Bush, Robert R., Eugene Galanter, and R. Duncan Luce. *Handbook of Mathematical Psychology*. New York: Wiley, 1963.

Fodor, Jerry, and J. Katz, eds. *Readings in Philosophy of Language*. Englewood Cliffs, N.J.: Prentice-Hall, 1964.

George, Alexander, ed. *Reflections on Chomsky*. Oxford: Blackwell, 1989.

—*Robert Barsky*

Church, Alonzo

1903–1995

U.S. Mathematician

Alonzo Church was one of the greatest logicians of the twentieth century and made many contributions to mathematics, including creation of the lambda calculus. Church was a professor of mathematics at Princeton University when in 1936 he solved a long-standing problem in the foundations of mathematics.

At the turn of the century, the German mathematician David Hilbert (1862–1943) posed what would eventually be called the set of *Hilbert problems*. These were open questions whose solutions, in Hilbert's opinion, would greatly assist the further advance of mathematics in the twentieth century. The tenth Hilbert problem was concerned with the decidability of mathematics—that is, if it is possible to find a "mechanical procedure" that, starting from the axioms of mathematics, could prove or refute any given mathematical proposition. (In fact, the original formulation of the problem given by Hilbert was less general, but in the following years he arrived at the formulation mentioned above.)

Church was able prove that the mechanical procedure, which Hilbert considered at least conceivable, did not exist. In the language of formal logic, arithmetic is "undecidable": it is not possible to write a mechanical procedure (i.e., a computer program) that can decide for *any* given arithmetical proposition, in every conceivable case, whether the proposition is true or false. Particular cases can be tested by programs; a program can be written to verify if "$1 + 1 = 2$" or if "$2 - 1 = 1$," but no matter how clever or complex the program is, the possibility of asking an arithmetical question for which the program will have no answer will always remain.

The main difficulty of disproving Hilbert's conjecture resides in defining precisely what is meant by a *mechanical procedure*. In 1936 there were no computers available and it was not easy to see how to formalize every step of a proof. **Alan Turing** (1912–54), who also proved the undecidability of arithmetic in 1936, followed a different path than Church: He defined a mechanism, the **Turing machine**, which resembles a computer and which is universal in the sense that it can implement any conceivable effective computing procedure. Church anticipated Turing's results by a few months, but by using a symbolic approach. He postulated that any effectively computable number/theoretic function can be expressed in the formalism of recursive functions. This is now called the "Church Thesis." Based on the theory of recursive functions, Church could show that arithmetic in undecidable.

An important contribution of Church's was the development of the lambda calculus, which is at the base of the functional programming languages developed decades later, like **LISP** or ML. In the lambda calculus, complex expressions are built out of simpler expressions. The two main concepts are function and function application. By proceeding recursively, functions can be applied to functions, and any computation can be expressed in this abstract language. The data on which lambda expressions act are also lambda expressions. Numbers, for example, are represented by functions in the lambda calculus.

BIOGRAPHY

Alonzo Church. Born 14 June 1903 in Washington, D.C. Received A.B. in mathematics from Princeton University, 1924; Ph.D., 1927, also from Princeton. Postdoctoral work in Göttingen, Germany, 1928–29. Professor of mathematics at Princeton, 1929–67. Solved Hilbert's tenth problem, 1936. Professor of Mathematics and philosophy at UCLA, 1967–90. Member of the American Academy of Arts and Sciences, the U.S. National Academy, and the British Academy. Died 11 August 1995 in Hudson, Ohio.

SELECTED WRITINGS

Church, Alonzo. *The Calculi of Lambda Conversion*. Princeton, N.J.: Princeton University Press, 1941.

———. *Introduction to Mathematical Logic*. Princeton, N.J.: Princeton University Press, 1944.

FURTHER READING
Davis, Martin. *The Undecidable.* Hewlett, N.Y.: Raven Press, 1965.
Kleene, S. C. "Recursive Function Theory." *Annals of the History of Computing.* Vol. 3, No. 1, 1981, pp. 52–67.

—*Raúl Rojas*

CIM See Computer Integrated Manufacturing.

Cisco Systems

Cisco Systems builds the switching devices that make the **Internet** a network. Nearly every time someone sends an **electronic mail**, looks at a **World Wide Web** page, or downloads a music file, their bits go through a Cisco **router**, switch, hub, or server somewhere along the line. Cisco dominates these core businesses in much the same fashion as **Intel** dominates personal computer chips and **Microsoft** dominates personal computer **operating systems** and office applications.

From its inception in 1985, Cisco grew to become one of the world's great corporations in a shorter time than any company before it—and in the process made thousands of people wealthy, from savvy Wall Street investors to individual engineers and coders in the company. The value of the company's stock at its initial public offering in 1990 grew 100-fold in a decade.

Cisco's internal culture and public image stands in vivid contrast to that of Microsoft. If Microsoft's software is often criticized as bloated, buggy, and vulnerable to security problems, Cisco's hardware, its Internet Operating System (IOS), and other code are seen as reliable and secure. If Microsoft has been seen as the "evil empire," a vindictive giant set on winning by forcing the industry into submission, Cisco has the opposite reputation: open and collaborative, bent on growing by responding to the customer. Despite its near-monopoly status in its core markets, Cisco has never suffered civil lawsuits or federal sanctions for anticompetitive practices.

Cisco's rapid growth has come despite two changes in top management and three distinct growth stages. The founders, Leonard Bosack and Sandra Lerner, had met and married on the Stanford University campus in Palo Alto, California, the heart of **Silicon Valley**. Bosack, a computer engineer, would become head of computer services for the university, while Lerner become head of the computer services for its business school. They originally developed what would become their core product—a router capable of speaking and translating different computer protocols—to allow different departments' computers to communicate. When the university refused to allow them to license the technology to other institutions and corporations, they quit and founded Cisco.

They financed the company on credit cards and through a mortgage on their house, building the first routers and writing code with the help of friends in their bedroom and living room, and getting their first customers through what was probably the first commercial **spam** on the nascent Internet. Corporations and institutions were desperate for some way to connect their departments into compatible networks; other than the first spam announcement, the company did no advertising for its first seven years.

In 1986, Lerner and Bosack sought venture capital. Seventy-five firms turned them away. Finally, Don Valentine of Sequoia Capital, who had also invested in **Apple Computer**, invested U.S.$2.5 million, and in return, the founders gave him 32 percent of the company. When Cisco went public in 1990, Valentine's U.S.$2.5 million investment turned out to be worth over U.S.$10 billion. The investment also gave Valentine effective control and the right to appoint new management at will. Valentine appointed John Morgridge as the new chief executive officer.

Neither of the founders had run a business before, and Lerner had a famously aggressive and abrasive management style. In August 1990, all seven of the company's vice-presidents signed letters of resignation effective immediately unless Lerner left. In short order, both Lerner and Bosack were gone. Ironically, the company had never used the U.S.$2.5 million from Sequoia—its cash flow had been sufficient to sustain growth. But the change of management was probably necessary to Cisco's growth.

Morgridge's style was tight, but familial. He encouraged confrontation and argument in meetings, yet he was often wildly funny as well. He was tight-fisted with