

Annotated bibliography

March 16, 2020

References

- [ACPP91] Martín Abadi, Luca Cardelli, Benjamin C. Pierce, and Gordon D. Plotkin. Dynamic typing in a statically typed language. *ACM Trans. Program. Lang. Syst.*, 13(2):237–268, 1991.

Shows how existential types can be used for dynamic typing in otherwise statically typed languages. Values of dynamic type, i.e., existentially typed values, can be inspected with a "typecase" construct. The stack pointer type and the type pattern matching feature of Zee originate with this paper.
- [AKS10] Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors. *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, October 4-8, 2010*. ACM, 2010.
- [Aug85] Lennart Augustsson. Compiling pattern matching. In Jouannaud [Jou85], pages 368–381.
- [AW03] Amal J. Ahmed and David Walker. The logical approach to stack typing. In Shao and Lee [SL03], pages 74–85.
- [AZM10] Aslan Askarov, Danfeng Zhang, and Andrew C. Myers. Predictive black-box mitigation of timing channels. In Al-Shaer et al. [AKS10], pages 297–307.
- [BJ96] Hans-Juergen Boehm and Guy L. Steele Jr., editors. *Conference Record of POPL'96: The 23rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Papers Presented at the Symposium, St. Petersburg Beach, Florida, USA, January 21-24, 1996*. ACM Press, 1996.
- [Car84] Luca Cardelli. Compiling a functional language. In *Proceedings of the 1984 ACM Conference on LISP and Functional Programming, LFP 1984, August 5-8, 1984, Austin, Texas, USA* [DBL84], pages 208–217.

- [CDS11] Yan Chen, George Danezis, and Vitaly Shmatikov, editors. *Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS 2011, Chicago, Illinois, USA, October 17-21, 2011*. ACM, 2011.
- [CL95] Ron K. Cytron and Peter Lee, editors. *Conference Record of POPL’95: 22nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, San Francisco, California, USA, January 23-25, 1995*. ACM Press, 1995.
- [DBL84] *Proceedings of the 1984 ACM Conference on LISP and Functional Programming, LFP 1984, August 5-8, 1984, Austin, Texas, USA*. ACM, 1984.
- [DBL17] *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*. IEEE Computer Society, 2017.
- [DBL19] *32nd IEEE Computer Security Foundations Symposium, CSF 2019, Hoboken, NJ, USA, June 25-28, 2019*. IEEE, 2019.
- [DD77] Dorothy E. Denning and Peter J. Denning. Certification of programs for secure information flow. *Commun. ACM*, 20(7):504–513, 1977.
- [Ell02] Carla Schlatter Ellis, editor. *Proceedings of the General Track: 2002 USENIX Annual Technical Conference, June 10-15, 2002, Monterey, California, USA*. USENIX, 2002.
- [GMJ⁺02] Dan Grossman, J. Gregory Morrisett, Trevor Jim, Michael W. Hicks, Yanling Wang, and James Cheney. Region-based memory management in cyclone. In Knoop and Hendren [KH02], pages 282–293.

Mentions that "existential and universal polymorphism, conspire to allow pointers to escape the scope of their regions, just as closures allow pointers to escape [...]"
- [HM95] Robert Harper and J. Gregory Morrisett. Compiling polymorphism using intensional type analysis. In Cytron and Lee [CL95], pages 130–141.
- [JMG⁺02] Trevor Jim, J. Gregory Morrisett, Dan Grossman, Michael W. Hicks, James Cheney, and Yanling Wang. Cyclone: A safe dialect of C. In Ellis [Ell02], pages 275–288.
- [Jou85] Jean-Pierre Jouannaud, editor. *Functional Programming Languages and Computer Architecture, FPCA 1985, Nancy, France, September 16-19, 1985, Proceedings*, volume 201 of *Lecture Notes in Computer Science*. Springer, 1985.

- [KH02] Jens Knoop and Laurie J. Hendren, editors. *Proceedings of the 2002 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), Berlin, Germany, June 17-19, 2002*. ACM, 2002.
 - [LH83] Henry Lieberman and Carl Hewitt. A real-time garbage collector based on the lifetimes of objects. *Commun. ACM*, 26(6):419–429, 1983.
 - [LO98] Xavier Leroy and Atsushi Ohori, editors. *Types in Compilation, Second International Workshop, TIC '98, Kyoto, Japan, March 25-27, 1998, Proceedings*, volume 1473 of *Lecture Notes in Computer Science*. Springer, 1998.
 - [Mar08] Luc Maranget. Compiling pattern matching to good decision trees. In Sumii [Sum08], pages 35–46.
 - [MC98] David B. MacQueen and Luca Cardelli, editors. *POPL '98, Proceedings of the 25th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, San Diego, CA, USA, January 19-21, 1998*. ACM, 1998.
 - [MCG⁺99] Greg Morrisett, Karl Crary, Neal Glew, Dan Grossman, Richard Samuels, Frederick Smith, David Walker, Stephanie Weirich, and Steve Zdancewic. TALx86: A realistic typed assembly language. In *1999 ACM SIGPLAN Workshop on Compiler Support for System Software, Atlanta, GA, May 1999*, pages 25–35, 1999.
 - [MCGW98] J. Gregory Morrisett, Karl Crary, Neal Glew, and David Walker. Stack-based typed assembly language. In Leroy and Ohori [LO98], pages 28–52.
 - [MH97] J. Gregory Morrisett and Robert Harper. Typed closure conversion for recursively-defined functions. *Electron. Notes Theor. Comput. Sci.*, 10:230–241, 1997.
 - [MMH96] Yasuhiko Minamide, J. Gregory Morrisett, and Robert Harper. Typed closure conversion. In Boehm and Jr. [BJ96], pages 271–283.
- Shows how existential types can encode closures
- [MP88] John C. Mitchell and Gordon D. Plotkin. Abstract types have existential type. *ACM Trans. Program. Lang. Syst.*, 10(3):470–502, 1988.
 - [MWCG98] J. Gregory Morrisett, David Walker, Karl Crary, and Neal Glew. From system F to typed assembly language. In MacQueen and Cardelli [MC98], pages 85–97.

- [NCH⁺05] George C. Necula, Jeremy Condit, Matthew Harren, Scott McPeak, and Westley Weimer. Ccured: type-safe retrofitting of legacy software. *ACM Trans. Program. Lang. Syst.*, 27(3):477–526, 2005.
- [NZMZ10] Santosh Nagarakatte, Jianzhou Zhao, Milo M. K. Martin, and Steve Zdancewic. CETS: compiler enforced temporal safety for C. In Vitek and Lea [VL10], pages 31–40.
- [ÖED15] Özcan Öztürk, Kemal Ebcioglu, and Sandhya Dwarkadas, editors. *Proceedings of the Twentieth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '15, Istanbul, Turkey, March 14-18, 2015*. ACM, 2015.
- [PA17] Mathias V. Pedersen and Aslan Askarov. From trash to treasure: Timing-sensitive garbage collection. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017* [DBL17], pages 693–709.

Shows how automatic memory management can be leak information thorough timing. Acts as the main motivation behind the creation of the Zee language.
- [PA19] Mathias Vorreiter Pedersen and Aslan Askarov. Static enforcement of security in runtime systems. In *32nd IEEE Computer Security Foundations Symposium, CSF 2019, Hoboken, NJ, USA, June 25-28, 2019* [DBL19], pages 335–350.

Motivates and defines the Zee language along with case studies showing how Zee can be used for implementing a garbage collector and a thread scheduler
- [Ped19] Mathias V. Pedersen. *Enforcement of Timing-Sensitive Security Policies in Runtime Systems*. PhD thesis, Aarhus University, 10 2019.
- [Pey87] Simon L. Peyton Jones. *The Implementation of Functional Programming Languages*. Prentice-Hall, 1987.
- [Pey92] Simon L. Peyton Jones. Implementing lazy functional languages on stock hardware: The spineless tagless g-machine. *J. Funct. Program.*, 2(2):127–202, 1992.
- [Pie02] Benjamin C. Pierce. *Types and programming languages*. MIT Press, 2002.

Explains various type-related concepts. We mainly refer to this for its introduction to existential, universal, and recursive types.

- [Pie05] Benjamin C. Pierce. *Advanced Topics in Types and Programming Languages*. MIT Press, 2005.
- [RH84] William E. Riddle and Peter B. Henderson, editors. *Proceedings of the ACM SIGSOFT/SIGPLAN Software Engineering Symposium on Practical Software Development Environments, Pittsburgh, Pennsylvania, USA, April 23-25, 1984*. ACM, 1984.
- [SL03] Zhong Shao and Peter Lee, editors. *Proceedings of TLDI'03: 2003 ACM SIGPLAN International Workshop on Types in Languages Design and Implementation, New Orleans, Louisiana, USA, January 18, 2003*. ACM, 2003.
- [SM03] Andrei Sabelfeld and Andrew C. Myers. Language-based information-flow security. *IEEE Journal on Selected Areas in Communications*, 21(1):5–19, 2003.

Provides background for language-based information flow security. Surveys recent (contemporary to 2003) work in language-based information flow security. Provides an immense list of pointers into information-flow literature.
- [Sum08] Eijiro Sumii, editor. *Proceedings of the ACM Workshop on ML, 2008, Victoria, BC, Canada, September 21, 2008*. ACM, 2008.
- [TT97] Mads Tofte and Jean-Pierre Talpin. Region-based memory management. *Inf. Comput.*, 132(2):109–176, 1997.
- [TZ07] Stephen Tse and Steve Zdancewic. Run-time principals in information-flow type systems. *ACM Trans. Program. Lang. Syst.*, 30(1):6, 2007.
- [Ung84] David M. Ungar. Generation scavenging: A non-disruptive high performance storage reclamation algorithm. In Riddle and Henderson [RH84], pages 157–167.
- [VIS96] Dennis M. Volpano, Cynthia E. Irvine, and Geoffrey Smith. A sound type system for secure flow analysis. *Journal of Computer Security*, 4(2/3):167–188, 1996.

Establishes a type system based on Denning’s approach to information flow analysis using the lattice model. Programs that are well-typed in the type system have a property of non-interference. This is considered to be the first type system based on Denning’s approach.
- [VL10] Jan Vitek and Doug Lea, editors. *Proceedings of the 9th International Symposium on Memory Management, ISMM 2010, Toronto, Ontario, Canada, June 5-6, 2010*. ACM, 2010.

- [VLT12] Jan Vitek, Haibo Lin, and Frank Tip, editors. *ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI '12, Beijing, China - June 11 - 16, 2012*. ACM, 2012.
- [Wei69] Joseph Weizenbaum. Recovery of reentrant list structures in SLIP. *Commun. ACM*, 12(7):370–372, 1969.
- [ZAM11] Danfeng Zhang, Aslan Askarov, and Andrew C. Myers. Predictive mitigation of timing channels in interactive systems. In Chen et al. [CDS11], pages 563–574.
- [ZAM12] Danfeng Zhang, Aslan Askarov, and Andrew C. Myers. Language-based control and mitigation of timing channels. In Vitek et al. [VLT12], pages 99–110.
- [ZWSM15] Danfeng Zhang, Yao Wang, G. Edward Suh, and Andrew C. Myers. A hardware design language for timing-sensitive information-flow security. In Öztürk et al. [ÖED15], pages 503–516.