DOI 10.1007/s00165-010-0162-3
BCS © 2010
Formal Aspects of Computing (2010) 22: 485–487

Formal Aspects of Computing

Robin Milner: 13 January 1934–20 March 2010

Gordon Plotkin, Colin Stirling and Mads Tofte

Photo by Eaden Lilley

Robin Milner, who died of a heart attack at the age of 76, was a foremost computer scientist who became a Fellow of the Royal Society and received the Turing Award. His remarkable influence continued right up to his untimely death. Computer science would not be the same without his contribution; one thinks particularly of that part of our subject based on logic. Robin gave us his LCF system for manipulating assertions; his ML programming language for organising computations; and CCS, the π-calculus, and bigraphs for working with complex concurrent systems.

Robin was born at Yealmpton, near Plymouth, as Arthur John Robin Gorell Milner. He won scholarships to Eton College and to King's College, Cambridge. Before going to Cambridge in 1954 he did 2 years' national military service. He gained a first-class degree in mathematics after 2 years and then in 1956 studied philosophy (the moral sciences Tripos).

He went to London in 1958, wondering whether to take up music seriously, having done a lot of singing and oboe playing, some cello, and some composition while at Cambridge. He decided otherwise, and after teaching mathematics for a year, he took a programming job at Ferranti, in London. In 1963 he moved to a lectureship in mathematics and computer science at City University. In the same year he married Lucy. During the next 5 years while they lived in London their three children Gabriel, Barney, and Chloë were born.

At City he formed a deep interest in automata theory, programming languages, artificial intelligence, and the relationship of logic with computation. In 1968 he took a research position at Swansea University under David Cooper working on program correctness. There he learnt of Dana Scott's celebrated work with Christopher Strachey on programming language semantics.

In 1971 Robin moved to Stanford University as a researcher, joining John McCarthy's group at the Artificial Intelligence Project. He took up Scott's ideas as the basis of a system for program correctness, resulting in the Stanford LCF system—LCF stands for Scott's 'Logic of Computable Functions.' He also began his seminal work on concurrency, again in the tradition of Scott and Strachey, formulating a mathematical notion of process to model the behaviour of interacting computing agents. This work notably included a discussion of full abstraction at a general level.

In 1973, he was appointed to a lectureship at Edinburgh University, obtaining a Personal Chair in 1984. Edinburgh LCF was a development of the Stanford work, but now with a specially designed programming language, Edinburgh ML (or 'MetaLanguage'). The thought was that the language's type system, in particular its abstract type mechanism, would offer guarantees that, however computed, if a result had type 'theorem' it was indeed a theorem.

Reproduced with the permission of the EATCS.

In this way Milner and his group became pioneers of large-scale computer-assisted theorem proving. There, rather than attempting to provide automatic theorem proving abilities, one provides users with the means to write and employ strategies, called 'tactics', to search for theorems. LCF has influenced other proof assistants, such as Mike Gordon's HOL (Higher Order Logic), Larry Paulson's Isabelle, and Bob Constable's Nuprl; modern systems now routinely handle large systems of industrial importance.

As is evident from its name, the programming language ML was originally conceived of as a language for manipulating formal systems. However, it is assuredly of independent interest. It is a higher-order call-by-value language with side effects and exceptions. The type system stands out from among its many other features. Based on the Hindley–Milner polymorphic type checking algorithm, it enables users to be largely freed from specifying types, while nonetheless providing a fully type-checked language: the compiler calculates the missing types.

In an additional research effort, ML was developed much further by Robin and his many collaborators, incorporating ideas of Rod Burstall and his group, including parametric modules for programming-in-the-large—a contribution of David MacQueen. The result, in 1989, was the very influential Standard ML, perhaps the first large-scale functional programming language invented after LISP. It is especially notable for having precise semantics, given operationally. A revised and simplified definition was given 7 years later, incorporating several improvements.

ML has developed into a widely used language, both academically and commercially; it has several implementations, exists in several variants, and has several successors, e.g., Moscow ML, OCaml, and F#. It has also had much influence on the further development of functional programming languages.

His greatest research contribution was, perhaps, devoted to concurrency. At Edinburgh, and also while a Guest Professor at Aarhus University, he invented the Calculus for Communicating Systems (CCS), following on his initial work at Stanford. CCS enables one both to define and to specify interacting systems, where the communications between systems are synchronised, and the communication architecture is static. The creation of CCS was a long exploratory process, beginning with an algebra and ending with a small calculus, and moving from a denotational semantics to a structural operational semantics, based on labelled transition systems. As well as making the transition, now standard in the field, to operational semantics, Robin, jointly with David Park, introduced the central idea of bisimulation, whose equational theory he axiomatised. He also, with Matthew Hennessy, made the connection with modal logic, their Hennessy–Milner logic.

There then began a yet longer process of exploration of the world of concurrency which was to continue throughout the rest of Robin's career. In order to model mobile computing there was a pressing need to relax the static communication structure of CCS and move to a variable architecture, where communication links could themselves be passed between processes. This resulted in the π-calculus, developed together with Joachim Parrow and David Walker, and strongly influenced by previous work of Mogens Nielsen and Uffe Engberg. The main technical innovation of the π-calculus was, perhaps, the recognition of the need for a theory of names; as regards its semantics, an interesting move was the replacement of structurally defined labelled transition systems by structural congruence and reduction.

In 1995, Robin left Edinburgh to take up a chair at Cambridge (in fact the first established chair in Computer Science at Cambridge) and was head of department there from 1996 to 1999; he was then Research Professor before becoming Emeritus in 2001.

By the early 90's there was a plethora of calculi for concurrency, e.g., for modelling time, probability, location, and priority; further, the π-calculus allowed the representation of other calculi, particularly λ-calculi. So it was natural, rather than seeking a single, canonical, calculus, to look for a canonical framework for calculi that provides a uniform account of their syntax and dynamics.

Robin's first attempt at such a framework was his action calculi, begun while at Edinburgh and continued at Cambridge. Then, by freeing himself from the restrictions caused by the syntax of action calculi, he arrived at his final contribution to concurrency: his bigraphical model. A bigraph has two graphs on one set of nodes, and the dynamics are given by a set of transition rules. One graphical structure, the place graph, specifies the structure of a system; the other, the link graph, specifies the interconnections between the different parts. This work recalls the flowgraphs used for the static part of CCS, but provides a much richer formalism that directly encodes all his and many other previous calculi, as well as other graphical models of computation, such as Petri nets.

As with all his other contributions, Robin's work on concurrency has led to much further theoretical and practical research. For example, CCS led to work on realtime and embedded computation; the π-calculus led to work on cryptographic protocols and business process modelling; and bigraphs have already found application in computational systems biology.

Robin remained active until the end, continuing to promote the integration of advanced application with applicable theory energetically. His most recent work is his theory of bigraphs, published as a book 'The Space and Motion of Communicating Agents' by Cambridge University Press in 2009, and partly written while holding a Blaise Pascal chair held at École Polytechnique.

During the final year of his life he had resumed his connection with Edinburgh as a part-time professor, presenting courses on bigraphs. He had many plans: bigraphs were going to be further developed, working with people at Edinburgh and Copenhagen. He had had initial ideas for a theory of ubiquitous computing, and was going to visit the newly founded Institute of Science and Technology at Vienna to explore them further.

Robin contributed notably to our community. For example, in 1986 he was one of the founding members and the first director of the Laboratory for Foundations of Computer Science at Edinburgh. In 1996 he chaired the UK RAE (Research Assessment Exercise) panel for computer science. In 1997 he chaired the committee that led to the establishment of the Division, later School, of Informatics at Edinburgh University, bringing together the Department of Artificial Intelligence, the Centre for Cognitive Science, and the Department of Computer Science. With C.A.R. Hoare, he was a prime mover for identifying Grand Challenge Projects for the twenty-first century in computer science; he is particularly identified with the theme of ubiquitous computing, to which his work on bigraphs contributes.

Robin's research forms a very coherent body of work, with one idea or system or calculus, leading to another in what seems to be an entirely natural, if miraculous, progression. His research has had great influence on others, both direct and indirect and he received well-deserved recognition.

In 1987 he and his collaborators won the British Computer Society Technical Award for the development of ML. In 1 year, 1988, he became a founder member of Academia Europaea, Distinguished Fellow of the British Computer Society, and Fellow of the Royal Society. In 1991 he was accorded the ultimate accolade of the Turing Award. In 2005 he became Foreign Member of the Académie Francaise des Sciences, and in 2008 he became Foreign Associate of the US National Academy of Engineering. He also received many other honours, including nine honorary doctorates, among them ones from Chalmers University and the Universities of Aarhus, Bologna, and Paris Sud, and also, in 2004, the Royal Medal of the Royal Society of Edinburgh.

The coherence and strength of Robin's research can in part be attributed to a clear philosophical position: that computer science, like other sciences, has both an experimental and a theoretical aspect. The experimental one resides in computing systems in which humans, software, or hardware interact. These systems have a rich behaviour, demonstrate interesting and practically important phenomena, and provide the experiments at hand. The theoretical aspect is the provision of structural concepts and mathematics originating in the design and use of computing systems; these, in their turn, are crucial for the design of systems and make their analysis possible. In this way, the development of theoretical models is united with their experimental application.

Although a modest man who found enormous joy in discourse, Robin was a strong and persuasive leader, remarkably gracious and extremely encouraging to students, researchers, and colleagues. Lucy died just 3 weeks before Robin; they are survived by Barney and Chloëh, and two grandchildren, Amy and Jade.

Gordon Plotkin, Colin Stirling and Mads Tofte
May 2010.

Published online 18 August 2010