Ελληνική και ξενόγλωσση βιβλιογραφία για το μάθημα "Μεταγλωττιστές ΙΙ"

Νικόλαος Καββαδίας
nkavv@uop.gr

10 Νοεμβρίου 2009

[1] A. V. Aho, R. Sethi, and J. D. Ullman, Μεταγλωττιστές: Αρχές, Τεχνικές και Εργαλεία, με την επιμέλεια των: Άγγελος Σπ. Βώρος και Νικόλαος Σπ. Βώρος και Κων/νος Γ. Μασσέλος, Ed. Reading, MA: Εκδόσεις Νέων Τεχνολογιών, 2008. [Online]. Available: http://dragonbook.stanford.edu

[2] D. F. Bacon, S. L. Graham, and O. J. Sharp, "Compiler transformations for high-performance computing," ACM Computing Surveys, vol. 26, no. 4, pp. 345–420, December 1994.

[3] V. H. Allan, R. B. Jones, R. M. Lee, and S. J. Allan, "Software pipelining," ACM Computing Surveys, vol. 27, no. 3, pp. 367–432, September 1995.

[4] C. W. Fraser and D. Hanson, "A retargetable compiler for ANSI C," ACM SIGPLAN Notices, vol. 26, no. 10, pp. 29–43, 1991.

[5] D. Spinellis, "Declarative peephole optimization using string pattern matching," ACM SIGPLAN Notices, vol. 34, no. 2, pp. 47–51, February 1999.

[6] R. Leupers, O. Whalen, M. Hahenauer, T. Kogel, and P. Marwedel, "An executable intermediate representation for retargetable compilation and high-level code optimization," in Proceedings of the Third International Workshop on Systems, Architectures, Modeling, and Simulation (SAMOS 2003), Samos, Greece, July 21-23 2003, pp. 120–125.

[7] B. W. Kernighan and D. M. Ritchie, The C Programming Language, 2nd ed. Prentice Hall PTR, One Lake Street, Upper Saddle River, New Jersey 07458, USA: Prentice-Hall, 1988.

[8] T. Æ. Mogensen, Basics of Compiler Design: Extended Edition. Department of Computer Science, University of Copenhagen, Universitetsparken 1, DK-2100 Copenhagen, Denmark: Author's copyright, May 28 2009. [Online]. Available: http://www.diku.dk/hjemmesider/ansatte/torbenm/Basics/basics_lulu.pdf

[9] Παναγιώτης Πιντέλας and Παναγιώτης Αλεφραγκής, Μεταγλωττιστές (Πρακτική Εξάσκηση σε Θέματα Λογισμικού: Τόμος Ά). Ελληνικό Ανοικτό Πανεπιστήμιο, 2003. [Online]. Available: http://www.pli.eap.gr/pdf/PLH40/PLH40-1/pintellas2.pdf

[10] R. Leupers, "Compiler design issues for embedded processors," IEEE Design and Test of Computers, vol. 19, no. 3, pp. 51–58, July-August 2002.

[11] A. V. Aho, M. Ganapathi, and S. W. K. Tjiang, "Code generation using tree matching and dynamic programming," ACM Transactions on Programming Languages and Systems, vol. 11, no. 4, pp. 491–516, October 1989.

[12] G. J. Chaitin, "Register allocation and spilling via graph coloring," in Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation, June 1982, pp. 98–105.

[13] M. Poletto and V. Sarkar, "Linear scan register allocation," ACM Transactions on Programming Languages and Systems, vol. 21, no. 5, pp. 895–913, September 1999.

[14] R. Cytron, J. Ferrante, B. K. Rosen, M. N. Wegman, and F. K. Zadeck, "Efficiently computing static single assignment form and the control dependence graph," ACM Transactions on Programming Languages and Systems, vol. 13, no. 4, pp. 451–490, October 1991.

[15] J. Aycock and N. Horspool, "Simple generation of static single assignment form," in Proceedings of the 9th International Conference in Compiler Construction, March 2000, pp. 110–124.

[16] C. W. Fraser, D. R. Hanson, and T. A. Proebstring, "Engineering a simple, efficient code-generator generator," ACM Letters on Programming Languages and Systems, vol. 1, no. 3, pp. 213–226, September 1992.

[17] J. R. Allen, K. Kennedy, C. Porterfield, and J. Warren, "Conversion of control dependence to data dependence," in Proceedings of the Tenth ACM Symposium on the Principles of Programming Languages, January 1983, pp. 177–189.

[18] M. E. Conway, "Proposal for an UNCOL," Communications of the ACM, vol. 1, no. 10, pp. 5–8, 1958.

[19] W. M. McKeeman, "Peephole optimization," Communications of the ACM, vol. 8, no. 7, pp. 443–444, July 1965.

[20] M. A. Ertl, "Optimal code selection in DAGs," in Proceedings of the 26th ACM Symposium on the Principles of Programming Languages, San Antonio, Texas, USA, January 1999, pp. 242–249.

[21] H. Massalin, "Superoptimizer: A look at the smallest program," in Proceedings of the Second International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS II), January 1987, pp. 122–126.

[22] V. Bala and N. Rubin, "Efficient instruction scheduling using finite state automata," in Proceedings of the 28th Annual International Symposium on Microarchitecture, November 1995, pp. 46–56.

[23] D. Novillo, "GCC - an architectural overview, current status, and future directions," in Linux Symposium, vol. 2, 2006, pp. 185–200.

[24] A. W. Appel, "SSA is functional programming," vol. 33, no. 4, pp. 17–20, April 1998.

[25] C. Clark, "ASTs for optimizing compilers," vol. 36, no. 9, pp. 25–30, September 2001.

[26] V. Lefèvre, "Multiplication by an integer constant," INRIA Institute, Technical report No. 4192, May 2001.

[27] SUIF. [Online]. Available: http://suif.stanford.edu/suif/suif2/

[28] Machine-SUIF research compiler. [Online]. Available: http://www.eecs.harvard.edu/hube/research/machsuif.html

[29] "The Trimaran compiler research infrastructure for instruction level parallelism," Trimaran Consortium. [Online]. Available: http://www.trimaran.org

[30] PCC: The portable C compiler. [Online]. Available: http://pcc.ludd.ltu.se

[31] A COmpiler INfra-Structure home page. [Online]. Available: http://www.coins-project.org

[32] The GNU compiler collection homepage. [Online]. Available: http://gcc.gnu.org

[33] LANCE C compiler. [Online]. Available: http://www.lancecompiler.com

[34] LCC retargetable C compiler. [Online]. Available: http://www.cs.princeton.edu/software/lcc/

[35] The LLVM compiler infrastructure. [Online]. Available: http://www.llvm.org

[36] The Phoenix compiler infrastructure home page. [Online]. Available: http://research.microsoft.com/phoenix/

[37] Advanced compiler system for education. [Online]. Available: http://home.dei.polimi.it/agosta/doku.php?id\=teaching:alari

[38] bison homepage. [Online]. Available: http://www.gnu.org/software/bison/bison.html

[39] flex homepage. [Online]. Available: http://www.gnu.org/software/flex/flex.html

[40] IBURG, a tree parser generator. [Online]. Available: http://www.cs.princeton.edu/software/iburg/

[41] JBURG home page. [Online]. Available: http://jburg.sourceforge.net

[42] A simple retargetable peephole optimizer. [Online]. Available: ftp://ftp.cs.princeton.edu/pub/packages/lcc/contrib/copt.shar

[43] The GNU superoptimizer. [Online]. Available: ftp://ftp.gnu.org/pub/gnu/superopt/

[44] The Aha! superoptimizer. [Online]. Available: http://www.hackersdelight.org

[45] Graphviz. [Online]. Available: http://www.research.att.com/sw/tools/graphviz/

[46] VCG website. [Online]. Available: http://rw4.cs.ui-sb.de/~sander/html/gsvcg1.html

[47] Διομήδης Σπινέλλης. Σειρά διαλέξεων: "Computers For All". [Online]. Available: http://www.dmst.aueb.gr/dds/cfa/