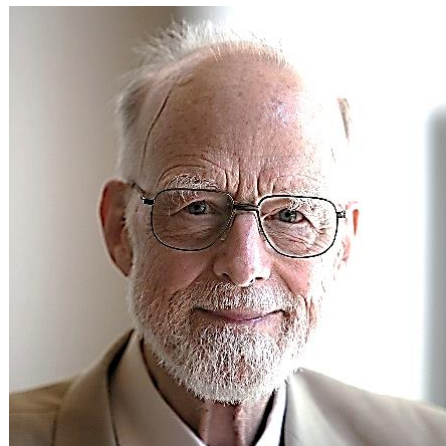# Tony Hoare @ 90

Dear readers,

Professor Sir Charles Antony Richard Hoare, FRS[1], reached his 90th birthday in January this year. Tony Hoare, as he is better known, was knighted for his services to theoretical computer science, and was one of the earliest computer scientists to be made a Fellow by the Royal Society. I believe the first computer scientist to be elected FRS was Maurice Wilkes, and for some years only he and Tom Kilburn — famed for the Manchester "Baby" and other historic computers — were Fellows. It took several years for more to be elected, and Tony himself was one of the first.

Sir Tony Hoare, FRS
(Source: Wikipedia)

Since those days, the Royal Society has recognised computer science as a "proper" science, and there are now a respectable number of CS Fellows. Soon after Tony Hoare's election, Robin Milner was elected an FRS, and following that, "the gates opened" and several more have joined their ranks. For example, Laurence Paulson[2] FRS — who did his post-doctoral work with the late Mike Gordon[3], FRS — gave our most recent annual FACS-LMS seminar this year, with Tony Hoare himself giving an earlier FACS annual Peter Landin Semantics Seminar[4]. Just recently, Glynn Winskel, who gave the 2021 Peter Landin Semantics seminar, was elected an FRS. For many years, Tony Hoare was series editor for the Prentice-Hall "Red and White" computer science series. For these and many other reasons, FACS felt that we would like to pay a tribute to Tony Hoare in our newsletter.

There have been a number of *festschriften* for Tony. I am grateful to several of our contributors for references to these, as listed below. In the light of that collection of technical material, the *FACS FACTS* editorial team invited a number of eminent computer scientists to give some more informal and personal recollec-

---

[1] https://en.wikipedia.org/wiki/Tony_Hoare
[2] Reported on by Andrei Popescu in this issue.
[3] Mike Gordon worked widely within the theorem-proving field, including mechanised support for proofs in Hoare logic.
[4] Denvir, T. (2013). Report on: Peter Landin Annual Semantics Seminar: Professor Sir Tony Hoare (December 3rd 2012). FACS FACTS, 2013(1):5–7, December 2013

tions of their work in collaboration with Tony or guided by him. It is a pleasure to record that six contributors we invited responded with enthusiasm: Jifeng He, Cliff Jones, Bill Roscoe, Joe Stoy, Bernard Sufrin, and the *FACS* chairman, Jonathan Bowen.

Tony Hoare read *Greats* at Oxford (*Literae Humaniores*, thanks to Bill Roscoe for this illumination!) and learned Russian during his national service with the Royal Navy. He then spent a year at Moscow University studying under the genius mathematician, Andrey Kolmogorov[5], who is famed for his original work on probability, statistics and intuitionistic or constructive logic, among much else. After that spell in Moscow, Tony moved to Elliott Bros. London Ltd., a British computer manufacturer, which is where I first met him, in my first job after graduating. On learning of Tony's professional path to that point, I was and still am amazed by the fact that he absorbed brand new technical and theoretical topics in mathematics and theoretical computing, in Russian, when he had only recently learned that language.

Since all the other contributors to this tribute to Tony refer to his later work in academia, I thought a brief note on his earlier 'industrial' time might be in order. I was at Elliott's from 1962 to 1965, Tony from 1960 to about 1968. Tony was designing and supervising the implementation of an early ALGOL 60 compiler[6], one of the first. I was rather envious of the ALGOL team, not being a member of it, but Tony orchestrated departmental seminars as in academic environments, which were very stimulating. Elliott's was at the same time bringing out a new computer. At that point, in the early 1960s, it didn't have a name. The computer was called "Project 41". We were sworn to public silence about that. The computer was to have a fuller-fledged time-sharing system than most of its predecessors. Individual user programs could proceed at the same time, and demands made by peripheral devices, such as card/paper tape readers, could take place simultaneously without seriously suspending the user programs. I was to program the kernel of the OS that would handle this time-sharing. This was a challenging task, but under Tony's guidance, I persevered. His core idea was that there should be two levels of interrupt, one less urgent but still needing attention in due time, which he called a 'hesitate', and the other, more urgent, which would inexorably lose information if not rapidly attend-

---

[5] https://en.wikipedia.org/wiki/Andrey_Kolmogorov
[6] Some 8 years later I too designed and implemented an ALGOL 60 compiler while at RADICS Ltd. I don't think I could have done so without the stimulus of witnessing Tony Hoare's earlier implementation. See Tim Denvir, More Recollections of ALGOL 60, in Resurrection No. 52, Autumn 2010, ISSN 0958-7403

ed to, a proper 'interrupt'. Our mutual manager, Sheila Quinn, long since deceased, almost warned me, "He's absolutely brilliant, and he expects everyone else to be the same!". Project 41 became the Elliott 4100 series, and I feel gratified that a few lines of crucial OS code that I wrote lie within those machines, if any of them are still in use.

Enough of this from me. I commend the personal memories from our six noble academic contributors, which follow on from here.

Tim Denvir
FACS co-editor

Festschriften

C A R Hoare and Cliff B Jones. *Essays in Computing Science.* Prentice Hall, 1989.

Bill Roscoe, editor. *A Classical Mind: Essays in Honour of CAR Hoare.* Pearson Education, 1994.

Cliff B Jones, A William Roscoe, and Kenneth R Wood, editors. *Reflections on the Work of C. A. R. Hoare.* Springer Science & Business Media, 2010.

Cliff B Jones and Jayadev Misra, editors. *Theories of programming: the life and works of Tony Hoare.* ACM, 2021.

Cliff B Jones. *An Interview with Tony Hoare: ACM 1980 A.M. Turing Award Recipient.* ACM, 2015. https://amturing.acm.org/pdf/HoareTuringTranscript.pdf

Jim Davies, Bill Roscoe, and Jim Woodcock, editors. *Millennial perspectives in computer science: Proceedings of the 1999 Oxford-Microsoft Symposium in honour of Professor Sir Tony Hoare.* Palgrave, 2000.

CARH retirement symposium, Oxford, 13–15 September 1999

For Tony's 90th Birthday

He Jifeng

Distinguished Professor, Tongji University, Shanghai

I was the visiting scholar in Oxford University from November 1983, when I first met Tony; he warmly welcomed me and gave me a tour of the computing laboratory. Under the recommendation of Bernard Sufrin, I officially joined PRG as a research fellow in November 1984 to participate in a project led by Tony Hoare. In a short period of time, he turned me from a newcomer to core member of the team, giving me the opportunity to participate in different projects, and encouraging me to participate in international conferences and summer schools, thereby exposing me to forefront of technology development. The academic exchange activities led by him gave me the opportunity to meet peers from different countries and that helped me in building an international network. In the more than 15 years of collaboration with Tony, he has always been a mentor to me, he had great skills as an applied logician; we had many meaningful discussions around software theoretical research and application technology. Additionally, he has this extraordinary ability to guide us in exploring links between fundamental research and engineering applications. He organized series of seminars to define the goal of the research, and discuss new approaches to improve the trustworthiness of software. We also had weekly meetings in which he inspired ideas from different groups, embracing a diversity of thoughts, and promoting a culture of openness and inclusion within PRG. Under his leadership, our projects with IBM Development laboratories and with microprocessor company Inmos culminated in technologically advanced products, and were recognized by Queen's awards. Based on our experience in the development and application of logical theories, we undertook a project to broaden the theoretical models to cover more aspects of Computing Science and its application to system design. In 1998, we published the results of ten year's research under the title "Unifying Theories of Programming".

On a personal level, our two families have many years of friendship. At Oxford, we enjoyed dinner at each other's home, and shared our respective cultural and personal experiences. Since leaving Oxford, we stayed in close contact and shared with each other our latest work findings. He came to China a number of times to support my new role of Dean of Software Engineering Institute. We last saw each other in 2018 in London at BCS symposium where we discussed the new research topics on Unifying Theories of Programming.

Till today, I still have many fond memories of the time that Tony and I shared, his wisdom, insight and kindness made a significant impact to my career and my life.

Congratulations Tony, to your remarkable lifetime achievement and highest international regard in the Computing Science field.

Happy 90th birthday.

*Jifeng He, Bernard Sufrin, and Tony, at a party given for Chinese scholars at St. Hilda's College.
Oxford in 1984, by the Ambassador of the People's Republic of China.
(The ambassador took the photograph himself.)*

Jifeng He and Tony at the UTP 2016 conference in Reykjavik, Iceland

Group photograph during the UTP 2016 conference in Reykjavik, Iceland.

Tony is in the centre with Jifeng He to the left and Leo Freitas, Ana Cavalcanti, and Huibiao Zhu to the right. Behind Tony to the right are Jonathan Bowen, Jim Woodcock, and Yixiang Chen.

Jifeng He and Tony were UTP 2016 keynote speakers. Jonathan Bowen and Huibiao Zhu were the proceedings editors.

Jifeng He and Tony at the BCS London office, 2018

Thanks to Tony Hoare

Cliff B Jones
School of Computing, Newcastle University

As well as the debt all computer scientists owe to Tony for his insightful scientific contributions, many of us have experienced his personal kindness. As someone who has benefited enormously from his support, I'd like to offer a personal note of thanks to this collection.

My most obvious debt to Tony is his arranging that I was accepted at Oxford in 1979 to undertake a doctorate with him. Since I had dropped out of Grammar School pre "A-levels", I did not have the normal prerequisites for post-graduate studies — but at the foot of the page detailing the requirements, the Examination decrees state how to get an exception approved — there's an organisation that has learned over centuries that to any rule there have to be exceptions. Tony presumably based the justification on publications and technical reports from my time in IBM. I should also add that Tony's college offered the perfect environment for a "mature student" and Wolfson remains a special place for me.

Pursuing my DPhil with Tony in Banbury Road was one of the most exhilarating phases of my long research career: in 1979 Tony was deeply into the evolution of CSP and did gently encourage me to consider it as an approach to the problems in which I was interested — but when he saw that I had my own approach to compositional design of concurrent programs he offered encouragement and support. I think I was the first of Tony's students to "be allowed to supplicate" for a DPhil in Oxford — my young sons sat with Tony during the ceremony and asked why Latin was being used — Tony was probably one of few people who understood it without difficulty but replied with humour "because the text is such nonsense".

A technical bonus of lunching with Tony at Wolfson College was a discussion over coffee with Robin Gandy when I was briefly considering using Temporal Logic for my rely-guarantee ideas: Robin was dismissive of TL and I didn't regret accepting his prompt.

However, Oxford was certainly not the first occasion for which my thanks are due. I'm almost certain that Tony prompted my invitation in 1973 to become a member of IFIP's prestigious Working Group on Programming Methodology (WG 2.3). We had certainly met as early as April 1969 when Tony presented his Axiomatic basis to WG 2.2 in Vienna. That part of Austria was important to both of us. I was able to be an "observer" at the WG 2.2 meeting because I was on a two-year assignment to the IBM Lab in Vienna. Tony had looked at their huge

VDL operational semantics of PL/I in his role on the ECMA standardisation committee; but well before that he had made a crucial comment at the 1964 Formal Language Description Languages conference in Baden-bei-Wien that foreshadowed his hunt for an implicit way of describing semantics.
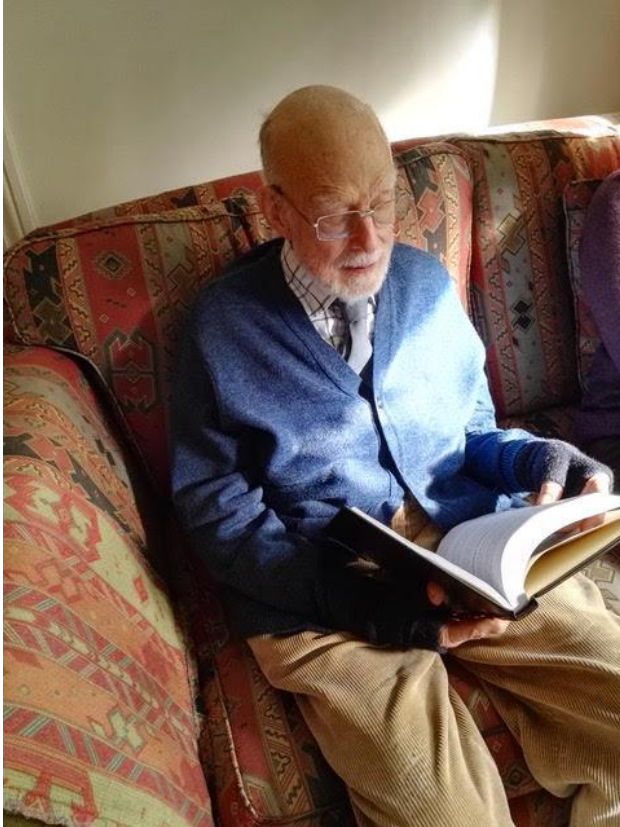
Tony in their Cambridge house studying `Theories of programming: the life and works of Tony Hoare'

Photo by Joanna Francis reprinted with her permission

It is also a pleasure to recall the crucial role that Tony's famous "Red and White" series of Prentice-Hall books played in the development of what many of us call "formal methods". My own 1980 book was key to the description of those aspects of VDM that relate to the design of general programs. But when it became obvious that LNCS 61 was not going to be reprinted by Springer, Tony immediately invited Dines Björner and I to update the material on programming language aspects of VDM's denotational approach and produce a new volume (published in his series in 1982).

In 1981, I was appointed to a chair in Manchester. One of the Manchester professors who interviewed me confided that mine was the only name in the intersection of the sets of recommended candidates and applicants. Since this was prior to my DPhil viva, I have little doubt about who made the recommendation.

A delightful personal recollection was our joint visit to China in April 1983. Since Tony had recently been elected an FRS and, of course, had received the 1980 Turing Award, our mutual friend Prof Zhou Chaochen had arranged that the Chinese Academy of Sciences invited Tony for a lecture tour; I was clearly the "support act" but the technical sessions worked well because Tony and I sat through each other's lectures; this was important because at that time it was considered impolite for members of a Chinese audience to ask questions during lectures but we were both besieged in the breaks — sitting in the other one's lectures gave us time to hone our next talks.

What made the Beijing/Nanjing/Hangzhou trip so special was that we were both accompanied by our families. My sons were only seven and nine years old and were periodically "kidnapped" to appear in Chinese photos as the "golden-haired boys" (= not completely black hair). Tony and Jill's daughter, Joanna is tall with vivid red hair and attracted many curious Chinese who in many places had seen few Westerners — remember 1983 was not long after the "cultural revolution". We were all (eight of us) taken on many interesting visits and during a trip to, for example, a communal farm Tony would ask "How many members of the communist party are there here?". Puzzled, I finally queried why Tony wanted to know to which he replied with a smile "I don't, it's just interesting that they always know"! (Remember that he had earlier spent time in the USSR.)

There was an amusing coda to the 1980 book (SDRA): Henry Hirschberg and Helen Martin from Prentice-Hall chose to mark the 50th book in the above-mentioned series with an annotated book of Tony's papers, and they invited me to edit the 1989 volume. They also wanted to display all of the series in the window at Blackwells' wonderful store. Despite Henry having stuck his neck out by causing over ten thousand copies of SDRA to be printed, Prentice-Hall could not find one to display. Ian Hayes to the rescue: having complained many times about the cost of textbooks in Australia, he was able to buy two copies – at bookshop prices – and post them to Prentice-Hall at their expense!

There are many other happy personal links to Tony and his family, but I should close these words of thanks by returning to my debt to his inspirational research; I know that this observation could be echoed and amplified by enormous numbers of computer scientists.

The photograph was taken during that April 1983 trip:
Family Hoare is sitting in front of the Nine-Dragon wall in Beijing.

References

1. C. A. R. Hoare and Cliff B. Jones. Essays in Computing Science. Prentice Hall, Inc., 1989.

2. Bill Roscoe, editor. A Classical Mind: Essays in Honour of CAR Hoare. Pearson Education, 1994.

3. Cliff B Jones, A William Roscoe, and Kenneth R Wood, editors. Reflections on the Work of C. A. R. Hoare. Springer Science & Business Media, 2010.

4. Cliff B. Jones and Jayadev Misra, editors. Theories of programming: the life and works of Tony Hoare. ACM, 2021.

5. Cliff B. Jones. Professor Sir Tony Hoare: ACM Turing Award Winner 1980. Online, 3 2016. Interview.
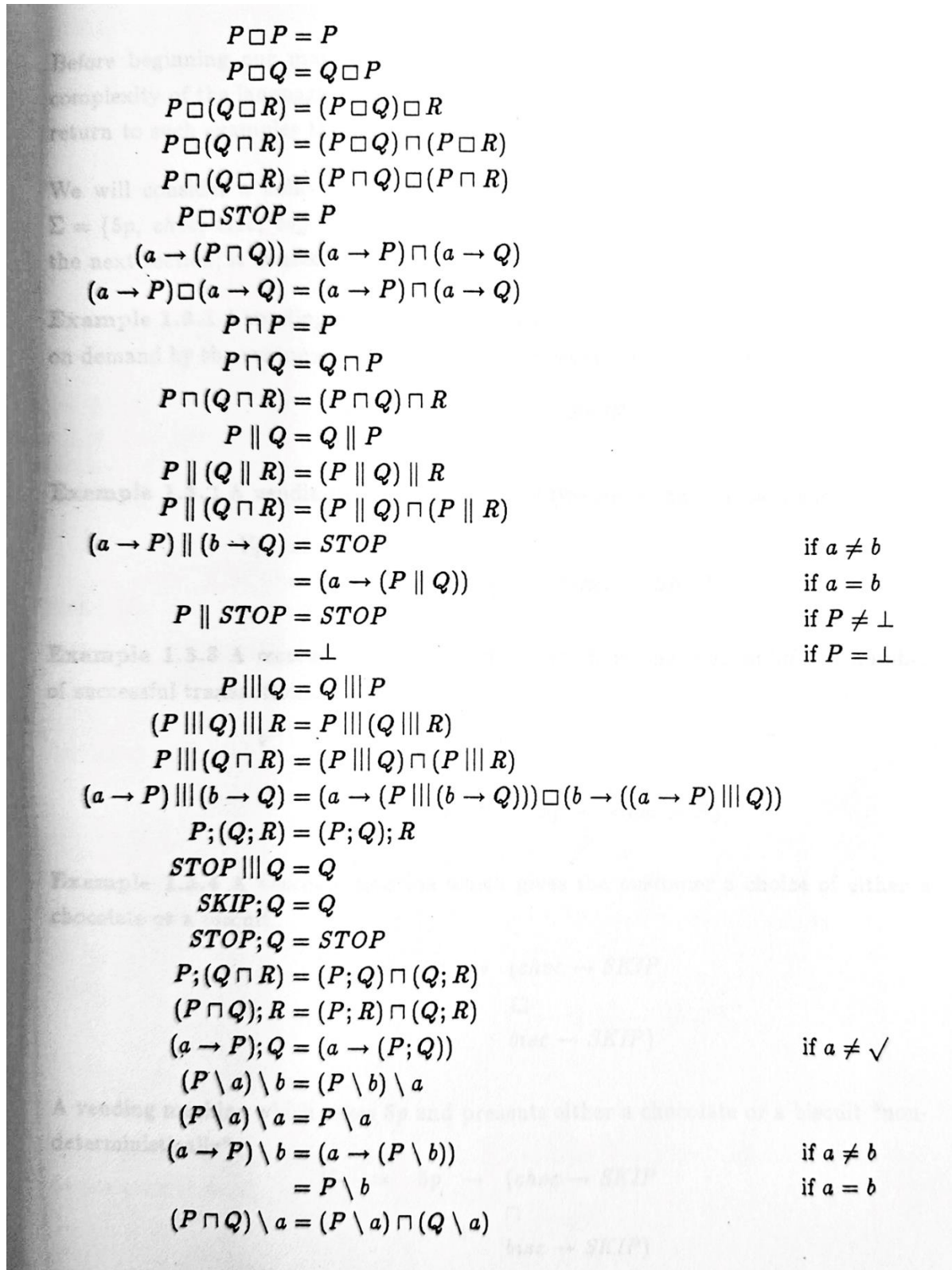
Working with Tony Hoare

Bill Roscoe
University of Oxford

In this memoir, I recall my interactions with Tony from Autumn 1978, when I first met him, up to 1989 when we completed the first phase of hardware verification with Inmos, together with more general memories. Much of the history of my projects with him over this period — the development of process algebra CSP, the semantics of occam and their application to hardware verification — has been well documented in my and others' previous writings such as [1,2,3,4] so here I am going to concentrate on what it was like to work with him and be supervised by him.

Tony has an amazing intuition for seeking a simple and elegant way of attacking problems. When I first met him, I had just emerged from my Oxford Maths degree — and delighted in creating complex structures in what most people would describe as pure maths. The Scott-Strachey approach to programming language semantics is certainly elegant, but as I learned it as a final-year undergraduate it seemed anything but simple. It therefore took me some months to realise the power of Tony's quest for simplicity. In particular, my first year of working with Tony and Steve Brookes on CSP showed how his design of the language itself was driven by his quest for the most elegant laws and models.

He was certainly driven by the desire that the CSP we were developing satisfies algebraic laws. These laws were, at the time, what you might call healthiness principles: pleasing properties that you would expect an elegant structure to satisfy. They were not motivated by the immediate desire to create an algebraic semantics in the style of ACP or as was later done for CSP by Brookes [5] and myself [6,7]. Laws of this second sort tend to be directed at transforming finite programs into a restricted normal form, whereas Tony's original laws such as those set out in [8], and reproduced elsewhere, have a more subjective feel to them. In developing Timed CSP with me, Mike Reed gathered together a large collection of Tony's laws, which you can find in [9].

Although Tony did not study mathematics as an undergraduate, he studied logic as part of the philosophy components of his Greats (*Literae Humaniores*) degree and told me a few years ago how much he valued his logic tutorials with the then young philosopher and logician John Lucas, a junior research fellow at Merton and later a tutorial fellow there. Tony was also one of a group of Merton students who regularly met to discuss mathematical topics.

$$P \square P = P$$
$$P \square Q = Q \square P$$
$$P \square (Q \square R) = (P \square Q) \square R$$
$$P \square (Q \sqcap R) = (P \square Q) \sqcap (P \square R)$$
$$P \sqcap (Q \square R) = (P \sqcap Q) \square (P \sqcap R)$$
$$P \square STOP = P$$
$$(a \rightarrow (P \sqcap Q)) = (a \rightarrow P) \sqcap (a \rightarrow Q)$$
$$(a \rightarrow P) \square (a \rightarrow Q) = (a \rightarrow P) \sqcap (a \rightarrow Q)$$
$$P \sqcap P = P$$
$$P \sqcap Q = Q \sqcap P$$
$$P \sqcap (Q \sqcap R) = (P \sqcap Q) \sqcap R$$
$$P \parallel Q = Q \parallel P$$
$$P \parallel (Q \parallel R) = (P \parallel Q) \parallel R$$
$$P \parallel (Q \sqcap R) = (P \parallel Q) \sqcap (P \parallel R)$$

$$(a \rightarrow P) \parallel (b \rightarrow Q) = STOP \qquad \text{if } a \neq b$$
$$= (a \rightarrow (P \parallel Q)) \qquad \text{if } a = b$$
$$P \parallel STOP = STOP \qquad \text{if } P \neq \perp$$
$$= \perp \qquad \text{if } P = \perp$$

$$P \mid\mid\mid Q = Q \mid\mid\mid P$$
$$(P \mid\mid\mid Q) \mid\mid\mid R = P \mid\mid\mid (Q \mid\mid\mid R)$$
$$P \mid\mid\mid (Q \sqcap R) = (P \mid\mid\mid Q) \sqcap (P \mid\mid\mid R)$$
$$(a \rightarrow P) \mid\mid\mid (b \rightarrow Q) = (a \rightarrow (P \mid\mid\mid (b \rightarrow Q))) \square (b \rightarrow ((a \rightarrow P) \mid\mid\mid Q))$$
$$P; (Q; R) = (P; Q); R$$
$$STOP \mid\mid\mid Q = Q$$
$$SKIP; Q = Q$$
$$STOP; Q = STOP$$
$$P; (Q \sqcap R) = (P; Q) \sqcap (Q; R)$$
$$(P \sqcap Q); R = (P; R) \sqcap (Q; R)$$
$$(a \rightarrow P); Q = (a \rightarrow (P; Q)) \qquad \text{if } a \neq \sqrt{}$$
$$(P \setminus a) \setminus b = (P \setminus b) \setminus a$$
$$(P \setminus a) \setminus a = P \setminus a$$
$$(a \rightarrow P) \setminus b = (a \rightarrow (P \setminus b)) \qquad \text{if } a \neq b$$
$$= P \setminus b \qquad \text{if } a = b$$
$$(P \sqcap Q) \setminus a = (P \setminus a) \sqcap (Q \setminus a)$$

Laws from Mike Read's DPhil thesis (1990)

A few years later, at the start of his career, Tony studied in Moscow in the school of Kolmogorov, one of the greatest mathematicians of the 20th century, and who was extraordinarily influential in developing modern probability. I only discovered recently that the standard axioms of probability[7] that I learned as an undergraduate were created by Kolmogorov about 25 years before Tony's work with him. It seems reasonable to wonder whether his spell in Moscow influenced Tony in favour of laws. Of course, while he was there Tony created what has been described as the first probabilistic algorithm, namely Quicksort — and reading Tony's paper [10] shows he was well aware of the probabilistic properties it has, so it would seem that there are two clear potential influences from Kolmogorov.[8]

Tony created the traces model of CSP and many of its laws, thereby demonstrating the utility of the search for simplicity. While we were well aware of its deficiencies it clearly demonstrated how wise it was to search for simplicity. One might characterise this as saying that it is better to find a solution that is too simple and extend it rather than tackle everything at once. Thus, he guided Steve and me to find minimal extensions to traces that had the required properties including respecting the laws. We had many discussions in Tony's office at 45 Banbury Road, either just the three of us or alongside other students such as John Kennaway, visitors such as E-R Olderog and David Park, and of course Dana Scott. This was very much an environment in which everyone was interested in what everyone else was doing.

My own interactions with him often took one of two forms: either I would be finding mathematical justifications for his intuitions, or he would be pressing me to find more elegant ways of presenting my own. An example of the first was him claiming in a lecture (1979) that a counter $Count_0$ defined by infinite mutual tail recursion, was equal "by induction" to the clever recursion

```
Zero = iszero -> Zero [] up -> Pos;Zero, where
Pos = down -> Skip [] up -> Pos;Pos.
```

I immediately realised that while this was a true equality, it could not be proved by conventional induction. However, determined to justify his insight, by the

---

next day I had formulated the metric theory of Unique Fixed Points and constructiveness which became central to CSP reasoning.

An example of the second form of interaction came a few years later (perhaps 1984/5) when he had asked me to create an algebraic semantics for occam, to which we had now switched our attention — I recall that our later discussions in 45 Banbury Road, which we left in Summer 1982, focussed on occam.

1990 Presentation of Queens Award for Technological Achievement for Inmos Ltd and the Oxford University Computing Laboratory

I was very proud to have created such a semantics (laws, normal form and reduction strategy) for occam that was congruent to my earlier denotational semantics. However, influenced by occam's syntactic structure, it was a syntactic mess containing too much ellipsis … He made it clear that this was not good enough. So, I went back with my tail between my legs to think again, and eventually we came up with the form now in the paper [11].

That paper led to our famous project on the creation of the occam transformation system and the verification of the T800 FPU. Tony was a wonderful mentor: by that time he was happy for me to take the lead and influenced the project only through occasional suggestions — such as the idea of building the transformation system — and taking part in conversations with David May and others at `inmos`.

By that time, he had moved on from CSP, though the Laws of Programming project was itself influenced by the occam work. And so, his original creation of CSP was left to me to take forward, a typically generous move on his part, as was the later involvement with occam and the transputer.

My own close research collaboration with him thus came to an end in about 1989, though he was always very interested the work of my group over the years, who of course usually revered him. His sheer enthusiasm for the young and their work shone through. I remember that about 15 years ago he came back to Oxford for a big dinner in Trinity College. He asked me to organise a research meeting first thing the next morning, something the proposed participants were not keen on, as it would be so soon after the previous evening's celebration. I dragged my feet, but he organised it anyway. This was one of several occasions where Jill unjustly criticised me for "forcing Tony to work so hard".

I have been lucky enough to have been made to feel a part of Tony and Jill's family, ever since visiting them a number of times at their home in Chalfont Road when I was a research student. It is a wonderful feeling to be an honorary Hoare in that way, largely because of Jill.

Tony and Jill Hoare – Picture used for the Hoare Room, Oxford University

Tony was always extremely loyal to his protégés, and the "family" he built at the PRG stayed together around him for many years, with a number, including me, appointed to permanent academic positions.

I do not think I ever saw him touch an actual computer until shortly before he left Oxford in 1999. Microsoft had given him a PC. We should have seen the writing on Oxford's wall.

References

1. Jones, C. B., & Roscoe, A. W. (2010). Insight, inspiration and collaboration. *Reflections on the Work of CAR Hoare*, 1--32.

2. Brookes, S. D., & Roscoe, A. W. (2021). CSP: A practical process algebra. In *Theories of Programming: The Life and Works of Tony Hoare* (pp. 187–222).

3. Hoare, C. A. R. (1991). The transputer and occam: A personal story. *Concurrency: Practice and Experience*, *3*(4), 249–264.

4. May, D. (2021). CSP, Occam, and Inmos. In *Theories of Programming: The Life and Works of Tony Hoare* (pp. 271–284).

5. Brookes, S. D. (1983). *A model for communicating sequential processes* (Doctoral dissertation, University of Oxford).

6. Roscoe, A. (1998). *The theory and practice of concurrency*, Prentice-Hall.

7. Roscoe, A. W. (2010). *Understanding concurrent systems*. Springer Science & Business Media.

8. Brookes, S. D., Hoare, C. A., & Roscoe, A. W. (1984). A theory of communicating sequential processes. *Journal of the ACM (JACM)*, *31*(3), 560–599.

9. Reed, G. M. (1990). *A uniform mathematical theory for real-time distributed computing* (Doctoral dissertation, University of Oxford).

10. Hoare, C. A.R. (1962). Quicksort. *The Computer Journal*, *5*(1), 10–16.

11. Roscoe, A. W., & Hoare, C. A. R. (1988). The laws of occam programming. *Theoretical Computer Science*, *60*(2), 177–229.

Tony Hoare

Joe Stoy
Formerly of the University of Oxford

The first time I met Tony was at MIT in 1974, when I was on sabbatical there. Barbara Liskov organised a symposium called "Languages and Systems to Support Structured Programming", which he attended. I remember his saying, to the surprise of some there, that if compilation were instantaneous, the "linking phase" would never have been invented — logically we should go straight from a text management problem to an executable program. This was of course long before the web, HTML, hypertext linking, and all that.

When Christopher Strachey died in 1975, I became the only established ("permanent") member of the Programming Research Group. We spent the next year getting Strachey's ad hominem professorship turned into a permanent ("statutory") chair. Many people and bodies were a great help in this: Leslie Fox, a numerical analyst and Head of the Computing Laboratory; the Mathematics Faculty Board; the head of the University's principal academic committee (the General Board); and several others. I actually attended the (very brief) meeting of Congregation which established the Chair.

After that a Board of Electors was appointed, and they eventually invited Tony to accept the Chair. Tony and Jill kindly invited me to spend a couple of days in Belfast to discuss the job. This was during the "troubles", and some Oxford friends suggested I should not go, or if I did, not to leave the airport. In the event I had a very pleasant stay at their home, talking about the PRG, and Oxford more generally (to which Tony was actually no stranger, having read Greats at Merton). Later in 1977, after Tony had accepted the post, he spent a couple of days with us, meeting various people and "casing the joint".

When Tony took up his post in September 1977, the PRG was almost empty of research staff. The graduate students were there, of course, and so were a couple of programmers; a new cohort on the Diploma course was arriving (for the Diploma in "Advanced Mathematics and Computation"). Dana Scott was also around as Professor of Mathematical Logic, albeit in the Philosophy Department; he had been a great support during the interregnum, and continued to be so. The ethos of the PRG still survived: seminars in the garden during good weather, and frequent conversations on the stairs of the house, which helped keep the group unified. But with Tony's arrival the research focus of the group gradually shifted, from functional programming and denotational semantics, to

imperative programming and predicate semantics. The practical work also gradually shifted, from work in BCPL on the PRG's own computer (by then an Interdata 8/32 machine) using our own hierarchical quasi-functional operating system (OS6, or the published version OSPub), to work on a new batch of DEC LSI-11 workstations running UCSD Pascal. A taught MSc course started in 1979 replacing (after a gap) the previous Diploma course.

Visitors also began to reappear, some supported by the SRC research grant, drawn to the new focus. For example, Cliff Jones, with whom I had had previous discussions of the difference between the Vienna Definition Language (operational) and the Vienna Development Method (denotational), arrived and began DPhil study under Tony's supervision, on the addition of "rely" and "guarantee" clauses to the pre- and post-conditions of Predicate Semantics. Nevertheless, Peter Henderson, a functional programming person, was appointed as a Lecturer in 1980 to run the new MSc course, and stayed until 1983. Mary Sheeran was his DPhil student, and Geraint Jones his research assistant, who worked on Peter's Lispkit system. Jean-Raymond Abrial was also here during that time, and his work on "Z" dominated much of the research conversation. He eventually left: I think Dana criticised Z as being too prolix, not mathematically concise enough to be really useful. In research, I myself became increasingly a loner, though Tony was always supportive. For example, when I wrote a piece for his Festschrift thirty years ago, he wrote me a detailed comment on it, which I greatly appreciated. Gradually, however, most of my research collaborators were at MIT. I went back there on sabbatical for the year 1981-82; but before I left, things at PRG were greatly overshadowed by sadness at the death of Tony and Jill's youngest child, Matthew, from leukaemia. Dana also left for CMU in 1981.

Teaching was a different matter. When I returned in 1982, the atmosphere had completely changed. For one thing, we had moved into the Keble Road site, along with the rest of the Computing Laboratory (the numerical analysts). That at once made things less personal and homely than it had been at Banbury Road. For another, the Thatcher Government had published the Alvey Report, and had proposed a great expansion of Computer Science teaching at British universities, including new posts for the PRG.

This would not be the PRG's first foray into undergraduate teaching: I had for several years offered courses in Functional Programming among the second-year options for Mathematics undergraduates, with accompanying tutorials. Bill Roscoe had been attracted into computing by attending these lectures (with a slight complication because he also wished to go to my wife's lectures on

Group Theory in an adjacent lecture room at the same times — she brought him to my room, and we all agreed a slight revision of the timetable); he later became a research student at PRG and subsequently Head of the Department. Another person attracted to computing in that way was John Launchbury, later the founder and now the Chief Scientist of Galois Inc.

But now the undergraduate teaching would go into a much higher gear. The Computing Laboratory had been allocated four new lecturerships initially, each accompanied by ten additional undergraduate places. These were to be "joint" tutorial posts, which implied that colleges had to be found for each of them. I myself had been taken care of, having been elected (with Leslie Fox's support) to a Research Fellowship at Balliol in 1975 — when the Honour Schools eventually began in 1984 and my Fellowship became Tutorial, I actually became Balliol's junior tutor and Senior Tutor in the same instant. But colleges had to be wooed for the newcomers. Three of the first tranche of appointments, in 1983 went to people who were already at PRG in one guise or another: Bill Roscoe, Bernard Sufrin and Ib Sørensen; the fourth went to Richard Bird from Reading. The ten additional places were a great bribe for the college administrations, but a college's support depended in large part on its Mathematics tutors. The college election committees had their own set of stories (for example, at those colleges which had hoped that the Computing tutors would be able to spend most of their time teaching classical Applied Mathematics. Fortunately, Tony had learnt the art of academic politics during his time at Belfast, but even so he was initially only able to get two colleges offering to give tutorial fellowships — to Bill at Univ and Bernard at Worcester. Ib, whose special remit was industrial liaison, joined Tony at Wolfson, and Richard went initially to a graduate college, St Cross, where Peter Henderson had been, and replaced him as Director of the MSc course; five years later, though, he moved to a Tutorial Fellowship at Lincoln. That set of appointments at one blow tripled the academic staff of the PRG. But other tranches followed. In 1984, John Hughes (St Edmund Hall) and his wife Mary Sheeran (Lady Margaret Hall), both no strangers to PRG, were appointed, and so was Ian Page (initially at St Hugh's, though when Richard moved to Lincoln, Ian, preferring to be at a graduate college, took his place at St Cross). In 1985 Mike Reed and Jeff Sanders replaced John and Mary at SEH and LMH, while Carroll Morgan (Pembroke) and Bill McColl (Wadham) were new appointments.

So for those years, Tony had a particularly heavy load of academic appointments to manage, as well as the usual graduate intake. But there was also the new undergraduate program to design. It was decided early on that we should not initially attempt to put on a single-subject Honour School — in fact, the

Honour School of Computation did not start until 1994. There were to be two joint courses, Mathematics and Computation, and Engineering and Computing Science (ECS). Alongside ECS there was another joint course, Engineering and Materials Science. It was also decided that the first years of both our courses should be entirely devoted, along with Mods at the end of that year, to the other subjects, with their existing first-year syllabuses. That gave the new staff an extra year of grace to prepare classes and tutorial sheets for the new courses, many of which were given in both Schools. But before that, the legislation establishing the new Schools had to be put through the Faculty Board and the General Board, and the second-year syllabuses designed, and then approved by the Faculty Board and its Subfaculty of Computation. Tony was on the Faculty Board and saw that through. All this meant that our periodic group meetings became weekly and more formalised (e.g., regular minutes): Tony's meticulous mind was invaluable. And of course, once the courses started in earnest, the tutors were often away from the Computing Lab giving tutorials or doing other college things: Wednesday afternoons in term time was the accepted time for college meetings.

The Maths and Computing School flourished: even when the Computing School started it remained (with Tony's strong approval) one of our flagship programmes. ECS, on the other hand, dwindled and eventually faded away. I think the Engineers didn't understand the way we taught programming: for example not using C, but preferring to start by using languages (Haskell and Modula) in which it was easier to teach good programming principles. Later they did better on their own when they appointed their own Professor of robotics and computing (Mike Brady) — he later did a stint as Head of the Engineering Science Department.

After the initial upheaval, the Lab continued to flourish and grow. A second Chair was established in 1988 -- Joe Goguen for ten years, followed by Richard Brent. My informal contact with Tony became less and less — no doubt the college and University administrative work I was doing didn't help (for example, at various times chairing the Faculty Board and a couple of University committees). But I know that the Computing Laboratory was in excellent shape when Tony handed over its leadership and left Oxford for Microsoft in 1999.

The last time Tony and I met was in Seattle, at the Federated Logic Conference in August 2006, where we (and Dana) were among the invited speakers. The three of us enjoyed a long on-deck conversation on the conference boat excursion around the Bay. I remember Tony's talk vividly — he spoke for about an hour, in elegant English, with not a single slide or use of the blackboard, to the bemusement of some in the audience, who were not used to presentations like that: "A Classical Mind" indeed.

Early Days at the PRG …

Bernard Sufrin
Emeritus: Worcester College and
Department of Computer Science, Oxford University

**… anecdotes from the Programming Research Group,
for Tony Hoare's 90th birthday**

It's a pleasure to be able to write here about Tony, and about our early days at the Programming Research Group; and it's a relief that it's going to be read by people who already know Tony's work, and for whom yet another technical appraisal of his oeuvre would be *de trop*. The invitation said "Don't be afraid to make it personal." So I haven't been. My goal is to show how strange the Oxford environment felt at first; and how adaptable, tolerant, and magnanimous Tony was.

The Interview

I met Tony Hoare and Joe Stoy for the first time in mid-1978, when I came to be interviewed at the PRG for an SRC research fellowship at the PRG that was associated with a Wolfson College fellowship. Tony had arrived as Professor of Computation about a year earlier.

At the time the PRG still occupied a large semi-detached house at 45 Banbury Road, and the first challenge for the three interviewees had been to believe that they were in the right place, for there was little or no indication of that on the outside of the building.

To our surprise, we all arrived at the house at exactly the same time, and Tony seemed completely unfazed that one of us was wearing swimming shorts and a singlet[9] "Do you mind all being interviewed at the same time", he greeted us without flinching, "like in the Civil Service?"

The ostensible goal of the project we were competing to staff was to publish the texts of high-quality software. I imagined that this would be along the lines

---

[9] For years afterwards he would tell people that it had been me wearing the swimming shorts: but it wasn't – my own noticeability stunt had been to arrive on a huge motorbike; and the third candidate is still memorable because he was wearing a tweed suit and waistcoat on the hottest day of that year.

pioneered by Strachey and Stoy when they published an annotated BCPL text of their operating system OS/Pub in 1972[10].

The reason I say "ostensible" is that in those days project proposals could promise insight without having to commit to a timetable of deliverables. And in those early years we were all to learn from Tony that progress in the general direction of a project's goals would nearly always generate enough insight to please its sponsors, and to carry us into the next project or two.

In fact the interview was a round-table discussion: though it never became clear what the role of the four grandees present beside Tony and Joe was, for they hardly spoke. Questions soon arose about what kinds of program should be published and what programming languages should be used.

I said that I thought we would need to publish in the highest-level language possible: "efficiency be damned." Whatever programs we would publish, it was the ideas behind them that mattered; and I recall saying that these would be more straightforwardly expressed in Lisp, because of the ease with which (what we would now call) domain-specific control and data abstractions could be described. Having not a clue about the extensive work initiated by Tony at Belfast to improve Pascal[11], when it came up as a candidate language I was pretty forthright about what I thought were its shortcomings as anything other than a pedagogical tool[12].

At the time I was effectively ignorant of the emerging disciplines of data and program refinement that Tony did so much to teach us about, so it had been the reputation of the PRG that had attracted me.

Almost all my research[13] had been conducted in the spirit of the "invisible college of Strachey." Most of the large programs I'd built had either been written in Lisp or BCPL[14]. I had also been an early and enthusiastic subscriber to

---

[10] BCPL is best known as a precursor of C. To the best of my knowledge, this was the first time the complete text of an operating system written in a high-level language had been published.
[11] Pascal Plus – a language whose relationship to Pascal was analogous to that of Simula67 to Algol60. See: https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.4380091109.

[12] Tony and Joe seemed to like this: but my co-interviewees really didn't, and competed vigorously to show who was the better defender of Pascal. The man in swim shorts also turned out to be an enthusiast for Algol68; and it is (now) clear that I wasn't the only one there who wasn't completely *au fait* with Tony's work.

[13] At the University of Essex, CMU, Bolt-Beranek and Newman, then Essex again.

[14] For example, I'd built a fast Gedanken implementation inspired by Landin's "mechanical evaluation of expressions" in BCPL soon after reading John Reynolds's paper. I'd completed an implementation of BBN-BCPL for PDP-11s when I worked in the USA, and I'd also implemented a

the PRG monograph series, and had followed the developing work on mathematical semantics, despite feeling ill-equipped to understand its theoretical foundations.

Junior?

Soon after the interview I got a job offer from Tony, and a letter from Wolfson College saying that I had been "elected to a junior research fellowship" there. Being a Sheffield and Essex alumnus I wasn't used to this terminology; and being 31 at the time, and the tenured head of a small group at Essex, I didn't really consider myself "junior".   Tony's very first professional kindness to me was to explain in our subsequent phone call that I mustn't take the "junior" too seriously, that it was his intention to build a proper academic department, that it would be advantageous to get in on the ground floor, and that I could have a few years to prove myself[15].

The PRG

I arrived at 45 Banbury Road on my motorbike on the first of September in 1978, to join a PRG that consisted of two academics (Tony and Joe), service staff (two programmers, a caretaker, and a PA), and around half a dozen first year D.Phil students.

I had packed and sent a hundred or so books and papers to the PRG in advance of finding somewhere to live; and was astonished to find that the staff had added them all to the machine-readable PRG library catalogue. They included a few grocery invoices, several books on radical US politics, and a dozen anti-war pamphlets; every item had been allocated an accession number, and given the corresponding library sticker. Service indeed!

During working hours Tony seemed to spend an inordinate amount of time in administration: as Professor of Computation he had inherited multifarious responsibilities, including chairing the management committee of the University Computing service, and overseeing the work of the Computing Teaching Centre – a large operation teaching not-for-credit courses to anybody in any department who was interested. I learned later that much of what I'd taken to be routine administration had actually been working through Oxford's ramified committee structures to divest the post of these responsibilities.

---

multithreading variant of BCPL. Other large projects had included a typesetting system at the core of whose operation was a microprocessor inspired by Strachey's GPM.

[15] The advantage of having been interviewed with the other two was that I was sure by the end of the day that if one of them were offered the post then I'd have been misreading Tony's body-language and couldn't have done the job anyway.
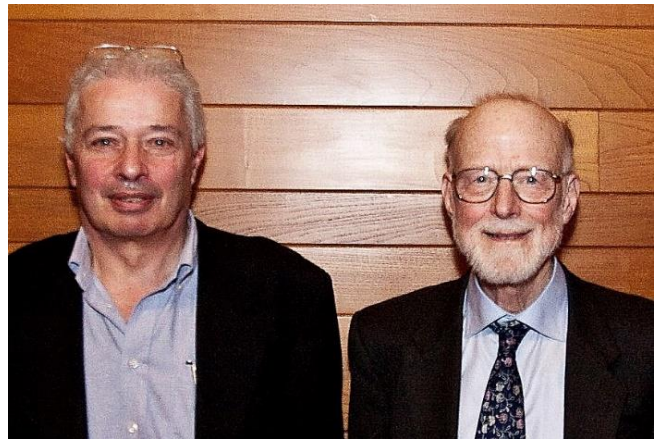
Tony outside my Magdalen College room, late summer 2015

At that time, and for years later, he did his creative thinking and writing at home: disappearing at five o clock, then returning in the morning with long manuscripts that he'd often share with us. His ability to switch from administration to scientific work without losing his focus was extraordinary. And his heuristic to avoid being overburdened by the pile of internal correspondence on his desk was audacious: "if I can't deal with something immediately, then I add it to the [discard] pile; if the senders think it important then they'll send it again."

Tony and I would walk to Wolfson for lunch nearly every day. It really was a free lunch, but the main attraction was having Tony to myself. He walked very quickly, and though a keen and strong walker myself I found it impossible to maintain his pace without breaking into a trot from time to time. I had asked him on my first day what he wanted me to do, and he had said that I should please myself, but that if I felt like it, I should teach a course.

Tony and me at Wolfson College after I had delivered (ventriloquising him) the second part of his 2011 Haldane Lecture: "Applied Logic".

Once I decided that I was going to teach a course on programming language implementation I was able to run some ideas past him – I'd done some thinking about targeting (what I called) tree interpreters implemented in microcode[16] and was keen on exploring these in a course.

His counterproposal was that I instead talk about targeting stack machines implemented in the same technology. This was not to be the last time during our work together that he said his piece then deployed his trademark phrase: "I

---

[16] User micro-programmable bit-slice processors had started to emerge a few years earlier.

hope you make the right decision[17]." It was the first of countless examples I recall of him providing clarity, without giving instructions: one of the real signs of academic leadership.

The most difficult thing about giving the course was the incredulous reaction of the University Examination Schools to my request for a lecture room to teach it in. "But you are not a matriculated member of the University", they said, "so we cannot." Being a matriculated member of two other Universities cut no ice with them. "Get your college to matriculate you", they suggested. But my college was no more helpful "But do you teach for us?", the senior administrator asked. Being told that most of the Doctoral students in the PRG who would be coming to the course were at Wolfson also cut no ice: unless I *already* taught for them, they would not matriculate me.

Tony wryly told me that he could do absolutely nothing about any of this and recounted his own difficulties in getting any resources at all from the University for the group; let alone stable commitments that would let us move in the direction of establishing an undergraduate degree[18].

---

[17] I suppose I must have, for I still own the front panel and four boards of a High-Level Hardware Orion: a machine designed and commercialized by two of the attendees at the lectures, along lines I preached about in the course. For a short time – before it was eclipsed by Sun – this machine was a *sine qua non* for UK CS researchers, particularly those working in functional and logic programming.

[18] I have written elsewhere about Tony's role in the PRG's long march to proper department-hood. See: https://doi.org/10.1145/3477355.3477366.

Second cohort of [M.Sc students] and their teachers.
From rear, LTR: _ _ *Abrial Sufrin [Bill Richardson] [Ian Cottam] _*
Front: *Stoy _ Jones Hoare _ [Mary Sheeran] _*

Joe's contribution was to send me an elegantly written nineteenth-century squib on the subject of not knowing what, exactly, it was that one had joined when one joined Oxford; but this only reinforced my views about the weirdness of the place, and the soundness of Tony's advice to be patient. Eventually I got a hint that I could "squat" in a lecture theatre in Atmospheric Physics; and the course went ahead.

Operating Systems Wars

In a tradition established by Strachey, the group would meet weekly around Tony's dining table or in the library upstairs, and discussions ranged from scientific to technical. The technical discussions were mostly about how to replace the superannuated small Modular One that provided our *only* in-house computing facility.

Its operating system had evolved from OS/Pub. It was a two-phase cooperatively-interactive system – in the sense that either *everybody* had to be editing, or *everybody* had to agree to stop editing and let the queued batch

jobs all run to completion[19]. The phase-change negotiations entailed much shouting up and down the four flights of stairs[20].

Tony chaired the technical meetings with an amused detachment that I understood once he told me that he had no need for a computer. His detachment was to be seriously tested during the "Operating System War" that followed us buying a large microprogrammed Interdata 8/32 machine, and inviting Richard Miller as a visitor – he had done the first port of Unix to another architecture, and it was for the 8/32. There were two camps: the utilitarians who wanted to use Unix on it; and the idealists, who had planned to microcode the BCPL abstract machine, and to run a derivative of the in-house system on it. It must have been difficult for the latter not to take matters personally. Joe, with the support programmers, had done a tremendous job in microcoding the BCPL abstract machine so that each user would have their own virtualised abstract machine; and Joe himself was admirably patient with the hot-headed utilitarians during this episode[21].

Tony eventually brought peace by making the judgment of Solomon: the working day would be cut in half: the in-house system would run for one half, and Unix would run for the other half[22].

Editors

During one of our walks, I spoke of my interest in interactive text editors (of which I had already built too many), and Tony suggested that I choose a well-known one and present a "rational reconstruction" of it. I struggled for a month or so, and eventually came up with something a bit like the Unix editor e, written in a highly modular form in Ada. I wasn't really satisfied with the outcome, and nor was Tony.

---

[19] One of its quirks was that when someone did a complex search on a large document, all the other terminals would stop echoing until the search had completed.

[20] A couple of the DPhil students were a bit too enthusiastic about the shouting. I'd been used to quieter surroundings and much better computing facilities and quickly decamped to a room in the High-Energy Physics particle-tracking lab, where I used their DEC10. I am still astonished that Peter Mosses had managed to build his **Semantics Implementation System** on the PRG system, and that Joe Stoy had used it to write his monumental work on Denotation Semantics. See:
https://scholar.google.co.uk/citations?view_op=view_citation&hl=en&user=fIK8JS8AAAAJ&citation_for_view=fIK8JS8AAAAJ:DBa1UEJaJKAC.

[21] One of us had described the PRG as "like a computing Galapagos Island."

[22] A slight complication was that in the afterglow of the publication of Joe's book on Denotational Semantics we had somehow convinced ourselves to buy a line-printer with a λ where the X should be, so reading "raw" UNIλ printer outputs could be disconcerting.

He suggested that we ask Steve Brookes, one of his DPhil students, to do the same, and Steve very quickly came up with the editor semantics in the denotational style. His key modelling decisions were to give the semantics of *whole sessions* with the editor, and to incorporate the filestore into its state. I thought this an impressive feat, but it seemed clear that the path from a continuation-based whole-session semantics to an actual implementation would not be straightforward. My response was to make several attempts at giving a "narrative" account of the editor based on successive approximations to the (intended) reality; but it was only after Jean-Raymond Abrial and Cliff Jones arrived at the PRG that I was to learn enough about abstract specifications and program and data refinement to make a successful attempt at this.

Jean-Raymond Abrial & Cliff Jones

Jean-Raymond Abrial & Cliff Jones arrived in time for the following academic year: J-R as a research fellow funded by an individual grant, and Cliff to do his first degree (a DPhil under Tony's supervision). This happened to be the first year of the M.Sc. in Computation that Tony had persuaded the University to offer despite the PRG not really having enough staff to give it. He had deployed an argument that I saw him use a few times while we worked together – we may not have had enough permanent staff to give the necessary courses, but had enough visiting and research staff.

Jean-Raymond gave a course in Program Specification, where he introduced the variant of Z that he, Steve Schumann, and Bertrand Meyer had written about earlier that year. Cliff and Tony gave a course on Programming Language Principles – Cliff using VDM for his part[23]. Tony encouraged me to go to J-R's and Cliff's lectures, and it was at these that I learned the basis for the most powerful and liberating intellectual tools I ever used.

STL, and 2G Z

In the spring of that academic year, Tony drove Jean-Raymond, Ib Sørensen (Joe's DPhil student), and me from Oxford to meet Bernard Cohen at Standard Telecommunications Laboratories at Harlow to discuss a project that they would fund to demonstrate the effectiveness of formal specification techniques in communicating ideas about systems[24]. The project was called CAVIAR[25].

---

[23] Joe Stoy and I gave courses on Functional Programming and on Programming Language Definition and Implementation.

[24] It was a terrifying journey: Tony drove fast, with what might have been called verve had he not also kept turning round to talk to the two of us in the back of his car.

[25] Computer Assisted Visitor Information And Resources

It was to specify a system to be used by STL to manage room scheduling and resource provision for the (very many) visitors to its buildings. J-R, Tim, Ib and I spent the next few months refining the existing **Z** notation, and thinking about CAVIAR with it. We'd meet every afternoon in the smoke-filled office[26] occupied by Jean-Raymond and Ib and our non-smoking research fellow, Tim Clement. And when we weren't changing the specification we'd change the language. Jean-Raymond had the big language ideas, and worked frighteningly fast, but listened to anybody with critical ideas. What emerged from our deliberations was the "second generation" **Z**, with semantics rooted in a generically-typed set theory, and a "basic library" presented in the style of Bourbaki.

We were pleased with the overall intelligibility of the notation, though our "class'", and "class-function" notions – intended to promote incremental and discursive specifications – soon turned out to be quite unsuitable for rigorous reasoning.

By then Tony's scientific energies were focussed on CSP; and for personal reasons he wasn't in a position to engage properly with our work as well. But he trusted us to make our own way, and at his suggestion we sent the language draft to Dana Scott. Dana's response was warm but unrelenting: the language was too verbose – for example it could take half a page to write down a simple quantified predicate. Of course, he was right: we had made too many concessions to avoiding making the language *appear* too mathematical, so as not to frighten programmers in flight from mathematics.

We soon understood that anybody who was going to use the language seriously would have to be able to think mathematically, and we began the retreat from our pretty-but-verbose notations to the more orthodox forms from which the "third generation" **Z** eventually evolved. This was to become, for a while, one of the pillars of the reputation of the PRG.

Skating on safe ice

I was "unpartnered" at the time and would have been indescribably lonely during my first Christmas in Oxford had it not been for Jill and Tony's invitation to join their three children and a couple of close friends for lunch on Boxing Day. It was a freezing cold day and gloriously sunny.

In those days Port Meadow, a large water meadow by the Thames in North Oxford, would flood and freeze for a couple of miles for a few days every year; and since the water was never more than a foot or so deep it was very safe to

---

[26] Those were the days!

skate on it. After lunch the children insisted that we all go skating. I had been an ice-hockey player until I was 18 and was very keen to explore the meadow: I'd never skated on anything larger than an ice rink.

But what about skates for me? Although the Hoares turned out to have enough skates to shoe a hockey team, my feet were unhoareishly huge. Here the law of spontaneous availability, known to all packrats, came to my rescue: if you don't throw old things away then one day they'll come in useful, if you can find them. The family had a mid-Victorian pair of skates that could be clamped to my shoes, and the attic they were hiding in was soon found.

The scene when we got to the meadow was like something out of Breugel. North Oxford was out on the ice: some folks skittering along on chairs, others falling about on skates. Tony himself skated joyfully and at breakneck speed in his everyday uniform of thigh-length grey coat and hat[27].

---

[27] The clamp-on experience wasn't completely satisfactory – they were quite blunt – but it was good enough to get me to revive my old hockey skates; and later to buy a more modern pair. I still skate on the frozen meadow whenever I can.

Linda Forrest (my partner), Tony and Jill Hoare: Addison's Walk, Magdalen College, late Summer 2015

My wife and I have felt part of the extended Hoare family for very many years now, and I've never forgotten the warmth of the welcome they gave me that day.

Recollections of Tony Hoare

Jonathan P. Bowen

Emeritus Professor, London South Bank University

Here I record some recollections of Professor Sir Tony Hoare [4]. As has been said elsewhere "Recollections may vary", but these are my memories as I remember them. Apologies now for any infelicities.

I first encountered Tony for a job interview in 1984 at the Programming Research Group in Oxford to work as a "Research Officer" (or research assistant in normal UK university parlance) using the Z notation, which was entirely new to me then [7]. We initially met in the Common Room at 11 Keble Road where he made me a coffee and chatted informally before the interview, which certainly helped me feel at ease. I was then formally interviewed by Tony and Bernard Sufrin. Tony asked me what I knew about garbage collection, and I said I had heard of it. This seemed to be enough to be offered the job, although I guess there must have been some other questions too! I had the background of an "elevated engineer" rather than a "fallen mathematician" (an important intersection for computer science in my view!), but enjoyed the connection of theory and practice, as Tony famously did too.

I initially worked on the Distributed Computing Project at the PRG with Carroll Morgan and Roger Gimson, specifying network services using Z. Tony was more involved with CSP at the time, so I did not see him so much for my first few years at the PRG, but the door of his office was always open, with the possibility to see him at any time. An early memory was dinner with Tony and Jill in their Chalfont Road home, with others such as Cliff Jones.

Then in 1989, Tony invited me to dinner at St John's College with the instruction to meet at the main gate of St John's [5]. There I met for the first Dines Bjørner, from DTU in Denmark and a larger-than-life character as anyone who knows him will tell you, Cliff Jones, then at the University of Manchester after working at the PRG with Tony and others, and Hans Langmaack, based at the Christian-Albrechts-University of Kiel in northern Germany. Each had "henchmen" in attendance for discussions on what was to develop into the European ESPRIT ProCoS I and II projects [2,10] and then a Working Group [11] on "Provably Correct Systems". I was to become very much involved with helping Tony on these projects, although I was employed on the parallel UK DTI SAFEMOS project on "Totally Verified Systems", collaborating with Mike Gordon at the University of Cambridge, David May and David Shepherd at Inmos, and Roger Hale and John Herbert at Cambridge SRI, with Tony as the Oxford

investigator. The research results were later more modestly recorded in a book entitled "*Towards Verified Systems*" [3]. Visits with Tony to Cambridge included meeting Roger Needham and David Wheeler. An important discussion was how to travel between Oxford and Cambridge, with Roger Needham favouring the four "Bs" route via Bicester, Buckingham, Bletchley (halfway), and Bedford, before reaching "C" (Cambridge). This may also be a reason for the location of Bletchley Park in World War II.

Meetings for the ProCoS projects took place in various European locations. I took on the role of Tony's "minder" during journeys, helping to ensure that Tony and his briefcase remained in the vicinity of each other and on one occasion ensuring that he boarded the train going in the right direction! Early in the project, Tony produced a typically beautifully elegant approach to compiling specifications in a relational style [17]. More modestly, I was able to show how easily these could be transformed into a Prolog logic program, for compiling and even for decompiling a high-level Occam-style program to/from Transputer-like machine instructions. On reflection, my job as Tony's minder may have been more important in the scheme of things, helping, in a small way, Tony to have higher thoughts.

As the PRG expanded, some of us were "banished" to 2 South Parks Road, still in the Science Area and not so far from the main location at 11 Keble Road but nicknamed "Tasmania" by Carroll Morgan (as an Australian!) due to its remoteness. Despite this, Tony regularly visited us to discuss research progress, when I was in a shared office with Paritosh Pandya and Mark Josephs. I had programmed my Sun workstation to issue me with audio reminders, recorded by my then very young daughter Alice (now a chemistry lecturer at the University of Manchester!). One pleaded, "Please come home Daddy" and of course, this was played back during one of Tony's visits towards the end of the day. Tony obligingly terminated the discussion on hearing this!

Other memories of Tony include his early use of electronic mail. Emails to Tony would be printed by his secretary, Julie Sheppard, and handed to him on paper. He answered these in his beautifully clear handwriting, and Julie typed the replies before emailing them back.

At one point, Tony gave a lecture course on CMOS circuits to undergraduates. These were beautifully modelled in a relational style at the transistor level, presented on handwritten acetates. With this approach, it was easy to transliterate the specifications very directly as logic programs, so I helped Tony by creating the practical laboratory worksheets based on Prolog for the students. I remember Tony sitting in front of a workstation near the start of the

course and asking, "What do I do?"! Of course, all his handwritten specifications worked perfectly in Prolog, and it was a delight to help in a small way on the course, demonstrating for students during practical sessions.

At one point, just before Tony went on sabbatical leave, a large cardboard box containing a PC, with a Visual Basic manual, was to be seen in his office. I never dared ask Tony what this was doing there. Tony and the PC disappeared for a year. On his return, he confided that he had opened the box, turned on the PC, and written a Visual Basic program. The only problem was that he did not know how to save it, so we will probably never know what the program did.

In 1995, I left the PRG and became a lecturer at the University of Reading. We held the last ProCoS Working Group meeting there in 1997, which Tony attended. In 1999, I attended Tony's retirement symposium in Oxford [13]. Such is Tony's reputation that there were many ACM Turing Award winners there. Don Knuth lay in the front row, apparently asleep, perhaps due to jetlag after flying from Stanford in California, but often asking a very pertinent question at the end of presentations. Ole-Johan Dahl gave a wonderful presentation during which the lights and overhead projector failed due to a temporary power cut. From the gloom of the emergency lighting, Ole-John said, "I think I can go on", which he duly did without slides, seamlessly continuing when the power returned to restore his slide presentation.

Tony subsequently moved to Microsoft Research in Cambridge, where he had to answer his emails on a computer, perhaps due to the lack of industrial secretarial support! In 2006, I was invited to interview Tony in Cambridge, for the Computer History Museum in California [8]. As an amateur concerning interviewing, the experience could have been nerve-racking for me, but the questions were well-prepared, and Tony put me at ease as he was able to do with most people in his always modest way. An abridged version later appeared in the *Communications of the ACM* [18].

Tony also attended the celebratory ProCoS reunion workshop in 2015 [15], which I helped to organize as Chair of FACS at the BCS London office, then in Covent Garden. Dines Bjørner presented his memories of how the ProCoS project started, including notes that Tony wrote in Dines' notebook on ideas to form the basis of research on the project. Tony kindly authored the foreword in the subsequent proceedings, written in his characteristically generous and eloquent style.

Walking with Tony was always an experience. He and I both like walking and at reasonable speed too. My last walk with Tony was in Reykjavik, Iceland, for the

UTP (Unifying Theories of Programming) Symposium there in 2016 [12]. He and his longtime collaborator Jifeng He, especially on UTP, were both keynote speakers. Tony's mind was as sharp as ever in our discussion.

The last time I saw Tony was in 2019 for the celebration of the 40th anniversary of FACS and the 20th anniversary of Tony's book on *Unifying Theories of Programming* with Jifeng He [16]. Jifeng gave the main talk, with Tony delivering some introductory commentary and Jim Woodcock providing a summary at the end [6]. It was gratifying that they could meet on this occasion, with Jifeng visiting from Shanghai in China.

After my job interview with Tony at the PRG in 1984, I spent the summer working at the then start-up company Silicon Graphics, in Mountain View at the heart of Silicon Valley in California. My Engineering Science tutorial partner at Oxford was an American and he returned to study for a master's degree at Stanford University in the US under Jim Clark, who invited him to become one of the seven founder members of the company. At the time I was also a research assistant in the Wolfson Microprocessor Unit at Imperial College, with the offer of a job at the PRG (at about a quarter the salary of that at Silicon Graphics). Should I stay on after three months at Silicon Graphics, receive stock options, and potentially become a millionaire? Or should I return to work under Tony Hoare at the PRG in Oxford, also my birthplace? Naturally, I chose the latter, such was the draw of Tony and Oxford! In any case, I would probably have lost my millions in the dotcom crash, knowing my lack of business acumen. This tribute is to thank Tony for that major decision point in my career. And of course, to congratulate him on his lifetime of contributions to computer science, always done in a very modest way, which I admire immensely.

Happy 90th birthday, Tony.

References

1. Bjørner, D. (2017). *ProCoS: How It All Began – as Seen from Denmark*. In [15], pp. 3–6.
    https://doi.org/10.1007/978-3-319-48628-4_1
2. Bjørner, D., Hoare, C.A.R., Bowen, J.P., He, J., Langmaack, H., Olderog, E.R., Martin, U., Stavridou, V., Nielson, F., Nielson, H.R., Barringer, H., Edwards, D., Løvengreen, H.H., Ravn, A.P., Rischel, H. (1989). A ProCoS project description: ESPRIT BRA 3104. *Bulletin of the European Association for Theoretical Computer Science*, 39:60–73.
    https://researchgate.net/publication/256643262
3. Bowen, J.P. (ed.) (1994). *Towards Verified Systems*. Real-Time Safety Critical Systems, vol. 2. Elsevier.

4.   Bowen, J.P. (2001). Hoare, C. Antony R. In: Rojas, R. (ed.), *Encyclopedia of Computers and Computer History*, pp. 368–370. Fitzroy Dearborn Publishers.

5.   Bowen, J.P. (2019a). *A Personal Formal Methods Archive*. ResearchGate. https://doi.org/10.13140/RG.2.2.31943.65447

6.   Bowen, J.P. (2019b). FACS Events, 2018–2020. *FACS FACTS*, 2019(1):7–18, December 2019. https://www.bcs.org/media/5204/facs-dec19.pdf

7.   Bowen, J.P. (2022). Review on Theories of Programming: The Life and Works of Tony Hoare. *Formal Aspects of Computing*, 34(3–4):1–3. https://doi.org/10.1145/3560267

8.   Bowen, J.P., Hoare, C.A.R. (2006). *Oral history of Sir Antony Hoare*. Tech. Rep. X3698.2007, Computer History Museum, California, USA, September 2006, Catalog Number 102658017. https://www.computerhistory.org/collections/catalog/102658017

9.   Bowen, J.P., Hoare, C.A.R., Hansen, M.R., Ravn, A.P., Rischel, H., Olderog, E.-R., Schenke, M., Fränzle, M., Müller-Ulm, M., He, J., Zheng, J. (1994). Provably Correct Systems – FTRTFT'94 tutorial. In: *Third International School and Symposium, Formal Techniques in Real Time and Fault Tolerant Systems*, Lübeck, Germany, September 1994. ProCoS document [COORD JB 7/1]. https://researchgate.net/publication/2420842

10. Bowen, J.P., Hoare, C.A.R., Langmaack, H., Olderog, E.R., Ravn, A.P. (1996). A ProCoS II project final report: ESPRIT Basic Research project 7071. *Bulletin of the European Association for Theoretical Computer Science*, 59:76–99. https://researchgate.net/publication/2255515

11. Bowen, J.P., Hoare, C.A.R., Langmaack, H., Olderog, E.-R., Ravn, A.P. (1998). A ProCoS-WG Working Group final report: ESPRIT Working Group 8694. *Bulletin of the European Association for Theoretical Computer Science*, 64:63–72. https://researchgate.net/publication/2527052

12. Bowen, J.P., Zhu, H. (eds.) (2017). *Unifying Theories of Programming: 6th International Symposium, UTP 2016, Reykjavik, Iceland, June 4–5, 2016*. LNCS, vol. 10134. Springer. https://doi.org/10.1007/978-3-319-52228-9

13. Davies, J., Roscoe, B., Woodcock, J. (2000). *Millennial Perspectives in Computer Science, Proceedings of the 1999 Oxford–Microsoft Symposium in honour of Sir Tony Hoare*. Cornerstones of Computing, Palgrave.

14. Denvir, T. (2013). Peter Landin Annual Semantics Seminar: Professor Sir Tony Hoare (December 3rd 2012). *FACS FACTS*, 2013(1):5–7, December 2013. https://www.bcs.org/media/3083/facs-dec13.pdf

15. Hinchey, M.G., Bowen, J.P., Olderog, E.-R. (eds.) (2017). *Provably Correct Systems*. NASA Monographs in Systems and Software Engineering, Springer. https://doi.org/10.1007/978-3-319-48628-4

16. Hoare, C.A.R., He, J. (1999). *Unifying Theories of Programming*. Prentice Hall International Series in Computer Science. http://www.unifyingtheories.org

17. Hoare, C.A.R., He, J., Bowen, J.P., Pandya, P. (1990). An algebraic approach to verifiable compiling specification and prototyping of the ProCoS level 0 programming language. In: Directorate-General XIII of the Commission of the European Communities (ed.), *ESPRIT '90 Conference*, pp. 804–818. Kluwer Academic Publishers. https://doi.org/10.1007/978-94-009-0705-8_65

18. Shustek, L. (2009). An interview with C.A.R. Hoare. *Communications of the ACM*, 52(3):38–41. https://doi.org/10.1145/1467247.1467261

Jonathan Bowen and Tony Hoare during the FTRTFT'94 conference in Lübeck, Germany, September 1994 [9].

*Tony Hoare and Jonathan Bowen at dinner
after Tony's BCS-FACS Peter Landin Semantics Seminar
in London, 3 December 2012 [14]. (Photograph by Sue Black.)*

*Jonathan Bowen, Tony Hoare, and Jifeng He
at the UTP Symposium, Reykjavik, Iceland, September 2016 [12].*