ITI Series (IUUK)

Institut teoretické informatiky
Institute for Theoretical
Computer Science

Informatický ústav Univerzity Karlovy
Computer Science Institute
of Charles University

I T I

I U U K

2025-689

Andrew Goodall, Jaroslav Nešetřil

The Making of a New Science

the early history of theoretical computer science

Institute for Theoretical
Computer Science (ITI)
Charles University

Computer Science Institute
of Charles University
(IUUK)

Malostranské náměstí 25
118 00 Praha 1
Czech Republic

http://iti.mff.cuni.cz/series/

http://iuuk.mff.cuni.cz/series/

The Making of a New Science

the early history of theoretical computer science

GALLERIA
CHODBA

Exhibition 18 June – 21 September 2025
at Galleria Chodba,
MFF UK, Malostranské náměstí 2/25, Praha 1

Andrew Goodall and Jaroslav Nešetřil

We could perceive that a new science was being born, arising from the roots of mathematical logic and projecting its light on the future of computers and computer programming.

— Giorgio Ausiello, *The Making of a New Science*

Exhibition guide

It is very rare that one can make an exhibition tracing the entire history of an established scientific discipline. Theoretical Computer Science (TCS for short) – which only emerged in the 1970s as a recognized field per se out of its origins in logic, mathematics and engineering – has developed and matured so quickly that this is possible; or, better, we can at least try to present the history of the field in a concise and yet uncrowded way.

TCS (or the Theory of Computing, Computability Theory, or even Algorithms and Complexity, to give just a few of the several names that are used to refer to the discipline) develops the abstract concepts and mathematical models that underpin computation, with a particular focus on the design, analysis, and efficiency of algorithms. It has developed in the last few decades from humble origins into a field which is taught at most universities worldwide, and which spans a large part of technological development today. One can say even more: perhaps nowhere in contemporary science is there such a direct line from theory to development and then to praxis and its consequent impact on everyday life. Our aim here can only be to sketch parts of this evolution of TCS. We concentrate on the early origins in the 1920s to c. 1990 (with occasional glimpses beyond); we thereby also limit our scope by concentrating on the theory that established itself in the twentieth century, omitting some trends which have proved to be very active more recently. This is more concretely described below.

We are fortunate that we can build our exhibition around two earlier instances. One is the excellent book *The Making of a New Science* (Springer 2023) by Giorgio Ausiello. This book, by one of the pioneering leaders and organizers of the development of TCS in Europe, is both a factual report of the history of TCS until 1980 and a memoir replete with personal details, making it fascinating reading for non-specialists and specialists alike. Prof. Ausiello put us in contact with the people behind our second source, the exhibition *50 Years of Theoretical Computer Science* shown at the ICALP '22 conference in Paris. We thank Sandrine Cadet and Sylvain Schmitz, organizers of this exhibition, for allowing us to use and modify their material.

Here we give a panel-by-panel overview of our exhibition. The first panel (bearing the title of the exhibition, **The Making of a New Science**) pays tribute to Prof. Giorgio Ausiello, featuring a brief quotation from his book along with quotes from other recognized computer scientists, and two of Vera Molnár's digital plotter drawings (from the series *Letters from my mother*).

The second panel (**On the shoulders of giants**) picks out some of the major milestones and key figures in the theory of computation. This of course goes back to star mathematicians (and scientists) of the early twentieth century (we omitted traces of the field to be found in previous centuries). This earlier history is recorded in the reels of black-and-white "film" while more recent names are recorded in the array of colour "polaroids". The panel is complemented by a list of awardees of the Nevanlinna-Abacus medal of the International Mathematical Union (established in 1982 and given at the International Congress of Mathematicians ever since).

The next panel **Pioneers in TCS** continues the focus on key figures of the field, featuring a spotlight on Maurice Nivat (transferred directly from the Paris exhibition *50 Years of Theoretical Computer Science*), whose individual story has counterparts for each and every one of the notable figures in TCS listed in the remainder of the panel. These lists give in chronological order all the winners of four major annual international prizes: the A. M. Turing Award, Gödel Prize, EATCS Award and Donald E. Knuth Prize. Seeing these lists, one gets a better feeling of how the development of TCS has been a product of collective endeavour by a remarkable and numerous set of individuals.

Brussels, 1972 reworks a panel from the Paris exhibition *50 Years of Theoretical Computer Science,* supplementing it by facsimiles of key documents reproduced in the appendix to

Ausiello's *The Making a New Science*. Described are the key events leading to the creation of the European Association of Theoretical Computer Science (EATCS), highlighting the role played by Maurice Nivat and others in bringing it to fruition.

The foundation of new institutions led to the creation of a new type of conference, one with refereed contributions selected by a programme committee. Such conferences became the dominant medium for scientific communication in TCS, and a (non-exhaustive) selection of the most long-running, active and prestigious are displayed in the panel **Early conferences in TCS**. The penultimate panel of the exhibition (see below) features an editorial by Moshe Vardi discussing the role of conferences in computer science, which is very different from mathematics.

The next five panels are devoted to particular areas of TCS in their early development. The choice of topics is of course rather arbitrary as current TCS is a very broad field. But on the history there is a consensus. We included material from the Paris panels on computational complexity, logic and complexity, automata and a schematic overview of algorithms that have shaped the world; on the other hand, we left out the Paris material on zero-knowledge proofs, fine-grained complexity, model checking, the science of programming, machine-checked proofs, and quantum computing. We are limited to thirteen panels and some of these topics will be treated in a follow-up exhibition.

With these constraints, our panels are **Automata theory**, describing the development of this area from Alan Turing on; **Computational complexity (I)**, describing the birth of complexity classes and "complexity" as we know it today; **Computational complexity (II)**, describing the influence of logic and the logical side of TCS; and **Algorithms (I)**, representing some of the plethora of beautiful algorithms from which the theory of algorithms has evolved. These four panels are adapted from the Paris panels, to which we have added photographs of some of the key figures involved. Additionally, in the Algorithms (I) panel we highlighted seven major algorithms from those "shaping the world", while, in the Automata theory panel we included diagrams illustrating the evolution of a pair of elementary cellular automata (rules 30 and 110).

The next two panels are the only panels devoted to specific problems. **Algorithms (II)** treats a topic dear to all Czech mathematicians and computer scientists as it describes the role of Borůvka and Jarník in the development of the minimum spanning tree algorithm (as well as the Steiner tree algorithm). The panel **Advanced graph theory in TCS** describes the story of expanders – another TCS saga. Expanders represent a key structure in the theory of algorithms and their construction involves beautiful and difficult mathematics.

The last two panels document some of the bewilderingly extensive activity in TCS. The panel **Publishing research** includes material from the Paris exhibition panel focusing on ICALP publications, to which we have added a discussion about the distinctive practice of theoretical computer of using conferences and their proceedings as the primary mode of communication rather than journals: we reproduce an editorial by Moshe Vardi in *Communications of the ACM,* concerning a still very current topic of debate. The final panel **Selected textbooks in TCS, 1966-1999** gives a panorama of various twentieth century textbooks, both basic and advanced, many appearing in new editions to the present day, and all of them influential in directing the course of TCS as it entered the twenty-first century.

We expect that many of our viewers may have remarks or questions related to this exhibition: please send them to our address galleriachodba@iuuk.mff.cuni.cz

Acknowledgments

We thank Giorgio Ausiello for alerting us to the exhibition *50 Years of Theoretical Computer Science* shown at the ICALP '22 conference in Paris and for putting us in touch with its coordinators, Sandrine Cadet and Sylvain Schmitz, who generously allowed us to use their material as a basis for our own exhibition. We are grateful to the School of Computer Science of the Faculty of Mathematics and Physics for supporting Galleria Chodba as well as the UNCE project 'Language, image, gesture: forms of discursivity' (a joint project with the Faculty of Arts, Charles University).

Andrew Goodall, Jarik Nešetřil

Curators, Galleria Chodba

Catalogue

As students, through Corrado's lectures, we could perceive that a new science was being born, arising from the roots of mathematical logic and projecting its light on the future of computers and computer programming — Giorgio Ausiello

In the 1980s, we the theoreticians owned the field of computer science in the sense that we knew how compilers should be designed, how operating systems should work, how databases should be organized for efficient access. Our responsibility and main mission was to outfit all areas of computer science with rigor and the power of mathematics. In a sense, we were exercising complete intellectual hegemony over the rest of computer science. We knew it. Then the internet happened. — Christos Papadimitriou

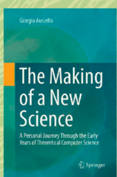
The Making of a New Science

The best practice is inspired by theory. — Donald Knuth

... Computer science is no more about computers than astronomy is about telescopes. There was this realization: a new science is emerging. The name may not be well chosen, but it is a new science. — Jacques Arsac

A theoretical computer scientist will have the same aim as any other computer scientist: namely to understand, analyse and hopefully clarify the fascinating concept of computation. The theoretical computer scientist distinguishes himself by his choice of the ways to achieve such an understanding and clarification – the choice being, after proper models have been formalized, to prove theorems that are meaningful. With the mathematician and logician, the theoretical computer scientist has in common the knowledge that sometimes the solution to a problem has to be obtained at some distance from the problem itself, and that an extensive knowledge of the relevant objects and concepts has to be developed before any major question can be answered. — Maurice Nivat

We are fortunate to have been helped by two sources in putting together this exhibition on the history of theoretical computer science in the twentieth century. The first is Giorgio Ausiello's excellent *The Making of a New Science* (Springer 2018). This book – a memoir of one of the pioneering leaders and organizers in the development of theoretical computer science – gives a factual account of the burgeoning new science in Italy, Europe, and the US. A history filled with personal recollections by one of the key eyewitnesses, and fascinating reading for outsiders. The second source is thanks again to Prof. Ausiello, who put us in contact with the coordinators of the exhibition *50 Years of Theoretical Computer Science* at ICALP'22 on July 6–8, 2022 at Université Paris Cité, devised for the occasion of the 50th anniversary of the ICALP conference and the creation of EATCS.



Giorgio Ausiello and Jarik Nešetřil

On the shoulders of giants

Episodes in the development of TCS

1900
David Hilbert's 23 Mathematical Problems. In the 2nd Hilbert called for a mathematical proof of the consistency of the axioms for arithmetic.

1907, 1912
Leonardo Torres Quevedo introduces a formal language for the description of mechanical drawings; builds the first decision-making automaton – a chess-playing machine.

1931
Kurt Gödel's incompleteness theorem - inherent limitations to formal systems and what can be proved within them.
Emil Post develops, independently of Turing, a mathematical model of computation. Post's rewrite technique is widely used in programming language specification and design.

1936
Alonzo Church proves undecidability of arithmetic using lambda-calculus.

1936
Alan Turing in 'On computable numbers' introduces the Turing machine, providing a foundational framework for understanding the limits and possibilities of computation.

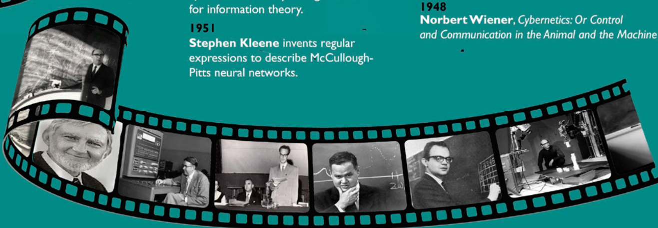
1943
Warren McCulloch and **Walter Pitts** describe a simplified neural network architecture for intelligence.

1944
John von Neumann, *Theory of Games and Economic Behavior*, written with Oskar Morgenstern (in 1928 he had instigated the theory of games, proving the minimax theorem).
1945 Merge sort; von Neumann architecture.
1950s Time complexity of computation.
1952 2-D cellular automata.

1948
Norbert Wiener, *Cybernetics: Or Control and Communication in the Animal and the Machine*

1948
Claude Shannon's *Mathematical theory of communication* lays the groundwork for information theory.

1951
Stephen Kleene invents regular expressions to describe McCullough-Pitts neural networks.



1950
Boris Trakhtenbrot 'The impossibility of an algorithm for the decidability problem on finite classes.' In 1964, proves the Gap Theorem – there are arbitrarily large computable gaps in the hierarchy of complexity classes.

1956
Edward McCluskey develops the first algorithm for designing combinational circuits – the Quine-McCluskey logic minimization procedure.

1959
Arthur Samuel proves that machines can learn from past errors.

1960
While studying machine translation of languages in Moscow, **Tony Hoare** develops Quicksort.

1965
John Tukey and James Cooley 'An algorithm for the machine calculation of complex Fourier series'

1969-
Donald Knuth, *The Art of Computer Programming*.

1969
Marvin Minsky and Seymour Papert *Perceptrons*, cause of a long-standing controversy in the study of artifical intelligence.

John McCarthy, Leonid Khachiyan, Shafi Goldwasser, Barbara Liskov, Robert Tarjan, Peter Shor

Leonard Adleman, Edsger Dijkstra, Leonid Levin, Stephen Cook, Whitfield Diffie, Martin Hellman, Mike Paterson, Noga Alon, Josep Díaz, Józef Gruska, Avi Wigderson

Abacus Medal

(Rolf Nevanlinna Prize before 2022) for all mathematical aspects of computer science, and for scientific computing and numerical analysis.

1982	**Robert Tarjan**
1986	**Leslie Valiant**
1990	**Alexander Razborov**
1994	**Avi Wigderson**
1998	**Peter Shor**
2002	**Madhu Sudan**
2006	**Jon Kleinberg**
2010	**Daniel Spielman**
2014	**Subhash Khot**
2018	**Constantinos Daskalakis**
2022	**Mark Braverman**

Pioneers in TCS

Turing and EATCS Awards, Gödel and Knuth Prizes

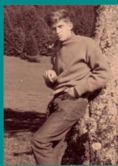
Turing Award

for contributions of lasting and major technical importance to computer science.

1966	Alan Perlis
1967	Maurice Wilkes
1968	Richard Hamming
1969	Marvin Minsky
1970	James (Jim) Hardy Wilkinson
1971	John McCarthy
1972	Edsger Dijkstra
1973	Charles Bachman
1974	Donald Knuth
1975	Allen Newell, Herbert Simon
1976	Michael Rabin, Dana Scott
1977	John Backus
1978	Robert Floyd
1979	Kenneth Iverson
1980	C.A.R. Hoare
1981	Edgar Codd
1982	Stephen Cook
1983	Dennis Ritchie
1984	Niklaus Wirth
1985	Richard Karp
1986	John Hopcroft
1987	John Cocke
1988	Ivan Sutherland
1989	William Kahan
1990	Fernando Corbato
1991	Robin Milner
1992	Butler Lampson
1993	Juris Hartmanis
1994	Edward Feigenbaum, Raj Reddy
1995	Manuel Blum
1996	Amir Pnueli
1997	Douglas Engelbart
1998	Jim Gray
1999	Frederick Brooks
2000	Andrew Chi-Chih Yao
2001	Ole-Johan Dahl, Kristen Nygaard
2002	Leonard Adleman, Ronald Rivest, Adi Shamir
2003	Alan Kay
2004	Vinton Cerf, Robert Kahn
2005	Peter Naur
2006	Frances Allen
2007	Edmund Clarke, Joseph Sifakis
2008	Barbara Liskov
2009	Charles Thacker
2010	Leslie Valiant
2011	Judea Pearl
2012	Shafi Goldwasser, Silvio Micali
2013	Leslie Lamport
2014	Michael Stonebraker
2015	Whitfield Diffie, Martin Hellman
2016	Tim Berners-Lee
2017	John Hennessy, David Patterson
2018	Yoshua Bengio, Geoffrey Hinton, Yann LeCun
2019	Edwin Catmull, Patrick Hanrahan
2020	Alfred Vaino Aho, Jeffrey David Ullman
2021	Jack Dongarra
2022	Robert Melancton Metcalfe
2023	Avi Wigderson
2024	Andrew Barto, Richard Sutton
2025	

Maurice Nivat

Nivat (right) with siblings and grandmother



Nivat aged 20

Decoration by Minister of Research Hubert Curien, 2002

Nivat with Schützenberger at ICALP'72

Maurice Nivat

As a mathematician, Nivat applied rigorous algebraic approaches to numerous domains, from formal languages to program semantics, from concurrent processes to discrete geometry. As a scientific leader, he undertook with incredible energy the mission of promoting study and research in the theory of computing.

Early years

- 1937 Born in Clermont-Ferrand, France.
- 1956 Enters *École Normale Supérieure*. His broad-mindedness and originality flourish, and he is the leader of a group of merry fellows which calls itself "Praesidium du Bordel Suprême"; he gets married and has his first son while still at ENS.
- 1959 Begins work at Institut Blaise Pascal and gets acquainted with computers and programming languages.
- 1969 Becomes professor at Université de Paris.

Founding the EATCS

- 1971 With Louis Nolin and Marcel-Paul Schützenberger, presents a "charter" of theoretical computer science, called *Rapport préliminaire sur l'Informatique Théorique*; proposes to establish a collaboration with the main European universities and research centres.
- 1972 Organizes the first International Colloquium on Automata, Languages and Programming (ICALP). He and Alfonso Caracciolo organize the Brussels meeting where the creation of the EATCS is approved.
- 1973 Elected President of EATCS and edits the first *Bulletin of the EATCS*; founds the journal *Theoretical Computer Science.*

Gödel Prize

for outstanding papers in the area of theoretical computer science

- L. Babai, S. Moran, S. Goldwasser, S. Micali, C. Rackoff
- Johan Håstad
- Neil Immerman, Róbert Szelepcsényi
- Mark Jerrum, Alistair Sinclair
- Joseph Halpern, Yoram Moses
- Seinosuke Toda
- Peter W. Shor
- Moshe Y. Vardi, Pierre Wolper
- S. Arora, U. Feige, S. Goldwasser, C. Lund, L. Lovász, R. Motwani, S. Safra, M. Sudan, M. Szegedy
- Géraud Sénizergues
- Yoav Freund, Robert Schapire
- M. Herlihy, N. Shavit, M. Saks, F. Zaharoglou
- Noga Alon, Yossi Matias, Mario Szegedy
- Manindra Agrawal, Neeraj Kayal, Nitin Saxena
- Alexander A. Razborov, Steven Rudich
- Daniel A. Spielman, Shang-Hua Teng
- Omer Reingold, Salil Vadhan, Avi Wigderson
- Sanjeev Arora, Joseph S.B. Mitchell
- Johan T. Hastad
- E. Koutsoupias, C. Papadimitriou, T. Roughgarden, E. Tardos, N. Nisan, A. Ronen
- Antoine Joux, Dan Boneh, Matthew K. Franklin
- Ronald Fagin, Amnon Lotem, Moni Naor
- Daniel A. Spielman, Shang-Hua Teng
- S. Brookes, P.W. O'Hearn
- C.Dwork, F. McSherry, K. Nissim, A. Smith
- Oded Regev
- Irit Dinur
- Robin A. Moser, Gábor Tardos
- A. Bulatov, M. Dyer, D. Richerby, J-Y. Cai, X. Chen
- Z. Brakerski, C. Gentry, V. Vaikuntanathan
- S. Fiorini, S. Massar, S. Pokutta, H. R. Tiwary, R. de Wolf, T. Rothvoss
- Ryan Williams
- Eshan Chattopadhyay, David Zuckerman

EATCS Award

in recognition of a distinguished career in theoretical computer science

- Richard Karp
- Corrado Böhm
- Maurice Nivat
- Grzegorz Rozenberg
- Arto Salomaa
- Robin Milner
- Mike Paterson
- Dana S. Scott
- Leslie G. Valiant
- Gérard Huet
- Kurt Mehlhorn
- Boris (Boaz) Trakhtenbrot
- Moshe Y. Vardi
- Martin Dyer
- Gordon Plotkin
- Christos Papadimitriou
- Dexter Kozen
- Éva Tardos
- Noam Nisan
- Thomas Henzinger
- Mihalis Yannakakis
- Toniann (Toni) Pitassi
- Patrick Cousot
- Amos Fiat
- Samson Abramsky
- Rajeev Alur

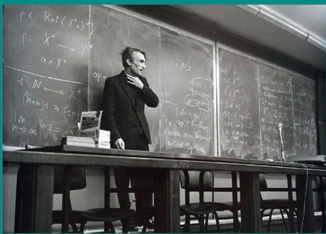
Knuth Prize

for outstanding contributions to the foundations of computer science.

- Andrew Yao
- Leslie Valiant
- László Lovász
- Jeffrey Ullman
- Christos Papadimitriou
- Miklós Ajtai
- Mihalis Yannakakis
- Nancy Lynch
- Volker Strassen
- David Johnson
- Ravi Kannan
- Leonid Levin
- Gary Miller
- Richard Lipton
- László Babai
- Noam Nisan
- Oded Goldreich
- Johan Håstad
- Avi Wigderson
- Cynthia Dwork
- Moshe Vardi
- Noga Alon
- Éva Tardos
- Rajeev Alur
- Micha Sharir

Brussels, 1972

Where, when and why EATCS and ICALP started



Marcel-Paul Schützenberger at the first ICALP conference, July 3-7 1972

Informatics or computer science was seen by other disciplines and by many politicians as simply a technology to support other enterprises. It was already clear, however, that to improve the correctness and efficiency of large-scale programs, theoretical studies were needed to investigate the principles and properties of computing. At the time, such work in Europe tended to be local and national. New funding for inter-European collaboration would be required.

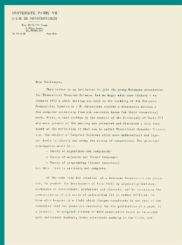
There was a very special spirit in the air; we knew that we were witnessing the birth of a new scientific discipline centered on the computer. – R. Karp

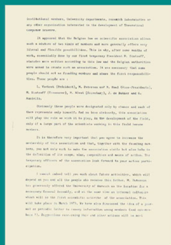
There was absolutely no appreciation of the work on the issues of computing. Mathematicians did not recognize the emerging new field. – M. Rabin

RAPPORT PRÉLIMINAIRE SUR L'INFORMATIQUE THÉORIQUE

First European report on the field of Theoretical Computer Science

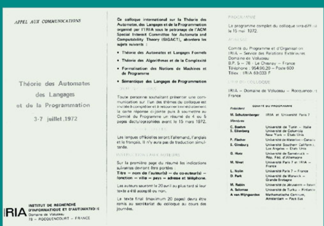
Rapport préliminaire sur l'Informatique Théorique (M. Nivat, L. Nolin, M.-P. Schützenberger, 1971) outlines the main pillars of the new science and, for each pillar, describes the research subject addressed, with reference to specific authors:
— Algorithms, in particular those concerning arithmetic operations (Winograd), sorting (Knuth, Floyd), graph algorithms (Rabin);
— Automata and formal languages, with reference to equations on the free monoid (Lentin), codes, finite automata and regular languages (Kleene, Krohn & Rhodes), push-down automata and context-free languages (Schützenberger), tree automata;
— Formal semantics of programming languages, where with experience from the syntactic and semantic definition of Algol 68, the need to provide precise formulations of the semantics of programming languages is discussed, based on the early works on axiomatic semantics (Floyd), operational semantics (McCarthy), approaches to semantics based on lambda-calculus (Scott) and combinatory logic (Nolin), and the theory of program schemes (Ianov, Luckham, Park & Paterson, and Strong).
The report underlines the theory of operating systems, of parallel concurrent and cooperating processes, and of the corresponding computation models (Dijkstra, Naur, Wirth) expected to play an important role in the future.





Foundation of the EATCS

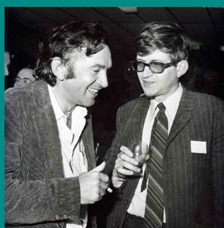
A meeting on January 27-28, 1972 at the Berlaymont building of the EU Commission in Brussels, chaired by Alfonso Caracciolo, is attended by M. Nivat, L. Nolin, M. Gross (F), H. Langmaack, K. H. Böhling (D), I. Verbeek, J. de Bakker (NL), M. Paterson (UK), M. Sintzoff (B), C. Böhm, U. Montanari, and G. Ausiello (I). After presenting the report of M. Nivat, L. Nolin and M.-P. Schützenberger, they approve the proposal prepared by Maurice Nivat on cooperation among European universities, which leads in September to the creation of the European Association for Theoretical Computer Science (EATCS).



APPEL AUX COMMUNICATIONS

Théorie des Automates des Langages et de la Programmation 3-7 juillet 1972

IRIA



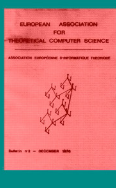
Maurice Gross and Maurice Nivat at the first ICALP

First ICALP

On July 3-7, 1972, at IRIA (Rocquencourt, Paris) the first ICALP takes place. The Program Committee of C. Böhm, S. Eilenberg, P. Fischer, S. Ginzburg, G. Hotz, M. Nivat, L. Nolin, D. Park, M. Rabin, A. Salomaa, and A. van Wijngaarden is chaired by M.-P. Schützenberger.
The programme includes 45 accepted papers (29 in English, 14 in French, 2 in German) on automata theory, theory of programming, theory of formal languages, and complexity of algorithms.

First Bulletin

On December 1973, Maurice Nivat prepares the first Bulletin of EATCS at IRIA, Rocquencourt. The bulletin includes the minutes of the first general assembly and council meeting; reports on the second MFCS; and provides activity reports of the Mathematisch Centrum, Amsterdam, the Technological University, Delft, the Technological University, Twente, the Istituto di Scienza dell'Informazione, Università di Torino and the Institut de Programmation, Université Paris VI.



association européenne d'informatique théorique

european association for theoretical computer science

BULLETIN N°1 décembre 1973

Early conferences in TCS

Across computer science – in both Theoretical Computer Science and Software Engineering – the tradition became established to submit new results to conferences in the form of an extended abstract. This is evaluated by the Programme Committee, often by a single or double blind review process. Accepted papers are then delivered at the conference itself, and subsequently published in the conference proceedings. Conferences are now ranked (A*,A,B,C) and acceptance of a paper in a top-ranked conference is considered very prestigeous, comparable to appearing in a top mathematical journal. On the other hand, this practice of preferring conferences to journals is very different from mathematics and other academic fields. This discrepancy is discussed e.g. in M. Vardi (2009) 'Conferences vs. journals in computing research' Communications of the ACM 52:5.

1960 Symposium on Foundations of Computer Science (FOCS)

1969 Symposium on Theory of Computing (STOC)

1972 International Colloquium on Automata, Languages, and Programming (ICALP)

1972 International Symposium on Mathematical Foundations of Computer Science (MFCS)

1975 International Workshop on Graph-Theoretic Concepts in Computer Science (WG)

1977 International Symposia on Fundamentals of Computation Theory (FCT)

1984 Symposium on Theoretical Aspects of Computer Science (STACS)

1986 Symposium on Logic in Computer Science (LICS)

1986 Computational Complexity Conference (CCC)

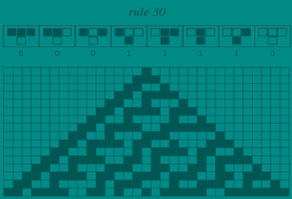
1990 Symposium on Discrete Algorithms (SODA)

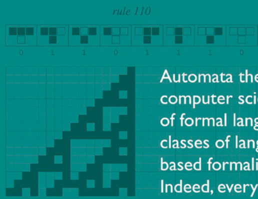
1990 Intern'l Symposium on Algorithms and Computation (ISAAC)

1993 Annual European Symposium on Algorithms (ESA)

Automata theory

Abstract machines and their computational power





Automata theory is one of the oldest research areas in computer science. Historically, it developed with the theory of formal languages, since automata were classified by classes of languages they can recognize. Today, automata-based formalisms are widely applied in modern computing. Indeed, every computing device has "automata inside"!

Automata theory concerns abstract computing devices and their computational power. It emerged from Turing's study of the power of general-purpose computation and from Kleene's formalization of an earlier proposal by McCulloch and Pitts, the latter motivated by the study of networks of neurons.

Automata theory permeates computer science. Initially their study was motivated by, and has an immediate application in, fields such as computer design, compilation of programming languages, and search and pattern matching. Their use then spread across the whole field.

Automata theory uses increasingly sophisticated mathematical techniques to study the power of abstract computational devices. It has close connections with classic and novel fields of mathematics such as group theory and the theory of algebraic structures, logic, (finite) model theory, number theory, (automatic) real function theory, symbolic dynamics, and topology.

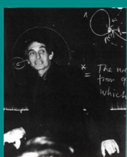

John Horton Conway

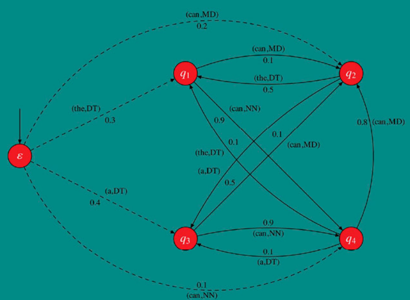

Michael O. Rabin


John von Neumann


Noam Chomsky


Dana Scott


Marcel-Paul Schützenberger



A weighted word automaton for part-of-speech tagging in English

Selected key milestones in automata theory


Alan Turing

1936 **A. Turing:** Turing machines

1943 **W. McCulloch, W. Pitts:** Nerve nets as finite automata

1948 **J. von Neumann:** The general and logical theory of automata

1951 **S.C. Kleene:** Regular expressions, Kleene's Theorem

1955 **M.P. Schützenberger:** Algebraic theory of automata: Syntactic semigroups and variable-length codes

1956 **E.F. Moore:** Minimal automata

1957 **J. Myhill:** Non-deterministic automata and determinisation.

1958 **A. Nerode:** Nerode equivalence
J.R. Büchi, C.C. Elgot, B.A. Trakhtenbrot: Finite automata and monadic second-order logic (MSO)

1959 **M.O. Rabin, D. Scott:** Finite automata and their decision problems

1963 **N. Chomsky, M.P. Schützenberger:** Context-free languages and pushdown automata

1965 **M.P. Schützenberger:** Star-free expressions and group-free monoids
K. Krohn and J. Rhodes: Decomposition of automata

1969 **M.O. Rabin:** Automata on infinite trees and MSO

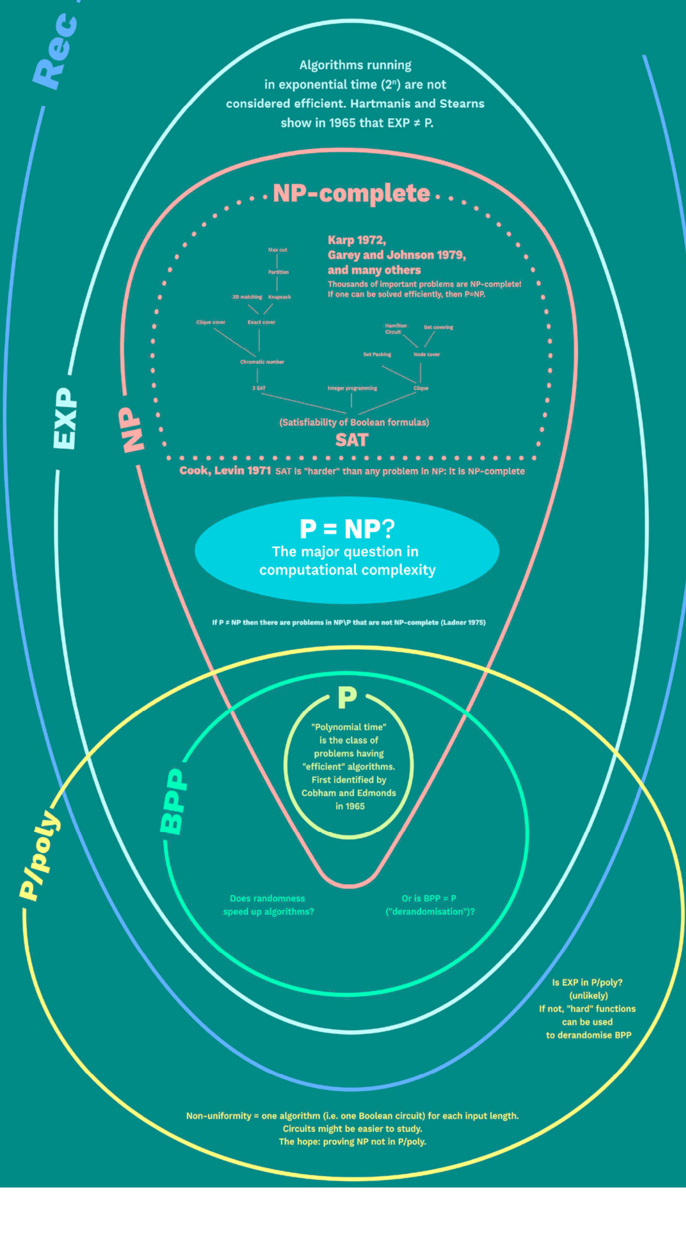
1982 **Y. Gurevich, L. Harrington:** Trees, automata and games
W. Thomas: Classifying regular events in symbolic logic

1988 **N. Immerman, R. Szelepszenyi:** Complementation of linear bounded automata
K. Hashiguchi: Solution of the restricted star-height problem

Computational complexity (I)

Classifying problems by hardness

In the 1930s, Church, Turing and others proposed the "right" notion of algorithm and studied what is recursive, i.e. what can be solved by all computers. Later, with the first computers, the efficiency of algorithms became crucial. Computational complexity was born.




Richard Karp


Leonid Levin


Stephen Cook


Leslie Valiant

Rec
problems that can be solved by computers

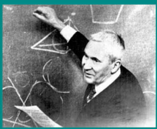
EXP
Exponential time

NP
Nondeterministic Polynomial time: solutions can be verified efficiently

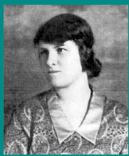
BPP
Bounded-error Probabilistic Polynomial time

Computational complexity (II)

A perfect match with logic



Andrey Kolmogorov



Rózsa Péter

The unity of logic and computation has manifested itself in the development of computability theory from the 1930s onward, and the development of computational complexity from the 1960s onward. Computability theory delineates the boundary between decidability and undecidability; computational complexity that between tractability and intractability. Logic provides prototypical complete problems for complexity classes and led to descriptive complexity, a framework for characterizing complexity classes using logical resources.

Complete problems

1936
Church-Turing Theorem
First-Order Validity is computably enumerable (c.e.)-complete.

1949
Trakhtenbrot's Theorem
First-Order Finite Satisfiability is computably enumerable (c.e)-complete.

1971
Cook-Levin Theorem
SAT is NP-complete.

Descriptive complexity

1974
Fagin's Theorem
NP = ESO. In words, a decision problem Q is in NP if and only if Q is expressible in existential second-order logic ESO.

"machine-free characterisation of NP with no mention of polynomial"

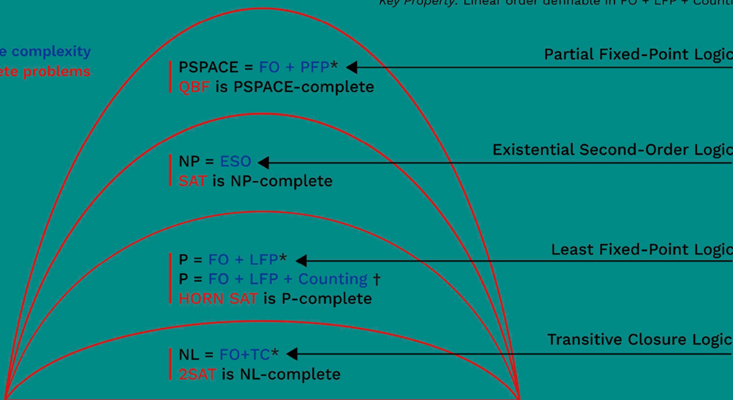
Example: SAT is definable by the ESO-formula

$$\exists S \forall c \exists v ((P(c,v) \wedge S(v)) \vee (N(c,v) \wedge \neg S(v)))$$

1982
Immerman-Vardi Theorem
P = FO+LFP on classes of ordered finite structures.

2010
Grohe's Theorem
If **C** is a class of graphs with at least one excluded minor, then on **C**
P = FO + LFP + Counting.
Key Property: Linear order definable in FO + LFP + Counting on C.

Descriptive complexity and complete problems



*on classes of *ordered* finite structures
† on classes of finite structures *excluding at least one minor*

Long-standing Open Problem in Descriptive Complexity
[Chandra & Harel (1982) – Gurevich (1988)]

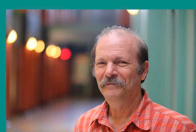
↓ ?

Is there a logic for P on the class of all finite structures?

Martin Grohe

Neil Immerman

Ronald Fagin



Moshe Vardi

Algorithms (I)

Shaping the world

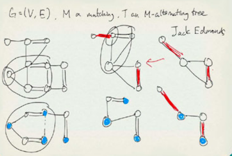
Algorithms are the heart of computing systems. They are not usually visible to the user, but they keep the systems going and provide functionality and speed. Without algorithms there would be no systems. Not surprisingly, every computer scientist is taught algorithms. The design and analysis of algorithms is a subject of intellectual depth and beauty with a wide-ranging impact on the real world.

dynamic
Distributed Hash Tables
The boosting algorithm
in machine learning
(due to Schapire)

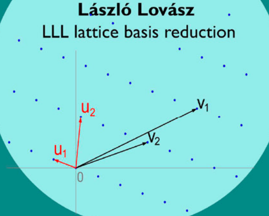
Weighted Fair Queueing Balanced Trees
Metropolis Algorithm Matching algorithms
Ford-Fulkerson Max-Flow, Edmonds-Karp

Ford
Fulkerson
Max-Flow
Gaussian
Elimination
Balanced Trees
Sorting, Quicksort
quantum
computing
FFT LLL Page rank
Linear Programming
Dynamic Programming
Interior Point, Simplex online matching Dijkstra
Cole Vishkin algorithm (mostly theoretically) Heaps Signal Processing
Simulated Annealing online algorithms Balanced Trees Edmonds-Karp
Counting algorithm with Hyperloglog
algorithm longest common subsequence online
Shor's factoring algorithm programming Gaussian Elimination
ranking Randomized Incremental Construction in machine learning
Brozozowski DFA minimisation algorithm Distributed Hash Tables
belief propagation Lamport Clocks
Matching algorithms
All APSP allgorithms The Berlekamp-Welch error correction
algorithm, error correction rsa approximate TSP
quantum computing genetic algorithms Lovasz
Matching algorithms Signal Processing and Communication
bipartite FFT and Viterbi (special case of dynam
programming)
general
codes
deep learning
trees
Greedy
rsa

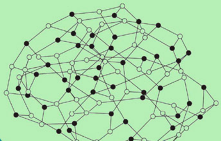
Jack Edmonds
maximum matchings



Arjen Lenstra, Henrik Lenstra, László Lovász
LLL lattice basis reduction

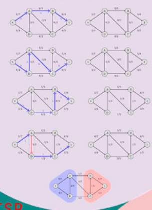


Michel Goemans, David Williamson
approximate max cut

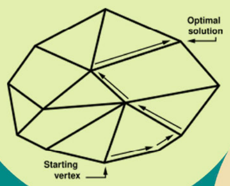


fides for
approximate TSP
codes
Christo
Gaussian
Elimination
ant colony
systems
genetic algorithms Lempel-Ziv
matrix multiplication
Distributed Hash Tables
codes algorithms, e.g.
Euclidean Algorithm
Gale-Shapley
Error-Correcting Codes
Approximation
Merkle
Trees algorith
in Reed-Solomon
Paxos
Dynamic
Programming
fides for
approximate TSP
codes Lovasz
Depth-First-Search
Balanced Trees
Sorting, Quicksort
Page rank
Hashing
Dijkstra
LLL
Heaps
FFT

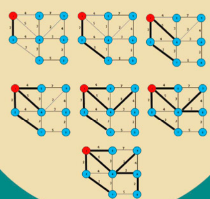
Lester Ford, Delbert Fulkerson
maximum flow



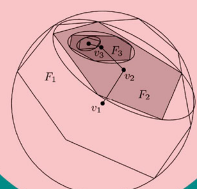
George Dantzig
simplex method



Edsger Dijsktra
shortest path



Leonid Khachiyan
ellipsoid method for LP



Algorithms (II)

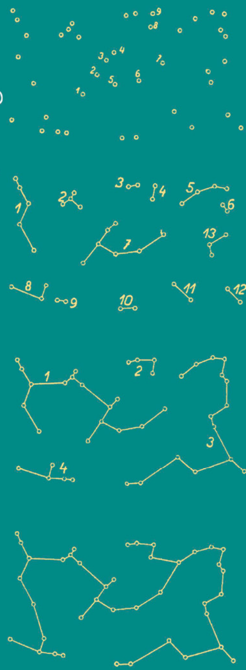
Minimum spanning tree: Borůvka and Jarník

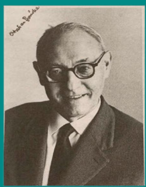
1926 **O. Borůvka,** Příspěvek k otázce ekonomické stavby elektrovodných sítí [Contribution to the solution of a problem of economical construction of electrical networks]. *Elektrotechnický obzor* 15, 1926, 153–154.

O. Borůvka, O jistém problému minimálním. *Práce Moravské přírodovědecké společnosti* sv. III, spis 3, 1926, 37–58 (in Czech, German summary).

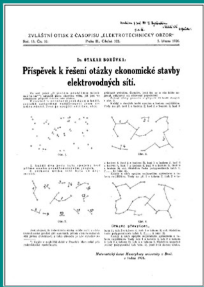
BUBLINKOVÝ BORŮVKŮV POSTUP

1. KAŽDÝ BOD JE BUBLINKA (JE TOLIK BUBLINEK JAKO JE BORŮVEK V LESE)
2. SPOJ KAŽDOU BUBLINKU S NEJBLIŽŠÍ BUBLINKOU
3. NAJDI NOVÉ BUBLINKY.
4. POKUD JSOU ALESPOŇ 2 BUBLINKY OPAKUJ 2.
POKUD JE POUZE JEDNA BUBLINKA JSI HOTOV.

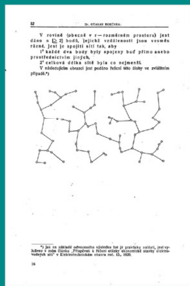




Otakar Borůvka



First page of O. Borůvka, Příspěvek k otázce ekonomické stavby elektrovodných sítí.



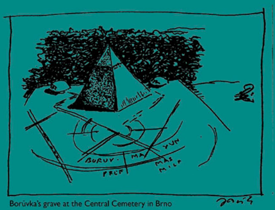
Last page of O. Borůvka, O jistém problému minimálním.

1923 scheme of South Moravian electric power line network

JARNÍK = PRIM

JARNÍK KÖSSLER = STEINER

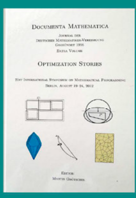
BORŮVKA = SOLLIN (1962!)



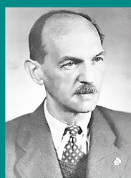
Borůvka's grave at the Central Cemetery in Brno

1930 **V. Jarník,** O jistém problému minimálním. (Z dopisu panu O. Borůvkovi) (Czech) [On a certain problem of minimization]. Práce moravské přírodovědecké společnosti 6, fasc. 4, 1930, 57–63.

While its typical implementation using data structures such as a priority queue does not achieve linear time, variations and improvements have led to algorithms that approach linear time complexity for certain graph structures Linearity of the MST problem is a long-standing open problem.



Jarník's formula for MST. O jistém problému minimálním.



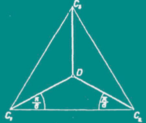
Vojtěch Jarník

First page of V. Jarník and M. Kössler, O minimálních grafech obsahujících *n* daných bodů

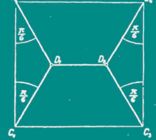
R. L. Graham and P. Hell, On the history of the minimum spanning tree problem, *Ann. Hist. Comput.* 7:1 (1985), 43–57
M. Mareš, The Saga of Minimum Spanning Trees, *Computer Science Review*, 2: 3, (2008,) 165–221
J. Nešetřil and H. Nešetřilová, The Origins of Minimal Spanning Tree Algorithms, *Documenta Mathematica*, Extra Volume ISMP (2012), 127–141

1934 **V. Jarník and M. Kössler,** O minimálních grafech obsahujících *n* daných bodů [On minimal graphs containing *n* given points]. *Časopis pro pěstování matematiky a fysiky* 63: 8 (1934), 223–235

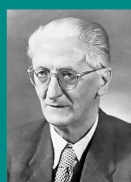
Jarník and Kössler introduce a problem – the first journal publication to do so – later called the Steiner problem, which they solve for regular *n*-gons.



Obr. 2.



Obr. 3.



Miloš Kössler

Advanced graph theory in TCS

Expander graphs

The tension between sparsity and connectivity in expander graphs makes them especially important to computer science, e.g. in constructing good pseudorandom number generators, derandomising probabilistic algorithms, constructing error-correcting codes, and in building probabilistically checkable proofs.

- RAMANUJAN GRAPHS ARE HIGHLY CONNECTED SPARSE LARGE GRAPHS.
- THE TENSION BETWEEN SPARSE AND HIGHLY CONNECTED IS WHAT MAKES THEM SO USEFUL IN VARIED APPLICATIONS.

1916
S. Ramanujan
On certain arithmetic functions

1956
J. von Neumann
Probabilistic logics and the synthesis of reliable organismsfrom unreliable components

1965
A. Selberg
On the estimation of Fourier coefficients of modular forms

1967
A. Kolmogorov and Y. M. Barzdin
On the realization of networks in three-dimensional space

1971
J. Cheeger
A lower bound for the smallest eigenvalue of the Laplacian

1973
M. Fiedler
Algebraic connectivity of graphs
M. Pinsker
On the complexity of a concentrator
G.A. Margulis
Explicit constructions of concentrators

1979
O. Gabber and Z. Galil
Explicit constructions of linear size superconcentrators

1986
N. Alon
Eigenvalues and expanders

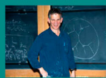
1988
A. Lubotzky, R.S. Phillips, P. Sarnak
Ramanujan graphs
G.A. Margulis
Explicit group-theoretical constructions of combinatorial schemes and their application to the design of expanders and concentrators

2000
O. Reingold, S. Vadhan, A. Wigderson
Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors

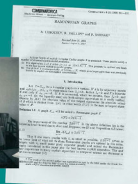
THEY EXIST!

DEFINITION (OPTIMAL EXPANDERS)
A GRAPH (SEQUENCE WITH $n \to \infty$) IS
a) RAMANUJAN IF FOR $j = 1, \ldots, n-1$
$$|\lambda_j(X_{n,d})| \leq 2\sqrt{d-1}.$$
b) BIPARTITE RAMANUJAN IF IT IS BIPARTITE $\lambda_{n-1}(X) = -d$ AND
$$|\lambda_j(X_{n,d})| \leq 2\sqrt{d-1} \quad \text{FOR } j = 1, \ldots, n-2.$$


Alexander Lubotzky


Peter Sarnak


Ralph S. Phillips




Grigory Margulis

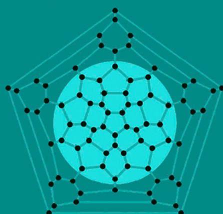
Over a century ago, Ramanujan made a deceptively simple conjecture about modular forms, subsequent work on which led to the development of sophisticated number-theoretic tools, Lubotzky, In1986-8 Phillips and Sarnak wielded these tools in the construction of Ramanujan graphs – an explicit family of expander graphs with optimal sparsity and connectivity.

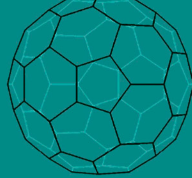
"The existence of expanders is counterintuitive. Very well-known mathematicians made conjectures which, once pruned down and understood, were saying essentially, 'Expanders don't exist.'" – P. Sarnak

The notion of an expander can be traced back to J, von Neumann's investigation into fault-tolerant circuits and Y.M. Barzdin and A. Kolmogorov's realization of networks demanding high connectivity subject to sparsity constraints. The latter authors come very close to producing expander graphs of uniformly bounded degree, and show in essence that a random graph is an expander. It wasn't until 1973 that the notion of expander was formally defined by M. Pinsker, who established their existence by a probabilistic argument: for a deterministic construction, Pinsker refers to a paper of G.A. Marsulis that appeared later that year, in which Margulis "*gave the first explicit construction of an infinite family of expander graphs. In that paper the proof was based on the theory of discrete subgroups of Lie groups. In particular I used arguments related to Kazhdan's work on property (T). This was probably unexpected for people working in computer science.*" (At that time Pinsker, Kazhdan, and Margulis were all members of the renowned IPPI in Moscow.)

In 1986 N. Alon and R. Boppana, building on the work of Fiedler and Cheeger, reformulated the measure of a graph's expansion properties in terms of its 'spectral gap.' Alon and Boppana showed that this gap was bounded, so the expansion ratio of a family of expanders could not exceed a certain threshold: they asked whether an optimal family of expander graphs could be found. Given this linear algebra reformulation of the graph's expansion measure, Lubotzky suggested that Sarnak and Phillips explore some seemingly apropos methods from number theory; exploiting a series of results related to the Ramanujan conjecture, they were able to produce an infinite family of optimal expander graphs – the Ramanujan graphs. Two years later, Lubotzky, Phillips and Sarnak published their result in *Combinatorica*.

At IPPI in the mid 1980s, around the time Margulis was constructing explicit families of expanders using quaternions and looking for high girth examples, he "*realized, based on some deep work by Deligne, that they were also expander graphs which are in a certain sense better than the previous constructions. Slightly later and completely independently, Lubotzky, Phillips and Sarnak gave basically the same construction, but with some variations.*"





This Ramanujan graph has 80 vertices, which is close to the largest known planar Ramanujan graph of 84 vertices. Its girth is 5, its expansion constant 1/4 (as indicated by the shaded circle), and λ_1 has been calculated by A. Gamburd to be 2.81811... It may be constructed by shrinking pentagons on a dodecahedron.

Publishing research

ICALP, a case study

editor's letter

DOI:10.1145/1506409.1506410

Moshe Y. Vardi

Conferences vs. Journals in Computing Research

An old joke tells of a driver, returning home from a party where he had one drink too many, who hears a warning over the radio about a car careening down the wrong side of the highway. "A car?" he wondered aloud, "There are lots of cars on the wrong side of the road!"

I am afraid that driver is us, the computing-research community. What I'm referring to is the way we go about publishing our research results. As far as I know, we are the only scientific community that considers conference publication as the primary means of publishing our research results. In contrast, the prevailing academic standard of "publish" is "publish in archival journals." Why are we the only discipline driving on the conference side of the "publication road?"

Conference publication has had a dominant presence in computing research since the early 1980s. Still, during the 1980s and 1990s, there was ambivalence in the community, partly due to pressure from promotion and tenure committees about conference vs. journal publication. Then, in 1999, the Computing Research Association published a Best Practices Memo, titled "Evaluating Computer Scientists and Engineers for Promotion and Tenure," that legitimized conference publication as the primary means of publication in computer research. Since then, the dominance of conference publication over journals has increased, though the ambivalence has not completely disappeared. (In fact, ACM publishes 36 technical journals.)

Recently, our community has begun voicing discomfort with conference publication. A Usenix Workshop on Organizing Workshops, Conferences, and Symposia for Computer Systems (WOWCS), held in San Francisco in April 2008, focused on the paper selection process, which is not working too well these days, according to many people. (You can find the proceedings at http://www.usenix.net/events/wowcs08/ and a follow-up wiki at http://wiki.usenix.org/bin/view/Main/Conference/CollectedWisdom.)

Two presentations at the workshop evolved into thought-provoking *Communications*' Viewpoint columns. In the January 2009 issue, we published "Scaling the Academic Publication Process to Internet Scale" by J. Crowcroft, S. Keshav, and N. McKeown (p. 27). In this issue, you will find "Program Committee Overload in Systems" by K. Birman and F.B. Schneider (p. 34). The former attempts to offer a technical solution to the paper-selection problem, while the latter points us to the nontechnical origins of the problem, expressing hope to "to initiate an informed debate and a community response."

I hope the outcome from WOWCS and the Viewpoint columns published here will initiate an informed debate. But I fear these efforts have not addressed the most fundamental question: Is the conference-publication "system" serving us well *today*? Before we try to fix the conference publication system, we must determine whether it is worth fixing.

My concern is our system has compromised one of the cornerstones of scientific publication—peer review. Some call computing-research conferences "refereed conferences," but we all know this is just an attempt to mollify promotion and tenure committees. The reviewing process performed by program committees is done under extreme time and workload pressures, and it does not rise to the level of careful refereeing. There is some expectation that conference papers will be followed up by journal papers, where careful refereeing will ultimately take place. In truth, only a small fraction of conference papers are followed up by journal papers.

Years ago, I was told that the rationale behind conference publication is that it ensures fast dissemination, but physicists ensure fast dissemination by depositing preprints at www.arxiv.org and by having a very fast review cycle. For example, a submission to *Science*, a premier scientific journal, typically reaches an editorial decision in two months. This is faster than our conference publication cycle!

So, I want to raise the question whether "we are driving on the wrong side of the publication road." I believe that our community must have a broad and frank conversation on this topic. This discussion began in earnest in a workshop at the 2008 Snowbird Conference on "Paper and Proposal Reviews: Is the Process Flawed?" (see http://doi.acm.org/10.1145/1462571.1462581).

I cannot think of a forum better than *Communications* in which to continue this conversation. I am looking forward to your opinions.

Moshe Y. Vardi, EDITOR-IN-CHIEF

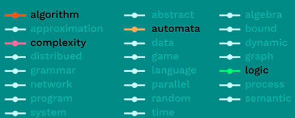
The outlet for computer science research continues to be through conferences and their proceedings: this was the case from the outset, when Theoretical Computer Science was beginning to be recognized as an adademic discipline, up to the present day.

"The reason conference publication is preferred to journal publication, at least for experimentalists, is the shorter time to print (7 months vs 1-2 years), the opportunity to describe the work before one's peers at a public presentation, and the more complete level of review (4-5 evaluations per paper compared to 2-3 for an archival journal). Publication in the prestige conferences is inferior to the prestige journals only in having significant page limitations and little time to polish the paper. In those dimensions that count most, conferences are superior."
— Computing Research Association, Best Practices Memo, 1999

But this way of doing things has not been without its critics. In 2009, M. Vardi observes in his editorial for the *Communications of the ACM* that fast dissemination is no longer an issue with on-line archives; the quality of reviewing is higher for journals than for conferences; and other academic fields use journals as their primary publication outlet.

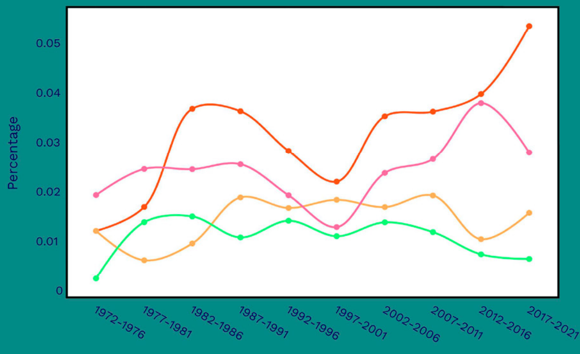
ICALP topics

Throughout its fifty-year history, ICALP has provided a broad coverage of topics in Theoretical Computer Science. How has the relevance of research topics within the theoretical-computer-science community changed since 1972?



ICALP publications

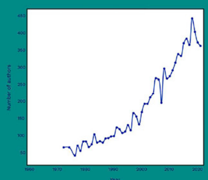
In the fifty years since its inception, the ICALP conference has evolved in pace with the scientific advances and the growth and maturation of the theoretical computer science community. An analysis of DBLP data gives a bird's-eye-view of that evolution.



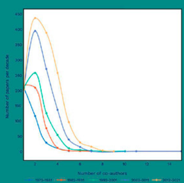
Percentage of ICALP papers whose titles mention the word algorithm, complexity, automata, or logic.

ICALP authorship

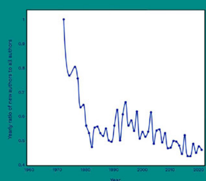
Like other major conferences in Theoretical Computer Science, the authorship at ICALP tends to stabilise over time.



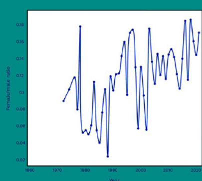
Number of authors per year, corresponding to a similar evolution in the number of accepted papers.



Number of papers with each co-authorship size, per decade. Papers with two, three, and even four authors have gradually become more common than single-author papers.



Percentage of new authors per year. Every year, approximately half the authors at TCS conferences are newcomers to the conference.



Ratio of women over men among authors. Still below 0.2 for almost all TCS conferences in 2021.

Selected textbooks in TCS

1966-

-1999

Exhibition curators

Dr. Andrew Goodall studied at the University of Oxford and since 2012 has been working at the Computer Science Institute of Charles University at MFF. He works mainly in combinatorics and algebra. He is known also for his photography, having had several exhibitions in Prague.

Prof. Jaroslav Nešetril is employed at the Computer Science Institute of Charles University at MFF. He works in many areas of mathematics and computer science. He collaborated with Jiří Načeradský for 20 years and together they created an extensive oeuvre (see, for example, J. Načeradský, J. Nešetřil: Antropogeometrie I, II, Rabasova Galerie 1998, ISBN 80-85868-25-3).

This catalogue was published by DIMATIA-IUUK MFF UK on the occasion of the exhibition *The Making of a New Science* held at Galleria Chodba, Malostranské nam. 25, Praha 1, from 18 June to 21 September 2025.