Andrey Rybalchenko

August 2014

Office	Microsoft Research 21 Station Road CB1 2FB Cambridge UK
Phone	+44 1223 745216
Email	rybal@microsoft.com
Web	research.microsoft.com/people/rybal/ (previously www7.in.tum.de/~rybal)

Research Interests

Temporal verification of reactive systems, program analysis and verification, automated deduction, constraint logic programming, type systems and inference, distributed systems, systems and information flow security, applied formal logics.

Work Experience

since Apr 2013	Senior Researcher Microsoft Research, Cambridge, UK
Jan 2010 – Jan 2014	Professor of Theoretical Computer Science (W2, tenured) Technische Universität München (TUM), Germany
Sep 2007 – Dec 2009	Researcher, Head of Verification Systems Group Max Planck Institute for Software Systems (MPI-SWS), Germany
Jan 2006 – Aug 2007	Postdoctoral Researcher Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland
Jun 2005 – Aug 2005	Summer Intern Microsoft Research, Cambridge, UK
Oct 2004 – Nov 2004	Visitor Microsoft Research, Redmond, USA
Oct 2002 – Aug 2007	Researcher Max Planck Institute for Computer Science (MPI-INF), Germany
May 2002 – Sep 2002	Research Assistant Max Planck Institute for Computer Science (MPI-INF), Germany
Jan 1999 – Apr 2002	Research Assistant German Research Center for Artificial Intelligence (DFKI), Germany

Education

Jun 2005	Ph.D. in Computer Science University of Saarland, Germany
Sep 2002	Diploma in Computer Science University of Saarland, Germany
Jun 2000	Diploma in Computer-Aided Mechanical Engineering Voronezh State Technical University, Russia

Prizes, Awards, and Distinctions

1. The gold winner in the Concurrency category of the 2nd Intl. Competition on Software Verification held at TACAS 2013.
2. ERC Starting Grant "Automatic synthesis of software verification tools from proof rules", 1,500,000 €, June 2012 – January 2014.
3. The bronze winner in the ControlFlowInteger category of the 1st Intl. Competition on Software Verification held at TACAS 2012.
4. Doctoral dissertation award for Ashutosh Gupta's thesis "Constraint solving for verification" from the alumni association of TUM, December 2011.
5. Capital Magazine: 40 Talents under 40 in Academia, November 2011.

6. Capital Magazine: 40 Talents under 40 in Academia, November 2010.
7. MIT TR35 Young Innovator Award, September 2010.
8. Visiting professorship at LIAFA, University Paris 7, March 2010.
9. Fellowship (PI) from Microsoft Research PhD Scholarship Programme, 100,000 €, January 2010.
10. Best paper at the 12th European Joint Conferences on Theory and Practice of Software, April 2009.
11. Microsoft Research European Fellowship, 250,000 €, September 2006.
12. Otto Hahn research group (W2) from the Max Planck Society, July 2006.
13. Otto Hahn Medal awarded by the Max Planck Society, 2005.
14. Ph.D. summa cum laude, Saarland University, 2005.
15. Scholarship from the International Max Planck Research School for Computer Science, 2002–2003.
16. Günther Hotz Medal and Prize from the University of Saarland, 2002.

Press Coverage

1. Andrey Rybalchenko: Leidenschaft für Software. E. Tsakiridou. In *INGENIEUR.de*[1], November 16, 2012.
2. Prof. Software – und wie er die Welt retten will. C. Seidl. In *Münchner Merkur*, Issue 188, August 17, 2011.
3. Der Terminator und Hilberts Würstchenfabrik. E. Tsakiridou. In *TUM Faszination Forschung*, Issue 8, July 2011.
4. Absolvent der Saar-Uni zählt zu den besten Informatikern der Welt. In *Saarbrücker Zeitung*, August 31, 2010.
5. Software under stress. U. Deffke. In *MaxPlanckResearch*, Issue 02, 2009.
6. Send in the Terminator: A Microsoft tool looks for programs that freeze up. G. Stix. In *Scientific American*, December 2006.
7. Testers aim to kill off dreaded blue screens. M. Branscombe. In *Financial Times*, November 22, 2006.

Software

1. ARMC – Abstraction Refinement Model Checker for safety and liveness.
   `http://www7.in.tum.de/~rybal/armc/`
2. Cardan – cardinality analysis tool for declarative networking applications.
   `http://www7.in.tum.de/~rybal/cardan/`
3. Clp-Prover – satisfiability checker and interpolant synthesizer for linear arithmetic combined with uninterpreted function symbols.
   `http://www7.in.tum.de/~rybal/clp-prover/`
4. DAHL – a distributed programming system.
   `http://www7.in.tum.de/tools/dahl/`
5. HSF – a constraint solver for Horn clauses.
   `http://www7.in.tum.de/tools/hsf/`
6. InvGen – an efficient invariant generator.
   `http://www7.in.tum.de/~rybal/invgen/`
7. RankFinder – well-foundedness checker and ranking function synthesizer.
   `http://www7.in.tum.de/~rybal/rankfinder/`
8. Terminator – termination checker for C programs that integrates transition invariant-based termination checking into Microsoft tool suite SDV.
   `http://research.microsoft.com/TERMINATOR/`

[1] http://www.ingenieur.de/Arbeit-Beruf/Arbeitsmarkt/Andrey-Rybalchenko-Leidenschaft-fuer-Software

Publications

Refereed Conference Papers

1. Reduction for compositional verification of multi-threaded programs. With C. Popeea and A. Wilhelm. In *FMCAD: Formal Methods in Computer-Aided Design*. IEEE, 2014.

2. A constraint-based approach to solving games on infinite graphs. With T. Beyene, S. Chaudhuri, and C. Popeea. In *POPL: Principles of Programming Languages*. ACM, 2014.

3. Separation logic modulo theories. With J. Navarro Pérez. In *APLAS: Asian Symp. on Programming Languages and Systems*. Springer, 2013.

4. Automation of quantitative information-flow analysis. With B. Köpf. In *SFM: Formal Methods for Dynamical Systems*. Springer, 2013.

5. An epistemic perspective on consistency of concurrent computations. With K. v. Gleissenthall. In *CONCUR: Concurrency Theory*. Springer, 2013.

6. On solving universally quantified Horn Clauses. With N. Bjorner and K. McMillan. In *SAS: Static Analysis Symposium*. Springer, 2013.

7. Solving existentially quantified Horn clauses. With T. Beyene and C. Popeea. In *CAV: Computer Aided Verification*. Springer, 2013.

8. Threader: a verifier for multi-threaded programs - (competition contribution). With C. Popeea. In *TACAS: Tools and Algorithms for the Construction and Analysis of Systems, (software verification competition contribution)*. Springer, 2013.

9. Binary reachability analysis of higher order functional programs. With R. Ledesma-Garza. In *SAS'2012*.

10. Synthesizing software verifiers from proof rules. With S. Grebenshchikov, N. Lopes, and C. Popeea. In *PLDI: Programming Language Design and Implementation*. ACM, 2012.

11. HSF(C): a software verifier based on Horn clauses. With S. Grebenshchikov, A. Gupta, N. Lopes, and C. Popeea. In *TACAS'2012, (software verification competition contribution)*.

12. Compositional termination proofs for multi-threaded programs. With C. Popeea. In *TACAS'2012*.

13. Solving recursion-free Horn clauses over LI+UIF. With A. Gupta and C. Popeea. In *APLAS'2011*.

14. HMC: Verifying functional programs using abstract interpreters. With R. Jhala and R. Majumdar. In *CAV'2011*.

15. Threader: a constraint-based verifier for multi-threaded programs. With A. Gupta and C. Popeea. In *CAV'2011*.

16. Separation logic + superposition calculus = heap theorem prover. With J. Navarro Pérez. In *PLDI'2011*.

17. Predicate abstraction and refinement for verifying multi-threaded programs. With A. Gupta and C. Popeea. In *POPL'2011*.

18. Distributed and predictable software model checking. With N. Lopes. In *VMCAI: Verification, Model Checking, and Abstract Interpretation*. Springer, 2011.

19. Alligators for arrays. With T. Henzinger, T. Hottelier and L. Kovacs. In *LPAR: Logic for Programming, Artificial Intelligence and Reasoning*. Springer, 2010.

20. Non-monotonic refinement of control abstraction for concurrent programs. With A. Gupta and C. Popeea. In *ATVA: Automated Technology for Verification and Analysis*. Springer, 2010.

21. Thread-modular counterexample-guided abstraction refinement. With A. Malkis and A. Podelski. In *SAS'2010*.

22. A multi-modal framework for achieving accountability in multi-agent systems. With S. Kramer. In *LIS: Logics in Security*. 2010.

23. Approximation and randomization for quantitative information-flow analysis. With B. Köpf. In *CSF: Computer Security Foundations*. IEEE, 2010.

24. Applying Prolog to develop distributed systems. With N. Lopes, J. Navarro Pérez and A. Singh. In *ICLP: Int. Conf. on Logic Programming*. CUP, 2010.

25. Finding heap-bounds for hardware synthesis. With B. Cook, A. Gupta, S. Magill, J. Simsa, S. Singh and V. Vafeiadis. In *FMCAD'2009*.

26. Cardinality analysis for declarative networking. With J. Navarro Pérez and A. Singh. In *CAV'2009*.

27. InvGen: an efficient invariant generator. With A. Gupta. In *CAV'2009*.

28. Automatic discovery and quantification of information leaks. With M. Backes and B. Köpf. In *S&P: Security and Privacy*. IEEE, 2009.

29. From tests to proofs. With A. Gupta and R. Majumdar. In *TACAS'2009*.

30. Operational semantics for declarative networking. With J. Navarro Pérez. In *PADL: Practical Aspects of Declarative Languages*. Springer, 2009.

31. Verifying liveness for asynchronous programs. With P. Ganty and R. Majumdar. In *POPL'2009*.

32. Heap assumptions on demand. With A. Podelski and T. Wies. In *CAV'2008*.

33. Proving conditional termination. With B. Cook, S. Gulwani, T. Lev-Ami and M. Sagiv. In *CAV'2008*.

34. Proving non-termination. With A. Gupta, T. Henzinger, R. Majumdar and R. Xu. In *POPL'2008*.

35. Precise thread-modular verification. With A. Malkis and A. Podelski. In *SAS'2007*.

36. Path invariants. With D. Beyer, T. Henzinger and R. Majumdar. In *PLDI'2007*.

37. Proving thread termination. With B. Cook and A. Podelski. In *PLDI'2007*.

38. Proving that programs eventually do something good. With B. Cook, A. Gotsman, A. Podelski and M. Vardi. In *POPL'2007*.

39. ARMC: the logical choice for software model checking with abstraction refinement. With A. Podelski. In *PADL'2007*.

40. Constraint solving for interpolation. With V. Sofronie-Stokkermans. In *VMCAI'2007*.

41. Invariant synthesis for combined theories. With D. Beyer, T. Henzinger and R. Majumdar. In *VMCAI'2007*.

42. Thread-modular verification is Cartesian abstract interpretation. With A. Malkis and A. Podelski. In *ICTAC: Int. Colloq. on Theoretical Aspects of Computing*, . Springer, 2006.

43. Model checking Duration Calculus: a practical approach. With R. Meyer and J. Faber. In *ICTAC'2006*.

44. Using predicate abstraction to generate heuristic functions in UPPAAL. With J. Hoffmann, J. Smaus, S. Kupferschmid and A. Podelski. In *MoChArt: Model Checking and Artificial Intelligence*. 2007.

45. Terminator: Beyond safety. With B. Cook and A. Podelski. In *CAV'2006*.

46. Termination proofs for systems code. With B. Cook and A. Podelski. In *PLDI'2006*.

47. Abstraction refinement for termination. With B. Cook and A. Podelski. In *SAS'2005*.

48. Separating fairness and well-foundedness for the analysis of fair discrete systems. With A. Pnueli and A. Podelski. In *TACAS'2005*.

49. Transition predicate abstraction and fair termination. With A. Podelski. In *POPL'2005*.

50. Transition invariants. With A. Podelski. In *LICS: Logic in Computer Science*. IEEE, 2004.

51. A complete method for the synthesis of linear ranking functions. With A. Podelski. In *VMCAI'2004*.

Refereed Workshop Papers

52. CTL+FO verification as constraint solving. With T. Beyene and M. Brockschmidt. In *SPIN: Model Checking of Software*. Springer, 2014.

53. Program verification as Satisfiability Modulo Theories. With K. McMillan and N. Bjørner. In *SMT: Satisfiability Modulo Theory*. 2012.

54. Subsumer-first: steering symbolic reachability analysis. With R. Singh. In *SPIN'2009*.

Refereed Journal Papers

55. From tests to proofs. With A. Gupta and R. Majumdar. Journal version of [29]. In *STTT: J. of Software Tools for Technology Transfer*. Springer, 2013.

56. Constraint solving for interpolation. With V. Sofronie-Stokkermans. Journal version of [40]. In *JSC: J. of Symbolic Computation*. Elsevier, 2010.

57. Proving program termination. With B. Cook and A. Podelski. In *CACM: Communications of ACM*. ACM, 2010.

58. Summarization for termination: no return! With B. Cook and A. Podelski. In *FMSD: Formal Methods in Systems Design*, 35(3). Springer, 2009.

59. Model checking Duration Calculus: a practical approach. With R. Meyer, J. Faber and J. Hoenicke. Journal version of [43]. In *FAC: Formal Aspects of Computing*, 20(4–5). Springer, 2008.

60. Transition predicate abstraction and fair termination. With A. Podelski. Journal version of [49]. In *TOPLAS: Transactions on Programming Languages and Systems. Selected papers of POPL'2005*, 29(3). ACM, 2007.

Editor

61. Verified Software: Theories, Tools, Experiments (VSTTE), 5th International Conference. With E. Cohen. Springer, 2013.

62. Formal Verification of Distributed Algorithms (Dagstuhl Seminar 13141). With B. Charron-Bost, S. Merz, and J. Widder. Dagstuhl Reports, 2013.

63. Verification, Model Checking, and Abstract Interpretation (VMCAI), 13th International Conference. With V. Kuncak. Springer, 2012.

64. Computer Science - Theory and Applications, Fourth International Computer Science Symposium in Russia (CSR). With A. E. Frid, A. Morozov, and K. W. Wagner. Springer, 2009.

Invited Papers

65. Transition invariants and transition predicate abstraction for program termination. With A. Podelski. In *TACAS'2011*.

66. Constraint solving for program verification: Theory and practice by example. In *CAV'2010*.

Technical Reports

67. An Epistemic Perspective on Consistency of Concurrent Computations. With K. von Gleissenthall. arXiv:1305.2295, 2013.

68. Separation Logic Modulo Theories. With J. A. Navarro-Peréz. arXiv:1303.2489, 2013

69. Generalised Interpolation by solving recursion-free Horn clauses. With A. Gupta and C. Popeea. arXiv:1303.7378, 2013

70. Refinement type inference via abstract interpretation. With R. Jhala and R. Majumdar. arXiv:1004.2884, 2010.

71. Path invariants. With D. Beyer, T. Henzinger, and R. Majumdar. Technical Report MTC-REPORT-2006-003, EPFL, 2006.

72. Software model checking of liveness properties via transition invariants. A. Podelski and A. Rybalchenko. Technical Report MPI-I-2003-2-004, Max Planck Institute for Computer Science, 2003.

Theses

73. Temporal Verification with Transition Invariants. A. Rybalchenko. PhD thesis, University of Saarland, June 2005.

74. A model checker based on abstraction refinement. A. Rybalchenko. Master's thesis, University of Saarland, September 2002.

Invited Presentations

Lectures at Conferences

1. (Quantified) Horn Constraint Solving for Program Verification and Synthesis. Keynote at the 30th International Conference on Logic Programming (ICLP). July 2014.

2. Horn Constraint Solving for Program Verification and Synthesis. Tutorial at the 35th annual ACM SIGPLAN conference on Programming Language Design and Implementation. June 2014.

3. Horn Constraint Solving for Program Verification and Synthesis. Invited talk the Workshop on Software Correctness and Reliability. October 2013.

4. Horn Constraint Solving for Program Verification. Invited talk at the Deduktionstreffen. September 2013.

5. Solving Quantified Horn Clauses. Invited talk at the 4th Workshop on Tools for Automatic Program Analysis (TAPAS). May 2013.

6. Solving Quantified Horn Clauses. Invited talk at Alpine Verification Meeting (AVM). May 2013 .

7. Towards automatic synthesis of software verification tools. Invited keynote at the 19th International SPIN Workshop on Model Checking Software (SPIN). July 2012.

8. Towards automatic synthesis of software verification tools. Invited talk at the 13th International ACM SIGPLAN Symposium on Principles and Practice of Declarative Programming (PPDP). July 2011.

9. Constraint Solving for Program Verification: Theory and Practice by Example. Invited talk at the 19th EACSL AnnualConference on Computer Science Logic (CSL). August 2010.

10. Constraint Solving for Program Verification: Theory and Practice by Example. Invited tutorial at the 22nd International Conference on Computer Aided Verification (CAV). July 2010.

11. Automated Methods for Proving Program Termination and Liveness. Invited tutorial at the 11th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC). September 2009.

12. From Tests to Proofs. Keynote talk at the 2nd International Workshop on Invariant Generation (WING). March 2009.

13. Joining Forces for Program Verification. Invited talk at the THEORY Workshop at the 2nd IFIP Working Conference on Verified Software: Theories, Tools, and Experiments (VSTTE), Toronto, Canada. October 2008.

14. Automated Termination Proofs for Systems Code. Invited talk at the 8th Int. Workshop on Verification of Infinite-State Systems (INFINITY). August 2006.

Lectures at Summer and Winter Schools

15. The 3rd Summer School on Formal Techniques. Menlo College, Atherton, CA. May 2013.

16. Program Verification. Ferienakademie Sarntal. September 2011.

17. Proving Program Termination. The 3rd Winter School on Hot Topics in Distributed Computing, La Plagne, France. March 2010.

18. Software Verification. The 7th Int. Summer School on Modelling and Verifying Parallel Processes (MOVEP). June 2008.

Other Invited Lectures

19. Horn Clauses and Verification. Summer School on Computational Logic. July 2014.
20. Synthesizing Software Verifiers from Proof Rules. CREST Seminar, University College London, UK. January 2012.
21. A Proof Rule for Multi-Threaded Programs. Seminar at Harvard University, USA. September 2010.
22. Constrained Environment Assumptions for Multi-Threaded Programs. Seminar at Microsoft Research Cambridge, UK. May 2010.
23. Automatic Discovery and Quantification of Information Leaks. Dagstuhl seminar, Germany. September 2009.
24. From Tests to Proofs. Seminar talk at LIAFA, University Paris 7, France. March 2009; Seminar talk Microsoft Research, Cambridge, UK. June 2008; Seminar "Semantics and Abstract Interpretation" at Ecole Normale Supérieure, Paris, France. Feb 2008.
25. Path Invariants. Seminar "Modélisation et Vérification" at LABRI, Bordeaux, France. February 2008; Seminar on Deduction and Decision Procedures, Dagstuhl, Germany. October 2007.
26. Automation of Abstraction for Software Verification: GIBU Meeting "Embedded Systems", Dagstuhl, Germany. April 2007.
27. Automated Termination Proofs for Systems Code. Alpine Verification Meeting 2006, Monte Verita, Switzerland. May 2006; Seminar on Software Verification: Infinite-State Model Checking and Static Program Analysis. Dagstuhl, Germany. February 2006; Model Checking Day at Theoretical Computer Science Chair, TU München. December 2005. Seminar at LIAFA, University Paris 7, France. November 2005; Fundamental Computer Science F.N.R.S. Contact Group, Montefiore Institute, University of Liège, Belgium. October 2005.
28. Transition Predicate Abstraction and Fair Termination. TRESOR seminar at EPFL, Lausanne, Switzerland. April 2005.
29. Transition invariants. Seminar "Semantics and Abstract Interpretation" at Ecole Normale Supérieure, Paris, France. April 2004.
30. Experiments on automated termination checks for C programs. Seminar on Applied Deductive Verification, Dagstuhl, Germany. November 2003.

Academic Advising

Post-doctoral Researchers

1. Corneliu Popeea. "Modular verification of multi-threaded programs", since September 2008.

Students

1. Klaus von Gleissenthall, PhD student. Thesis "Verification of security properties of functional programs", since Jun 2012.
2. Tewodros Beyene, PhD student. Thesis "Discovery and automation of proof rules for software verification", since Jan 2012.
3. Sergey Grebenshchikov, M.Sc. student. Thesis "Machine learning and data mining for softare verification", since Nov 2011.
4. Ruslán Ledesma Garza, PhD student. Thesis "Verification of temporal properties of functional programs", since Sep 2011.
5. Nuno Lopes. Visiting PhD student (IST Lisbon). Thesis "Software verification and compiler optimizations". Since Aug 2009.

Graduated Students and Post-Docs

1. Juan Navarro Pérez. Post-doctoral researcher "Verification of declarative programs", February 2008–January 2012. Lecturer at Queen Mary University of London.
2. Andreas Wilhelm, Master's thesis "Efficient Verification of Multi-Threaded Programs " (co-advised with C. Popeea), Oct 2013–Dec 2013.
3. Klaus von Gleissenthall, Master's thesis "Epistemic characterization of concurrent computations", Jun 2012.
4. Sergey Grebenshchikov, B.Tech. thesis "Machine learning and data mining for efficient abstraction refinement", Aug 2011–Oct 2011.
5. Ruslán Ledesma Garza, Master's thesis "Automata-theoretic approach Verification of functional programs", Apr 2009–Aug 2011.
6. Ashutosh Gupta, PhD thesis "Constraints for verification", Sep 2007–Jul 2011. Post-doc at IST, Austria.
7. Nuno Lopes, Master's thesis "Design and implementation of a declarative distributed programming system", February–July 2009.
8. Rishabh Singh, Bachelor thesis "Incremental Predicate Abstraction", Sep 2007–Feb 2008. PhD student at MIT.
9. Ashutosh Gupta, Master thesis "Checking (Non)-Termination", Oct 2006–Aug 2007.

Interns

1. Rati Gelashvili, Tbilisi University. "Computation of Hilbert bases" at MPI-SWS, 2009.
2. Vaibhav Goel, IIT Kharagpur. "Lazy Summarization" at MPI-SWS, 2008.
3. Pragyesh, IIT Roorkee. "Distributed Software Model Checking" at MPI-SWS, 2008.
4. Rishabh Singh, IIT Kharagpur."Incremental Predicate Abstraction" at EPFL, 2007.

Doctoral Thesis Reviewer

1. Sascha Böhme, Technische Universität München. May 2012.
2. Eric Koskinen, University of Cambridge. Apr 2011.
3. Nassim Seghir, University of Freiburg. Abstraction Refinement Techniques for Software Model Checking. 2010.

Teaching

1. Model Checking. Graduate course (15 participants). TUM. Summer term 2013.
2. Fundamental algorithms. Undergraduate course (30 participants). TUM. Winger term 2012/13.
3. Model Checking. Graduate course (15 participants). TUM. Summer term 2012.
4. Verification of Concurrent Progams. Seminar (6 participants). TUM. Summer term 2012.
5. Informatics II: Functional programming and introduction to program verification. Undergraduate course (550 participants). TUM. Winter term 2011/12.
6. Pearls of Computer Science 1. Undergraduate course (10 participants). TUM. Winter term 2011/12.
7. Pearls of Computer Science 4. Undergraduate course (10 participants). TUM. Summer term 2011.
8. Model Checking. Graduate course (15 participants). TUM. Summer term 2011.
9. Informatics II: Functional programming and introduction to program verification. Undergraduate course (300 participants). TUM. Winter term 2010/11.
10. Model Checking. Graduate course (15 participants). TUM. Summer term 2010.

11. Proving program termination. PUMA graduate school lecture series. TUM. January 2010.
12. Software Model Checking. Graduate course (10 participants). University of Saarland. Winter term 2007/08.
13. Verification. Tutorials for graduate course. University of Saarland. Summer term 2003.
14. Programming I. Tutorials for undergraduate course. University of Saarland. Winter term 2003/04.

Professional activities

Program Chair

1. The 5th International Conference on Verified Software: Theories, Tools, Experiments (VSTTE), 2013. Co-chair.
2. The 13th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI), 2012. Co-chair.
3. The 4th International Symposium Computer Science Russia (CSR), 2009. Application track chair.

Member of Program Committees

4. The 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), 2014.
5. The 14th International Conference on Runtime Verification (RV), 2014.
6. The 17th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS), 2014.
7. The 3rd Workshop On GRAPH Inspection and Traversal Engineering (GRAPHITE), 2014.
8. The 9th International Symposium on Trustworthy Global Computing (TGC), 2014.
9. The 35th annual ACM SIGPLAN conference on Programming Language Design and Implementation (PLDI), external review committee, 2014.
10. The 9th Ershov Informatics Conference (PSI), 2014.
11. The 28th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2013.
12. The 4th Workshop on Program Semantics, Specification & Verification: Theory and Applications (PSSV), 2013.
13. The 11th Internation Symposium in Software Testing and Analysis (ISSTA), 2013.
14. The 22nd International Symposium on Logic-Based Program Synthesis and Transformation (LOPSTR), 2012.
15. The 18th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), 2012.
16. The 39th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL), external review committee, 2012.
17. The 9th International Symposium on Automated Technology for Verification and Analysis (ATVA), 2011.
18. The 7th Annual Doctoral Workshop on Mathematical and Engineering Methods in Computer Science (MEMICS), 2011.
19. The 23rd International Conference on Computer Aided Verification (CAV), 2011.
20. ACM SIGPLAN Symposium on Programming Language Design and Implementation (PLDI), external review committee, 2011
21. The 12th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI), 2011

22. The 20th European Symposium on Programming (ESOP), 2011.
23. The 9th Workshop on Specification and Verification of Component-Based Systems (SAVCBS), 2010
24. The 3rd International Workshop on Invariant Generation (WING), 2010.
25. The 16th International Conference on Logic for Programming Artificial Intelligence and Reasoning (LPAR), 2010.
26. The 9th ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering (PASTE), 2010.
27. The 17th International Static Analysis Symposium (SAS), 2010.
28. The 3rd International Conference on Verified Software: Theories, Tools and Experiments (VSTTE), 2010.
29. The 37th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages (POPL), 2010.
30. The Fourth ACM SIGPLAN Workshop on Programming Languages meet Program Verification (PLPV), 2010.
31. The 14th International Workshop on Formal Methods for Industrial Critical Systems (FMICS), 2009.
32. The 21st International Conference on Computer-Aided Verification (CAV), 2009.
33. The Symposium on Automatic Program Verification (APV) in Argentina, 2009.
34. The 10th International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI), 2009.
35. The 15th International Conferences on Logic for Programming, Artificial Intelligence and Reasoning (LPAR), 2008.
36. The 9th International ACM Symposium on Principles and Practice of Declarative Programming (PPDP), 2007.

Refereeing

Journals Acta Informatica, Formal Methods Letters, Infomation Processing Letters, Logical Methods in Computer Science, Transactions on Programming Languages and Systems, Theoretical Computer Science, Information Processing Letters.

Conferences Concurrency Theory; Conference on Automated Deduction; Computer-Aided Verification; Foundations of Software Science and Computation Structures; Foundations of Software Technology and Theoretical Computer Science; Logic in Computer Science; Logic-based Program Synthesis and Transformation; Principles of Programming Languages; Static Analysis Symposium; Tools and Algorithms for the Construction and Analysis of Systems; Verification, Model Checking and Abstract Interpretation.