Publications: [4], [1], [14], [13], [12], [15], [7], [17], [2], [16], [6], [8], [11], [10], [9], [5], [3].

References

[1] Michael D. Bond, Nicholas Nethercote, Stephen W. Kent, Samuel Z. Guyer, and Kathryn S. McKinley. Tracking bad apples: Reporting the origin of null and undefined value errors. In *Proceedings of the ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA 2007)*, pages 405–422, Montreal, Canada, October 2007.

[2] Kevin Bush, Mark Gebhart, Eric Wei, Natalie Yudin, Bertrand Maher, Nicholas Nethercote, Doug Burger, and Stephen W. Keckler. Evaluation and optimization of signal processing kernels on the TRIPS architecture. In *Proceedings of the 4th Annual Workshop on Optimizations for DSP and Embedded Systems (ODES-4)*, New York, New York, USA, March 2006.

[3] María García de la Banda, David Jeffery, Kim Marriott, Nicholas Nethercote, Peter J. Stuckey, and Christian Holzbaur. Building constraint solvers with HAL. In *Proceedings of the 17th International Conference on Logic Programming (ICLP'01)*, Paphos, Cyprus, November 2001.

[4] Kim Marriott, Nicholas Nethercote, Reza Rafeh, Peter J. Stuckey, Maria Garcia de la Banda, and Mark Wallace. The design of the Zinc modelling language. *Constraints, Special Issue on Abstraction and Automation in Constraint Modelling*, 13(3), 2008.

[5] Nicholas Nethercote. The analysis framework of HAL. Master's thesis, Department of Computer Science and Software Engineering, University of Melbourne, Australia, April 2002.

[6] Nicholas Nethercote. *Dynamic Binary Analysis and Instrumentation*. PhD thesis, Computer Laboratory, University of Cambridge, United Kingdom, November 2004.

[7] Nicholas Nethercote, Douglas C. Burger, and Kathryn S. McKinley. Convergent compilation applied to loop unrolling. *Transactions on High-Performance Embedded Architectures and Compilers, Special Issue: Future Directions in Embedded Systems Compilation*, 1:140–158, September 2006.

[8] Nicholas Nethercote and Jeremy Fitzhardinge. Bounds-checking entire programs without recompiling. In *Informal Proceedings of the Second Workshop on Semantics, Program Analysis, and Computing Environments for Memory Management (SPACE 2004)*, Venice, Italy, January 2004.

[9] Nicholas Nethercote and Alan Mycroft. The cache behaviour of large lazy functional programs on stock hardware. In *Proceedings of the ACM SIGPLAN Workshop on Memory System Performance (MSP 2002)*, pages 44–55, Berlin, Germany, July 2002.

[10] Nicholas Nethercote and Alan Mycroft. Redux: A dynamic dataflow tracer. *Electronic Notes in Theoretical Computer Science*, 89(2), 2003.

[11] Nicholas Nethercote and Julian Seward. Valgrind: A program supervision framework. *Electronic Notes in Theoretical Computer Science*, 89(2), 2003.

[12] Nicholas Nethercote and Julian Seward. How to shadow every byte of memory used by a program. In *Proceedings of the Third International ACM SIGPLAN/SIGOPS Conference on Virtual Execution Environments (VEE 2007)*, pages 65–74, San Diego, California, USA, June 2007.

[13] Nicholas Nethercote and Julian Seward. Valgrind: A framework for heavy-weight dynamic binary instrumentation. In *Proceedings of ACM SIGPLAN 2007 Conference on Programming Language Design and Implementation (PLDI 2007)*, pages 89–100, San Diego, California, USA, June 2007.

[14] Nicholas Nethercote, Peter J. Stuckey, Ralph Becket, Sebastian Brand, Gregory J. Duck, and Guido Tack. MiniZinc: Towards a standard CP modelling language. In *Proceedings of the 13th International Conference on Principles and Practice of Constraint Programming (CP2007)*, Providence, Rhode Island, USA, September 2007.

[15] Nicholas Nethercote, Robert Walsh, and Jeremy Fitzhardinge. Building workload characterization tools with Valgrind. In *Invited tutorial, IEEE International Symposium on Workload Characterization (IISWC 2006)*, San José, California, USA, October 2006.

[16] Julian Seward and Nicholas Nethercote. Using Valgrind to detect undefined value errors with bit-precision. In *Proceedings of the USENIX'05 Annual Technical Conference*, Anaheim, California, USA, April 2005.

[17] Aaron Smith, Jim Burrill, Jon Gibson, Bertrand Maher, Nicholas Nethercote, Bill Yoder, Doug Burger, and Kathryn S. McKinley. Compiling for EDGE architectures. In *Proceedings of the 4th International Symposium on Code Generation and Optimization (CGO-4)*, New York, New York, USA, March 2006.