Andrew D. Gordon

Science Advisor, Cogna, Chancery House, 53-64 Chancery Lane, London WC2A 1QS
Honorary Professor, School of Informatics, University of Edinburgh, Edinburgh EH8 9LE
https://www.linkedin.com/in/andrew-d-gordon/

November 2025

Qualifications

BSc (Computer Science, First Class Honours) 1987, University of Edinburgh.

PhD (Computer Science) 1992, University of Cambridge.

Employment

Summer intern, Digital Systems Research Center, Palo Alto, summer 1989.

Postdoc researcher, Chalmers University, Gothenburg, Jan-Dec 1993.

Research associate, University of Cambridge, Jan 1994-Sep 1994.

Royal Society University Research Fellow, University of Cambridge, Oct 1994-Oct 1997.

Consultant, Digital Systems Research Center, Palo Alto, Aug 1996 and Mar 1997.

Researcher, Microsoft Research, Nov 1997-Aug 2002.

Visiting Professor at the University of Provence, Marseille, Apr 1998.

Senior Researcher, Microsoft Research, Sep 2002-Aug 2007.

Visiting Professor at the University of Newcastle, Mar 2007-Mar 2010.

Principal Researcher, Microsoft Research, Sep 2007-Dec 2012.

Professor of Computer Security, University of Edinburgh, Oct 2010-Jul 2023.

Principal Research Manager, Microsoft Research, Dec 2012-Dec 2017.

Senior Principal Research Manager, Microsoft Research, Jan 2018-Aug 2023.

Partner Research Manager, Microsoft Research, Sep 2023 – Nov 2023.

Chief Science Officer, Cogna, Nov 2023 – Feb 2025.

Science Advisor, Evara AI, since Mar 2024.

Science Advisor, Cogna, since Mar 2025.

Science Advisor, Advanced Research + Innovation Agency, since August 2025.

Awards and Honours

Distinguished Dissertation in Computer Science, Functional Programming and Input/Output [2], jointly awarded by the British Computer Society and the Conference of Professors and Heads of Computing, 1993.

Invited speaker at IEEE Logic in Computer Science Provable Implementations of Security Protocols (LICS'06), Seattle, August 2006.

Most Influential ETAPS 1998 Paper, Mobile Ambients [6], with L. Cardelli, awarded by the European Association for Programming Languages and Systems, 2007.

Member of UK Computing Research Committee, since June 2008.

Microsoft award for transfer of SecPAL: Design and semantics of a decentralized authorization language [10] technology to Microsoft Vine product, 2009.

Most Influential ACM POPL 2000 Paper, Anytime, Anywhere: Modal Logics for Mobile Ambients [7], with L. Cardelli, awarded by ACM SIGPLAN, 2010.

Best Paper ETAPS 2013, Deriving Probability Density Functions from Probabilistic Functional Programs [13], with S. Bhat, J. Borgström, and C. Russo, awarded by the European Association for Programming Languages and Systems, 2013.

Unifying Speaker on Structure and Interpretation of Probabilistic Programs (ETAPS, Eindhoven, April 2016). (The unifying speakers address a plenary session of all five constituent conferences of ETAPS.)

Calc.ts in Excel for the web inducted into Microsoft Wall of Fame, July 2020.

ACM Fellow 2020, “For contributions to programming languages: their principles, logic, usability, and trustworthiness.”

(ACM is the leading international association of computing professionals. ACM's most prestigious member grade recognizes the top 1% of ACM members for their outstanding accomplishments in computing and information technology and/or outstanding service to ACM and the larger computing community.)

Honorable mention for ACM CHI 2023 paper “What it wants me to say” [20].

Honorary Professor, University of Edinburgh, since August 2023.

2024 ETAPS Test of Time Award with Luca Cardelli for 1998 paper Mobile Ambients [6].

2024 Test of Time Award of the Symposium on Computer Security Foundations with Alan Jeffrey for 2001 paper Authenticity by Typing for Security Protocols [9].

Keynote Speaker on Requirements are all you need, (ICFP, Milan, September 2024).

Gordon has delivered over 40 invited talks at research workshops and conferences.

Leadership and Service in Academic Research Community

Gordon has chaired numerous technical program committees, including the Symposium on Principles of Programming Languages **(POPL 2017)** (one of the two top research conferences in programming languages) and twice chairing constituent conferences of **ETAPS** (European Joint Conferences on Theory and Practice of Software, the primary European forum for researchers working on Software Science): **FOSSACS 2003** and **ESOP 2010**. He co-chaired the Workshop on Formal Methods in Security Engineering (**FMSE 2005**) and the Workshop on Probabilistic Programming Languages, Semantics, and Systems (**PPS 2018**). He served on the Steering Committees of the International Conference on Functional Programming (**ICFP**) and the Symposium on Principles of Programming Languages (**POPL**).

Gordon is founder and organiser of many conferences on hot research topics, including the HOOTS series of workshops on Higher Order Operational Techniques in Semantics (Cambridge 1995, Stanford 1997, Paris 1999, and Montreal 2000), and many others (Algebraic Process Calculi 2005, R2D2 2008, CryptoForma 2009, RADICAL 2010, and the Dagstuhl Seminar on Probabilistic Programming 2013). He has graduated 7 PhD students and examined 26 PhDs.

Leadership within Microsoft Research

Gordon was one of the earliest hires at MSR Cambridge in 1997 and helped build the Research Area on **Programming Principles and Tools**. He led the group from 2012 until the abolition of research areas in 2018. As Research Area Leader, he hired and managed a set of world-class researchers. He steered the team in the direction of programming language research applied to machine learning, and vice versa. He served on the lab **Senior Leadership Team** for five years, with responsibility for PPT and for early-stage but highly ambitious Seedling Projects across the lab – he organised training activities on research innovation including delivering talks on Strategic Thinking for Researchers. Gordon convened the Microsoft Research University of Edinburgh Joint Initiative in Informatics (2011-2018), which funded more than a dozen PhD students.

Advisory and Editorial Boards

Gordon served on the Scientific Advisory Board of the Cluster of Excellence on "**Multimodal Computing and Interaction**", Saarbrucken from 2008 to 2016. He sat on the editorial board of the Springer book series on **Information Security and Cryptography** and of the journal **Logical Methods in Computer Science.** He was steering committee chair for **Languages for Inference** (LAFI), the workshop series on programming language research techniques applied in machine learning. He sat on the Executive Committee of the **UK Computing Research Committee** 2021 – 2024.

Resumé of Achievements

Five Lines of Research and their Impact on Best-in-Class Programming Languages

Gordon has sustained over his career a high level of contribution to programming languages and applications, with enduring influence across the computing industry.

1. PhD research on monadic I/O was adopted by Haskell, the most widely used lazy functional language.
2. Innovation of symbolic crypto in the spi calculus, which led to ProVerif, the most widely used tool for symbolic verification of crypto protocols.
3. Innovation of refinement types for verifying security-critical code in his F7 typechecker was adopted by F*, the most widely used language for verified cryptographic code.
4. Innovation of information-flow levels for probabilistic programs in his research language Tabular looks to be adopted by Stan, the most widely used probabilistic programming language.
5. Research on semantics and AI generation of spreadsheet formulas productized to accelerate web spreadsheets and autosuggest formulas with Excel, the most widely used programming language ever, used by hundreds of millions of users.

1 Monadic I/O – impact on Haskell

Gordon's prize-winning PhD work on input/output in functional languages contributed to the design of monadic I/O in Haskell [2,4], now widely used across industry and in education. Haskell owes its success to many factors, but it would have had little practical impact without the I/O mechanism that Gordon joined the committee to standardize, based on his PhD work, for Haskell version 1.3. (A measure of the importance of monads in Haskell is that Gordon's symbol ">>=" for monadic bind forms half the Haskell logo – the other half is Church's lambda symbol.)

2 Process Calculi for Security and Mobility – impact on ProVerif

Gordon's pioneering papers on the spi calculus [5] (with Abadi) and the ambient calculus [6,7] (with Cardelli) influenced a generation of theoreticians who built on his work. These formalisms built on the theory of concurrency introduced by Milner et al's pi-calculus, using its private names to represent encryption (in the spi calculus) and accessibility (in ambient calculus). But Gordon **went beyond pi by showing how to solve real-world practical problems, such as errors in web services security protocols, by proofs within the theory**. The spi-calculus paper [5] has over 2000 citations; the core ideas for representing cryptography introduced by spi led to the applied pi calculus, and hence are a basis for popular tools such as Blanchet's

ProVerif, whose original input language was essentially spi.[1] The ETAPS 1998 paper on mobile ambients [6] has over 2600 citations, the most widely cited paper ever at ETAPS, the major European research conference on theory and practice of software.

3 Refinement Types for Security – impact on F*

Gordon pioneered the use of refinement types to prove security properties, first in the setting of the spi-calculus (the Cryptyc typechecker) [9] but subsequently for programming languages in general. He led the development of the Minim checker for M [12] (the language at the heart of Microsoft's PowerQuery), and the F7 typechecker [11], whose lasting impact is as a direct basis for the F* language developed at Microsoft and INRIA.[2] The refinement types of F7 are a key feature of F*, though it has many features not present in F7. The crypto library HACL* is a great success for F* and ships in the Linux and Windows kernels.[3]

4 Probabilistic Programming in Tabular – impact on Stan

Gordon worked on formal semantics of probabilistic programming for machine learning [13,14,15,16]. His research language Tabular [15], to support probabilistic models for data within Excel workbooks, aimed at democratizing machine learning to non-developers such as scientists or business analysts. Tabular introduced the idea of applying an information flow type system to probabilistic programs. Under his supervision, PhD student Maria Gorinova adopted this idea in their SlicStan language [17] to allow a simpler, more compositional syntax for Stan, the most widely used probabilistic programming language. SlicStan is well received by the Stan community, and its "nice composition-friendly syntax" is slated for inclusion in Stan 3.

5 End-User Programming – impact on Excel

Semantics of Formulas: Calc.ts is a new calculation engine for Excel formulas, built from scratch on programming language principles from research on a simple reference implementation of Excel formulas, first in F# and ported to TypeScript. Gordon invented Calc.ts to **accelerate Excel on the web to desktop performance:** it solves the problem of sluggish user experience over slow networks by evaluating formulas within the browser. Work that began as programming language research transformed into a serious engineering effort over several years to deliver a Microsoft Excel feature, that is 100% in production since June 2018 for the many millions of users of Excel on the web.

---

[1] Abadi, Blanchet, and Fournet write about applied pi calculus, "we represent fresh channels, nonces, and keys as new names, and primitive cryptographic operations as functions, obtaining a simple but useful programming-language perspective on security protocols (much as in the spi calculus)." "The spi calculus developed the idea that the context represents an active attacker, and equivalences capture authenticity and secrecy properties in the presence of the attacker." And: "Since 2001, the applied pi calculus has been the basis for ... useful software, such as the tool ProVerif."

[2] Swamy et al write: "F* subsumes two previous languages, F7 and Fine."

[3] See this file for usages of F7-style refinement types in HACL*, eg, n2:size_t{v n1 * v n2 <= max_size_t}.

It is a prize-winning project inducted in July 2020 into Microsoft's **Wall of Fame**. In terms of direct counts of end users, Calc.ts is the **most impactful contribution of his career:** it saves seven person-years of waiting time every single day.

Gordon's invention of Calc.ts arose from his collaboration with Simon Peyton Jones on Project Yellow, a long-term partnership between Excel and Gordon's Calc Intelligence team at MSR Cambridge on enhancing **Excel as a programming language.** Yellow features ship in production to millions of customers, including dynamic arrays, data types, LET and LAMBDA, and have made a lasting impact to spreadsheeting in Excel.

Gordon's paper on elastic sheet-defined functions [18] generalises the classic idea of sheet-defined functions in spreadsheets to allow for variable-sized inputs. The paper is remarkable in that it both provides sophisticated solutions to a range of technical difficulties that arise when achieving the intended generalisation, and conducts a user-study to determine whether spreadsheet users would find the solution understandable and preferable to the best alternative (they do). The paper epitomises the effectiveness of multi-disciplinary work in end-user programming. The journal editors say it is the first paper with a user study ever to appear in the Journal of Functional Programming.

AI generation of formulas: Gordon and his team researched techniques to help end users generate formulas from natural language. The GridBook system [19] generates formulas for processing tables from natural language queries written in the grid. The Formula Autosuggest feature automatically inserts aggregation formulas given natural language context in the grid. A research prototype [20] generates column formulas via AI – it received an award at ACM CHI 2023, and ships as a feature in Excel Copilot.

Other Research Highlights

His paper on representing programming languages in theorem provers [1] introduced the distinction between deep and shallow embeddings, now standard vocabulary.

His paper on mechanizing proofs about variable binding [3], introduces what's now known as the locally nameless representation of lambda terms, considered by some to be the best representation of syntax for programming language metatheory.[4]

His paper on typing a multi-language intermediate code [8] describes the bytecode verifier for the .NET Runtime, to which Gordon contributed. It was the first research paper ever on .NET, still a core piece of the Microsoft developer strategy 20 years on.

His work on the logic language SecPAL [10] for decentralized access control, including delegation and revocation, was shipped in the Microsoft Vine product, and is widely cited in the research literature on authorization languages.

---

[4] Charguéraud writes: "In the context of formal proofs, Gordon [1993] appears to be the first to have used the locally nameless representation."

Statement of applied nature of his work

Since 2016, Gordon has focused on applied research in close partnership with the Microsoft Excel team. With hundreds of millions of users, Excel is the world's most widely used programming language. Gordon shifted his research focus to serve the needs of **end-user programmers**, people like educators, doctors, salespeople, etc., who write formulas to process spreadsheet data without having a professional interest in programming. He began by collaborating with Simon Peyton Jones on Project Yellow, a long-term project to modernize Excel as a programming language. Subsequently, he led and recruited the Calc Intelligence team from scratch: a unique mix of engineers and researchers, with skills in AI, programming languages, and human-computer interaction. Calc Intelligence is one of the most successful and visible projects at Microsoft Research in Cambridge: its work is publicised by Microsoft Research in this and this blog, this podcast, in the history of research collaborations with Microsoft Excel, and the Microsoft Wall of Fame. Building on research papers (including [18,19,20]), his team has delivered feature after feature to the many hundreds of millions of users of Excel. All the links below give credit to Gordon or his team.

- *Project Yellow*, started 2016: dynamic arrays, entities, and the LET and LAMBDA functions. (Excel's LAMBDA is celebrated in the popular web comic XKCD.)
- *Calc.ts*, started 2017, evaluates your formulas in the browser to accelerate your experience of calculation in the web version of Excel to desktop performance.
- *New formula bar architecture*, begun 2020, to add AI-powered augmentations, live now in Excel for the web, coming soon to Excel desktop.
- *Advanced Formula Environment* (AFE), released in 2022 as a Microsoft Garage "beta" feature, facilitates authoring of LAMBDAs.
- *Formula Autosuggestions*, shipped in 2022, automatically predicts aggregate formulas using neural net research from his team.
- *Excel Labs a Microsoft Garage Project*, released in 2023, bundles experimental features including AFE, the function LABS.GENERATIVEAI() to access AI models within the grid, sheet-defined functions [18], and the Python Editor (to write and edit Python Formulas).

Brian Jones, former Product Manager for Microsoft Excel speaks here with Gordon on YouTube about the "thought leadership we've had from the Calc Intel team". (See also this YouTube video and this one about Excel Labs features.)

Abigail Sellen FRS, director of MSR Cambridge, said: "Andy has consistently led a highly impactful research team delivering a pipeline of cutting-edge technologies to Excel through a strong partnership with that team. This includes features like Yellow, Calc.ts, Formula Autosuggest, and LET/LAMBDAS. He has just successfully pivoted his team around LLMs, contributing a highly successful feature in Excel Co-Pilot."

Top 20 Most Significant Publications

1. Richard J. Boulton, Andrew D. Gordon, Michael J. C. Gordon, John Harrison, John Herbert, John Van Tassel: **Experience with Embedding Hardware Description Languages in HOL**. TPCD 1992: 129-156 [1] (317 citations)
2. Andrew D. Gordon: **Functional programming and input/output**. University of Cambridge, 1992. Published as book by Cambridge University Press, 1994 [2] (206 citations)
3. Andrew D. Gordon: **A Mechanisation of Name-Carrying Syntax up to Alpha-Conversion.** HUG 1993: 413-425 [3] (91 citations)
4. Simon L. Peyton Jones, Andrew D. Gordon, Sigbjørn Finne: **Concurrent Haskell.** POPL 1996: 295-308 [4] (649 citations)
5. Martín Abadi, Andrew D. Gordon: **A Calculus for Cryptographic Protocols: The Spi Calculus**. Inf. Comput. 148(1): 1-70 (1999) [5] (2145 citations)
6. Luca Cardelli, Andrew D. Gordon: **Mobile ambients**. Theor. Comput. Sci. 240(1): 177-213 (2000) [6] (2623 citations)
7. Luca Cardelli, Andrew D. Gordon: **Anytime, Anywhere: Modal Logics for Mobile Ambients**. POPL 2000: 365-377 [7] (528 citations)
8. Andrew D. Gordon, Don Syme: **Typing a multi-language intermediate code**. POPL 2001: 248-260 [8] (139 citations)
9. Andrew D. Gordon, Alan Jeffrey: **Authenticity by Typing for Security Protocols**. J. Comput. Secur. 11(4): 451-520 (2003) [9] (405 citations)
10. Moritz Y. Becker, Cédric Fournet, Andrew D. Gordon: **SecPAL: Design and semantics of a decentralized authorization language**. J. Comput. Secur. 18(4): 619-665 (2010) [10] (472 citations)
11. Jesper Bengtson, Karthikeyan Bhargavan, Cédric Fournet, Andrew D. Gordon, Sergio Maffeis: **Refinement types for secure implementations**. ACM Trans. Program. Lang. Syst. 33(2): 8:1-8:45 (2011) [11] (348 citations)
12. Gavin M. Bierman, Andrew D. Gordon, Catalin Hritcu, David E. Langworthy: **Semantic subtyping with an SMT solver**. J. Funct. Program. 22(1): 31-105 (2012) [12] (84 citations)
13. Sooraj Bhat, Johannes Borgström, Andrew D. Gordon, Claudio V. Russo: **Deriving Probability Density Functions from Probabilistic Functional Programs.** TACAS 2013: 508-522 [13] (49 citations)
14. Andrew D. Gordon, Thomas A. Henzinger, Aditya V. Nori, Sriram K. Rajamani: **Probabilistic programming**. FOSE 2014: 167-181 [14] (562 citations)
15. Andrew D. Gordon, Thore Graepel, Nicolas Rolland, Claudio V. Russo, Johannes Borgström, John Guiver: **Tabular: a schema-driven probabilistic programming language**. POPL 2014: 321-334 [15] (69 citations)

16. Johannes Borgström, Ugo Dal Lago, Andrew D. Gordon, Marcin Szymczak: **A lambda-calculus foundation for universal probabilistic programming**. ICFP 2016: 33-46 [16] (146 citations)
17. Maria I. Gorinova, Andrew D. Gordon, Charles Sutton: **Probabilistic programming with densities in SlicStan: efficient, flexible, and deterministic**. Proc. ACM Program. Lang. 3(POPL): 35:1-35:30 (2019) [17] (25 citations)
18. Matt McCutchen, Judith Borghouts, Andrew D. Gordon, Simon Peyton Jones, Advait Sarkar: **Elastic sheet-defined functions: Generalising spreadsheet functions to variable-size input arrays**. J. Funct. Program. 30: e26 (2020) [18] (12 citations)
19. Sruti Srinivasa Ragavan, Zhitao Hou, Yun Wang, Andrew D. Gordon, Haidong Zhang, Dongmei Zhang: **GridBook: Natural Language Formulas for the Spreadsheet Grid**. IUI 2022: 345-368 [19] (22 citations)
20. Michael Xieyang Liu, Advait Sarkar, Carina Negreanu, Benjamin G. Zorn, Jack Williams, Neil Toronto, Andrew D. Gordon: **"What It Wants Me To Say": Bridging the Abstraction Gap Between End-User Programmers and Code-Generating Large Language Models**. CHI 2023: 598:1-598:31 [20] (149 citations)