

Curriculum Vitae

Prof. Dr. Byron Cook

Affiliations: Microsoft Research Cambridge and Queen Mary, University of London

Address: Roger Needham Building, 7 J J Thomson Avenue, Cambridge CB3 0FB, United Kingdom

Email: byroncook@gmail.com

Telephone: +44 (0)1223 479795

URL: <http://research.microsoft.com/users/bycook>

Short biography

I am a senior researcher at Microsoft's laboratory at Cambridge University, and full professor of computer science at Queen Mary, University of London. My research interests include topics in formal methods, logic and programming languages. I also have an interest and expertise in the areas of hardware and systems, based on previous work experience in Intel's microprocessor design division and Microsoft's Windows kernel team. To date I have worked actively in the areas of functional programming, hardware modeling and design, SAT-solving, symbolic model checking for finite-state systems, decision procedures, and automatic program verification.

My recent research in automatic program verification has gained significant attention (*e.g.* a substantial publication record, numerous keynote speaker invitations, and press hits in *Scientific American*, *Vogue*, *Financial Times*, *Economist*, and *Wired*). I am particularly well known for my work on automatic methods for proving program termination—this work represents a breakthrough, challenging the prevailing opinion in computer science that automatic termination proving is impossible. I am also well known for my contributions to SLAM, which is often credited for the recent revival of automatic program verification research. My new work on hardware synthesis for C programs promises to lead to further high impact in the area of computer design and verification.

As well as traditional industrial research activities (publishing research papers, working with product groups, supervising interns, internal Microsoft lectures, etc), I have been active in academic outreach. For example, I have taught several courses at top universities (CMU, Berkeley, Cambridge) on termination and was invited to give a set of lectures at the prestigious Marktoberdorf summer school. I also supervise Ph.D. students and postdoctoral researchers at Cambridge University as well as Microsoft researchers and developers.

Awards

- Roger Needham Award, 2009. The Roger Needham Award is made annually by the British Computer Society for a distinguished research contribution in computer science by a UK based researcher within ten years of their PhD. The award includes a 5,000 GBP cash prize, and a public lecture at the UK's Royal Society in London.

Publications

Journal articles

1. *Summarization For Termination*
Byron Cook, Andreas Podelski, and Andrey Rybalchenko
International Journal on Formal Methods and System Design (*forthcoming*)
2. *Termination proving is not impossible after all*
Byron Cook
Communications of the ACM (*forthcoming*)
3. *Software engineering and formal methods*
Mike Hinchey, Michael Jackson, Patrick Cousot, Byron Cook, Jonathon P. Bowen
Communications of the ACM, Vol. 51, 2008, pp. 54-59
4. *Verification of Boolean programs with unbounded thread creation*
Byron Cook, Daniel Kroening, and Natasha Sharygina
Journal of Theoretical Computer Science, Vol. 388, 2007, pp. 227-242
5. *Predicate abstraction via symbolic decision procedures*
Shuvendu Lahiri, Tom Ball, and Byron Cook
Journal of Logic Methods in Computer Science, Vol. 3(1:2), 2007, pp. 1-20
6. *Design automation with mixtures of proof strategies for propositional logic*
Gunnar Andersson, Per Bjesse, Byron Cook, and Ziyad Hanna
IEEE Transactions on CAD, Vol. 22(8), 2003, pp. 1042-1048
7. *A framework for microprocessor correctness statements*
Mark Aagaard, Byron Cook, Nancy Day, and Robert Jones
International Journal on Software Tools for Technology Transfer, Vol 4(3), 2002, pp. 298-312

Conference articles

8. *Finding heap-bounds for hardware synthesis*
Byron Cook, Ashutosh Gupta, Stephen Magill, Andrey Rybalchenko, Jiri Simsa, Satnam Singh, and Viktor Vafeiadis
FMCAD [International Conference on Formal Methods in Computer Aided Design], Austin, 2010
9. *Proving that non-blocking algorithms don't block*
Alexey Gotsman, Byron Cook, Matthew Parkinson, and Viktor Vafeiadis
POPL [International Symposium on Principles of Programming Languages], Savannah, 2009
10. *Proving conditional termination*
Byron Cook, Sumit Gulwani, Tal Lev-Ami, Andrey Rybalchenko, and Mooly Sagiv
CAV [International Conference on Computer-Aided Verification], Princeton, 2008
11. *Scalable shape analysis for systems code*
Hongseok Yang, Oukseh Lee, Josh Berdine, Cristiano Calcagno, Byron Cook, Dino Distefano, and Peter O'Hearn
CAV [International Conference on Computer-Aided Verification], Princeton, 2008
12. *Ranking abstractions*
Aziem Chawdhary, Byron Cook, Sumit Gulwani, Mooly Sagiv, and Hongseok Yang
ESOP [European Symposium on Programming], Budapest, 2008

13. *Proving thread termination*
Byron Cook, Andreas Podelski, and Andrey Rybalchenko,
PLDI [International Conference on Programming Language Design and Implementation], San Diego, 2007
14. *Thread-modular shape analysis*
Alexey Gotsman, Josh Berdine, Byron Cook, and Mooly Sagiv,
PLDI [International Conference on Programming Language Design and Implementation], San Diego, 2007
15. *Local reasoning for storable locks and threads*
Alexey Gotsman, Josh Berdine, Byron Cook, Noam Rinetzky, and Mooly Sagiv
APLAS [Asian Symposium on Programming Languages and Systems], Singapore, 2007
16. *Proving that programs eventually do something good*
Byron Cook, Alexey Gotsman, Andreas Podelski, Andrey Rybalchenko, and Moshe Vardi
POPL [International Symposium on Principles of Programming Languages], Nice 2007
17. *Variance analyses from invariance analyses*
Josh Berdine, Aziem Chawdhary, Byron Cook, Dino Distefano, and Peter O’Hearn
POPL [International Symposium on Principles of Programming Languages], Nice 2007
18. *Shape analysis for composite data structures*
Josh Berdine, Cristiano Calcagno, Byron Cook, Dino Distefano, Peter O’Hearn, Thomas Wies,
and Hongseok Yang
CAV [International Conference on Computer-Aided Verification], Berlin, 2007
19. *Arithmetic strengthening for shape analysis*
Stephen Magill, Josh Berdine, Edmund Clarke, and Byron Cook,
SAS [International Static Analysis Symposium], Denmark, 2007
20. *Proving termination by divergence*
Domagoj Babic, Byron Cook, Alan Hu, and Zvonimir Rakamaric
SEFM [International Conference on Software Engineering and Formal Methods], London, 2007
21. *Shape analysis by graph decomposition*
Roman Manevich, Josh Berdine, Byron Cook, Ganesan Ramalingam, and Mooly Sagiv
TACAS [International Conference on Tools and Algorithms for the Construction and Analysis of
Systems], Braga, 2007
22. *Over-approximating Boolean programs with unbounded thread creation*
Byron Cook, Daniel Kroening, Natasha Sharygina
FMCAD [International Conference on Formal Methods in Computer Aided Design], San Jose,
2007
23. *Interprocedural shape analysis with separated heap abstractions*
Alexey Gotsman, Josh Berdine, and Byron Cook
SAS [International Symposium on Static Analysis], Seoul, 2007
24. *Automatic termination proofs for programs with shape-shifting heaps*
Josh Berdine, Byron Cook, Dino Distefano, and Peter O’Hearn
CAV [International Conference on Computer-Aided Verification], Seattle, 2006
25. *Terminator: Beyond safety*
Byron Cook, Andreas Podelski, and Andrey Rybalchenko
CAV [International Conference on Computer-Aided Verification], Seattle, 2006

26. *Repair of Boolean programs with an application to C*
Andreas Griesmayer, Roderick Bloem, and Byron Cook
CAV [International Conference on Computer-Aided Verification], Seattle, 2006
27. *Termination proofs for systems code*
Byron Cook, Andreas Podelski, and Andrey Rybalchenko
PLDI [International Conference on Programming Language Design and Implementation], Ottawa, 2006
28. *Thorough static analysis of device drivers*
Thomas Ball, Ella Bounimova, Byron Cook, Vladimir Levin, Jakob Lichtenberg, Con McGarvey, Bohus Ondrusek, Sriram K. Rajamani, Abdullah Ustuner
EuroSys [European Systems Conference], Leuven, 2006
29. *Abstraction-refinement for termination*
Byron Cook, Andreas Podelski, Andrey Rybalchenko
SAS [International Symposium on Static Analysis], London, 2005
30. *Using Stålmarck's algorithm to prove inequalities*
Byron Cook and George Gonthier
ICFEM [International Conference on Formal Engineering Methods], Manchester, 2005
31. *Predicate abstraction via symbolic decision procedures*
Shuvendu Lahiri, Thomas Ball, and Byron Cook
CAV [International Conference on Computer-Aided Verification], Edinburgh, 2005
32. *Cogent: Accurate theorem proving for program verification*
Byron Cook, Daniel Kroening, and Natasha Sharygina
CAV [International Conference on Computer-Aided Verification], Edinburgh, 2005
33. *Zapato: Automatic theorem proving for predicate abstraction refinement*
Thomas Ball, Byron Cook, Shuvendu K. Lahiri, and Lintao Zhang
CAV [International Conference on Computer-Aided Verification], Boston, 2004
34. *Refining approximations in software predicate abstraction*
Thomas Ball, Byron Cook, Satyaki Das, and Sriram K. Rajamani
TACAS [International Conference on Tools and Algorithms for the Construction and Analysis of Systems], Barcelona, 2004
35. *SLAM and Static Driver Verifier: technology transfer of formal methods inside Microsoft*,
Thomas Ball, Byron Cook, Vladimir Levin and Sriram K. Rajamani.
IFM [International Conference on Integrated Formal Methods], Kent, 2004
36. *Accurate theorem proving for program verification*,
Byron Cook, Daniel Kroening, Natasha Sharygina,
ISoLA [Leveraging Applications of Formal Methods], Paphos, 2004
37. *A symbolic approach to predicate abstraction*
Shuvendu K. Lahiri, Randall E. Bryant, and Byron Cook
CAV [International Conference on Computer-Aided Verification], Boulder, 2003
38. *A proof engine approach to solving combinational design automation problems*
Gunnar Andersson, Per Bjesse, Byron Cook, and Ziyad Hanna
DAC [Design Automation Conference], Las Vegas, 2002
39. *A framework for microprocessor correctness statements*
Mark Aagaard, Byron Cook, Nancy Day, and Robert Jones
CHARME [Advanced Research Working Conference on Correct Hardware Design and Verification Methods], Edinburgh, 2001

40. *Combining stream-based and state-based verification techniques for microarchitectures*
Mark Aagaard, Byron Cook, and Nancy Day
FMCAD [International Conference on Formal Methods in Computer Aided Design], Austin, 2000
41. *Formal verification of explicitly parallel microprocessors*
Byron Cook, John Launchbury, John Matthews, and Dick Kieburtz
CHARME [Advanced Research Working Conference on Correct Hardware Design and Verification Methods], Bad Herrenalb, 1999
42. *On embedding a microarchitectural design language within Haskell*
John Launchbury, Jeff Lewis and Byron Cook
ICFP [International Conference on Functional Programming], Paris, 1999
43. *Microprocessor specification in Hawk*
John Matthews, John Launchbury, and Byron Cook
ICCL [International Conference on Computer Languages], Chicago, 1998

Books

44. *Program termination*
Byron Cook
Cambridge University Press (forthcoming)
45. *Proceedings of the 8th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI)*
Byron Cook and Andreas Podelski (Eds.)
Springer

Journal special issues

46. *Special Issue on Satisfiability Modulo Theories*
Byron Cook and Roberto Sebastiani (Eds.)
Journal on Satisfiability, Boolean Modeling and Computation

Workshop articles

47. *Symbolic model checking for asynchronous Boolean programs*
Byron Cook, Daniel Kroening, and Natasha Sharygina
SPIN [SPIN Workshop on Model Checking of Software], San Francisco, 2005
48. *Specifying superscalar microprocessors in Hawk*
Byron Cook, John Launchbury, and John Matthews
FTH [International Workshop on Formal Techniques for Hardware], Marstrand, 1998
49. *Disposable memo functions*
Byron Cook and John Launchbury
Haskell Workshop, Amsterdam, 1997

Invited articles

50. *Principles of program termination*
Byron Cook
Lecture notes from 2008 Marktoberdorf summer school (Marktoberdorf)

51. *Advances in Program Termination and Liveness*
Byron Cook
VMCAI [International Conference on Verification, Model Checking, and Abstract Interpretation], Savannah, 2009
52. *Computing bounds on space and time for hardware compilation*
Byron Cook
FMCAD [Formal Methods in Computer Aided Design], Portland, 2008
53. *Automatically proving program termination*
Byron Cook
CAV [International Conference on Computer-Aided Verification], Berlin, 2007
54. *Bringing hardware and software closer together with termination analysis*
Byron Cook
MEMOCODE [International Conference on Formal Methods and Models for Codesign], Nice, 2007
55. *Automatically Proving Concurrent Programs Correct*
Byron Cook
SEFM [IEEE International Conference on Software Engineering and Formal Methods], London, 2007
56. *Finding bugs in device drivers with Static Driver Verifier*
Byron Cook
ASM [International Conference on Abstract State Machines], Paris, 2005
57. *Finding API usage rule violations in Windows device drivers using Static Driver Verifier*
Byron Cook
ISoLA [Leveraging Applications of Formal Methods], Paphos, 2004

Press coverage

- *A Good Sign*, Angela Saini, Science, July 2009
- *Optic Nerve*, Dodie Kazanjain, Vogue, January 2009
- *All Shook Down*, Hiya Swanhuysen, San Francisco Weekly, December 2008
- *Byron Cook: Terminator - Proving good things will eventually happen* (Video interview), Charles Torre, MSDN, July 2007
- *Byron Cook: Inside Terminator* (Video interview), Charles Torre, MSDN, September 2007
- *Send in the Terminator*, Gary Stix, Scientific American, December 2006
- *Testers aim to kill off dreaded blue screens*, Mary Branscombe, Financial Times, November 22nd, 2006
- *Microsoft creates an application terminator*, Andy Patrizio, Internet News, August, 2006
- *Microsoft researcher aims to make software more predictable*, Tony Baer, Computer Wire, August, 2006,
- *Microsoft bug-checking tools promise fewer crashes*, Joris Evens, CNET, May 26, 2006
- *Microsoft's secret bug squasher*, Simson Garfinkel, Wired, November, 2005

- *Researching a path to fewer bugs*, Patrick Meader, Visual Studio Magazine, February 2003
- *Building a better bug-trap* , Economist, June 19th, 2003

Teaching

- *Program termination*, 4 graduate-level lecture hours at University of California, Berkeley, 2009
- *Program termination*, 6 graduate-level lecture hours at the International Summer School on Trends in Concurrency (Prague), 2008
- *Program termination*, 5 graduate-level lecture hours at the Marktoberdorf Summer School, 2008
- *Program termination*, 6 graduate-level lecture hours at the International Summer School on Trends in Concurrency (Prague), 2008
- *Program termination*, 24 graduate-level lecture hours at Carnegie Mellon University, 2008
- *Program termination*, 6 graduate-level lecture hours at Imperial College, 2008
- *Program termination*, 6 graduate-level lecture hours at Cambridge University, 2007
- *Introduction to C++*, 20 undergraduate-level lecture hours at The Evergreen State College, 1998
- *Constructing Applets in Java*, 10 undergraduate-level lecture hours at The Evergreen State College, 1998
- *C++ and Java, Object Oriented Programming*, 20 undergraduate-level lecture hours at The Evergreen State College, 1997
- *C programming language: Introduction*, 20 undergraduate-level lecture hours at The Evergreen State College, 1997
- *Web programming in Perl*, 10 undergraduate-level lecture hours at The Evergreen State College, 1997
- *Introduction to Computer Science II*, 20 undergraduate-level lecture hours at Portland Community College, 1997
- *Introduction to Computer Programming*, 20 undergraduate-level lecture hours at Oregon Institute of Technology, 1997
- *Introduction to Data Structures*, 20 undergraduate-level lecture hours at Oregon Institute of Technology, 1997
- *Introduction to Programming for the World Wide Web*, 20 undergraduate-level lecture hours at The Evergreen State College, 1996
- *Introduction to UNIX Operating System*, 20 undergraduate-level lecture hours at The Evergreen State College, 1996
- *Introduction to Computer Science I*, 20 undergraduate-level lecture hours at Portland Community College, 1996

Tools developed

- T2 : Program termination prover
(in development)
- TERMINATOR : Program termination prover
<http://research.microsoft.com/TERMINATOR>
- SLAYER : Shape analysis engine
<http://research.microsoft.com/SLayer>
- SLAM : Symbolic software model checker
<http://research.microsoft.com/SLAM>
- Static Driver Verifier : Device driver correctness tool (Microsoft Windows product released through the Windows Device Driver Development Kit)
<http://www.microsoft.com/whdc/devtools/tools/SDV.aspx>
- ZAPATO : Decision procedure framework
(Microsoft internal)
- POINDEXTER : Pointer analysis engine
(Microsoft internal)
- PROVER CL : Propositional SAT solver
<http://www.prover.com/products/ppi/cl.xml>
- PROVER SL : Symbolic model checker for finite-state systems
<http://www.prover.com/products/ppi/sl.xml>
- HLSPEC : High-level microprocessor design language and tools
(Intel internal)
- HAWK : High-level microprocessor design language and tools
<http://www.cse.ogi.edu/PacSoft/projects/Hawk/>

Invited keynotes and tutorials

- IFIP Working Group 2.3, Boston, 2009
- HCSS [International Conference on High Confidence Software and Systems], Baltimore, 2009
- PSY [CAV Workshop on Practical Synthesis for Concurrent Systems], Grenoble, 2009
- NFM [NASA Formal Methods Symposium], Moffett Field, 2009
- VMCAI [International Conference on Verification, Model Checking, and Abstract Interpretation], Savannah, 2009
- IFM [Integrated Formal Methods], Dusseldorf, 2009
- Infinity [International Workshop on Verification of Infinite-State Systems], Bologna, 2009
- Workshop on Applied Logic: Inductive and Deductive Reasoning, 2009
- IFIP Working Group 2.3, Cambridge, 2008
- Marktoberdorf Summer School, 2008

- FMCAD [Formal Methods in Computer Aided Design], Portland, 2008
- HCSS [International Conference on High Confidence Software and Systems], Baltimore, 2008
- International Summer School on Trends in Concurrency (Prague), 2008
- CAV Workshop on Numerical Abstractions for Software Verification, 2008
- Science of Security Workshop, Berkeley, 2008
- QCon Enterprise Software Development Conference, San Francisco, 2007
- CAV [International Conference on Computer-Aided Verification] Berlin, 2007
- HCSS [International Conference on High Confidence Software and Systems], Baltimore, 2007
- MEMOCODE [International Conference on Formal Methods and Models for Codesign] (Nice), 2007
- SEFM [IEEE International Conference on Software Engineering and Formal Methods] London, 2007
- Seminar on the Challenge of Software Verification, Dagstuhl, 2006
- WST [International Workshop on Termination], Seattle, 2006
- SVV [International Workshop on Software Verification and Validation], Seattle, 2006
- AVoCS [International Workshop on Automated Verification of Critical Systems], Nancy, 2006
- ARW [Automated Reasoning Workshop], Bristol, 2006
- ICSSR [International Computer Science Symposium in Russia], St. Petersburg, 2006
- ESCAR [CADE Workshop on Empirically Successful Classical Automated Reasoning], 2005
- DISPROVING [Workshop on Disproving - Non-Theorems, Non-Validity, Non-Provability] Tallinn, 2005
- ASM [International Workshop on Abstract State Machines], Paris, 2005
- Combination of Decision Procedures Summer School, Stanford, 2004
- HCSS [Conference on High Confidence Software and Systems], Baltimore, 2004
- ISoLA [Leveraging Applications of Formal Methods] (Paphos)
- Colloquium L'ingnierie du logiciel, Paris, 2004
- DAC [Design Automation Conference], Las Vegas 2001
- University colloquium lectures at Stanford, Berkeley, Carnegie Mellon, Harvard, MIT, ETH, INRIA, University of Toronto, University of Birmingham, University of British Columbia, University of Utah, University of Colorado, Oxford, Cambridge, Chalmers, and New York University.
- Internal symposiums at Compaq, Intel, the US National Security Agency, and Siemens.
- Microsoft Techfest¹ in 2009, 2007, 2006, and 2005

¹Techfest is a Microsoft event in which researchers give lectures and make demos available to Microsoft employees and the press. The event is high-profile (>30,000 attendees) and the lectures are selected using a competitive process.

Panel discussions

- NASA Formal Methods Symposium, Moffett Field, 2009
- Science of Security Workshop, Berkeley, 2008
- SMT [International Workshop on Satisfiability Modulo Theories], Princeton, 2008
- HCSS [International Conference on High Confidence Software and Systems], Baltimore, 2008
- University of Illinois Affiliates Conference, 2006
- MEMOCODE [International Conference on Formal Methods and Models for Codesign] Verona, 2005

Professional activities

- Program committee appointments:
 - SAS [International Static Analysis Symposium], 2010
 - CAV [International Conference on Computer-Aided Verification], 2010
 - CAV [International Conference on Computer-Aided Verification], 2009
 - FMICS [Formal Methods for Industrial Critical Systems], 2009
 - WST [International Workshop on Termination], 2009
 - POPL [Symposium on Principles of Programming Languages], 2008
 - TACAS [International Conference on Tools and Algorithms for the Construction and Analysis of Systems], 2008
 - SSV [International Workshop on Systems Software Verification], 2008
 - LPAR [International Conference on Logic for Programming Artificial Intelligence and Reasoning], 2007
 - VMCAI [International Conference on Verification, Model Checking, and Abstract Interpretation], 2007
 - TACAS [International Conference on Tools and Algorithms for the Construction and Analysis of Systems], 2007
 - SMT [International Workshop on Satisfiability Modulo Theories], 2007
 - SV [International Workshop on System Verification], 2007
 - TV [Thread Verification Workshop], 2006
 - PDPAR [Pragmatics of Decision Procedures in Automated Reasoning], 2006
 - SoftMC [Software Model Checking Workshop], 2005
 - CUFP [ICFP Workshop on Commercial Users of Functional Programming], 2004
 - SoftMC [Software Model Checking Workshop], 2003
 - CHARME [Advanced Research Working Conference on Correct Hardware Design and Verification Methods], 2003
- Steering committee, International Workshop on Satisfiability Modulo Theories
- Future co-chair, CAV [International Conference on Computer-Aided Verification], Edinburgh, 2010

- Future co-chair, FMICS [Formal Methods for Industrial Critical Systems], Eindhoven, 2009
- Workshops chair, CAV [International Conference on Computer-Aided Verification], Princeton, 2008
- Guest co-editor, Journal on Satisfiability, Boolean Modeling, and Computation (Special Issue on Satisfiability Modulo Theories),
- Program chair for tools papers, TACAS [International Conference on Tools and Algorithms for the Construction and Analysis of Systems], 2008
- Co-organizer, *Seminar on Deduction and Decision Procedures*, Schloss Dagstuhl, 2007
- Co-chair, AHA [International Symposium on Automatic Heap Analysis], 2007
- Co-chair, VMCAI [International Conference on Verification, Model Checking, and Abstract Interpretation], 2007
- Co-organizer, SSPV [Symposium on SAT-solvers and Program Verification], 2006
- Co-chair, PDPAR [Pragmatics of Decision Procedures in Automated Reasoning], 2006
- Program chair for tools papers, TACAS [International Conference on Tools and Algorithms for the Construction and Analysis of Systems], 2007
- Co-organizer of SoftMC'05 and SoftMC'03 [CAV Workshop on Software Model Checking]
- Co-organizer of CFDP'05 [Cambridge Forum on Decision Procedures]

Work history

Industrial positions

- Microsoft Research, Senior researcher, 2004-Current
- Monoidics Ltd, Corporate Board Member, 2008-Current
- Microsoft Base OS group, Software developer, 2002-2004
- Prover Technology AB, Software developer, 1999-2002
- Intel Strategic CAD Labs, Software developer, 1998-1999

Academic positions

- Queen Mary, University of London, Full Professor (joint with Microsoft), 2008-Current
- Carnegie Mellon University, Visiting professor, 2008
- Queen Mary, University of London, Visiting professor, 2006-2008
- Chalmers University, Visiting lecturer, 2005-2007
- The Evergreen State College, Adjunct professor, 1996-1998
- Portland Community College, Adjunct professor, 1996-1997
- Oregon Institute of Technology, Adjunct professor, 1997

Education

- Ph.D. The Oregon Graduate Institute of Science and Technology, Defended 2002, Advisor: John Launchbury
- B.Sci. The Evergreen State College, Graduated 1995.

References

- Sir Prof. Dr. Tony Hoare – Microsoft Research
thoare@microsoft.com
- Prof. Dr. Peter O’Hearn – Queen Mary, University of London
ohearn@dcs.qmul.ac.uk
- Prof. Dr. Andreas Podelski – Freiburg University
podelski@informatik.uni-freiburg.de
- Prof. Dr. Mooly Sagiv – Tel Aviv University
msagiv@post.ta.ac.il
- Dr. Don Syme – Microsoft Research
dsyme@microsoft.com
- Prof. Dr. Moshe Vardi – Rice University
vardi@cs.rice.edu