Strachey 100

Celebrating the life and research of Christopher Strachey

University of Oxford
Department of Computer Science

18th–19th November 2016
Introduction

Christopher Strachey (1916–1975) was a pioneering computer scientist and the founder of the Programming Research Group, now part of the Department of Computer Science at Oxford University. Although Strachey was keenly interested in the practical aspects of computing, it is in the theoretical side that he most indelibly left his mark, notably by creating with Dana Scott the denotational (or as he called it, ‘mathematical’) approach to defining the semantics of programming languages. Strachey also spent time writing complex programs and puzzles for various computers, such as a draughts playing program for the Pilot ACE in 1951. He developed some fundamental concepts of machine-independent operating systems, including an early suggestion for time-sharing, and was a prime mover in the influential CPL programming language. Strachey came from a notable family of intellectuals and artists, perhaps most famous for Christopher’s uncle Lytton, a writer and member of the Bloomsbury group.

The conference will open with a morning of talks dedicated to a historical exploration of Strachey’s life, chaired by Cliff Jones. The morning session will end with a panel chaired by Bernard Sufrin. After lunch, there will be a video presentation from Strachey’s long-time collaborator Dana Scott, who regrets he is unable to be in the United Kingdom for this event. The afternoon session will consist of talks about the future of research inspired by Strachey at Oxford and further afield, which will be chaired by Samson Abramsky.

This booklet will provide some information on the proceedings of the conference, including a schedule of talks and short abstracts from those contributors who provided them. There is also a series of appendices with some longer papers, and a list of the items shown at an exhibition at the Bodleian Library.
# Program

<table>
<thead>
<tr>
<th>Start</th>
<th>Name</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Morning session</strong> <em>(Chair: Cliff Jones)</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09:00</td>
<td>Mike Wooldridge</td>
<td>Introduction</td>
</tr>
<tr>
<td>09:15</td>
<td>Martin Campbell-Kelly</td>
<td>Strachey: the Bloomsbury years</td>
</tr>
<tr>
<td>09:45</td>
<td>Joe Stoy</td>
<td>Strachey and the Oxford Programming Group</td>
</tr>
<tr>
<td>10:15</td>
<td>Martin Richards</td>
<td>Strachey and the development of CPL</td>
</tr>
<tr>
<td>10:45</td>
<td>break</td>
<td></td>
</tr>
<tr>
<td>11:15</td>
<td>Peter Mosses</td>
<td>SIS, a semantics implementation system</td>
</tr>
<tr>
<td>11:45</td>
<td>Robert Milne</td>
<td>Semantic relationships: reducing the separation between practice and theory</td>
</tr>
<tr>
<td></td>
<td>David Hartley</td>
<td></td>
</tr>
<tr>
<td>12:15</td>
<td>Michael Jackson</td>
<td>Panel <em>(Chair: Bernard Sufrin)</em></td>
</tr>
<tr>
<td></td>
<td>Roger Penrose</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Chris Wadsworth</td>
<td></td>
</tr>
<tr>
<td>13:00</td>
<td>lunch</td>
<td></td>
</tr>
<tr>
<td><strong>Afternoon session</strong> <em>(Chair: Samson Abramsky)</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14:15</td>
<td>Dana Scott</td>
<td>Video presentation</td>
</tr>
<tr>
<td>14:45</td>
<td>Jane Hillston</td>
<td>A mathematical approach to defining the semantics of modelling languages</td>
</tr>
<tr>
<td>15:15</td>
<td>Philip Wadler</td>
<td>Christopher Strachey, first-class citizen</td>
</tr>
<tr>
<td>15:45</td>
<td>break</td>
<td></td>
</tr>
<tr>
<td>16:15</td>
<td>Hongseok Yang</td>
<td>Probabilistic programming</td>
</tr>
<tr>
<td>16:45</td>
<td>Uday Reddy</td>
<td>Parametric polymorphism and models of storage</td>
</tr>
<tr>
<td>17:15</td>
<td>Jeremy Gibbons</td>
<td>What are types for?</td>
</tr>
<tr>
<td>17:45</td>
<td>Closing remarks</td>
<td></td>
</tr>
</tbody>
</table>
Talk abstracts

Martin Richards:
‘Christopher Strachey and the Development of CPL’

Christopher Strachey was the most significant contributor to the design and implementation of the programming language CPL. At the time there was little understanding of the complexities of computer language design and how type systems could cope with lists and the kinds of structures needed to represent, for instance, parse trees. The CPL project cannot be regarded as being successful since it did not result in a usable CPL compiler. The reasons being that the language became too large and complicated, there were insufficient people to implement the compiler and, in the middle of the three year project, all work had to be transferred from Edsac 2 to Titan, a newly designed version of the Ferranti Atlas computer which as yet had no operating system. Even so, we can be proud of the work that went into CPL and its influence on the design of many later languages.

Peter Mosses:
‘SIS, A semantics implementation system’

During my DPhil studies, supervised by Christopher Strachey, I developed a prototype of a system for executing programs based on their denotational semantics. It involved partial evaluation of lambda-notation, implemented using Wadsworth’s call-by-need algorithm. I continued the development of the system as a postdoc at Oxford, and subsequently at Aarhus, Denmark. The system was called SIS: Semantics Implementation System.

Robert Milne:
‘Semantic relationships: reducing the separation between practice and theory’

Christopher Strachey believed that the gap between theory and practice was impeding the development of computing science. In my talk I shall consider how our work together on the essay that ultimately became our book tried to narrow the gap, by formalising, and reasoning about, the implementation concepts for programming languages. A particular focus will be the proof techniques for imperative programs that use storage, which were implicit, but not very easy to discern, in the book.
Jeremy Gibbons:
‘What are types for?’

Types in programming languages are commonly thought of as a way of preventing certain bad things from happening, such as multiplying a number by a string. But this is only half of the benefit of types: it is what types are against. Types in programming languages are also what enable some good things to happen, such as selecting the right implementation of a heterogeneous operation like comparison or printing based on type information; this is what are types for. This ability is surprisingly powerful, and gives rise to a variety of highly expressive generic programming techniques. I will illustrate with some examples based on the rank-polymorphic array operations introduced in Iverson’s APL: not only does the type information prevent array shape errors, it is what directs the lifting of operations across array dimensions.

Exhibition items

Some information here has been taken from the catalogue of the Strachey archive at the Bodleian Library, collected by the Contemporary Science Archives Centre. This is available online at http://www.bodley.ox.ac.uk/dept/scwms/wmss/online/modern/strachey-c/strachey-c.html.

   Tabulated account of his life and activity, drawn up by Strachey.
   The information, itemised by year and by month, is compressed in Strachey’s very tiny handwriting on to 2 and a half sheets of writing-paper. The earliest entry is for 1926, and the entries have been added in blocks of two or three minutes at a time, to the end of 1964.
   Personal and professional events are recorded, and the account is remarkable for the amount of information encompassed.

   A photo of Strachey aged 6 lying by the side of the pool at Mud, the Strachey holiday home in Sussex. Another of Strachey aged 8 wearing a hat and carrying a forkful of hay. Four photos of Strachey looking relaxed reading a newspaper in 1973, probably also taken at Mud. One computer generated image of Strachey, composed of various symbols including a moustache of λs.

3. Diagrams of the school gramophone at St. Edmund’s School, Canterbury. Date: late 1940s.
   Strachey was employed as a schoolmaster at this small public school,
where he taught physics and mathematics. He was also involved in extra-curricular school activities, including the school’s radio club, where he designed and built a combined radio and gramophone.

   After leaving St. Edmund’s school, Strachey took a post at the rather more prestigious Harrow. Here as well he took part in a number of activities, including teaching musical appreciation, and often gave lectures at the science society, including on the topic of interplanetary travel.

   During the last years of his time at Harrow, Strachey became interested in computers, and spent some time at the National Physical Laboratory learning about their Pilot ACE machine. He then spent quite a few years working on a program which would enable a computer to play draughts.

   Strachey’s copy of his letter to A.M. Turing, 15 May 1951. Written immediately after hearing Turing’s talk ‘Can Digital Computers think?’, broadcast on BBC Third Programme on same day. In his letter Strachey discusses the general problems of ‘thinking machines’, their application to the Manchester machine, and his modification of his ACE draughts program.

   Whilst working at the National Research Development Corporation, Strachey was responsible for large parts of the design of Ferranti’s new computer, the Pegasus.

   Strachey describes the paper as follows: “‘Time Sharing in Large Fast Computers’ was probably the first paper to discuss time-sharing and multi-programming as we now know them. It is a matter of history that the time-sharing idea became extremely fashionable in the middle sixties and dominated much of the work on computing at that time. When I wrote the paper in 1959 I, in common with everyone else, had no idea of the difficulties which would arise in writing the software to control either the time-sharing or multi-programming. If I had I should not have been so enthusiastic about them.”

The CPL Working Papers were issued in July 1966 following a meeting of the CPL authors in Oxford on 24 June. They consisted of 2 parts: 1. The second edition of the Elementary Programming Manual and 2. An unfinished Reference Manual. The concluding paragraphs of the minutes of the 24 June 1966 meeting state: “The circulation of the CPL Technical Report will conclude the phase of effort on CPL which has continued for the past four years, and at this stage most authors will probably wish to discontinue their efforts in a formal way... CPL of course needs major further work, particularly on data structures and program segmentation. C.S. (Strachey) and D.P. (Park) propose to study this at Oxford, and M.R. (Richards) may implement it at M.I.T.”

Strachey’s preface to the Working Papers, which gives a useful account of the history of CPL and its documentation, is on display here.

   In an autobiographical note written in 1971, Strachey described this as follows:
   “An interesting by-product of the work on CPL and its compiler was the General Purpose Macrogenerator which is in effect a simple but very general string processor. This program has now been implemented in almost every country in which computing is a research subject—I certainly know of implementations in Australia, Japan, Poland, Norway, Denmark, Germany, France, Italy, Holland as well as several in England and America. It gives an elegant example of the use of a stack and shows the power of functional composition to construct complicated operations from simple primitives. It is not, however, of any very deep theoretical interest and I now regard it as a rather beguiling time-waster.”

   Strachey’s work on time sharing culminated in US and UK patents, filed at the end of the 1950s and only awarded in the mid 1960s.

   Strachey described this as ‘a popular article ... it describes, amongst other things, a draughts-playing program’. Strachey found difficulty in writing, and in adequately illustrating, the article, of which many drafts, revisions and alternatives survive in the folders both of working papers and editorial correspondence.

   One of the features of Strachey and Scott’s mathematical semantics which must have particularly pleased Strachey is the compactness with
which languages could be defined. This is an example of such a single page (two-sided) definition.

Strachey was determined to be elected to the Royal Society towards the end of his life, and partly in an attempt to increase his publication count and win some notoriety, he co-wrote with one of his students this extremely lengthy essay and submitted it for the Cambridge Adams prize. It represents the ultimate achievement of Strachey’s work on semantics, and explains the concept and metalanguage as well as giving exhaustive worked examples on a sizeable programming language. The Strachey archive contains a large quantity of manuscript material, written in both Strachey’s and Milne’s hands. Displayed here is only the introduction and the note to the assessor’s apologising for the length of the essay.

A preliminary typed version of the essay, although still not completed. Strachey continued to revise the essay extensively, and there were plans to publish in 1975, although these were changed by Strachey’s sudden death. The submission did not win the prize.

After Strachey’s death, Milne rewrote the material of the Adams Essay and it was published in book form.

17. ‘The word games of the night bird’. Date: 1974.

Appendix: papers
