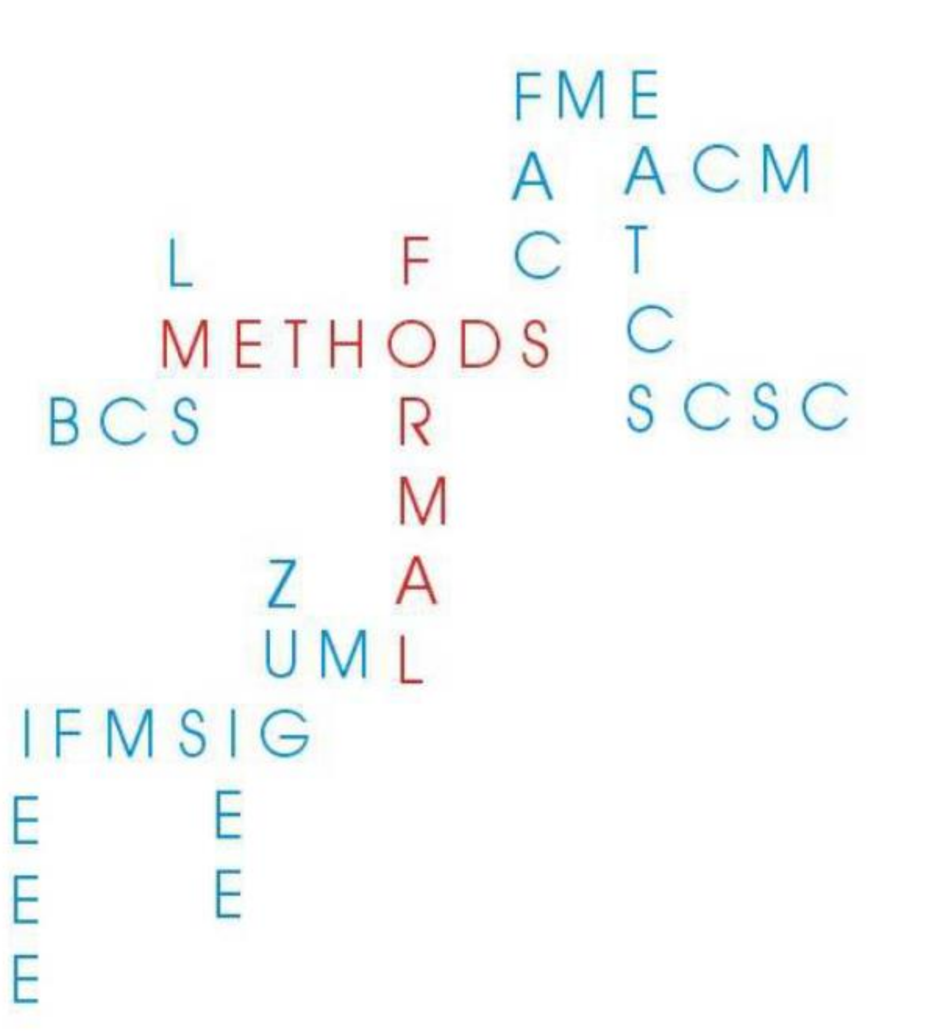
Issue 2023-2
July 2023

FACS

F A C T S



Formal Aspects of Computing Science Specialist Group

The Newsletter of the Formal Aspects of Computing Science (FACS) Specialist Group

ISSN 0950-1231

About FACS FACTS

FACS FACTS (ISSN: 0950-1231) is the newsletter of the BCS Specialist Group on Formal Aspects of Computing Science (FACS). FACS FACTS is distributed in electronic form to all FACS members.

Submissions to FACS FACTS are always welcome. Please visit the newsletter area of the BCS FACS website for further details at:

https://www.bcs.org/membership/member-communities/facs-formal-aspects-of-computing-science-group/newsletters/

Back issues of FACS FACTS are available for download from:

https://www.bcs.org/membership/member-communities/facs-formal-aspects-of-computing-science-group/newsletters/back-issues-of-facs-facts/

The FACS FACTS Team

Newsletter Editors

Tim Denvir timdenvir@bcs.org
Brian Monahan brianqmonahan@googlemail.com

Editorial Team

Jonathan Bowen, John Cooke, Tim Denvir, Brian Monahan, Margaret West.

Contributors to this issue

Jonathan Bowen, Jack Copeland, Tim Denvir, Marie Farrell, Mario Gleirscher, Matt Luckcuck, Brian Monahan, Maike Schwammberger

BCS-FACS websites

BCS: http://www.bcs-facs.org
LinkedIn: https://www.linkedin.com/groups/2427579/
Wikipedia: http://en.wikipedia.org/wiki/BCS-FACS

If you have any questions about BCS-FACS, please send these to Jonathan Bowen at jonathan.bowen@lsbu.ac.uk.

Editorial

Dear readers,

Welcome to Issue 2023-2 of the *FACS FACTS* newsletter. This is the second issue of 2023.

In this issue we have several feature papers, a conference notice, meeting reports and book reviews. But first, we would like to bring your attention to a proposed EU law on Artificial Intelligence. This is important because of the impact it will have on international standards.

It may not be widely known, but the EU is proposing a European law on AI - "the first law on AI by a major regulator anywhere":

https://www.theguardian.com/technology/2023/jun/13/artificial-intelligence-us-regulation.

The link https://artificialintelligenceact.eu provides a downloadable copy of the Act in assorted European Languages and also includes on-going information about limitations to the Act, feedback, etc.

A further link to a 2021 publication with details of the proposed EU Law is

https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3900378

FACS is planning to set up a working group to examine the implications of this in more detail and report on it in our next Newsletter in January 2024.

So now for the present issue of *FACS FACTS*: Jonathan Bowen and Jack Copeland tell us of the recent discovery and significance of a hand-written mathematical manuscript of Alan Turing; Matt Luckcuck, and colleagues Marie Farrell, Maike Schwammberger, and Mario Gleirscher discuss the challenges of autonomous systems for formal methods; Tim Denvir relates the 100-year history of hyperoperations, apparently initiated by Wilhelm Ackermann in 1924 in his search for a computable function that is not primitive recursive; There are plans in September this year to celebrate the 80th birthday of Prof. Jifeng He, an important contributor to the field of formal methods: Jonathan Bowen gives us a biography of Jifeng He and a preview of the Festschrift volume in the *LNCS* series which he has co-edited, along with the associated symposium; Alvaro Miyazawa gives us two reports on *FACS* seminars, one on *Formal Development of Cyber-Physical Systems: The Event-B Approach* by Paulius Stankaitis from the University of Newcastle in April 2023, and the other on *Topological Proofs* by Claudio Menghi from the University of Bergamo in July 2023; and Brian Monahan has provided a book review on *Human Compatible* by Stuart Russell about AI safety. Stuart Russell is a professor of Computer Science at the University of California, Berkeley, where he directs the Centre for Human Compatible Artificial Intelligence. He is an Honorary Fellow of Wadham College, University of Oxford and the vice-chair of the

World Economic Forum's Council on AI and Robotics.

As usual, we end finally with Forthcoming Events, details of the Committee etc.

We very much appreciate and look forward to contributions, including comments, from you, our readers.

We hope you enjoy *FACS FACTS* issue 2023-2!

Tim Denvir
Brian Monahan
Margaret West

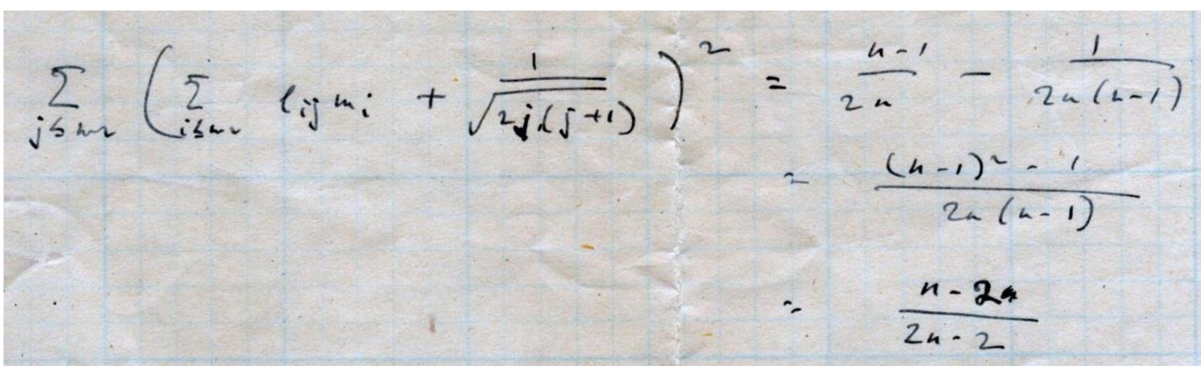
Table of Contents

A newly discovered mathematical manuscript by Alan Turing

Jonathan P. Bowen & B. Jack Copeland

JB: I am a member of the History of Mathematics Forum which meets periodically at Queen's College in Oxford to discuss work in progress relating to mathematical history in general, including some computer science. For example, I gave a talk on Alan Turing's connections with Oxford, which resulted in feedback from Chris Hollings, co-organizer of the Forum. This led to correspondence with a former student of Queen's College, Prof. Ioan James, who remembered attending a mathematical talk by Turing concerning his concept of a Turing machine (Bowen, 2022); in follow-up communication, I learned that Turing delivered the lecture in 1950 at Magdalen College, Oxford (James, 2022).

In November 2022, Chris Hollings sent me a query from Bonhams auction house about a newly discovered two-page hand-written untitled manuscript by Alan Turing, consisting of some draft mathematical notes and seemingly written during the Second World War. These were found among the papers of Rolf Noskwith (1919–2017), a fellow Bletchley Park codebreaker who worked side by side with Turing. A sample of the manuscript is included below, enhanced for readability from a yellowing image provided by Bonhams.



$$\sum_{j\le n}\left(\sum_{i\le n} c_{ij}u_i + \frac{1}{\sqrt{2j(j+1)}}\right)^2 = \frac{n-1}{2n} - \frac{1}{2n(n-1)} = \frac{(n-1)^2-1}{2n(n-1)} = \frac{n-2}{2n-2}$$

Original image ©Bonhams. Sample postprocessed by J. P. Bowen for legibility. Reproduced with permission of Bonhams and King's College, Cambridge.

I emailed Bonhams to say that I thought the manuscript appeared authentic and that the content looked like 'Turing having fun with the mathematics of equilateral triangles in n dimensions rather than official Bletchley Park work.' I also suggested that they should contact my colleague, the philosopher and historian of computing Jack Copeland, editor of *The Essential Turing* and lead co-author of a book on Turing's life and work, *The Turing Guide* (Copeland, 2004, Copeland et al., 2017). He subsequently provided Bonhams with a more detailed report and now takes up the story.

JC: Bonhams sent me JPEG scans of the two sheets of notes, which were written on standard issue HMSO graph paper. They described these as ‘our exciting Turing find’, and wanted my opinion about what Turing was working on, and whether it had any bearing on his codebreaking work at Bletchley Park. Of course, if the answer to the second of these questions were affirmative, it would presumably mean that the notes had been removed from Bletchley Park in contravention of the Official Secrets Act. The brave (or foolhardy) did from time to time smuggle documents out through the (leaky) iron curtain of secrecy—for example, the wiring plans of the Colossus computer. This was very fortunate for historians of technology, since the plans of Colossus kept in the official safe were eventually burned, on orders from above.

We still do not know everything about Turing’s codebreaking activities at Bletchley Park—in fact, that is something of an understatement—and the possibility of fresh information was an intriguing prospect, as I clicked eagerly on the JPEG images. It soon turned out, though, to be vanishingly unlikely that there was any connection between Turing’s notes and codebreaking. The contents of the notes bear no discernible relationship to any of the ciphers or analytical machinery that Turing was involved with at Bletchley Park.

Despite this initial disappointment, I found the notes very interesting. For one thing, they are in Turing’s own hand. On the back of one of the folded sheets he had scrawled ‘ROLF’. Rolf Noskwith, who magnanimously helped me years ago with the translation of some Enigma U-boat messages, was a Bletchley Park veteran with extensive first-hand knowledge of how German Naval Enigma was broken. Born in Germany, he fled to England in 1932, and then in the summer of 1941 was recruited for Turing’s Hut 8, from Cambridge, where he was by then a mathematics undergraduate. Noskwith related that, by the time he came to Hut 8, the codebreakers were reading Enigma-encrypted U-boat messages ‘almost as quickly as the Germans’. Also scribbled on the back of one of the sheets of notes, alongside Turing’s ‘ROLF’ but this time in Noskwith’s own hand, was ‘Temple Bar 3878’. Some sleuthing revealed this to be the wartime phone number of the New Theatre, in St Martin’s Lane, near London’s Leicester Square (now the Noel Coward Theatre). Turing enjoyed the theatre. Did he and Noskwith perhaps go together? It is nice to think of them jumping on the London train at Bletchley station, in pursuit of a spot of recreation.

The notes themselves also seem clearly to be recreational in nature. There was much off-duty maths going on at Bletchley Park, some of it merely fun, but some breaking new ground—for instance Turing’s correspondence with Max Newman (published in *The Essential Turing*), and his 1942 paper with Newman in *The Journal of Symbolic Logic*, about Alonzo Church’s theory of types. The aim of Turing’s two pages of notes was evidently to set out and solve a problem in n-dimensional geometry. He begins: ‘We construct an equilateral ê in n dimensions’. Then he describes some conditions that the triangle must meet, and he poses the problem of how many sets of integers there are satisfying a given equation involving the triangle’s coordinates. He reasons his way to a solution, expressed as a function of *n*. How the problem arose, why Turing and Noskwith were discussing it, and what joy Noskwith took from Turing's treatment of it in the notes, are questions with no known answers.

Also unanswered is the question of precisely when the notes were written. Noskwith and Turing worked alongside each other in Hut 8 until the autumn of 1942, when Turing left for the United States to liaise with codebreakers there. Perhaps the notes date from that period, or perhaps later. Turing arrived back at Bletchley Park from the US in March 1943, and a few months after that his workplace shifted to Hanslope Park, about ten miles north of Bletchley Park (an easy bicycle ride for Turing).

The notes somehow resemble Turing himself—cryptic, little context, precise, brimming with detail, and effortlessly tackling a problem that might defeat a lesser mind.

JB: Subsequently the manuscript was put up for auction by Bonhams in March 2023 (Lindberg, 2023). Bonhams have auctioned Turing manuscripts previously, most notably a 56-page notebook sold in New York on 13 April 2015 for US$1,025,000. After Turing's death, this had been owned by his PhD student and friend Robin Gandy (Copeland, 2017, Hodges, 2015). The two-page manuscript under discussion here was sold on 29 March 2023 for £35,000 (McKay, 2023), or £44,000 with commission (Bonhams, 2023). It was bought by King's College, Cambridge (McGuire, 2023a), which is entirely apt since this was Turing's college as a student and fellow. The college holds an extensive archive of works by Turing, largely accessible online as *The Turing Digital Archive* (https://turingarchive.kings.cam.ac.uk). The manuscript is now available on the archive's website for those that wish to peruse it in more detail (Turing, 2023) and annotated notes have also been produced (McGuire, 2023b).

References

Bonhams (2023). *Lot 230: Turing (Alan)*. Fine Books and Manuscripts. Bonhams, London. https://www.bonhams.com/auction/28320/lot/230/

Bowen, J. P. (2022). Alan Turing and Oxford. *Resurrection: The Journal of the Computer Conservation Society*, 97:11–18, March. https://www.computerconservationsociety.org/resurrection/res97.htm#e

Copeland, B. J., ed. (2004). *The Essential Turing*. Oxford University Press.

Copeland, B. J. (2017). Alan Turing's Lost Notebook, *London Mathematical Society Newsletter*, 468:22–23, April. https://www.lms.ac.uk/sites/lms.ac.uk/files/Publications/newsletter/201704.pdf

Copeland, B. J., Bowen, J. P., Wilson, R., Sprevak, M., et al. (2017). *The Turing Guide*. Oxford University Press.

Hodges, A. (2015). Turing point. *Bonhams Magazine*, 42:17, Spring. https://www.bonhams.com/magazine/18629/

James, I. (2022). Personal communication. Email via S. Dopson to J. P. Bowen, 1 January.

Lindberg, S. (2023). Alan Turing's Bletchley Park Recreational Maths At Bonhams Book Sale. Bonhams, 29 March. https://www.bonhams.com/press_release/36258/

McGuire, P. (2023a). Recreational mathematics. *King’s Treasures*, Special Collections of King’s College, Cambridge, 28 April. https://kcctreasures.com/2023/04/28/recreational-mathematics/

McGuire, P.(2023b). Alan Turing’s note to Rolf Noskwith on unit equilateral triangles in *n* dimensions. *King’s Treasures*, King’s College, Cambridge, April. https://kcctreasures.files.wordpress.com/2023/04/ttranscript.pdf

McKay, I. (2023). Recreational maths: Handwritten Alan Turing document sold at Bonhams. *Antiques Trade Gazette*, 2592, 13 May. https://www.antiquestradegazette.com/print-edition/2023/may/2592/auction-reports/recreational-maths-handwritten-alan-turing-document-sold-at-bonhams/

Turing, A. M. (2023). Untitled manuscript, AMT/D/19. *The Turing Digital Archive*. Archive Centre, King’s College, Cambridge. https://turingarchive.kings.cam.ac.uk/correspondence-amtd/amt-d-19

Challenges of Autonomy for Formal Methods

Matt Luckcuck
University of Derby

Marie Farrell
University of Manchester

Mario Gleirscher
University of Bremen

Maike Schwammberger
Karlsruhe Institute of Technology

July 2023

Abstract: We discuss the challenges that autonomy presents for formal verification and our peer-reviewed workshop that focusses on tackling them.

Autonomy is proving to be a useful trait for computer systems to have. Autonomy, or the ability for a system to make decisions without human intervention, enables the system to react flexibly to its environment. This is helpful for tasks that are dull, dirty, dangerous, or distant. Applications of autonomy are much wider than the example of driverless cars, including supporting civil nuclear decommissioning [1] and grasping space waste [2]. A system's autonomous decision-making component(s) can support a human user's decisions, without being linked to any physical actuation, in sectors like healthcare, finance, and administration. However, autonomy is often embedded within a robotic system, enabling the whole system to navigate and interact with its environment on its own. In either case, autonomous systems need careful verification to ensure that their behaviour is safe, secure, and free of other ethical issues.

There are broadly two types of Artificial Intelligence (AI) used to build autonomous systems: symbolic and sub-symbolic. Symbolic approaches include techniques like logic programming (e.g. with Prolog) and agent programming (e.g. with Jason), where statements and rules are used to describe the world and the actions that the system can decide to take. Because symbolic approaches are explicit and based on mathematical logic, they are easier to tackle with formal methods in ways similar to the formal verification of other software-based systems. Sub-symbolic (or connectionist) approaches to AI include machine learning and neural networks, where conclusions are inferred from large amounts of existing data to build knowledge about the world, often using statistical methods. Sub-symbolic approaches can be useful where we don't really understand the criteria for a decision or the criteria aren't very easy to describe logically. However, sub-symbolic approaches are much less amenable to formal verification because their decision-making mechanism is represented in a comparatively opaque way.

Given this comparison, the available routes to formally verifying an autonomous system will depend on what kind of AI approach(es) it uses. It seems likely that an autonomous system will need both approaches, for example using a symbolic approach for high-level decisions like route planning and using a sub-symbolic approach to classify sensor data. There are academic approaches that are already investigating combinations of symbolic and sub-symbolic AI [3], [4], [5]. It is clear that to deal with modern autonomous systems, formal methods will have to be geared to tackle both types of AI, be able to specify modular systems, and be understandable by people without a background in formal methods. This last point is essential for autonomous systems used in regulated environments, where they must be assessed or certified before they can be used.

Unique Verification Challenges

Autonomous Systems present several challenges to formal modelling and verification, after a comprehensive survey of academic publications [6] we categorised these challenges as being either external or internal to the autonomous system under consideration.

External Challenges

External challenges are problems *independent* of the way that the system is designed and implemented.

Modelling the Physical Environment of the system is one of the most prominent challenges when applying formal methods to autonomous systems that will interact with the real world. At design-time, assumptions are often made about the environment that the autonomous robotic system will work in, but these often do not survive contact with the real world. The system's environment is usually unstructured, often 'noisy' (or messy), and a changing environment can easily invalidate the system's internal model of the world. This is further complicated by problems like inaccurate sensors, changes in actuator performance, and components that are degraded or damaged.

We use the phrase *reality gap* to refer to this broad set of problems caused by the real-world conditions being different to the design-time environmental assumptions. The phrase is borrowed from other model-based engineering approaches, but applies equally well here in the world of formal models.

Adequate Abstractions for Formal Verification are needed in addition to adequate environment models. The abstractions that our formal methods provide should be adequate, in the sense that we can transfer our observations or conclusions from our models into reality. While environment models may be internalised in the autonomous system, the models are also part of the abstractions used for verification. So, abstractions are specifications that include both the model of the system and its environment, as well as a notion of the requirements to be verified. As such, abstractions and their representations (as models, programs, or properties) serve as the key inputs to a (formal) verification effort. Also here, the reality gap needs to be kept as small as possible (e.g. by model validation) to be able to transfer verification results back to reality but without losing the ability to apply a certain verification technique successfully.

Evidence for Regulators is crucial for autonomous systems that will be used in regulated sectors such as the nuclear and aerospace industries. As we mentioned earlier, it's essential that formal methods in this area keep pace with the regulatory environment of the sorts of systems that we want to be able to model and verify. This means two things: 1) formal methods should be capable of producing evidence in ways that are understandable (or at the very least *explainable*) to people with a technical, but not a formal methods, background; 2) we should strive to raise the profile of formal methods with regulators so that they understand what modern formal methods are capable of. This second point is important, because while formal methods may not be new to some regulators, their understanding of them may be stuck in a more old-fashioned view of what our approaches and community can do.

Internal Challenges

Conversely, the internal challenges posed by autonomy *depend on* how the system is built.

Symbolic Autonomy covers approaches like logic programming and agent-based approaches. As already mentioned, these AI approaches are more explicit and so are easier to model and verify. However, they still present challenges to formal methods when specifying and verifying the mechanism that chooses the rule or plan to execute. In general, enabling formal verification of agent programs remains an open problem. Though for a specific family of languages, the MCAPL framework enables program model checking for agent programs.

Sub-Symbolic Autonomy refers to machine learning and neural networks. As we outlined earlier, these approaches make decisions in a more opaque way and this means that they are more difficult to model and verify. As the editorial in BCS FACS FACTS issue 2023-1 says, decision-making components built on machine learning "do not possess a readily analysable schema of behaviour from which its behaviour can be assessed and assurance gained through an explicit analysis". Formal verification for these sorts of systems often leans towards runtime verification methods, essentially guarding or governing the acceptable outputs of the system (we can compare this with the idea of centrifugal governors for steam engines [7]). These approaches work – assuming that they have a way to prevent incorrect or unsafe decisions being acted on by the system – but it might not suffice or be effective to introduce a decision-making component that needs another component running constantly just to ensure that the system makes safe decisions.

Multi-Entity Systems are composed of swarms of multiple identical robots or teams of heterogeneous robots. More broadly, they are *Systems of Systems*. Multi-Entity Systems present challenges for coordination, usually solved by simple algorithms out of which group behaviour emerges. But this emergence, and the sheer number of communicating agents, can often cause problems for formal methods tools. Model checking can be severely limited if state space explosion cannot be properly dealt with. Often, methods like theorem proving or runtime verification are needed for this sort of system.

Self-Adaptive and Reconfigurable Systems are often used to enable the capability to adapt to the sort of environmental *reality gap* problems that we've already discussed. However, this can also provide another route for the system to choose incorrect or unsafe behaviour, and reconfiguring or adapting itself must be implemented with care. For formal approaches tackling adaptation and reconfiguration, the challenge lies in how configurations can be specified, analysed, and compared.

Tackling the Unique Challenges with Formal Methods

In previous work [8] we argue that autonomous systems need the rigour of formal methods to ensure that they are safe, secure, and free from other ethical harms; and that the formal methods community should seize the opportunity provided by autonomous systems to demonstrate the power of modern formal methods.

This argument, and the observations in our survey paper [6], prompted us to run the first workshop on Formal Methods for Autonomous Systems https://fmasworkshop.github.io, which was held at the Third World Congress on Formal Methods in 2019. From the first edition, the workshop has focused on presenting research that uses formal methods to tackle these unique challenges of autonomous systems.

The Formal Methods for Autonomous Systems Workshop

We founded the Formal Methods for Autonomous Systems (FMAS) workshop to provide a peer-reviewed venue for academic research that focusses on applying formal methods to the unique verification challenges posed by autonomous systems. The event is now in its fifth year and has built a friendly community of researchers who are working in this fast-paced subdomain. FMAS provides a venue to present publications, discuss the key challenges of autonomous systems, and stimulate collaboration between researchers. The papers published in previous editions of FMAS are listed on DBLP: https://dblp.dagstuhl.de/db/conf/fmas/index.html.

FMAS 2023 will be co-located with the international conference on integrated Formal Methods (iFM) on the 15th and 16th of November 2023 in Leiden, the Netherlands. All the details for FMAS 2023 can be found at:

https://fmasworkshop.github.io/FMAS2023/.

We accept papers in four categories:

- **Vision papers** *6 pages (excluding references)* describe directions for research into Formal Methods for Autonomous Systems;
- **Research previews** *6 pages (excluding references)* describe well-defined research ideas that are in their early stages, and may not be fully developed yet. Work from PhD students is particularly welcome;
- **Experience report papers** *15 pages (excluding references)* report on practical experiences in applying Formal Methods to Autonomous Systems, focusing on the experience and lessons to be learnt;
- **Regular papers** *15 pages (excluding references)* describe completed applications of Formal Methods to an Autonomous System, new or improved approaches, evaluations of existing approaches, and so on.

Important Dates

- Submission: 17th of August 2023 (Anywhere on Earth https://www.timeanddate.com/time/zones/aoe)
- Notification: 15th of September 2023
- Final Version due: 20th of October 2023
- Workshop: 15th and 16th of November 2023

In addition to the proceedings from this year's workshop, we are applying for a special issue with ***Science of Computer Programming*** (https://www.sciencedirect.com/journal/science-of-computer-programming) that will collect invited papers from the first five years of FMAS workshops and solicit new work through an open call for papers.

References

[1] Robotics and AI in Nuclear Hub, https://rainhub.org.uk/

[2] Farrell M, Mavrakis N, Dixon C, and Gao Y. Formal verification of an autonomous grasping algorithm. In International symposium on artificial intelligence, robotics and automation in space. European Space Agency. 2020. https://www.hou.usra.edu/meetings/isairas2020fullpapers/pdf/5040.pdf.

[3] Anderson M and Anderson SL. GenEth: a general ethical dilemma analyzer. Paladyn, Journal of Behavioral Robotics 2018; 9(1): 337–357. https://doi.org/10.1515/pjbr-2018-0024.

[4] Aitken JM, Veres SM, Shaukat A et al. Autonomous nuclear waste management. IEEE Intelligent Systems 2018; 33(6): 47–55.https://doi.org/10.1109/mis.2018.111144814.

[5] Shukairi H A, and Cardoso R C. ML-MAS: A Hybrid AI Framework for Self-Driving Vehicles. In Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '23). International Foundation for Autonomous Agents and Multiagent Systems. 2023. https://dl.acm.org/doi/10.5555/3545946.3598762

[6] Luckcuck M, Farrell M, Dennis L A, Dixon C, and Fisher M. Formal Specification and Verification of Autonomous Robotic Systems: A Survey. ACM Computing Surveys 2019; 52(5): 1–41. http://dl.acm.org/citation.cfm?doid=3362097.3342355.

[7] Maxwell JC. I. On governors. Proceedings of the Royal Society of London 1868; 16: 270–283. https://royalsocietypublishing.org/doi/10.1098/rspl.1867.0055.

[8] Farrell M, Luckcuck M and Fisher M. Robotics and Integrated Formal Methods: Necessity Meets Opportunity. In Furia C and Winter K (eds.) Integr. Form. Methods, LNCS, volume 11023. Heidelberg, Germany: Springer. https://doi.org/10.1007/978-3-319-98938-9_10

A Brief History of Hyperoperations

Marking a Centenary

Tim Denvir

July, 2023

Introduction

Leopold Kronecker stated (c1893) that God made the integers, that there is something "natural" about integers and natural numbers. About ten years ago I wanted to question this by constructing an impossibly large number, one which was larger than anything in the physical universe. Without realising it I reinvented hyperoperations – I had not heard of them at that point. In fact the concept has been with us for a century, although the term was coined somewhat later.

So in this present article I first enlarge on what I perceive was Kronecker's idea, revisiting briefly those earlier thoughts of mine, and then summarise the references to hyperoperations by four noted researchers over the last hundred years, namely Wilhelm Ackermann (1924), Reuben Goodstein (1947), Donald Knuth (1976) and Ronald Graham (c1977). I conclude with thoughts on the relevance of all this to formal semantics and some pointers to possible future work.

Kronecker's dictum

Leopold Kronecker (1823-1891) was a German mathematician who specialised in number theory and algebra. His birthplace, Liegnitz or Legnica, is now in Poland. Kronecker was a contemporary of Dirichlet, who was his PhD supervisor, Weierstrass, Riemann and Gauss. These days his name is probably most associated with the Kronecker Delta and the Kronecker Product.

In 1893 Heinrich Weber quoted Kronecker as having said "God made the whole numbers, everything else is of human construction" (my translation). Why did Kronecker think that the real numbers and others are a human construction? I suspect that he may have thought something along the following lines.

We have invented Real numbers in order to measure objects that we find in the world: their weight, length, width, magnetic properties etc. Objects do not in general weigh an exact whole number of grams or measure a whole number of metres. However, being able to describe a mass or a length down to an indefinite precision is an overkill. If we expressed the mass of an object to 30 decimal places in grams, the final few digits would be a number of tenths, hundredths, and thousandths of the mass of an electron, meaningless because an electron is an elementary particle. Likewise, if we express the length of an object to more than, say, 30 decimal places in metres, we are way below the size of the smallest atom, and near the Planck length, which is the smallest scale at which quantum effects can become apparent. It makes no sense to characterise the position or the boundary of an object to such precision.

Furthermore, Real numbers of course include non-algebraic numbers, non-computable numbers and numbers that cannot be defined at all.

Allowing indefinite precision in numbers is a mathematically neat and coherent way to cater for all needs for measurement of the physical properties of objects, but it is an idealisation; the total gamut of those numbers do not reflect the real world. They are "unnatural".

Thinking about the quotation from Kronecker led me ten years ago to write a small article for FACS on Very Large Numbers[1]. I wanted to show that the integers, indeed, the Natural Numbers, are also not natural at all and so too are a human invention. Along the way I reinvented hyperoperations without realising it, for I had not heard of them ten years ago. The Appendix gives an edited extract from that earlier article.

Hyperoperations

The first operation on numbers we learn about is Peano's successor, which primary school children are taught to call "counting on". Succ(0) is 1, Succ(5) is 6, etc. If we apply Succ repetitively we get addition: $Succ^4(5)$ is 5+4. If we apply addition repetitively we get multiplication: 5+5+5+5=5×4. If we apply multiplication repetitively we get raising to a power or exponentiation: $5\times5\times5\times5=5^4$ or 5^4. There conventionally we stop. But there is no necessity to do so. We could define a further operation by applying exponentiation repetitively thus: 5^5^5^5 for which we need a notation, perhaps 5^^4. We could keep on indefinitely, and define a general operation f_n. Defining such a general operation mathematically is easy enough (see the Appendix for an example).

Such operations are called hyperoperations. The earliest reference to hyperoperations I have found is implicit in Ackermann's function, dating from 1924, although that was not Ackermann's primary aim. Subsequent revisits to hyperoperations came with R. L. Goodstein in 1947, Donald Knuth's arrows in 1976, and Graham's number in, probably, 1977.

Ackermann's Function

As far as I can tell, the first to devise hyperoperations was the mathematician Wilhelm Ackermann (1896-1962) in 1928, although he seems to have been thinking about it earlier, in about 1924. Ackermann was a German mathematician who was primarily noted for his work on mathematical logic. He produced a joint work with David Hilbert, *Principles of Mathematical Logic*[2]. However, if Ackermann "invented" hyperoperations,

---

[1] *Rambling Thoughts on VLNI*, in FACS FACTS issue 12-2, page 8 https://www.bcs.org/media/3088/facs-nov12.pdf

[2] 1950 (1928). (with David Hilbert) *Principles of Mathematical Logic*. Chelsea. Translation from the 1938 German edition.

he did so only in passing, in his studies into computability. Ackermann's objective was to show that there are computable functions that are not primitive recursive.

We are all familiar with recursive functions. A, perhaps overworked, example is factorial. It is easily defined recursively as follows:

```
fact(n) = if n = 0 then 1 else n × fact(n - 1)
```

A function in a computer language which supports recursion, can be programmed and will look almost the same as that definition of a mathematical function. But the factorial function can be programmed without recursion, using a **for** loop or similar, thus (using an obvious, I hope, pseudocode):

```
result := 1;
for k = 1 step 1 until n do
        result := k × result;
end
fact(n) := result;
```

In general, a computable function is primitive recursive if it can be computed using **for** loops. Primitive recursive functions are a subset of general recursive functions[3]. Computable functions which are not primitive recursive can be difficult to dream up, but Ackermann's function was designed to be an example of just that. Ackermann's original formulation involved three arguments, but later simpler formulations have been devised, all frequently known as Ackermann's function. One of the most common versions is the two-argument Ackermann-Péter function, defined as follows:

$$A(0, n) = n + 1$$
$$A(m + 1, 0) = A(m, 1)$$
$$A(m + 1, n + 1) = A(m, A(m + 1, n))$$

The first A(0, n) is clearly Peano's successor function, and each subsequent A(1, n) etc. involves a repetitive application of the previous one. Thus, each involves a hyperoperation in the sequence, successor, addition, multiplication, exponentiation and onwards. This might be clearer with the formulation of Ackermann's function as a sequence of *unary* functions, each defined in terms of its predecessor; here, the superscript denotes iterative application of the function:

$$A_0(n) = n + 1$$
$$A_{m+1}(n) = A_m^{n+1}(1)$$

[3] A helpful article on primitive recursive functions can be found in Wikipedia at https://en.wikipedia.org/wiki/Primitive_recursive_function

A_0 is once again the successor function and each A_{m+1} is a repetitive application of A_m. With this formulation, the relation between Ackermann's function and hyperoperations is more direct. However, again I should emphasise, Ackermann's purpose was to find a computable function that was not primitive recursive. He was exploring computability in the 1920s; I continually find it remarkable that people were researching computability long before any electronic computers were on the scene.

In 1926 David Hilbert conjectured that Ackermann's function was not primitive recursive, but Ackermann himself proved it in 1928. Wikipedia has a good illuminating page on Ackermann's function[4]. Also, Appendix A of John Barnes' book, *Nice Numbers*, is a very accessible account of Ackermann[5].

Goodstein

R. L. Goodstein (1912-1985) was a British mathematical logician with a strong interest in the philosophy of mathematics. He graduated from Magdalene College Cambridge, held posts at the Universities of Reading and Leicester, and gained his PhD from Birkbeck College, London. Goodstein published a paper in the *Journal of Symbolic Logic* in 1947, *Transfinite Ordinals in Recursive Number Theory*[6]. This paper focuses mainly on recursive functions. In the last two of its eight pages he points out, perhaps as an afterthought to his Theorem 11 in the paper, a function that defines hyperoperations, without using that term: it had probably not yet been coined then. He shows that his function $G(k,a,n)$ defines addition, multiplication and exponentiation for $k = 1, 2, 3$ respectively. He then points out that for $k \geq 4$, $G(k,a,n)$ defines "successive new processes, which we may call tetration, pentation, hexation, and so on". He continues and cites Ackermann's 1928 paper. For more biographical information on Goodstein, see his informative Wikipedia page[7].

Knuth's Arrows

Donald Knuth, 1938- , was very active during the earlier years of computer science[8]. In those early days he concentrated on the analysis of algorithms, and on his continuing

---

[4] https://en.wikipedia.org/wiki/Ackermann_function.

[5] John Barnes, *Nice Numbers*, Birkhäuser, 2016, ISBN 978-3-319-46830-3, which I reviewed in the FACS newsletter 2022-2.

[6] R. L. Goodstein (Dec 1947). "Transfinite Ordinals in Recursive Number Theory". Journal of Symbolic Logic. 12 (4): 123–129.

[7] https://en.wikipedia.org/wiki/Reuben_Goodstein

[8] https://en.wikipedia.org/wiki/Donald_Knuth

multi-volume series, *The Art of Computer Programming*, (TAoCP), with volume 1 published in 1968 and subsequent volumes produced in various editions over the years, said to be still incomplete. In the late sixties and early seventies, TAoCP was a highly useful reference book for programmers and software engineers, especially for those engaged in systems programming. Knuth's expertise was wide, ranging over compiler design, cryptography, numerical analysis, linear programming and much more. His original education gave him a solid foundation, with a first degree in physics and subsequent Master's and PhD in mathematics. I counted 22 awards and prizes from his Wikipedia page, including the ACM Turing Award in 1974. Donald Knuth, despite his age of 85 at the time of writing, is still academically active, as can be seen from his "Computer Musings", in his personal webpage at Stanford[9].

Bearing FACS readers in mind, I can't resist quoting one sentence from his Wikipedia page: He once warned a correspondent, "Beware of bugs in the above code; I have only proved it correct, not tried it."

Knuth devised his "up-arrow" notation in 1974, apparently for the specific purpose of defining very large integers. I have taken much of what follows from the Wikipedia page on Knuth's up-arrows, which I gratefully acknowledge[10]. A single up-arrow ↑ represents exponentiation, a double up-arrow represents iterated exponentiation, ↑↑, three ↑↑↑ represent iterated ↑↑, etc. and $\uparrow^n$ represents *n* arrows[11]. Some examples might help.

$2 \uparrow 4 = 2 \times (2 \times (2 \times 2)) = 2^4 = 16$

$2 \uparrow\uparrow 4 = 2 \uparrow (2 \uparrow (2 \uparrow 2)) = 2^{16} = 65{,}536$

$2 \uparrow\uparrow\uparrow 4 = 2 \uparrow\uparrow (2 \uparrow\uparrow (2 \uparrow\uparrow 2)) = 2 \uparrow\uparrow (2 \uparrow\uparrow (2 \uparrow 2)) = 2 \uparrow\uparrow (2 \uparrow\uparrow 4)$

$= 2 \uparrow (2 \uparrow (2 \uparrow \ldots)$ **Note:** $2 \uparrow\uparrow 4$ or 65,536 copies of 2

$= 2\verb|^|2\verb|^|2\verb|^|\ldots\verb|^|2$ **Note:** 65,536 2's

It is worth observing that Knuth's arrows associate to the right; that while addition and multiplication are commutative and associative, exponentiation is neither:
$a \verb|^| (b \verb|^| c) \neq (a \verb|^| b) \verb|^| c$

Knuth's work on this specific topic appears to be based on, or at least related to, the much earlier work by R. L. Goodstein in 1947 as already mentioned.

[9] https://cs.stanford.edu/~knuth/

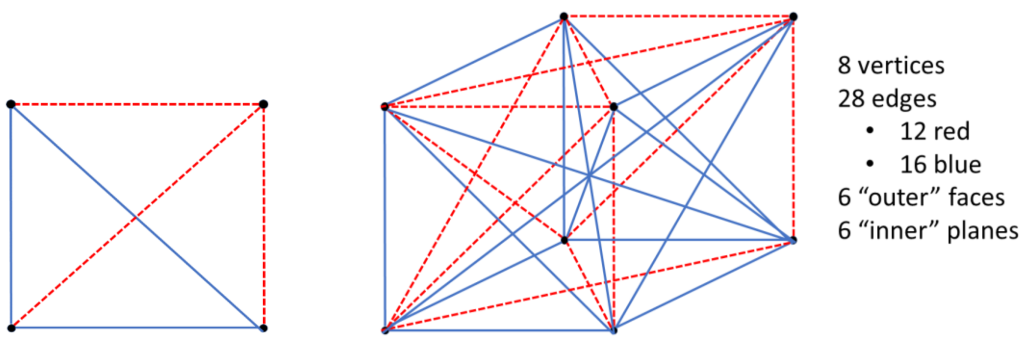
[10] https://en.wikipedia.org/wiki/Knuth%27s_up-arrow_notation

[11] Knuth, Donald E. (1976). "Mathematics and Computer Science: Coping with Finiteness". Science. 194 (4271): 1235–1242.

Graham

Ronald Graham (1935 – 2020) was an American mathematician of considerable note[12], credited with influencing the rapid development of discrete mathematics in the recent past. He was president of both the American Mathematical Society and the Mathematical Association of America. He was awarded the Leroy P. Steele Prize for lifetime achievement and elected to the U.S. National Academy of Sciences. He did collaborative work with Paul Erdõs[13] and many other mathematicians, including Donald Knuth[14]. Ronald Graham spent the majority of his professional academic life at Bell Labs and the University of California at San Diego. He supported himself while a graduate student by trampolining in a circus and later was president of the International Jugglers' Association!

Like Wilhelm Ackermann, Ronald Graham utilised hyperoperations in pursuit of another problem, in about 1977. Suppose you have an n-dimensional cube. So a 2-dimensional cube is a square, with four, or 2^2, vertices. In addition to the usual four edges, we draw the diagonals, in other words we connect all the vertices to each other with edges. There are therefore (4×3)/2 edges. We colour them with one of two colours, say red or blue. The **rule** is not to colour all the edges that lie in any one plane with the same colour. This is trivially easy with a square or two-dimensional cube. It is also easy with a familiar three-dimensional cube: there are 8 vertices, therefore (8×7)/2 = 28 edges.



For a cube of n dimensions, there are 2^n vertices and therefore $(2^n \times (2^n-1))/2 = 2^{n-1} \times (2^n-1)$ edges, where "edges" include lines joining any pair of vertices. It has been proved that the **rule** can be followed for dimensions up to n=12. For n=13, it is uncertain and has not yet been proved one way or the other. Note that a simplistic trial of all colourings is impractical even using a computer for more than a few n, because the number of colourings is $2\uparrow(2^{n-1} \times (2^n-1))$. Ronald Graham has proved that the **rule**

[12] See https://en.wikipedia.org/wiki/Ronald_Graham

[13] https://en.wikipedia.org/wiki/Paul_Erdõs

[14] See Concrete Mathematics - Wikipedia

cannot be followed for $n = \mathbf{G}$, where **G** is Graham's number. Graham's number is huge and can be defined using hyperoperations once again, as follows:

Using Knuth's up-arrow notation (see earlier section of this article),

$$\mathbf{G} = \mathtt{f}^{64}(4), \text{ where } \mathtt{f}(n) = 3 \uparrow^n 3$$

So for example, $\mathtt{f}^2(4) = \mathtt{f}(\mathtt{f}(4)) = 3 \uparrow^{\mathtt{f}(4)} 3 = 3 \uparrow^{3\uparrow\uparrow\uparrow\uparrow 3} 3$ where exponentiation in the last expression is taken to associate to the right, in the same way as Knuth's arrows. So $3^{3^3} = 3^{27} = 7,625,597,484,987$ and $3^{3^{3^3}} = 3^{3^{27}} = 3^{7,625,597,484,987}$, which is too large to expand with a calculator; however, it is the number of Knuth's arrows involved in the expansion of $\mathtt{f}^2(4)$. Graham's number is $\mathtt{f}^{64}(4)$, far larger. In 1977 it was considered to hold the record for the largest number involved in a serious mathematical proof: A good explanation of Graham's number with many more details can be found on its Wikipedia page[15].

There are a couple of delightful YouTube videos interviewing Ronald Graham on this topic[16].

Conclusion

This article has briefly surveyed a century of hyperoperations, large numbers, and the quests of some of their principal explorers. Numbers of various kinds are used in the formal modelling of computation. How well do numbers, particularly integers, fill this rôle?

Formal semantics should be seen, not just as an essential of formal development of software, but also, perhaps primarily, as a philosophical enterprise. Philosophy is the endeavour to understand what we mean and what we are talking about when we express ourselves in speech or writing on various topics – truth, nature, reality, "life, the universe and everything". So formal semantics is the attempt to clarify what we mean when we write software, programs and other scripts in the arena of computing.

I think it is better to see this philosophical enterprise as the context for formal models, not merely the more pragmatic aim of proof of correctness. It is a more general context, and helps our understanding in ways that may yet be unseen.

The judgement of whether a particular mathematical discipline is appropriate, sufficient for the needs, insufficient, or overkill, should be made in the light of our endeavour to understand all aspects of the computational artefacts that we construct.

---

[15] [Graham's number - Wikipedia]

[16] See "[What is Graham's number?]" and "[How big is Graham's number?]"

Future Work: Universal Hyper-algebras

Algebraic specifications are based on the principles of universal algebras. Universal Algebra generalises specific algebras as having a set of carrier sets and a set of operations, which are functions whose domains and range are selected from the carrier sets, together with a set of axioms. The data types and functions/procedures in a program fit nicely into this scheme.

Perhaps the canonical exposition of this view of formal specifications was originated by Hartmut Ehrig and Bernd Mahr in their two-volume work, *Fundamentals of Algebraic Specification*, in particular the first volume[17]. In the Introduction – *Historical Remarks*, the authors refer to work in the 1960s by Parnas, Hoare, Liskov, Zillies, Burstall and Goguen, and the ADJ Group among many others. They also reference the more purely mathematical work by Paul Cohn, *Universal Algebra*[18], which I believe may well have been a primary stimulus for this whole approach to formal specification. If the axioms are in the form of equations, the task of programming automated theorem provers is much less of a hard problem, but still by no means easy.

In Ehrig and Mahr's terminology, carrier sets are called "sorts" and functions are called "operations". A **signature** SIG = (S, OP) consists of a set S of **sorts** and a set OP of **constant and operation symbols**, and is the union of pairwise disjoint subsets: K_s, the set of **constant symbols of sorts $s \in$ S**, and $OP_{w,s}$, the set of **operation symbols with argument sorts $w \in S^+$** and **range sort $s \in$ S** (definitions quoted directly from Ehrig and Mahr). Variables, Terms and Equations are defined, and a specification SPEC = (S, OP, E) consisting of a signature SIG = (S, OP) and a set of equations E. Much more detail and definitions, of course, follow in that meticulous but clear work.

Work on algebraic specification has carried on continuously since Ehrig and Mahr's book. A more recent example is by Donald Sannella and Andrzej Tarlecki[19], which uses "elements of universal algebra, category theory and logic", and has attracted considerable praise.

The question that inevitably arises is that, with hyperoperations, we have a (countably) infinite set of operations, so are any extensions needed to the concept and treatment of algebraic specifications? Re-examining Ehrig and Mahr's work, I could find no stipulation that OP needs to be a finite set. But if hyperoperations were seriously to be incorporated

---

[17] H. Ehrig, B. Mahr, *Fundamentals of Algebraic Specification 1, Equations and Initial Semantics*, EATCS Monographs on Theoretical Computer Science, Volume 6, Springer-Verlag 1985.

[18] Paul M. Cohn, *Universal Algebra*, original edition: Harper & Rowe, 1965; revised edition: D. Reidel Publishing Company 1981.

[19] Donald Sannella and Andrzej Tarlecki *Foundations of Algebraic Specification and Formal Software Development*, EATCS Monographs in Theoretical Computer Science. Springer, 2012. ISBN 978-3-642-17335-6.

into the algebraic specification framework, it would be desirable to incorporate the generating equations that inductively define them into the framework. Would it be pretentious to call these extended algebras, "hyper-algebras"? More work is needed!

Appendix: Hyperoperations and Very Large Numbers

We think of integers as straightforward, a model of things one can count: the natural world is full of them: stars, planets, sheep, the pennies in one's bank balance. Indeed, the positive integers are called the "Natural Numbers". But even the integers include numbers one could never in practice write down or define, because we have only a finite quantity of paper (or other recordable medium, electronic, molecular etc.) in the universe. Even if one could write a symbol on every particle of matter in existence, there are only about 10^{43} of them. Assuming an alphabet of 256 characters, there is a finite upper limit of $256^{10^{43}}$ sequences of symbols, or sentences, with which we can express any thought whatsoever. Our expressible thoughts are bounded and finite! Yet we can construct some very large numbers indeed, much larger than $256^{10^{43}}$.

The general notation g^m denotes applying some given function g, m times: so $g^1(n) = g(n)$ and $g^2(n) = g(g^1(n))$. In general, $g^{1+m}(n) = g(g^m(n))$.

Let's define a sequence of operators, **F**, inductively

$$
\begin{array}{llll}
\mathsf{Succ}(n) & = 1 + n & & \textit{Successor} \\
\\
\mathsf{F}_1(m,n) & = (\lambda x \bullet \mathsf{Succ}(x))^m(n) & (= \mathsf{Succ}^m(n)) & \textit{Addition: } m+n \\
\mathsf{F}_2(m,n) & = (\lambda x \bullet \mathsf{F}_1(n,x))^m(0) & & \textit{Multiplication: } m \times n \\
\mathsf{F}_3(m,n) & = (\lambda x \bullet \mathsf{F}_2(n,x))^m(1) & & \textit{Exponentiation: } n^m \\
\mathsf{F}_{i+1}(m,n) & = (\lambda x \bullet \mathsf{F}_i(n,x))^m(n) & & \textit{General case}
\end{array}
$$

As noted, these F operators begin with the familiar integer arithmetic operators.

Let's now define a second sequence of operators, **G**, making use of **F**:

$$\mathsf{G}_i(n) = \mathsf{F}_i(n,n)$$

This clearly gives, for example: $\mathsf{G}_1(n) = \mathsf{F}_1(n,n) = n+n$, $\mathsf{G}_2(n) = \mathsf{F}_2(n,n) = n \times n$, $\mathsf{G}_3(n) = \mathsf{F}_3(n,n) = n^n$.

Now, let's take a biggish number like a million: denote 10^6 by **M**. Then $\mathbf{G_4(M)}$ is **M** repetitively raised to the power of **M** a million times over, a number so large that it severely taxes the imagination.

Finally, consider:

$$\mathbf{V} = \mathsf{G}_{\mathbf{M}}(\mathbf{M})$$

$\mathbf{G_M}$ is the millionth operator in the sequence and $\mathbf{G_M(M)}$ is $\mathbf{G_M}$ applied to one million. **V** ("Very Large Number Indeed") is a number of inconceivable magnitude, yet it is nonetheless an integer. If we represent the integers from 0 to **V** by a line segment, the vast majority of the integers represented by those points on the line will be impossible to define on any piece of universal "paper", just as the vast majority of numbers on the real line cannot be defined at all. The integers cease to look like some naturally occurring set, like a model of things one may in any way "count", for the vast majority of integers cannot in practice be defined.

This quest to name and define a very large integer uses the notion of hyperoperations almost perforce!

Jifeng He's 80th birthday Festschrift

Jonathan P. Bowen

July 2023

In September 2023, it is planned to celebrate the 80th birthday of Prof. Jifeng He, an important contributor to the field of formal methods. We provide a brief biography (Bowen & Zhu, 2023) followed by a description of the Festschrift volume that is to be published (Bowen et al. 2023). The associated symposium will be held in hybrid mode. We intend to provide FACS members the possibility of registering for online access to the symposium and also time-limited free access to the proceedings. Further information is available on the FACS website (http://facs.bcs.org).

Brief biography

Jifeng He was born in Shanghai China, in August 1943. He graduated in 1965 from the Department of Mathematics at Fudan University, located in Shanghai. Since 1965, he has held a position at East China Normal University (ECNU) in Shanghai, successively serving as a teaching assistant and then lecturer, and was promoted to full professor in 1986. In 1988, he was awarded the title of National Young and Middle-aged Experts with Outstanding Contributions. From 1980 to 1981, he was a visiting scholar at Stanford University and the University of San Francisco in California, United States. From 1983 to 1998, he worked as a senior researcher in the Programming Research Group (PRG) at the Oxford University Computing Laboratory (OUCL) in the United Kingdom, collaborating extensively with Tony Hoare, based in Oxford although retaining his position at ECNU. He was an important researcher on the European ESPRIT ProCoS project on "Provably Correct Systems" from 1989 and an essential collaborator with Tony Hoare on the *Unifying Theories of Programming* (UTP) approach (Hoare & He, 1998) that has spawned an important subfield of formal methods.

From 1998, Jifeng worked as a senior researcher at the International Institute of Software Technology of the United Nations University (UNU/IIST) in Macau. During 2002 to 2019, he was the Dean of the Software Engineering Institute at East China Normal University (ECNU). In 2002, he joined the first group of lifelong professors of ECNU. He was elected in 2005 as an academician of the Chinese Academy of Sciences, the highest scientific recognition in China. He received an honorary doctorate from the University of York in 2009. In December 2015, he was awarded the French National Palm Education Knight Medal. From 2017, Jifeng started to consider the issues around trustworthiness in Artificial Intelligence (AI). In 2019, he was appointed as the Distinguished Professor at Tongji University located in Shanghai. Jifeng's research interests have included sound methods for the specification of computer systems, communications, applications, and standards, as well as techniques for designing and implementing those specifications in software and/or hardware with high reliability.

Jifeng He speaking at a 2018 FACS event in London,
celebrating 20 years of UTP (Bowen, 2019)

Festschrift symposium

The 80th birthday Festschrift symposium for Prof. Jifeng He ("Jifeng@80") is to be held during 15–16 September 2023, at the Shanghai Science Hall in Shanghai, China (see below). This historic venue was built in the early 20th century within the French Concession area of Shanghai, previously as a school and for other purposes. In the 1950s, it became the Science Hall, and the facility has since been extended with newer buildings for scientific meetings and related activities.

View of the Shanghai Science Hall.
(Photograph by Jean-Pierre Dalbéra. Source: Wikimedia Commons.)

A Festschrift volume is due to appear in the Springer Lecture Notes in Computing Science series (LNCS 14080), containing papers relevant to formal methods by colleagues of Jifeng He, many of whom have been coauthors of academic papers with him (Bowen et al., 2023). This proceedings is a follow-on volume to that associated with Jifeng He's 70th birthday Festschrift symposium (Liu et al., 2013).

A public poster of Jifeng He in a Shanghai walkway during his 70th birthday Festschrift in 2013, celebrating his contributions to computer science (Bowen & Zhu, 2023).

Reviewing of papers was undertaken by a mix of authors and external international researchers. It is an indication of Jifeng's far-reaching reputation that reviewers are based in all the main continents of the world, with countries represented including Austria, Australia, Brazil, China, Denmark, France, Germany, India, Ireland, New Zealand, South Africa, the United Kingdom, and the United States.

Festschrift papers

In the initial section of the Festschrift volume, three papers present aspects of Jifeng He's contributions to computer science. The first paper provides a lifetime overview of Jifeng's research contributions, especially regarding formal methods. The following two papers provide more information with respect to developments in UTP (Unifying Theories of Programming) and rCOS (refinement calculus of object systems), two approaches in which Jifeng provided foundational underpinning. In the next two sections are papers by colleagues and coauthors with Jifeng while he was at the University of Oxford and also on the European ProCoS project on Provably Correct Systems during this time. The following sections include colleagues of Jifeng from China and Europe. The final section includes a paper related to Jifeng's recent roadmap for UTP in the future.

The following papers will be published in the Festschrift proceedings (Bowen et al., 2023) and presented during the symposium.

Jifeng He's research influence

Jifeng He at Oxford and Beyond: An Appreciation
Jonathan P. Bowen and Huibiao Zhu

UTP, Circus, and Isabelle
Jim Woodcock, Ana Cavalcanti, Simon Foster, and Augusto Sampaio

Linking Formal Methods in Software Development – A Reflection on the Development of rCOS
Zhiming Liu

Oxford colleagues

Consciousness by Degree
Yifeng Chen and J. W. Sanders

Specifying and Reasoning about Shared-Variable Concurrency
Ian J. Hayes, Cliff B. Jones, and Larissa A. Meinicke

The Consensus Machine: Formalising Consensus in the Presence of Malign Agents
A. W. Roscoe, Pedro Antonino, and Jonathan Lawrence

ProCoS colleagues

Domain Modelling: A Foundation for Software Development
Dines Bjørner

Concurrent Hyperproperties
Bernd Finkbeiner and Ernst-Rüdiger Olderog

Chinese colleagues

Characterizations of Parallel Real-Time Workloads
Xu Jiang, Jinghao Sun, and Wang Yi

Towards Efficient Data-flow Test Data Generation
Ting Su, Chengyu Zhang, Yichen Yan, Lingling Fan, Yang Liu, and Zhendong Su

European colleagues

Assume-Guarantee Reasoning for Additive Hybrid Behaviour
Pieter J. L. Cuijpers, Jonas Hansen, and Kim G. Larsen

Time: It is only Logical!
Frédéric Mallet

Applying Formal Verification to an Open-Source Real-Time Operating System
Andrew Butterfield and Frédéric Tuong

KnowLang -- A Formal Specification Model for Self-Adaptive Systems
Mike Hinchey and Emil Vassev

The Future Roadmap

A Coq Implementation of the Program Algebra in Jifeng He's New Roadmap for Linking Theories of Programming
Rundong Mu and Qin Li

Conclusion

The above papers will be presented during the planned hybrid Festschrift symposium, to be held 15–16 September 2023. Further information on the event and access to the proceedings will be provided on the FACS website (http://facs.bcs.org).

References

Bowen, J. P. (2019). FACS events, 2018–2020. *FACS FACTS*, 2019(1), 7–21, December. https://www.bcs.org/media/5204/facs-dec19.pdf

Bowen, J. P., Li, Q., and Xu, Q. (eds.) (2023). *Theories of Programming and Formal Methods: Essays Dedicated to Jifeng He on the Occasion of His 80th Birthday*. Springer, LNCS 14080.

Bowen, J. P. and Zhu, H. (2023). *Jifeng He at Oxford and Beyond: An Appreciation*. In Bowen et al. (2023).

Hoare, C. A. R. and He, J. (1998). *Unifying Theories of Programming*. Series in Computer Science, Prentice Hall.

Liu, Z., Woodcock, J., and Zhu, H. (eds.) (2013). *Theories of Programming and Formal Methods: Essays Dedicated to Jifeng He on the Occasion of His 70th Birthday*. Springer, LNCS 8051.

Formal Development of Cyber-Physical Systems: The Event-B Approach

Paulius Stankaitis, Newcastle University

Webinar presented: 04/04/2023

https://www.youtube.com/watch?v=i_Ha33q588Y

Reported by: Alvaro Miyazawa, University of York

Introduction

On 4 April 2023, FACS hosted a webinar on the formal development of cyber-physical systems presented by Dr Paulius Stankaitis, Research Associate at Newcastle University. The webinar had 52 registered attendees, and the abstract of the talk is below.

> **Abstract**: Cyber-physical systems (CPS) are complex computer-based systems which have closely intertwined physical processes, computation and networking system aspects. Some of their development and safety assurance difficulties arise from the need to model and reason at a system-level, and hybrid behaviours which are best captured by hybrid automata models. In this seminar I will present a formal development approach of cyber-physical systems using the Event-B formal method which is based upon refinement and proof-based verification. To improve the level of automation in the deductive verification of the resulting hybridised Event-B CPS models, the seminar will describe an approach of integrating reachability analysis in the proof process.

Summary

Paulius starts the talk by defining cyber-physical systems (CPS) as systems containing computer-based systems that control the physical processes of physical systems. Next, Safety-critical systems are identified as particularly important since failures can lead to casualties and financial loss.

Safety-critical railway signalling systems such as the European Train Control System (ETCS) and the Communication-based Train Control (CBTC) are discussed, and some goals are identified. In particular, the interesting goal of reducing the space between trains and, consequently, increasing the capacity of the system is mentioned.

Paulius covers the current application of formal methods in the railway domain, in particular, the verification of control tables and interlocking software and the use of the B method in the development of the Paris metro.

The talk proceeds to discuss a formal development framework for cyber-physical systems based on abstraction and refinement. The particular instance presented in this talk is based on Event-B, but other methods could be used.

The framework should enable a multifaceted and scalable CPS design, including verification, validation, reachability analysis, and animation. Automation is especially important to increase the scalability of the approach.

Paulius briefly describes the B and Event-B methods, with Event-B providing a high level of automation for the verification of discrete systems. Next, extensions of Event-B that support the modelling of cyber-physical systems are discussed.

The framework is developed based on the work of Dupont et al.; this extension, Hybridised Event-B, allows the definition of differential equations and includes a number of operators tailored to hybrid systems. A controller-plan-loop example is presented, and new proof obligations, such as continuous invariant preservation and continuous feasibility, are discussed.

An example of the development of the speed controller of a cyber-physical railway signalling system is presented. The first refinement develops an abstract Hybridised Event-B specification of the system based on the Davis resistance equation and specifies the properties the system should satisfy. A total of 55 proof obligations are derived from the first refinement, 36 of which are proved automatically, with 19 being proved using alternative means. These 19 proof obligations are, in general, related to the continuous aspects of the model.

Given the limitations of current proof techniques, alternative methods were investigated. One particular method is Reachability Analysis, which is used in a verification tactic that involves the translation of proof obligations to reachability problems. The reachability analysis is then performed using JuliaReach. This tactic allowed the automatic verification of 48 of the original 55 proof obligations.

The second refinement introduces sub-systems such as communication centres, interlocking, infrastructure and communication protocol. Since the hybrid aspects of the system are restricted to the first refinement, the level of automation in the verification of proof obligations of the second refinement is much higher, with 71 out of 85 proof obligations automatically discharged.

As the next steps, the automatic translation of hybridised Event-B models into JuliaReach needs to be improved, and new Event-B theories need to be developed.

About the speaker

Biography: Paulius Stankaitis is a Research Associate at Newcastle University where he is part of the Advanced Model-Based Engineering and Reasoning (AMBER) research group. He received his PhD from Newcastle University on the topic of formal modelling and verification of heterogeneous railway signalling systems. His current research focuses on integrating different formal verification methods for reasoning about safety of cyber-physical and autonomous systems.

The Independence Day of Witnessing the Correctness of Systems: From Topological Proofs and Beyond

Claudio Menghi, University of Bergamo

Webinar presented: 04/07/2023

https://www.youtube.com/watch?v=AjWdACIPS1o

Reported by: Alvaro Miyazawa, University of York

Introduction

On 4 July 2023, FACS hosted a webinar on Topological Proofs presented by Dr Claudio Menghi, Assistant Professor at the University of Bergamo and an Adjunct Professor at McMaster University. The webinar had 27 registered attendees and the abstract of the talk is below.

Abstract: Model checking provides developers with helpful information to improve their models when a property is not satisfied by a counterexample. However, engineers need helpful information also when a property is instead satisfied. This seminar will introduce Topological Proofs: slices of the model that witness the correctness of systems by justifying property satisfaction. It will present TOrPEDO, an approach that supports their use. It will discuss two complementary versions of this approach: the first based on unsatisfiable cores of LTL formulae and an enhanced and more efficient version that relies on a novel encoding of LTL formulae based on Bit-Vectors. The seminar will also report on the support provided by TOrPEDO for reasoning on incomplete and partial models, where property satisfaction can depend on unknown parts of the model, and discuss how Topological Proofs can be transferred and adapted to other settings, e.g., to generate diagnostic information for signal-based temporal properties. Finally, the seminar will reflect on the desire and the challenges to reach the Independence Day of witnessing the correctness of systems, when formalisms-independent frameworks can witness the correctness of different software artifacts.

Summary

The talk was structured around four of the author's publications: *From Model Checking to a Temporal Proof for Partial Models* [1], *Integrating Topological Proofs with Model Checking to Instrument Iterative Design* [2], *TORpEDO: Witnessing Model Correctness with Topological Proofs* [3], and *Trace Diagnostics for signal-based temporal properties* [4].

The complementary natures of model checking and theorem proving are given as the context for the work, where model checking provides counter-examples for negative results, and theorem proving provides proof for positive results. Claudio discusses the standard assumption that models are completely specified and gives an example of a light-traffic controller where this assumption does not hold (it contains an unspecified state).

Claudio raises the question of how model checking and theorem proving can be applied to partial models and provides a solution in the form of the THRIVE framework, which combines model checking and theorem proving to produce three possible classes of outcomes: definitive counter-examples, proofs, or proofs with possible counter-examples.

The THRIVE framework is based on Partial Kripke Structures and Linear Temporal Logic with three-valued semantics and uses pessimistic and optimistic approximations to evaluate properties of interest. Starting from the computed optimistic and pessimistic models, a two-valued model-checker is used to evaluate a property against both models, with a satisfied property being further investigated with a theorem prover. If a property holds for the pessimistic approximation, the result is **true**; if it is not satisfied by the optimistic approximation, the result is **false**; otherwise, the result is **maybe**.

Next, Claudio discusses issues related to the complexity and size of proofs and the impact of model changes. The notion of topological proofs as a portion of a model that satisfies a property is discussed, as well as their usefulness to engineers in guiding how models can be changed while preserving the properties of interest.

The TOrPEDO framework is introduced, incorporating the THRIVE framework with model revisions and automated re-verification. The framework is evaluated with respect to the size of proofs (compared to the size of models) and scalability using 60 combinations of models and requirements. The evaluation shows that topological proofs are approximately 60% smaller than the models, that in 78% of cases, the re-verification confirmed that revised models are compliant with the topological proof, and that computing topological proofs for existing tests take, on average, less than 10 seconds. In contrast, the computation for a larger model takes about 1 minute.

The evaluation of TORpEDO concludes that scalability is still an issue, and this is further investigated in the third paper, where TOrPEDO-SMT is proposed. In this work, Linear Temporal Logic (LTL) formulae are translated to propositional Logic, and the unsatisfiable cores are computed and converted back from propositional logic to LTL. TOrPEDO-SMT is evaluated against the original approach and shows promising results, being significantly faster and able to analyse larger models. Furthermore, a large benchmark model of gene regulatory networks is used to investigate the usefulness of the approach, producing evidence that topological proofs provide helpful information to designers.

Finally, following in the line of providing helpful information to designers, issues of traceability and explainability are addressed. Claudio discusses the analysis of execution traces and properties to provide explanations of violations. The approach is evaluated and shown to be effective in calculating diagnoses.

The talk concludes with a few reflections on the synergy between theory and practice, teamwork, the importance of reviewers, and the many open challenges, such as different modelling formalisms and requirement languages, trade-offs between expressiveness and performance, completeness, and usability.

About the speaker

Claudio Menghi received his BSc and MSc degrees in computer science from the Politecnico di Milano, where he later obtained his Ph.D. degree under the supervision of Prof. Carlo Ghezzi in 2015. He was a Postdoctoral Researcher at the University of Gothenburg and Chalmers (2017-2018), a Research Associate at the Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg (2018-2021), and an Assistant Professor at McMaster University (2021-2022). He is an Assistant Professor at the University of Bergamo and an Adjunct Professor at McMaster University (2023-now).

Claudio Menghi's research interests are formal methods and software engineering, focusing on cyber-physical systems, robotics, and formal verification. He has spent several years doing research with industry and applying formal methods and software engineering techniques in real-world and industrial contexts. He has led research projects with four industry partners: BOSCH and PAL Robotics in the robotics domain and LuxSpace and QRA Corp in the aerospace and cyber-physical domain.

References

[1] Bernasconi, A., Menghi, C., Spoletini, P., Zuck, L.D., Ghezzi, C. (2017). From Model Checking to a Temporal Proof for Partial Models. In: Cimatti, A., Sirjani, M. (eds) Software Engineering and Formal Methods. SEFM 2017. Lecture Notes in Computer Science(), vol 10469. Springer, Cham. https://doi.org/10.1007/978-3-319-66197-1_4

[2] Menghi, C., Rizzi, A.M., Bernasconi, A. (2020). Integrating Topological Proofs with Model Checking to Instrument Iterative Design. In: Wehrheim, H., Cabot, J. (eds) Fundamental Approaches to Software Engineering. FASE 2020. Lecture Notes in Computer Science(), vol 12076. Springer, Cham. https://doi.org/10.1007/978-3-030-45234-6_3

[3] Menghi, C., Rizzi, A. M., Bernasconi, A., and Spoletini, P. (2021). TOrPEDO: witnessing model correctness with topological proofs. Form. Asp. Comput. 33, 6 (Dec 2021), 1039–1066. https://doi.org/10.1007/s00165-021-00564-1

[4] Boufaied, C., Menghi, C., Bianculli, D., and Briand, L. C. (2023). Trace Diagnostics for Signal-Based Temporal Properties. In: IEEE Transactions on Software Engineering, vol. 49, no. 5, pp. 3131-3154, 1 May 2023, doi: 10.1109/TSE.2023.3242588

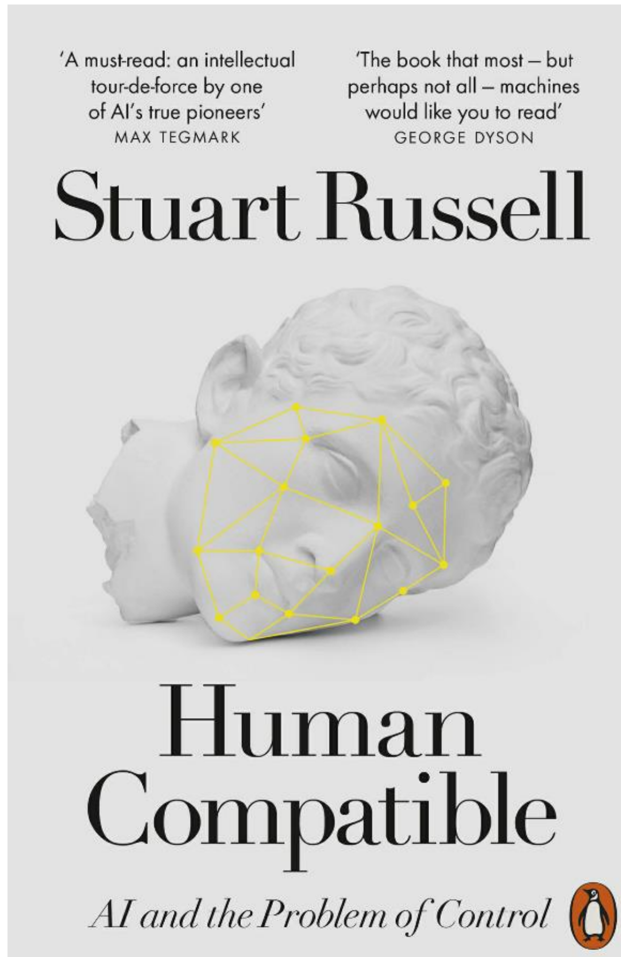
Human Compatible

AI and the Problem of Control

by Stuart Russell

Book review

Brian Monahan



"**Human Compatible: AI and the problem of control**" by Stuart J. Russell, Penguin, 2020, 336 pages, ISBN: 978-0-241-33524-6

The first few paragraphs of the preface go as follows:

> *"This book is about the past, present, and future of our attempt to understand and create intelligence. This matters, not because AI is rapidly becoming a pervasive aspect of the present but because it is the dominant technology of the future.*
>
> *...*
>
> *Everything civilization has to offer is the product of our intelligence; gaining access to considerably greater intelligence would be the biggest event in human history. The purpose of the book is to explain why it might be the last event in human history and how to make sure that it is not."*

In short, this is about Artificial Intelligence and how it affects *us* now and in the probable future – and by *us*, the author means *all* of us. In the jargon, this book is about *AI safety*[20]. taken here to mean the interdisciplinary field concerned with reducing the potential for AI to harm people, either accidentally or otherwise, whilst at the same time enabling AI technology to deliver potentially enormous benefits to humanity as a whole. This book's content has only become more relevant since its first publication in 2019 in the US, well before the arrival of ChatGPT and friends. However, in spite of it being

[20] The term is *AI safety*, rather than *AI ethics*, as it concerns making AI technology safe to use and interact with.

published over three years ago, the core issues tackled here remain unrelentingly relevant, standing somewhat presciently above various pronouncements made by several others since its publication.

The author is Stuart J. Russell OBE, a well-known, award-winning AI research professor based at UC Berkeley in the States, with well over 200 AI-related publications to his name. Russell coauthored the leading textbook in the area (updated 2021), "*Artificial Intelligence: A Modern Approach*" together with Peter Norvig, former director of research at Google. He currently leads the Centre for Human-Compatible Artificial Intelligence based at Berkeley, and gave the Reith Lectures for the BBC in 2021 and a well-received TEDx talk from 2017, all concerned with AI safety.

The work naturally splits into 3 parts, where the first 3 chapters are concerned with both the history and the nature of intelligence arising naturally in people and actively pursued in machines; chapters 4 to 6 give an exploration of a number of broad issues and problems arising from this pursuit and is perhaps the most specialised part of the book. Finally, the remaining 4 chapters are given over to explaining "a new way to think about AI and to ensure that machines remain beneficial to humans, forever." In order to avoid distracting technical clutter within the main text, four appendices are given that provide a slightly more technical summary account of ideas and techniques needed to underpin modern AI. Quite clearly, this work is certainly not short on ambition!

It is perhaps worth noting that the book contains precious little about Computer Science or even programming as such. For example, the first mention of "neural network" appears quite late on page 42, only appearing three times in the main text and then only in the first chapter![21] Instead, the hoped-for readership will be rather broad, ranging from the tech-savvy lay person, curious to get an inside peek into the world of AI and AI safety, all the way to people more focused upon the social implications of widely deployed AI systems, such as commentators, social scientists, policy makers and, of course, law makers and politicians. If one was looking for an introductory *technical* account of AI systems and their general operating principles, I wouldn't turn to this volume particularly (even with the appendices) – instead, Russell and Norvig's text book seems a far better starting point.

Given the portentous and sweepingly broad nature of the subject matter, it is fortunate, and probably essential, that Russell overall adopts a fairly breezy style of writing, seeking to avoid ponderousness — and largely achieving it. This style allows Russell to cover a lot of ground, but also gives him licence to vary the presentation as occasion demands to provide a deeper level of discussion. While this style opens itself to the criticism of patchiness, it seemed to me that this approach tackled important issues and topics in sufficient depth and adequacy, without overstaying one's welcome as it were. Inevitably, there will be particular topics that, in the opinion of some, could have been tackled in greater depth – but I didn't detect any really glaring omissions.

---

[21] The appendices contain marginally more technical descriptions of "neural network", "deep learning" and "reinforcement learning".

Chapters 3 and 4 provide an important contrast in that the first chapter provides a glowing summary of the wonderful benefits that the widespread use of AI technology could bestow upon humanity, whereas the second one provides a sober review of how AI could wind up harming us all significantly, including through mass unemployment, pervasive surveillance and enforcement of restrictions of basic civil liberties. This quite even-handed approach covers the ground in a way that deflects criticism of partisan bias, either for or against AI. This is important for any book broadly advocating AI safety– too much bias in either direction naturally compromises and sullies the overall message, making it harder for advocates on each side of the debate to at least listen and consider the ramifications.

Following the warm-up of chapter 4 concerning the potential for misuses of AI, chapter 5 addresses something that is of widespread concern, while being neither overtly positive nor overtly negative towards humanity: the emergence of *superintelligent* AI. Russell characterises this issue well in terms of what he calls "The Gorilla Problem". This goes as follows (page 132):

> *Around ten million years ago, the ancestors of the modern gorilla created (accidentally, to be sure) the genetic lineage leading to modern humans. How do the gorillas feel about this? Clearly, if they were able to tell us about their species' current situation vis-à-vis humans, the consensus opinion would be very negative indeed. Their species has essentially no future beyond that which we deign to allow.*

Russell is suggesting that humanity could perhaps be in a very similar position to the gorilla's vis-a-vis superintelligence, once that eventually emerges -- as it most surely will. According to Russell, the root underlying issue here is what he terms "value alignment". If we could *guarantee* that the actions of a superintelligence are necessarily entirely *aligned* and shared with humanity's own values then all would be well – benefits to humanity would undoubtedly then accrue. However, concerns arise when there is doubt about the availability of such a guarantee or even if it is feasible. This is a complex area -- the philosopher Nick Bostrom has devoted an entire book, "Superintelligence: Paths, Dangers, Strategies", to the discussion of this topic.

The final 4 chapters are concerned with motivating and explaining Russell's own preferred approach to dealing with this core issue – this is the notion of a "Provably Beneficial AI" which forms the subject of chapters 8 and 9, continuing to the end of the book in chapter 10. Admittedly, it was the word "provably" that initially caught my eye here. By provable, Russell does indeed mean mathematical provability, outlined in chapter 8 in the form of a game theoretical approach. Broadly the idea is to avoid providing an explicit set of detailed objectives (which the system could learn to evade and exploit), but instead to exploit *uncertainty* about the possibility of reward and failure. The idea broadly is then that such a system would *necessarily* need to constructively explore ways to succeed and be rewarded through providing benefit to people, and thus avoid failure.

To summarize briefly, this is an important work, which gives a foundational study of need for AI safety. More than that, it offers some plausible hope that, despite all the difficulties, the problem of AI safety may not be as intractable as is commonly feared.

Forthcoming Events

We have a new Seminar Organiser on the FACS committee, Alvaro Miyazawa at the University of York. If you have suggestions for future FACS seminar speakers or other events, especially if you are willing to help with co-organisation or even give a talk, please contact Alvaro on Alvaro.Miyazawa@york.ac.uk.

Events Venue (unless otherwise specified):

BCS, The Chartered Institute for IT
Ground Floor, 25 Copthall Avenue, London, EC2R 7BP

The nearest tube station is Moorgate, but Bank and Liverpool Street are within walking distance as well. The new Elizabeth Line is now very convenient for the BCS London office, by alighting at the Liverpool Street stop and leaving via the Moorgate exit.

15-16 September 2023	**Jifeng He's 80th birthday Festschrift Symposium** Eventbrite registration for *online* attendance at this Festschrift Symposium, to be held at the Shanghai Science Hall on 15-16 September 2023, celebrating Jifeng He's 80th birthday, is now available. See information and booking under: https://FACE1516092023.eventbrite.co.uk A webpage is also available on the FACS website linked from the http://facs.bcs.org main page. There are 50 online places available, so do book if you would like to attend, but please note the early start to fit in with the time in China!

Details of all forthcoming events can be found online here:

https://www.bcs.org/membership/member-communities/facs-formal-aspects-of-computing-science-group/

Please revisit this site for updates as and when further events are confirmed.

FACS Committee

bcs The Chartered Institute for IT

Formal Aspects of Computing Science Specialist Group

Jonathan Bowen
FACS Chair and
BCS Liaison

John Cooke
FACS Treasurer and
Publications

Roger Carsley
Minutes Secretary

Alvaro Miyazawa
FACS Seminar
Organiser

Tim Denvir
Co-Editor, FACS
FACTS

Brian Monahan
Co-Editor, FACS
FACTS

Margaret West
Inclusion Officer and
BCS Women Liaison

Andrei Popescu
LMS Liaison

Keith Lines
Government and
Standards Liaison

Ana Cavalcanti
FME Liaison

Brijesh Dongol
Refinement
Workshop Liaison

FACS is always interested to hear from its members and keen to recruit additional helpers. Presently we have vacancies for officers to help with fund raising, to liaise with other specialist groups such as the Requirements Engineering group and the European Association for Theoretical Computer Science (EATCS), and to maintain the FACS website. If you are able to help, please contact the FACS Chair, Professor Jonathan Bowen at the contact points below:

BCS-FACS
c/o Professor Jonathan Bowen (Chair)
London South Bank University
Email: jonathan.bowen@lsbu.ac.uk
Web: www.bcs-facs.org

You can also contact the other Committee members via this email address.

Mailing Lists

As well as the official BCS-FACS Specialist Group mailing list run by the BCS for FACS members, there are also two wider mailing lists on the Formal Aspects of Computer Science run by JISCmail.

The main list <facs@jiscmail.ac.uk> can be used for relevant messages by any subscribers. An archive of messages is accessible under:

http://www.jiscmail.ac.uk/lists/facs.html

including facilities for subscribing and unsubscribing.

The additional <facs-event@jiscmail.ac.uk> list is specifically for announcement of relevant events.

Similarly, an archive of announcements is accessible under:

http://www.jiscmail.ac.uk/lists/facs-events.html

including facilities for subscribing and unsubscribing.

BCS-FACS announcements are normally sent to these lists as appropriate, as well as the official BCS-FACS mailing list, to which BCS members can subscribe by officially joining FACS after logging onto the BCS website.