



Jacob T. Schwartz

JACOB T. SCHWARTZ

1930–2009

Elected in 2000

“For contributions to the theory and practice of programming language design, compiler technology, and parallel computation.”

BY BUD MISHRA

SUBMITTED BY THE NAE HOME SECRETARY

JACOB T. SCHWARTZ, a mathematician and computer scientist from New York University’s (NYU) Courant Institute and a polymath with pioneering and seminal contributions to a wide spectrum of subjects ranging from pure mathematics to applied engineering, died on March 2, 2009, at the age of 79.

Jack founded NYU’s Department of Computer Science, chairing it from 1964 to 1980 and headed DARPA’s (Defense Advanced Research Projects Agency) IPTO (Information Processing and Techniques Office) program from 1987 to 1989. He was the designer of the SETL programming language; the NYU Ultracomputer; the NYU Four-Finger Manipulator robot; a novel multimedia visualizer, the Fractal Computer User Centerface; and COMBAT, a whole genome sequence comparison tool.

Jack’s research interests were wide ranging: the theory of linear operators, von Neumann algebras, quantum field theory, theory of money, time sharing, parallel computing, programming language design and implementation, motion planning for robots, dexterous manipulation and grasping with robot hands, set-theoretic approaches in computational logic, proof and program verification systems, multimedia authoring

tools, experimental studies of visual perception, multimedia, and algorithms for analysis and visualization of genomic DNA sequences. He authored 18 books and more than 100 papers and technical reports. His students included Jerry Hobbs, Ken Kennedy, Robert Kupperman, Stanley Osher, Gian-Carlo Rota, Shmuel Winograd, Clyde Kruskal, and Larry Rudolph. His collaborators included Nelson Dunford (advisor), Martin Davis, John Cocke, Frances Allen, Ralph Grishman, Robert Dewar, Edmond Schoenberg, Alan Gottlieb, W. Daniel Hillis, Micha Sharir, Bud Mishra, Domenico Cantone, Alfredo Ferro, Eugenio Omodeo, Kenneth Perlin, and Michael Wigler.

Jack was elected to the National Academy of Engineering in 2000 for contributions to the theory and practice of programming language design, compiler technology, and parallel computation. Since his election, he made active contributions to various additional subfields of engineering: robotics with connections to mechanical engineering; multimedia and visualization technologies; and genomics, bioinformatics, and biotechnology. Jack was elected to the National Academy of Sciences in 1976. His honors included Sloane Fellow, Wilbur Cross Medal (Yale University), Townsend Harris Medal (City University of New York), Steele Prize (American Mathematical Society), and Mayor's Medal for Contributions to Science and Technology (New York City).

Jack was born to a working-class, immigrant family in the Bronx, New York, on January 9, 1930. He received his bachelor of science degree in 1949 from the City College of New York and his master's degree in 1949 and Ph.D. degree in 1951 from Yale University. He began his career as a mathematics instructor at Yale and in 1953 was promoted to assistant professor. In 1957, Schwartz joined the faculty at NYU as an associate professor and in 1958 was promoted to the position of full professor. He retired in 2005 but remained a professor emeritus of computer science at the Courant Institute, until his death. He also served as chairman of the Computer Science Board of the National Research Council and was chairman of the National Science Foundation Advisory Committee for Information, Robotics, and Intelligent Systems.

In 1958, as a mathematics graduate student at Yale, Jack collaborated with his Ph.D. advisor Nelson Dunford to co-author a three-volume treatise, *Linear Operators*, which has since become the standard text in the field and has remained continuously in print. Soon after publication of the book, in 1960s, Jack turned his attention to the field of computer science, which was still in its embryonic state, without a well-formed foundational theory or a well-tested engineering basis. By connecting the goals of computer science (in being able to perform computations with both efficiency in time and space use as well as in achieving accuracy rigorously) to pure and applied mathematics, Jack began a novel and unique style of problem solving in computer science: He showed how the design of programming language can be connected to set theory, program verification and compiler optimization to theorem proving, design of data motion in a parallel computer to group theory, robot motion planning to algebraic geometry, and dexterous manipulation to convexity theory.

For instance, when Jack became interested in the field of parallel computation in the 1960s, he started by focusing on the design of scalable shared-memory architecture and converged on an elegant mathematical solution by the late 1970s. In 1980 he wrote the influential paper, "Ultracomputers," published in October of that year in the *ACM Transactions on Programming Languages and Systems*. His solution was based on the notion of an ideal shared-memory parallel computer, which was based on a shuffle-exchange network and could be implemented to a good approximation in an architecture he dubbed Ultracomputer. Many of the fundamental building blocks of this architecture, most notably the use of "fetch and add" operation as a scalable synchronization primitive, have become ubiquitous in almost all parallel computing systems.

He created the NYU Ultracomputer project to implement this parallel architecture and actively encouraged industry to collaborate on research on parallel computation. For instance, in the early 1980s he collaborated with IBM to initiate the IBM RP3 project, which led to a 64-node prototype that was used for research in parallel OS, parallelizing compilers, and parallel

applications; as director of DARPA/IPTO, he influenced the research agenda on parallel computing; and he became a consultant for the Thinking Machines Corporation, an early maker of massively parallel thinking machines.

Another beautiful example of this style of research occurs in Jack's work on robot motion planning, which appeared in a series of papers, all entitled "On the Piano Mover's Problem" and all co-authored with his computational geometer colleague Micha Sharir. There the problem they tackle concerns a robot that wishes to move from a source configuration (say, consisting of a position and an orientation) to a destination configuration but must move through a space cluttered with obstacles while avoiding them. The simplest version of the problem is in a three-dimensional space, where the robot and the obstacles are all polyhedral (geometric solids with straight edges and flat faces); this is the same problem that a "piano mover" faces when removing a polyhedral piano from a rectoidal New York apartment, furnished disorderedly with polyhedral furniture. The basic idea is to simply consider all possible configurations of the robots (say, all possible ordered pairs of positions and orientations) in a configuration space (a six-dimensional space of positions and orientations), which can then be divided into admissible (those configurations not obstructed by any obstacles) and inadmissible (those configurations obstructed by some obstacles) components. Because of the algebraic nature of the configurations, there are only finitely many admissible and inadmissible components, and their boundary is described by algebraic equalities and inequalities (the so-called semialgebraic sets).

All these ideas can be expressed mathematically as sentences in Tarski algebra and then verified algorithmically. In particular, the sentence that asks whether there is a semialgebraic connected component containing both the source and destination configurations also answers the piano mover's problem. This elegant solution not only energized the research in robot motion planning but also rekindled an interest in "algorithmic algebra," which has since found many other applications of similar nature: geometric theorem

proving, computer-aided design, model checking in systems biology, and so forth. Jack seemed to have returned over and over to the core essence of mathematics to look for simple and elegant solutions to a diverse set of applications.

In an essay on computer science, Jack pithily described what he considered to be this essence of mathematics. “To find the simple in the complex, the finite in the infinite,” Jack wrote, “that is not a bad description of the aim and essence of mathematics.” To find the simple in the complex is not a bad description of what Jack’s work continues to represent to his friends, colleagues, mentees, advisees, and students.

Jack’s personal interests were wide and varied and included etymology, world music, music composition using software, middle school mathematics curriculum development, 3-D vision, Paleolithic stone implements, visual illusions, knots, extensive reading and cooking. In his final year Jack became interested in the basics of geometric lens optics and delved into the subject deeply enough to have started a paper on the topic. Up until one month prior to his death he was still working on transfer methods using wood, marble and clay.

Besides his wife, Diana, survivors include his daughters, Abby Schwartz of Manhattan and Rachel Fainman of Winnipeg, Manitoba; two grandchildren, Adam and Adrienne Fainman of Winnipeg, Manitoba; and a sister, Judith Dunford, the widow of the literary critic Alfred Kazin.