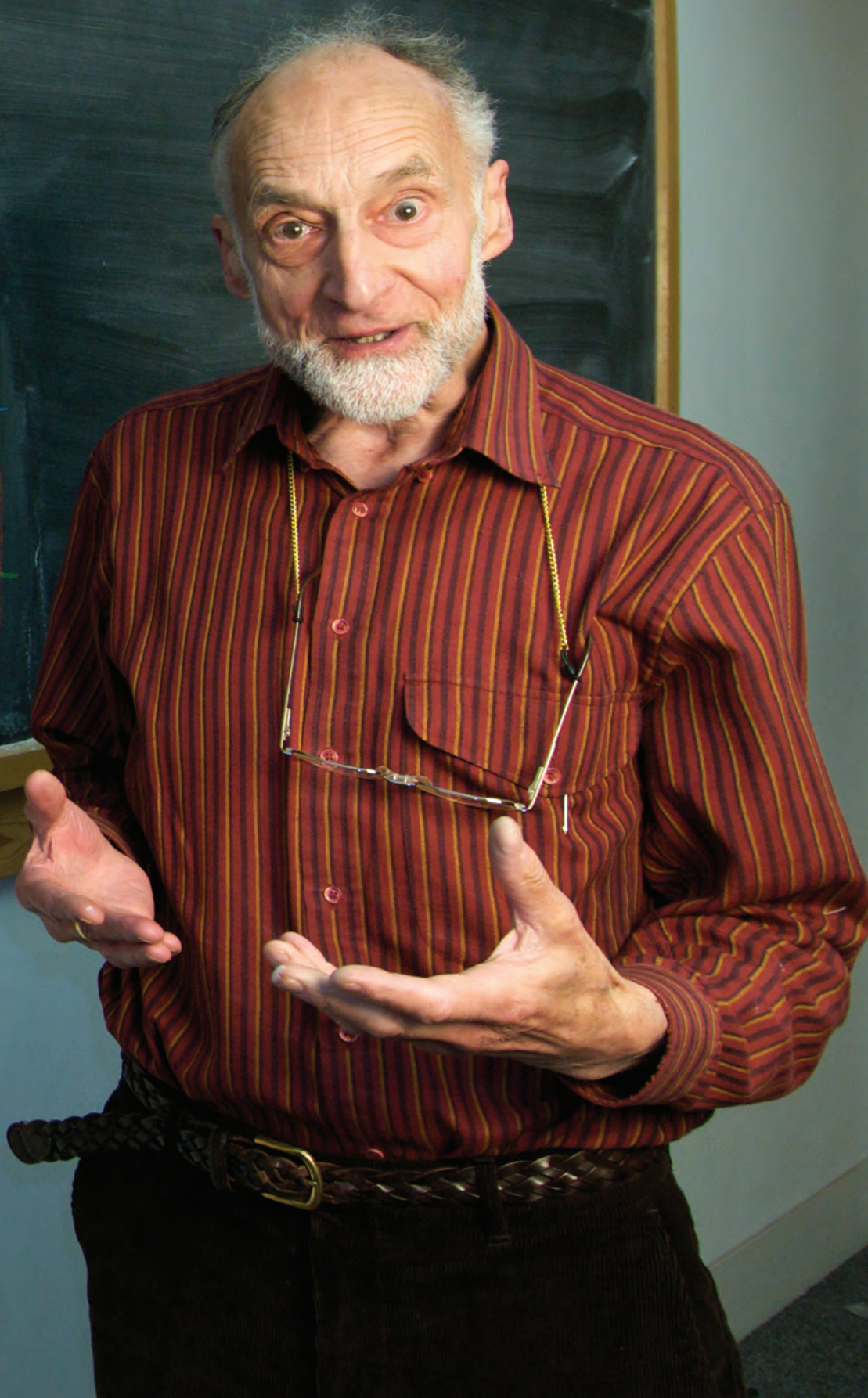
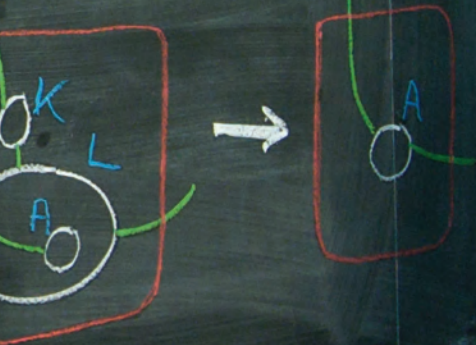


A key  $K$   
unlocks a lock  $L$



# Robin Milner: The Elegant Pragmatist

*Remembering a rich legacy in verification, languages, and concurrency.*

**I**T CAME AS a surprise even to those who knew him well: the death of Arthur John Robin Gorell Milner, known to friends simply as Robin, of a heart attack on March 20. Milner's wife, Lucy, had died three weeks before. Yet the pioneering computer scientist remained active, and seemingly in good health, until the end, maintaining close connections with his colleagues and even co-authoring a paper with a postdoctoral student he supervised at the IT University of Copenhagen.

A man of modest background and quiet brilliance, Milner made groundbreaking contributions to the fields of verification, programming languages, and concurrency. He was born in 1934 near Plymouth, England, and won scholarships to Eton—where he developed an enduring love of math as well as a prodigious work ethic—and King's College, Cambridge. It was during his time at Cambridge that Milner was introduced to programming, though the subject didn't interest him initially. "I regarded programming as really rather inelegant," he recalled in an interview in 2001 with Martin Berger, a professor at the University of Sussex. "So I resolved that I would never go near a computer in my life." Several years later, in 1960, Milner broke that resolution with a programming job at Ferranti, a British computer company; from there, it wasn't long before he moved to academia with positions at City University London, Swansea University, Stanford University, and the Universities of Edinburgh and Cambridge.

Inspired by Dana Scott's famous formulation of domain theory, Milner began working on an automatic theorem prover, hoping to find a way to mechanize a logic for reasoning about programs. The work culminated in the development of Logic for Computable

**"Robin initiated a revolution in what computing languages are and could be," says Robert Harper.**

Functions (LCF), an interactive system that helps researchers formulate proofs. "It was a novel idea," says Robert Harper, a professor of computer science at Carnegie Mellon University. "The approach before then was that computers would search for your proof. Robin recognized that the computer is a tool to help you find the proof." During that research, Milner also laid the foundations of ML, a metalanguage whose original intent was to enable researchers to deploy proof tactics on LCF. The innovative concepts it introduced, however—such as polymorphic type inference and type-safe exception handling—were soon recognized. Ultimately, ML evolved into a powerful general programming language (with Milner, Harper, and others working to specify and standardize it) and led to languages like F# and Caml.


"Robin initiated a revolution in what computing languages are and could be," asserts Harper.

Although he was closely involved in ML's development throughout the 1980s and 1990s, Milner also began working on the problem of concurrency, looking for a mathematical treatment that could rival theories of sequential computation. The Calculus of Communicating Systems (CCS) was the first solution he devised: a process calculus for a network that was

programmed to cooperate on a single task. CCS was succeeded by a more general theory of concurrency called pi calculus, which incorporated dynamic generation, and bigraphs, a theory of ubiquitous computing. For his work on LCF, ML, and CCS, Milner received the ACM A.M. Turing Award in 1991.

"Robin had this ability to think large and translate that all the way down to new theorems," says Mads Tofte, vice chancellor of the IT University of Copenhagen and a former graduate student. The sentiment is echoed by many of Milner's collaborators, who cite his unfailing ability to shift from a grand vision of computing to subtle mathematical particulars. Coupled with his rigorous attention to detail, the trait gave Milner's work a firm grounding in practical applications. "He was very concerned with making things work," says Gordon Plotkin, a former colleague at the University of Edinburgh.

Milner also labored on behalf of institutions and initiatives that shaped the future of the field. He helped establish the Laboratory for Foundations of Computer Science at the University of Edinburgh and served as the first chair of the University of Cambridge's Computer Laboratory. In 2002, Milner and Tony Hoare, a senior researcher at Microsoft Research in Cambridge (and a fellow Turing Award winner), began working on a grand challenge initiative to identify research topics that would drive science in the 21st century.

He was a natural leader, according to Tofte. "I'm not sure he was always aware of it," says Tofte, "but he was so good at getting the best out of people, which is exactly what a good leader does." 

Leah Hoffmann is a Brooklyn, NY-based technology writer.

© 2010 ACM 0001-0782/10/0600 \$10.00