*Journal of Universal Computer Science, vol. 23, no. 1 (2017), 1-4*
*submitted: 16/1/17, accepted: 20/1/17, appeared: 28/1/17 © J.UCS*

# Functional Programming: Past, Present, and Future. Contributions dedicated to D. A. Turner on his 70th Birthday

# J.UCS Special Issue

**Rafael Dueire Lins**
(Centro de Informática, Universidade Federal de Pernambuco, Recife, Brazil
and
Dept. de Estatística e Informática, Universidade Federal Rural de Pernambuco, Recife, Brazil
rdl.ufpe@gmail.com)

David Turner´s contributions to functional programming language design and implementation were seminal. He is perhaps best known for his pioneering work in combinator graph reduction and for the design and implementation of an influential series of pure, non-strict, functional programming languages: SASL, KRC and Miranda. David invented or co-invented many of the basic techniques of lazy functional programming and his ideas and notations have passed into later languages such as Haskell.

The papers in this special issue are all dedicated D.A.Turner on his 70th birthday and relate to his many contributions to the field. All the papers have been refereed to J.UCS standard. A short scientific biography of David precedes the papers.

**Board of Reviewers**

Many people have helped in various ways in the development of this volume. Among them special gratitude is due to the board of reviewers listed below.

Paul Bailes, The University of Queensland, Australia
Luis Soares Barbosa, Universidade do Minho, Portugal
Mariza Bigonha, Universidade Federal de Minas Gerais, Brazil
Roberto Bigonha, Universidade Federal de Minas Gerais, Brazil
Richard Bornat, Middlesex University, England, U.K.
Edwin Brady, The University of St Andrews, Scotland, U.K.
Francisco Heron de Carvalho Junior, Universidade Federal do Ceará, Brazil
Fernando Castor, Universidade Federal de Pernambuco, Brazil
Francesco Cesarini , Erlang Solutions Ltd, London, U.K.
Sergiu Dascalu, University of Nevada, Reno, U.S.A.
Roy Dyckhoff, The University of St Andrews, Scotland, U.K.
Kevin Hammond, The University of St Andrews, Scotland, U.K.
Robert Harper, Carnegie Mellon University, U.S.A.
Ralf Hinze, Radboud University, Nijmegen, The Netherlands

Gareth Howells, The University of Kent, England, U.K.
John Hughes, Chalmers University, Sweden
Roberto Ierusalimschy, Pontifícia Univ. Católica do Rio de Janeiro, Brazil
Delia Kesner, Université Paris Diderot, France
Rafael Dueire Lins, Universidade Federal de Pernambuco, Brazil
Conor McBride, University of Strathclyde, Scotland, U.K.
Greg Michaelson, Heriot-Watt University, Scotland, U.K.
Peter Mosses, Swansea University, Wales, U.K.
Alberto Pardo, Universidad de la Republica, Uruguay
Simon Peyton Jones, Microsoft Research, England, U.K.
João Saraiva, Universidade do Minho, Portugal
Mary Sheeran, Chalmers University, Sweden
Simon Thompson, The University of Kent, England, U.K.
Phil Trinder, University of Glasgow, Scotland, U.K.
Rosita Wachenchauzer, Universidad de Buenos Aires, Argentina
Philip Wadler, University of Edinburgh, Scotland, U.K.

**Acknowledgements**

The editor and the authors of this volume are grateful for the enthusiasm of Prof. Dr. Christian Gütl and Dana Kaiser that made it possible.

Rafael Dueire Lins
January, 2017

## Short Biography

David Alan Turner was born in London in January 1946 and educated at Selhurst Grammar School in Croydon followed by Oxford University, at Brasenose College, from October 1965. After gaining an Honours degree in Theoretical Physics at Oxford in June 1968 he decided to enter the then relatively new academic field of Computing.

In September 1968 David joined the recently established Oxford University Programming Research Group, led by Christopher Strachey, as a graduate student. In 1968-9 David gained a Diploma in Advanced Mathematics, specialising in Theory of Programming Languages and Logical Theory of Computation. His dissertation, submitted in September 1969, was in the area of Artificial Intelligence and entitled "Programs that Learn".

David remained at the Programming Research Group until September 1972. He attended the lectures of Strachey and Scott on Denotational Semantics and Domain Theory and undertook research in the semantics and implementation of higher order programming languages, writing a compiler/interpreter for the ISWIM-like language PAL, developed at MIT. He also took part in the development of OS6, an operating system for the group's Modula-1 computer, which was possibly the first operating system to be written entirely in a high level language (BCPL).

David's doctoral thesis, submitted some time after leaving Oxford, concerned the implementation of the lambda-calculus based language, SASL, by compilation to SK combinator code and the execution of the latter by normal graph reduction. The thesis included denotational semantics for both source and target languages and from these a formal proof of the correctness of the compiler by structural induction over the syntax of the source language. His thesis supervisor was Christopher Strachey and after Strachey's death in 1975, Dana Scott. David was awarded the degree of D.Phil by Oxford University in 1981. He also holds an M.A. in Physics from Oxford.

In October 1972 David was appointed lecturer in Computational Science at the University of St Andrews in Scotland, where he took part in the development and teaching of the undergraduate syllabus, preparing new courses in functional programming, theory of programming languages and theory of computation. He designed and implemented the purely applicative language, SASL (St Andrews Static Language), a sugared lambda calculus, as an alternative to LISP for teaching functional programming. David also worked on portable compilers for Algol-like languages and jointly with Ron Morrison designed a new language similar to Algol-W, for which David wrote a very compact one-pass compiler. This was further developed by Ron Morrison and Jack Cole after David left St Andrews and, under the name

S-Algol, was adopted by the Scottish Education Department for use in Scottish Schools as a replacement for BASIC.

In January 1977 David moved to Canterbury with his family, taking up a lectureship in Computer Science at the University of Kent, where he was promoted to Senior Lecturer in 1981 and to full professor in 1982. At Kent David played a full part in the teaching of the undergraduate syllabus and was for a period Director of the taught masters programme in Computing. He continued his research in functional programming and developed a long collaboration with Burroughs Corporation, who adopted SASL for a major development project at their research centre in Austin, Texas.

By the mid-80's the SK combinators implementation of SASL ran at 27 sites, being used for teaching by a number of universities and also by several companies including Burroughs. David also designed and implemented an extremely simple functional language based on recursion equations, Kent Recursive Calculator, which was used for teaching functional programming at Kent and at Oxford. In 1983 he established a company, Research Software Limited, for the purpose of developing a commercially supported non-strict functional language. This became Miranda, first released in 1985, which drew on the ideas of SASL and KRC but with the addition of a polymorphic type discipline similar to that of ML. The development of Miranda was supported by grants from International Computers Ltd and later the Amoco Production Company, which enabled David to take leave from his teaching duties at Kent to work on the project.

Miranda was widely adopted for both teaching and research and had a significant influence on the development of the field. Haskell, which emerged a few years later, has many similarities.

Since leaving St Andrews in January 1977 David has spent most of his career at the University of Kent in Canterbury, where he remains Emeritus Professor of

Computation, but has also held professorships at the University of Texas at Austin (visiting 1992), Queen Mary, University of London (visiting 2000-2004) and Middlesex University (2001-2006). He is also an Emeritus Professor of Middlesex University.

David enjoys teaching and his lectures were always popular with students. Active in the wider Computer Science community, he served for extended periods as external examiner to Computing degrees at other UK universities, mostly at taught master's level, and supervised a series of successful PhD candidates at Kent. He was for fourteen years an active member of IFIP Working Group 2.1 on algorithmic languages, and a founder member in 1987 of Working Group 2.8 on Functional Programming, whose meetings he continues to attend.

David is married to Una (born Una Williams), whom he met in Oxford in 1968. Una was a laboratory technician at that time but then trained as a nurse and a health visitor and later qualified as a lawyer, specialising in mental health law. They married in 1970 and have three children, Sarah, Daniel and Benjamin and, at the time of writing, five grandchildren. Una and David remain in Canterbury where they have lived since 1977.

David was born into a Jewish family and has two brothers and twenty living cousins. He is actively engaged in wider society and has served on the committees of a number of human rights organisations and NGO's. For much of his adult life David was a dedicated runner and participated in many half-marathons. He is a Fellow of the British Computer Society and of the Royal Society of Arts.

*David Turner at a meeting in Zermatt in 1997*