Bibliography

[Amm92] Z. Ammarguellat. A control-flow normalization algorithm and its complexity. *IEEE Transactions on Software Engineering*, 18(3):237–251, 1992.

[ASU86] A. V. Aho, Ravi Sethi, and Jeffrey D. Ullman. *Compilers — Principles, Techniques, and Tools.* Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

[Bak77] B. Baker. An algorithm for structuring flowgraphs. *Journal of the ACM*, 24(1):98–120, January 1977.

[Ban76] U. Banerjee. Data dependence in ordinary programs. Master's thesis, University of Illinois at Urbana-Champaign, 1976.

[Ban79] J. P. Banning. An efficient way to find the side effects of procedure calls and the aliases of variables. In *Conference Record of the Sixth Annual ACM Symposium on Principles of Programming Languages*, pages 29–41, San Antonio, Texas, January 29–31, 1979. ACM SIGACT and SIGPLAN.

[Bar78] J. M. Barth. A practical interprocedural data flow analysis algorithm. *Communications of the ACM*, 21(9):724–736, September 1978.

[BC86] M. Burke and R. Cytron. Interprocedural dependence analysis and parallelization. In *Proceedings of the SIGPLAN '86 Symposium on Compiler Construction*, pages 162–175, Palo Alto, California, June 25–27, 1986. ACM SIGPLAN. *SIGPLAN Notices,* 21(7), July 1986.

[CBC93] J. D. Choi, M. G. Burke, and P. Carini. Efficient flow-sensitive interprocedural computation of pointer-induced aliases and side effects. In *Conference Record of the Twentieth Annual ACM Symposium on Principles of Programming Languages*, pages 232–245, January 1993.

[CHK92] K. D. Cooper, M. W. Hall, and K. Kennedy. Procedure cloning. In *Proceedings of the IEEE'92 International Conference on Computer Languages*, pages 96–105, April 1992.

[CK89] K. D. Cooper and K. Kennedy. Fast interprocedural alias analysis. In *Conference Record of the Sixteenth Annual ACM Symposium on Principles of Programming Languages*, pages 49–59, Austin, Texas, January 11–13, 1989. ACM SIGACT and SIGPLAN.

[Coo85] K. D. Cooper. Analyzing aliases of reference formal parameters. In *Conference Record of the Twelfth Annual ACM Symposium on Principles of Programming Languages*, pages 281–290, New Orleans, Louisiana, January 13–16, 1985. ACM SIGACT and SIGPLAN.

[Coo89] B. G. Cooper. Ambitious data flow analysis of procedural programs. Master's thesis, University of Minnesota, May 1989.

[Cou86] D. S. Coutant. Retargetable high-level alias analysis. In *Conference Record of the Thirteenth Annual ACM Symposium on Principles of Programming Languages*, pages 110–118, St. Petersburg Beach, Florida, January 13–15, 1986. ACM SIGACT and SIGPLAN.

[CR82] A. L. Chow and A. Rudmik. The design of a data flow analyzer. In *Proceedings of the SIGPLAN '82 Symposium on Compiler Construction*, pages 106–113, Boston, Massachusetts, June 23–25, 1982. ACM SIGPLAN. *SIGPLAN Notices,* 17(6), June 1982.

[CWZ90] D. R. Chase, M. Wegman, and F. K. Zadeck. Analysis of pointers and structures. In *Proceedings of the SIGPLAN '90 Conference on Programming Language Design and Implementation*, pages 296–310, White Plains, New York, June 20–22, 1990. ACM SIGPLAN. *SIGPLAN Notices,* 25(6), June 1990.

[Deu92] A. Deutsch. A storeless model of aliasing and its abstractions using finite representations of right-regular equivalence relations. In *Proceedings of the IEEE'92 International Conference on Computer Languages*, April 1992.

[Don93] C. M. Donawa. Overview of the McCAT development environment. McCAT Design Note 13, School of Computer Science, McGill University, Montréal, Québec, August 1993.

[Ero93] A. Erosa. A program structurer for the McCAT C compiler. Master's thesis, McGill University, expected December 1993.

[Ghi92] R. Ghiya. Interprocedural analysis in the presence of function pointers. Technical report, McGill University, December 1992. ACAPS Technical Memo 62.

[Ghi93a] R. Ghiya. Practical techniques for interprocedural heap analysis. Report for 308-622B project with Prof. G. Gao, 1993.

[Ghi93b] R. Ghiya. Practical techniques for interprocedural heap analysis. Technical report, McGill University, May 1993. ACAPS Technical Memo 46.

[HDE+92] L. J. Hendren, C. Donawa, M. Emami, G. R. Gao, Justiani, and B. Sridharan. Designing the McCAT compiler based on a family of structured intermediate representations. In *Fifth Workshop on Languages and Compilers for Parallel Computing*, pages 261–275. Yale University, 1992. Also available as ACAPS Technical Memo No. 46.

[HEGV93] L. Hendren, M. Emami, R. Ghiya, and C. Verbrugge. A practical context-sensitive interprocedural analysis framework for C compilers. ACAPS Technical Memo 72, School of Computer Science, McGill University, Montréal, Québec, July 1993.

[Hen90] L. J. Hendren. *Parallelizing Programs with Recursive Data Structures.* PhD thesis, Cornell University, January 1990.

[HGS92] L. Hendren, G. R. Gao, and V. C. Sreedhar. ALPHA: A family of structured intermediate representations for a parallelizing C compiler. ACAPS Technical Memo 49, School of Computer Science, McGill University, Montréal, Québec, November 1992.

[HHN92a] L. J. Hendren, J. Hummel, and A. Nicolau. Abstractions for recursive pointer data structures: Improving the analysis and transformation of imperative programs. In *Proceedings of the SIGPLAN '92 Conference on Programming Language Design and Implementation*, pages 249–260, San Francisco, California, June 17–19, 1992. ACM SIGPLAN. *SIGPLAN Notices,* 27(7), July 1992.

[HHN92b] J. Hummel, L. J. Hendren, and A. Nicolau. Applying an abstract data structure description approach to parallelizing scientific pointer programs. In *Proceedings of the 1992 International Conference on Parallel Processing*, volume II, pages 100–104, St. Charles, Illinois, August 17–21, 1992.

[HN89] L. J. Hendren and A. Nicolau. Interference analysis tools for parallelizing programs with recursive data structures. In *Conference Proceedings, 1989 International Conference on Supercomputing*, pages 205–214, Crete, Greece, June 5–9, 1989. ACM.

[HN90] L. J. Hendren and A. Nicolau. Parallelizing programs with recursive data structures. *IEEE Transactions on Parallel and Distributed Systems*, 1(1):35–47, January 1990.

[HS92] L. J. Hendren and B. Sridharan. The SIMPLE AST - McCAT compiler. McCAT Design Note 2, School of Computer Science, McGill University, Montréal, Québec, July 1992.

[JM81] N. D. Jones and S. Muchnick. Flow analysis and optimization of LISP-like structures. In *Program Flow Analysis, Theory, and Applications*, pages 102–131. Prentice-Hall, 1981. Chapter 4.

[JM82] N. D. Jones and S. S. Muchnick. A flexible approach to interprocedural data flow analysis and programs with recursive data structures. In *Conference Record of the Ninth Annual ACM Symposium on Principles of Programming Languages*, pages 66–74, Albuquerque, New Mexico, January 25–27, 1982. ACM SIGACT and SIGPLAN.

[Jus93] Justiani. An array dependence framework for the McCAT C compiler. Master's thesis, McGill University, expected December 1993.

[Lan92] W. A. Landi. *Interprocedural Aliasing in the Presence of Pointers*. PhD thesis, Rutgers University, 1992.

[LH88] J. R. Larus and P. N. Hilfinger. Detecting conflicts between structure accesses. In *Proceedings of the SIGPLAN '88 Conference on Programming Language Design and Implementation*, pages 21–34, Atlanta, Georgia, June 22–24, 1988. ACM SIGPLAN. *SIGPLAN Notices,* 23(7), July 1988.

[LH93] C. Verbrugge L. Hendren. Generalized constant propagation. ACAPS Technical Memo 71, School of Computer Science, McGill University, Montréal, Québec, expected September 1993.

[LR92] W. Landi and B. G. Ryder. A safe approximate algorithm for interprocedural pointer aliasing. In *Proceedings of the SIGPLAN '92 Conference on Programming Language Design and Implementation*, pages 235–248, San Francisco, California, June 17–19, 1992. ACM SIGPLAN. *SIGPLAN Notices,* 27(7), July 1992.

[Mye81] E. W. Myers. A precise inter-procedural data flow algorithm. In *Conference Record of the Eighth Annual ACM Symposium on Principles of Programming Languages*, pages 219–230, Williamsburg, Virginia, January 26–28, 1981. ACM SIGACT and SIGPLAN.

[Sri92] B. Sridharan. An analysis framework for the McCAT compiler. Master's thesis, McGill University, Montréal, Québec, September 1992.

[Sta90] R. M. Stallman. Using and porting the GNU CC. Technical report, Free Software Foundation, Cambridge, Massachusetts, 1990.

[Wei80] W. E. Weihl. Interprocedural data flow analysis in the presence of pointers, procedure variables and label variables (extended abstract). In *Conference*

Record of the Seventh Annual ACM Symposium on Principles of Programming Languages, pages 83–94, Las Vegas, Nevada, January 28–30, 1980. ACM SIGACT and SIGPLAN.

[WO75] M. H. Williams and H.L. Ossher. Conversion of unstructured flow diagrams to structured form. *Comput. J.*, 21(2), 1975.

[Wol89] M. J. Wolfe. *Optimizing Supercompilers for Supercomputers.* Pitman, London and MIT Press, Cambridge, Massachusetts, 1989. In Research Monographs in Parallel and Distributed Computing; revised version of the author's Ph.D. dissertation published as Report No. UIUCDCS-R-82-1105, University of Illinois at Urbana-Champaign, 1982.

[ZC90] H. Zima and B. Chapman. *Supercompilers for Parallel and Vector Computers.* ACM Press, New York, New York, 1990.