Raphael M., 102
Rochester, Nathaniel, 112
Rosenbloom, Paul C., 93
Ross, Douglas Taylor, 99
Ross, Harold D., 112
Rosseland, Svein, 60
Rosser, J.B., 87
Rosza, Péter, 97
Roth, Didier, 10
Routh, Edward John, 71
Russell, Bertrand, 86
Rutishauser, Heinz, 86, 88, 90, 92–94, 96

Samelson, Klaus, 82, 84, 88, 93
Samuel, Arthur Lee, 102
Sauer, Robert, 60, 61, 81
Schäffler, Theodor Heinrich Otto, 35, 83
Scharlau, Winfried, 85
Schauffler, Rudolf, 49
Schecher, Heinz, 88, 98
Scherbius, Arthur, 47, 48
Scheutz, Edvard Raphael, 13
Scheutz, Pehr Georg, 13, 14, 35
Schickard, Wilhelm, 7, 8, 42
Schilling, Emil, 66
Schilling, Paul Pawel Lwowitsch, 31
Schilt, V., 11
Schönfinkel, Moses, 74
Schreyer, Helmut T., 108
Schröder, Ernst, 37
Schur, Issai, 71
Schuster, Johann Christian, 10, 11
Schützenberger, Marcel Paul, 95
Scott, Dana S., 88, 89, 96
Seeber, Robert E., 91
Selfridge, John L., 102
Selling, Eduard, 11, 44
Sforza, Francesco, 24
Shamir, Adi, 106
Shannon, Claude E., 26, 53–56, 71, 84, 102, 103, 107
Shaw, John Clifford, 95, 99, 102
Sheldon, John W., 81
Shepherdson, J.C., 96
Shestakov, Vitali Iwanowich, 56
Simon, Herbert Alexander, 99, 102
Simonetta, Cicco, 24
Sinkov, Abraham, 51
Sittler, F.J., 50
Slagle, James R., 101
Smirnov-Troyanskiy, P., 104
Smith, Francis O.J., 33
Solomonoff, Ray J., 107
Speiser, Ambros P., 112
Steiger, Otto, 12, 44, 63
Steinbuch, Karl, 101
Stibitz, George R., 17, 19, 69, 80, 81, 111
Stiefel, Eduard, 5, 6
Stoltenberg, Gerhard, 1
Strachey, Christopher S., 89, 96, 102
Studt, Heinrich, 46
Stuivenberg, Hendricus, 35
Sturgis, Howard Ewing, 96
Sundstrand, Oskar, 43
Svoboda, Antonin, 112

Swift, 90

Tarski, Alfred, 96, 97
Tatum, Liston, 81
Tauschek, Gustav, 45, 46, 65
Taussky-Todd, Olga, 6
Terebesi, Paul, 68
Thomas, Charles Xavier, 10
Thompson, A.J., 66
Thue, Axel, 87
Todd, John, 66, 67
Tootill, Geoff, 53
Tornig, Willi, 85
Trinks, Franz, 65, 66
Trithemius, Johannes, 24
Troncet, Louis, 42
Turing, Alan Mathison, 16, 17, 73, 75, 87, 91, 102, 104, 115
Tutte, William Thomas, 52

Uncapher, Keith W., 112
Upjohn, James, 62

Valtat, Raymond Louis André, 16, 45, 77, 78
van der Poel, Willem L., 112
van Duuren, Hendrik C.A., 53
van Wijngaarden, Adriaan, 89, 116
Venn, John, 27, 28
Vernam, Gilbert S., 26, 49
Viète, François, 25
von Hopfenberg, Anatol Wolf Holt, 99
von Neumann, John, 16, 17, 66, 70, 80, 81, 87, 91, 115
von Schubert, Friedrich Theodor, 101
von Steinheil, Carl August, 32

Wagner, R., 71
Wallis, John, 25
Walter, W. Grey, 103
Walther, Alwin, 61, 68
Watt, James, 63, 71
Weaver, Warren, 104
Weber, Heinrich, 39
Weber, Wilhelm, 31, 32
Weierstrass, Karl, 37
Weizenbaum, Joseph, 99, 100
Welchman, William Gordon, 104
Wertheimer, David Isaac, 10
Wheatstone, Charles, 25, 31–35, 46
Wheeler, David John, 91, 92
Whitehead, Alfred North, 87
Wiener, Norbert, 3, 54, 55, 71, 103, 104
Wilkes, Maurice Vincent, 17, 18, 70, 81, 88, 111
Wilkinson, James Hardy, 5, 6, 16, 18
Williams, Frederic Calland, 17, 111
Winograd, Shmuel, 96
Wirth, Niklaus Emil, 88, 99, 116
Wood, Benjamin Dekalbe, 108
Wynn-Williams, Charles Eryl, 17, 78

Yngve, Victor Huse, 116

Zemanek, Heinz, 96, 112, 125, 127
Zenger, Christoph, 121, 127
Zuse, Konrad, 16–18, 36, 41, 45, 56, 68, 69, 77, 78, 81, 84, 88, 91, 95, 99, 102, 108, 110, 118

Index of Figures

A Calculator Chronicle, IBM, Armonk, 11, 14, 42, 43
Abbildungen zur allgemeinen Bauzeitung, Jg. 13, Wien 1848, 32
Academy of Science, St. Petersburg: sketch by W. Schickard, 1624, 8
Annals of the History of Computing, 19, 52, 60, 83, 89, 90, 92, 99
Applied Mechanical Arithmetic as practised on the Comptometer. Felt and Terrant Comp. 1914, 42
Archäologisches Nationalmuseum, Athens, 62
Archiv für Post und Telegraphie, Berlin, 1888: J. Noebels: Geschichtlicher Entwicklungsgang der elektrischen Telegraphen, 33
Aspray, William et al.: Computing before Computers. Iowa State University Press, 1980, 2, 58, 66

Bassano and Vandyke Studios, London, 17
Bauer, F.L., personal archive, 37, 61, 69, 85, 90, 93, 108
Bell Telephone Laboratories, Murray Hill, NJ, 26
Belloc, Alexis: La Télégraphie Historique, Paris, 1888, 31
Bletchley Park Trust, 52, 105

California Institute of Technology Archives, 67
Campbell-Kelly, Martin et al.: The History of Mathematical Tables, Oxford University Press, 2003, 66
Charles Babbage Institute, University of Minnesota, Minneapolis, 43
Computer Laboratory, University of Cambridge, UK, 92
Crypto AG, 48, 49

Deutsches Museum München, 13
Duden Informatik: Schwill, Andreas et al., 109

Eckart, Mrs. Wallace J., private archive, 20

Fuchs, Walter R., Knaurs Buch der Denkmaschinen, 1968, 103, 104

Gardner, Martin, Logische Diagramme, 2
Gottfried Wilhelm Leibniz Bibliothek – Niedersächsische Landesbibliothek Hannover (LH XXXV, III B, 2 BI. Ir), 15

Handbuch der Elektronik, Leipzig, 1901: J. Noebels et al.: Telegraphie und Telephonie, 33

F.L. Bauer, *Origins and Foundations of Computing*,
DOI 10.1007/978-3-642-02992-9,
© Springer-Verlag Berlin Heidelberg 2010

Harvard University, News Office, 19
Herbrand, Jacques, Logical Writings, Dordrecht, 1971, 75
HNF Archives, 8, 9, 17, 18, 34, 44, 45, 47, 54, 68, 76, 88, 89, 96, 100, 110, 120

IBM Archives, 110, 117
Informatik-Führer Deutsches Museum, 2004, 21, 60, 94
Institute for Advanced Study Archives, Princeton, NJ, USA, 17
Iowa State University Archives, 79

J.H. Röll Verlag: Spurensuche. Wege zu Athanasius Kircher, Dettelbach 2002, 2
Joly, J.: Charles-Xavier Thomas. Un grand inventeur alsacien. In: La Vie en Alsace, 10, 1932, 11

Kahn, David, personal archive, 26, 47, 48, 50, 51

La Lumière electrique, 1883: copper engraving of Ch. Jocquet after the design of L. Leger, 10
Landesmuseum Württemberg, Foto: Peter Frankenstein, Hendrik Zwietasch, 10

Mauchly, Kathleen, 20
Meister, Aloys: Geschichte der Geheimschrift im Dienste der päpstlichen Kurie, Schöningh Verlag, Paderborn, 1905, 24
Metropolis, Nicholas et al.: A History of Computing in the Twentieth Century, p. 461, 92, 105
Monroe Calculating Machine Corporation, 1926, 12
Munz, Alfred: Philipp Matthäus Hahn, Sigmaringen 1987, 11
Museum für Kommunikation, Frankfurt, 31

Oberwolfach Photo Archives, 92

Palfreman, J., Swade, D.: The Dream Machine, BBC Books, 1991, 18

Royal Society, London, 61
Russell, Francis, The Secret War, 51

Santemases, J.G. Obra e Inventos de Torres Quevedo, Madrid, Instituto de Espana, 1980, 65
Science Museum, London, 13, 14, 64
sd&m-Konferenz 2001 “Software-Pioniere” (Videoarchiv HNF), 70
SIAM, Philadelpia, 106

Stadtarchiv Braunschweig, 66
Stanford University, News and Publication Service, 98

Technisches Museum, Wien, 10, 46
Turk, J.A.V.: Origin of Modern Calculating Machines – Technology and Society, Chicago, 1921, 12

United States Cryptographic Patents, 1861–1981, Jack Levine, 47
Universität Tübingen, 8
UPI/Corbis Bettmann Pictures, 19, 20

Wassén, Henry: The Odhner History – An Illustrated Chronicle of "A Machine to Count on", Gothenburg: Wezäta, 1951, 12
Wilkes, Maurice: Memoirs of a Computer Pioneer, The MIT Press, 70

Selected Readings

Aspray, William: *John von Neumann and the Origins of Modern Computing*, Cambridge, MIT Press, 1990
Augarten, Stan, *Bit by Bit: An Illustrated History of Computers*, Boston, Ticknor and Fields, 1984
Austrian, Geoffrey D.: *Herman Hollerith: Forgotten Giant of Information*, New York, Columbia University Press, 1982
Bauer, Friedrich L.: *Decrypted Secrets*, New York, Springer, 2000
Bauer, Friedrich L.: *Historische Notizen zur Informatik*, New York, Springer, 2009
Bowers, Brian: *Sir Charles Wheatstone, FRS, 1802–1875*, London, Science Museum, 2001
Brooks, Frederick P.: *The Mythical Man-Month*, Reading, Addison-Wesley, 1995
Ceruzzi, Paul E.: *Reckoners: The Prehistory of the Digital Computer*, Santa Barbara, Greenwood Press, 1983
De Beauclair, Wilfried: *Rechnen mit Maschinen*, Wiesbaden, Vieweg, 1968
Eames, Charles, Eames, Ray: *A Computer Perspective*, Cambridge, Harvard University Press, 1990
Findlen, Paula (ed.): *Athanasius Kircher. The Last Man Who Knew Everything*, London, Routledge, 2004
Gardner, Martin: *Logic Machines and Diagrams*, Chicago, Chicago University Press, 1982
Goldstine, Herman H.: *The Computer from Pascal to von Neumann*, Princeton, Princeton University Press, 1972
Grier, David Alan: *When Computers Were Human*, Princeton, Princeton University Press, 2005
Hashagen, Ulf, Keil-Slawik, Reinhard, Norberg, Arthur L.: *History of Computing: Software Issues*, New York, Springer, 2002
Hellige, Hans Dieter: *Geschichten der Informatik*, New York, Springer, 2004
Hodges, Andrew: *Alan Turing*, New York, Simon and Schuster, 1983
Hyman, Anthony: *Charles Babbage, Pioneer of the Computer*, London, Oxford University Press, 1982
Kleene, C. Stephen: *Mathematical Logic*, New York, Dover Publ., 2001
Knuth, Donald E.: *The Art of Computer Programming*, Vols. 1–3, Reading, Addison-Wesley, 1981
Lyons, John: *Noam Chomsky. Modern Masters*, New York, The Viking Press, 1970
McCorduck, Pamela: *Machines Who Think*, New York, W.H. Freeman & Co., 1979
Metropolis, N., Howlett, J., Rota, Gian-Carlo (eds.): *A History of Computing in the Twentieth Century*, San Deigo, Academic Press, 1980
Murray, Francis J.: *Mathematical Machines, Vol. II, Analog Devices*, New York, Columbia University Press, 1961
Nash, Stephen G.: *A History of Scientific Computing*, Reading, Addison-Wesley, 1990
Naumann, Friedrich: *Vom Abakus zum Internet*, Darmstadt, Primus Verlag, 2001

F.L. Bauer, *Origins and Foundations of Computing*,
DOI 10.1007/978-3-642-02992-9,
© Springer-Verlag Berlin Heidelberg 2010

Naumann, Friedrich, Schade, Gabriele: *Informatik in der DDR – Eine Bilanz*, Bonn, Gesellschaft für Informatik, 2006
Oberliesen, Rolf: *Information, Daten, Signale*, Reinbek, Rowohlt, 1987
Petzold, Hartmut: *Moderne Rechenkünstler*, Munich, C.H. Beck, 1992
Randell, Brian (ed.): *The Origins of Digital Computers*, New York, Springer, 1982
Reisig, Wolfgang, Freytag, Johann-Christoph: *Informatik – Aktuelle Themen im historischen Kontext*, New York, Springer, 2006
Shasha, Dennis, Lazere, Cathy: *Out of Their Minds – The Lives and Discoveries of 15 Great Computer Scientists*, Göttingen, Copernicus, 1995
Siefkes, Dirk, Braun, A., Eulenhöfer, P., Stach, H., Städtler, K.: *Pioniere der Informatik: Ihre Lebensgeschichte im Interview*, New York, Springer, 1999
Slater, Robert: *Portraits in Silicon*, Cambridge, MIT Press, 1992
Trogemann, Georg, Viehoff, Jochen: *Code@Art*, New York, Springer, 2005
Weaver, Warren, Shannon, Claude E.: *The Mathematical Theory of Communication*, Champaign, Illinois University Press, 1949
Weizenbaum, Joseph: *Computer Power and Human Reason*, New York, W.H. Freeman & Co., 1976
Wexelblat, Richard L. (ed.): *History of Programming Languages*, San Diego, Academic Press, 1969
Wiener, Norbert: *Cybernetics*, MIT Press
Wilkes, Maurice: *Automatic Digital Computers*, New York, John Wiley, 1956
Williams, Michael R.: *A History of Computing Technology*, New York, Prentice Hall, 1985
Zuse, Konrad: *Der Computer – Mein Lebenswerk*, New York, Springer, 1984