DOI 10.1007/s00165-012-0257-0

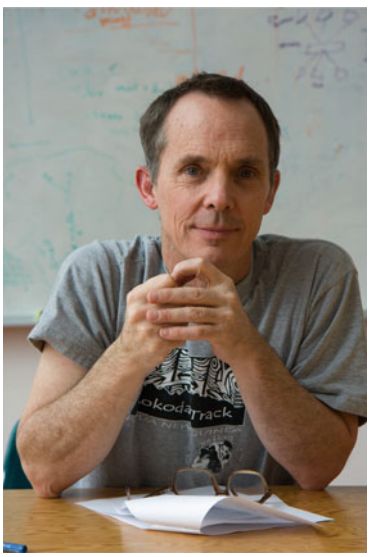
Formal Aspects of Computing (2012) 24: 417–422

Formal Aspects of Computing

Preface

Morgan: a suitable case for treatment[1]

"Can one charming madman save the only thing in the real world that's lived up to his best fantasies?" [Mor66]



This triple issue of *Formal Aspects of Computing* constitutes a Festschrift dedicated to Professor Charles Carroll Morgan, on the occasion of his sixtieth birthday. The Festschrift consists of an invited paper and 23 scientific research papers, all related to Carroll's own research interests. Three of these papers (below indicated by an asterisk (∗)) could not be included in these issues due to space limitations. They are available online and will appear in hard copy in a later issue of the journal.

This preface sheds some light on Carroll's life and scientific achievements. Due to his widespread interests, it can only give an impression of what he has achieved. Carroll is not only a brilliant computer scientist, but also a master of words: who else would come up with a title such as "Almost-Certain Eventualities" for a research paper. To acknowledge this creativity all forthcoming section headers are quotes of titles (or parts of titles) of research papers authored or co-authored by Carroll.

Carroll can be described as a researcher who is excited and attracted by problems that in some cases are not recognised as such: only the proposed solution reveals that a problem was there. He applies deep knowledge of one area to assist with difficulties in another. An example is given below. In sum, Carroll is a suitable case for (special) treatment.

The Shadow Knows

Charles Carroll Morgan was born in Washington, DC, USA, in 1952. At the age of 14, he moved to Australia—since there were no direct flights at that time, he travelled from the USA to Australia via Hawaii, Fiji and New Zealand. In 1970 he started his studies at the University of New South Wales (UNSW), Sydney, graduating with a BSc in Mathematics and Computer Science. His thesis was supervised by Ken Robinson. By winning the university medal for computer science, an award given to young students who showed highly distinguished merit in their program, he stepped into the light of computer science and not only the shadow knew that an excellent scientist entered the stage. In 1978 he stepped out from the shadow even further when finishing his PhD thesis on *"Parallel programming without synchronisation"*[2] [Mor78] at the University of Sydney, under supervision of Ian Jackson and Ross Quinlan. He has been interested in the general topic of concurrency ever since.

Correspondence and offprint requests to: P. Höfner, E-mail: peter.hoefner@nicta.com.au

[1] The sub-title is referring to Charles Carroll Morgan (born 1952), not to the movie shown first in 1966.

[2] There are rumours that Carroll's PhD thesis does not contain the word "computer"; we were not able to verify this.

Laws of Programming

In 1978, Carroll became computer programmer at Ascomp, where he had to program mainframe accounting systems in FORTRAN in the middle of the night, when the computers were not being used for accounting.

From 1980 to 1981, he was part-time lecturer in computer science at the University of Sydney.

Afterwards he joined the Programming Research Group under the leadership of Sir Tony Hoare at the University of Oxford. With his colleagues, Sir Tony Hoare, Jifeng He, Bill Roscoe and Jeff Sanders to name only a few, he worked on *"Laws of Programming"* [HHH+87].

Four papers in this Festschrift follow *Laws of Programming*:

- The invited paper *"In praise of algebra"* by C.A.R. Hoare and S. van Staden surveys the well-known algebraic laws of sequential programming and extends them with some less familiar laws for concurrent programming; relevant laws for concurrent separation logic, for Milner transitions, for the Back/Morgan refinement calculus, and for weakest preconditions are discussed. Some of these laws stem from [HHH+87].
- J.N. Oliveira gives an approach *"Towards a linear algebra of programming"*, an extension of the Algebra of Programming [BdM96] with probabilistic functions, expressing the likelihood of ambiguous, multiple outputs. He shows that matrices provide adequate support for the extension, while preserving the pointfree reasoning style typical of the Algebra of Programming.
- In *"Dijkstra, Floyd and Warshall meet Kleene"*, P. Höfner and B. Möller shed an algebraic light on shortest path algorithms. By combining shortest path problems with Kleene algebra, purely algebraic versions of the algorithms of Dijkstra, Floyd and Warshall are derived.
- By enriching probabilistic automata with a layered composition operator, M. Swaminathan, J.-P. Katoen and E.-R. Olderog adopt the communication closed layer concept to formal *"Layered reasoning for randomized distributed algorithms"*. Using layered separation in compositions of probabilistic automata, a randomised mutual exclusion algorithm is analysed, complementing an approach by Carroll et al. [MGCM08].

During his time at Oxford he held positions such as Junior Research Fellow, SERC Research Officer, Tutorial Fellow (Pembroke) and College Lecturer (Somerville). His major achievements were in the area of specification, mainly using the specification technique *Z* [Mor85]: *Z* was known and used since the late 1970s [ASM80] to describe large and complex static structures by combining building blocks known as *schemas*. However, dynamic properties could not be expressed by *Z*-schemas—instead λ-expressions were used. After identifying this major deficiency, Carroll proposed the notion of Dynamic-schemas (Δ-schemas), which are in line with the building-blocks used for static properties, and developed a corresponding calculus. Nowadays, Δ-schemas are widely used and included in the ISO/IEC13568-standard [ISO02]. While developing theoretical concepts for *Z*, he also used the specification language to describe real systems, such as the Unix filing system [MS84] and a telephone network [Hay87, Chapt. 3].

How to Brew-up a Refinement Order

Between 1989 and 1996 Carroll was a University lecturer in Computation at the University of Oxford. His three sons Tristan, Elliot and Ethan were born during this time. He spent a sabbatical at the University of Amsterdam in 1990 and started to teach himself Dutch, which he now speaks fluently. He proposed rigorous development techniques for small-scale, often safety-critical, programs. His research resulted in the *refinement calculus* [Mor88, WM90, MV92], which had been independently developed by Ralph-Johan Back [Bac78] and by Joseph M. Morris [Mor87]. The main idea is that "specifications and code are all programs; that there is a refinement order between programs; and that there is a specifically restricted sub-language called 'code' that allows programs written in it to be executed" [Mor90b]. Nowadays, refinement and refinement calculi form an entire sub-area within theoretical computer science; they are central concepts in many papers of this collection.

- The paper *"Reconciling real and stochastic time: the need for probabilistic refinement"* by J. Markovski, P.R. D'Argenio, J.C.M. Baeten and E.P. de Vink extends an ACP-style discrete-time process theory with discrete stochastic delays. Race conditions are resolved using conditionally-distributed unit delays.
- In *"Stepwise refinement of heap-manipulating code in Chalice"* K.R.M. Leino and K. Yessenov describe a system with automated tool support for refinement. They provide a view of heap-manipulating code based on a model of memory permissions, and a checking algorithm that encodes refinement proof obligations to harness the power of SMT solvers.

– *"Using refinement calculus techniques to prove linearizability"* by B. Jonsson presents rules for refinement of multi-threaded shared-variable concurrent programs. The work yields strengthenings of previous techniques for atomicity refinement. The developed refinement rules are applied to linearisability of Treiber's concurrent stack algorithm and Michael and Scott's concurrent queue algorithm.

Carroll presented his work on the refinement calculus at the Summer School Marktoberdorf 1992; the lecture notes appeared in [Bro93].

Besides [BvW98], Carroll's book *"Programming from Specifications"* [Mor90b]—translated into Chinese under the name 从规范出发的程序设计—can be seen as *the* reference for the refinement calculus: it "provide[s] a thorough treatment of elementary program constructs which were later incorporated as elements of B by Jean-Raymond Abrial".[3] In 1992 Michael J. Butler (currently at the University of Southampton) finished his PhD thesis *"A CSP Approach To Action Systems"*, which was supervised by Carroll.

– In *"External and internal choice with event groups in Event-B"* M.J. Butler modifies Carroll's CSP-like failures-divergence semantics for action systems [Mor90a] to allow both internal and external choice to be represented directly. This leads to a refinement rule for the preservation of choice between event groups while allowing for reduction of choice within event groups.

Carroll also was PhD supervisor of Steve King (now at the University of York) in the area of Z and the refinement calculus. Later, together with Ken Robinson, he co-supervised the PhD work of Thai Son Hoang (now at the ETH Zürich), on the B-method.

Refinement of Ignorance in Sequential Programs

In the early 1990s, Carroll was among the first who combined concurrent programming paradigms with sequential ones [Mor90a]. Although his main interests concern the foundations of programming, he has never lost touch with real programs and real applications. This volume contains five papers exploring and exploiting programming paradigms:

– *"Constraint-based correctness proofs for logic program transformations"* are discussed by A. Pettorossi, M. Proietti and V. Senni. They propose a method for the automatic generation of clause measures, which takes into account the particular program transformation at hand. Through some examples they show that their method is more powerful and practical than other methods proposed in the literature.

– In *"On termination and invariance for faulty channel machines"* P. Bouyer, N. Markey, J. Ouaknine, P. Schnoebelen and J. Worrell focus on channel machines with insertion errors and consider the problem of whether a given insertion channel machine has an infinite computation all of whose configurations satisfy a given predicate. Using their analysis of this problem they show that the satisfiability problem for the safety fragment of Metric Temporal Logic is non-elementary.

– R.S. Bird discusses work *"On building cyclic and shared structures in Haskell"*. In particular, he shows how to build certain cyclic and other multi-linked structures without the use of explicit pointers.

– K. Emoto, S. Fischer and Z. Hu equip MapReduce with a programming theory in calculational form in *"Filter-embedding semiring fusion for programming with MapReduce"*. By integrating the generate-and-test programming paradigm and semirings for aggregation of results, a novel parallel programming framework for MapReduce is proposed.

– With *"The safety-critical Java memory model formalised"* A. Cavalcanti, A. Wellings and J. Woodcock provide a formalisation of the model of memory regions of Safety-Critical Java, using Hoare & He's Unifying Theories of Programming. (∗)

In 1995 Carroll spent his sabbatical at the University of Utrecht, the Netherlands. A year later, he became University reader in computation at Oxford.

While his early work was mainly based on specification and refinement he turned to probabilistic semantics around 1995. His seminal paper *"Probabilistic Predicate Transformers"* [MMSS96] (co-authored by Annabelle McIver and Karin Seidel) was one of the first approaches combining refinement with probabilistic computations. It provides a dual formulation for programs containing nondeterministic and probabilistic choice: one formulation for testing, one in terms of executions. It is among his most cited papers.

[3] http://en.wikipedia.org/wiki/Carroll_Morgan_(academic) (1 June 2012).

The Thousand-and-one Cryptographers

In 2000 Carroll left Oxford, returned to Australia, and joined his Alma Mater UNSW as adjunct professor. In the early 2000s, he had another "excursion" to industry as software systems consultant at Craftware Solutions, Sydney, where he used J2EE to produce online superannuation advisors.

The paper [MM02] (co-authored by Annabelle McIver) gives an operational interpretation of the quantitative μ-calculus qMμ as a turn-based game between two players. It shows that, when the state space is finite, memoryless strategies suffice for achieving the minimax value of the quantitative game.

– *"Deterministic concurrent strategies"* are formalised by G. Winskel as maps in a bicategory. He shows that these strategies in a game can be identified with certain subgames, with the benefit that the bicategory of deterministic games becomes equivalent to a technically-simpler order-enriched category.

The book *"Abstraction, Refinement and Proof for Probabilistic Systems"* [MM05a] (co-authored by Annabelle McIver) summarises, substantially updates and rationalises much of Carroll's work done (with colleagues) in almost 10 years in the field of probabilistic semantics and refinement.

Between 2003 and 2007 Carroll held an ARC Australian Professorial Fellowship at UNSW, a research-only position that is highly regarded. In this period, he also paid an extended visit to the Laboratoire de Recherche en Informatique (LRI) at the Université Paris-Sud in Orsay, France.

Eleanor, his youngest child, was born.

Reading books and solving (logic and mathematical) puzzles is one of Carroll's hobbies. One day while sitting in a café in Manly (a Sydney beach-side suburb) Carroll was reading a book on mathematical puzzles [Hon03]. While reading and thinking about a puzzle involving an old wise man, he discovered that the puzzle was closely related to Herman's Ring, an algorithm for self-stabilisation of N identical processors connected in a ring. This sudden inspiration yielded an elementary proof that Herman's Ring is in $O(N^2)$ and has a worst-case running time of $\Theta(N^2)$ [MM05b]. This paper (co-authored by Annabelle McIver) conjectured that the worst-case behaviour results from a ring configuration of three evenly spaced tokens.

– M. Kwiatkowska, G. Norman and D. Parker analyse this conjecture in *"Probabilistic verification of Herman's self-stabilisation algorithm"* by means of probabilistic verification techniques, using the probabilistic model checker PRISM. The conjecture is confirmed for all ring sizes that can be exhaustively analysed.

– In *"Three tokens in Herman's algorithm"* S. Kiefer, A.S. Murawski, J. Ouaknine, B. Wachter and J. Worrell show for rings with three tokens that the probability of stabilisation within a given time is minimised when initially the tokens are evenly spaced. This strengthens a corollary of [MM05b], which states that the *expected* stabilisation time is minimised by the equidistant configuration.

In 2007/2008 Carroll spent his sabbatical in the Software Design Group at the MIT computer science and artificial intelligence laboratory.

Security, Probability and Nearly Fair Coins in the Cryptographers' Café

Since 2008 Carroll has been a professor at UNSW, while also maintaining an office at NICTA (National ICT Australia Ltd.). In this period he started to apply the insights in combining probability and nondeterminism gathered in the sequential world, to the world of concurrency, in particular to pCSP, an extension of CSP with probabilistic as well as internal and external choice [DvGH+07, DvGMZ07, DvGHM08].

Four papers in this Festschrift follow this line of research:

– *"Using schedulers to test probabilistic distributed systems"* by R.M. Hierons and M. Nuñéz presents a formal testing framework to deal with systems that interact with their environment at physically distributed interfaces, and where choices between different possibilities are probabilistically quantified.

– Two *"Characterisations of testing preorders for a finite probabilistic π-calculus"* are given by Y. Deng and A. Tiu: one based on probabilistic weak simulations and one on a probabilistic extension of a fragment of the Milner–Parrow–Walker modal logic. It is shown that in order to capture the testing preorders, one needs to use the "earliest" simulation relation.

– In *"Probabilistic may/must testing: retaining probabilities by restricted schedulers"* S. Georgievska and S. Andova propose and characterise a testing preorder for processes having internal, external and probabilistic choices, which is insensitive to the exact moment the internal and the probabilistic choices occur.
– With *"Exploring probabilistic bisimulations, part I"*, M. Hennessy takes a fresh look at strong probabilistic bisimulations, moving from relations between states in a probabilistic labelled transition system to relations between distributions. He also shows that a very simple extension to the Hennessy–Milner Logic provides finite explanations for inequivalences between distributions.

The papers [Mor06, Mor09] propose a notion of refinement that facilitates reasoning about knowledge, security and privacy, by ensuring that implementations reveal no more than their specifications. This "ignorance-preserving" refinement was the first bridge between refinement-based development and software security/privacy. It provides a "logic of ignorance", as well as a qualitative model for noninterference security for sequential programs, the *shadow semantics*.

– R. van der Meyden deals with *"Architectural refinement and notions of intransitive noninterference"*. He studies the soundness of such refinements with respect to a spectrum of different semantics for information flow policies.
– *"A secure voting scheme based on rational self-interest"* is proposed by J. Misra. The scheme involves the candidates themselves in implementing the voting system, and exploits the competing interests (rivalry) and mutual distrust among the candidates to force an honest election.
– In *"Checking non-interference in Timed CSP"*, A.W. Roscoe and H. Jian argue that the well-established notion of noninterference in CSP is not suitable for Timed CSP, and propose an improvement. They show how to automate the resulting timed noninterference check within the context of a recent extension of the refinement checker FDR. (∗)
– *"Security invariants in discrete transition systems"* are studied by T.S. Hoang. He extends the shadow semantics of Carroll to Event-B to reason about discrete transition systems with noninterference security properties. (∗)

Recently, Carroll spent his sabbatical at the Radboud University Nijmegen, the Netherlands.

Sums and Lovers

Let us sum up Carroll's love of academia. During his career he published more than 75 papers. He is best known for his work on Z, on the Refinement Calculus and on probabilistic semantics. Everyone who has had the pleasure to listen to one of his talks or lectures will recognise him as an excellent teacher. Indeed, last year one of his students said: "[He has a] definite interest and enthusiasm in the subject matter, and the ability to transfer this to the students".

Next to his sabbaticals in the Netherlands, France and the USA, he spent time at the University of Queensland, Australia, and at NICTA.

He is an editor of ACM Transactions on Computational Logic (TOCL), and member of the IFIP working groups 2.1 (Algorithmic Languages and Calculi) and 2.3 (Programming Methodology).

Almost-Certain Eventualities

It is almost certain that eventually some acknowledgements are given. Our thanks go out to Annabelle McIver for essential advice and bibliographic information. Other information about Carroll's life was taken from the webpage of the software design group of the MIT. We are grateful to Daniel Jackson for providing the photograph. Furthermore, we thank the editors and the publishers of *Formal Aspects of Computing* for publishing this Festschrift—in particular John Cooke, Jim Woodcock and Sabine Taeger for their enormous help, as well as Eerke Boiten who handled the paper written by P. Höfner and B. Möller. Last but not least we send acknowledgements to the authors for their submissions and the reviewers for their work, the latter often done under great time pressure.

Peter Höfner
Rob van Glabbeek
Ian Hayes

References

[ASM80] Abrial J-R, Schuman SA, Meyer B (1980) A specification language. In: On the construction of programs. Cambridge University Press, Cambridge, pp 343–410

[Bac78] Back R-J (1978) On the correctness of refinement steps in program development. PhD thesis. Report A-1978-4, Department of Computer Science, University of Helsinki

[BdM96] Bird R, de Moor O (1996) The algebra of programming. Prentice-Hall, Englewood Cliffs

[Bro93] Broy M (ed) (1993) Program design calculi. In: NATO ASI series, series F: computer and system sciences, vol 118. Springer, Berlin

[BvW98] Back R-J, von Wright J (1998) Refinement calculus: a systematic introduction. In: Graduate texts in computer science. Springer, Berlin

[DvGH$^+$07] Deng Y, van Glabbeek RJ, Hennessy M, Morgan CC, Zhang C (2007) Remarks on testing probabilistic processes. Electron Notes Theor Comput Sci 172:359–397

[DvGHM08] Deng Y, van Glabbeek RJ, Hennessy M, Morgan CC (2008) Characterising testing preorders for finite probabilistic processes. Logical Methods Comput Sci 4(4)

[DvGMZ07] Deng Y, van Glabbeek RJ, Morgan CC, Zhang C (2007) Scalar outcomes suffice for finitary probabilistic testing. In: De Nicola R (ed) European symposium on programming languages and systems (ESOP'07). Lecture notes in computer science, vol 4421. Springer, Berlin, pp 363–378

[Hay87] Hayes I, editor (1987) Specification case studies. Prentice Hall, Englewood Cliffs (2nd edition published in 1992)

[HHH$^+$87] Hoare CAR, Hayes IJ, He J, Morgan CC, Roscoe A, Sanders JW, Sørensen IH, Spivey JM, Sufrin BA (1987) Laws of programming. Commun ACM 30(8):672–686

[Hon03] Honsberger R (2003) Mathematical diamonds. Dolcani mathematical expositions, vol 26. Mathematical Society of America, Providence

[ISO02] ISO (2002) Information technology—Z formal specification notation—syntax, type system and semantics. Technical Report ISO/IEC 13568, International Organization for Standardization, 2002 (International Standard)

[MGCM08] McIver AK, Gonzalia C, Cohen E, Morgan CC (2008) Using probabilistic Kleene algebra pKA for protocol verification. J Logic Algebr Program 76(1):90–111

[MM02] McIver AK, Morgan CC (2002) Games, probability, and the quantitative μ-calculus qMμ. In: Logic for programming, artificial intelligence, and reasoning, LPAR '02. Springer, Berlin, pp 292–310

[MM05a] McIver AK, Morgan CC (2005) Abstraction, refinement and proof for probabilistic systems (Monographs in computer science). Springer, Berlin

[MM05b] McIver AK, Morgan CC (2005) An elementary proof that Herman's ring is $\Theta(N^2)$. Inf Process Lett 94(2):79–84

[MMSS96] Morgan CC, McIver AK, Seidel K, Sanders JW (1996) Refinement-oriented probability for CSP. Formal Asp Comput 8(6):617–647

[Mor66] Morgan: a suitable case for treatment. Film Poster (1966)

[Mor78] Morgan CC (1978) Parallel programming without synchronisation. PhD thesis, University of Sydney

[Mor85] Morgan CC (1985) Specification of a simplified network service in Z (problem 2). In: Denvir BT, Harwood WT, Jackson MI, Wray MJ (eds) The analysis of concurrent systems. Lecture notes in computer science, vol 207. Springer, Berlin, pp 318–344

[Mor87] Morris JM (1987) A theoretical basis for stepwise refinement and the programming calculus. Sci Comput Programm 9(3): 287–306

[Mor88] Morgan CC (1988) The specification statement. ACM Trans Programm Lang Syst 10(3):403–419

[Mor90a] Morgan CC (1990) Of wp and CSP. In: Gries D (ed) Beauty is our business: a birthday salute to Edsger W. Dijkstra. Springer, Berlin, pp 319–326

[Mor90b] Morgan CC (1990) Programming from specifications. Prentice-Hall, Englewood Cliffs

[Mor06] Morgan CC (2006) The shadow knows: refinement of ignorance in sequential programs. In: Mathematics of program construction, MPC'06. Springer, Berlin, pp 359–378

[Mor09] Morgan CC (2009) The shadow knows: refinement and security in sequential programs. Sci Comput Programm 74(8):629–653

[MS84] Morgan CC, Sufrin B (1984) Specification of the UNIX filing system. IEEE Trans Softw Eng 10(2):128–142 (republished as Chapt. 4 in [Hay87]).

[MV92] Morgan CC, Vickers T (eds) (1992) On the refinement calculus. Springer, Berlin

[WM90] Woodcock J, Morgan CC (1990) Refinement of state-based concurrent systems. In: Proceedings of the third international symposium of VDM Europe on VDM and Z—formal methods in software development, VDM'90. Springer, Berlin, pp 340–351

Published online 2 July 2012