A fast algorithm to compute heap memory bounds of Java Card applets

Pham T.-H., Truong A.-H., Truong N.-T., Chin W.-N.

College of Technology, Vietnam National University, 144 Xuan Thuy, Hanoi, Viet Nam; School of Computing, National University of Singapore, 3 Science Drive 2, Singapore 117543, Singapore

Abstract: We present an approach to find upper bounds of heap space for Java Card applets. Our method first transforms an input bytecode stream into a control flow graph (CFG), and then collapses cycles of the CFG to produce a directed acyclic graph (DAG). Based on the DAG, we propose a linear-time algorithm to solve the problem of finding the single-source largest path in it. We also have implemented a prototype tool, tested it on several sample applications, and then compared the bounds found by our tool with the actual heap bounds of the programs. The experiment shows that our tool returns good estimation of heap bounds, runs fast, and has a small memory footprint. ?? 2008 IEEE.

Index Keywords: Clustering algorithms; Computer programming languages; Computer software; Java programming language; Software engineering; Applets; Control Flow graphs; Directed Acyclic graphs; Fast algorithms; Prototype tools; Small memory footprints; Time algorithms; Upper bounds; Formal methods

Year: 2008
Source title: Proceedings - 6th IEEE International Conference on Software Engineering and Formal Methods, SEFM 2008
Art. No.: 4685813
Page : 259-267
Link: Scorpus Link
Correspondence Address: Pham, T.-H.; College of Technology, Vietnam National University, 144 Xuan Thuy, Hanoi, Viet Nam
Sponsors: IEEE Computer Society;Int. Inst. Software Technology of the United Nations Univ.;Formal Methods Europe;University of Cape Town
Conference name: 6th IEEE International Conference on Software Engineering and Formal Methods, SEFM 2008
Conference date: 10 November 2008 through 14 November 2008
Conference location: Cape Town
Conference code: 74875
ISBN: 9.78E+12
DOI: 10.1109/SEFM.2008.30
Language of Original Document: English
Abbreviated Source Title: Proceedings - 6th IEEE International Conference on Software Engineering and Formal Methods, SEFM 2008
Document Type: Conference Paper
Source: Scopus

Authors with affiliations:

1. Pham, T.-H., College of Technology, Vietnam National University, 144 Xuan Thuy, Hanoi, Viet Nam
2. Truong, A.-H., College of Technology, Vietnam National University, 144 Xuan Thuy, Hanoi, Viet Nam
3. Truong, N.-T., College of Technology, Vietnam National University, 144 Xuan Thuy, Hanoi, Viet Nam
4. Chin, W.-N., School of Computing, National University of Singapore, 3 Science Drive 2, Singapore 117543, Singapore

References:

1. (2006) Principles, Techniques, and Tools, , A. V. Aho, M. S. Lam, R. Sethi, and J. D. Ullman. Compilers:, 2nd Edition, Addison Wesley, August
2. Albert, E., Arenas, P., Genaim, S., Puebla, G., Zanardini, D., Cost Analysis of Java Bytecode (2007) LNCS, 4421, pp. 157-172. , R. D. Nicola, editor, ESOP, of, Springer
3. Albert, E., Genaim, S., Gomez-Zamalloa, M., Heap Space Analysis for Java Bytecode (2007) ISMM '07: Proc. of the 6th International Symposium on Memory Management, pp. 105-116. , New York, NY, USA, ACM
4. Cachera, D., Jensen, T., Pichardie, D., Schneider, G., Certified Memory Usage Analysis (2005) LNCS, , Proc. of 13th International Symposium on Formal Methods FM'05, Springer
5. W.-N. Chin, H. H. Nguyen, S. Qin, and M. Rinard. Memory Usage Verification for OO Programs. In C. Hankin and I. Siveroni, editors, SAS, 3672 of LNCS, pages 70-86. Springer, 2005Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., (2001) Introduction to Algorithms, , Second Edition. The MIT Press, September
6. Giambiagi, P., Schneider, G., Memory consumption analysis of Java smart cards (2005) Proc. of XXXI Latin American Informatics Conference (CLEI 2005), p. 12. , Cali, Colombia, October
7. Hofmann, M., Jost, S., Static Prediction of Heap Space Usage for First-Order Functional Programs (2003) POPL '03: Proc. of the 30th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, pp. 185-197. , New York, NY, USA, ACM
8. Hofmann, M., Jost, S., Type-Based Amortised Heap-Space Analysis (for an Object-Oriented Language) (2006) LNCS, 3924, pp. 22-37. , P. Sestoft, editor, Proc. of the 15th European Symposium on Programming ESOP, Programming Languages and Systems, of, Springer
9. Hughes, J., Pareto, L., Recursion and Dynamic Data-structures in Bounded Space: Towards Embedded ML Programming (1999) Proc. of the fourth ACM SIGPLAN International Conference on Functional Programming (ICFP '99), pp. 70-81
10. S. Microsystems. Virtual Machine Specification, Java Card Platform, v3.0, Classic Edition. March 2008S. Microsystems. Virtual Machine Specification, Java Card platform, v3.0, Connected Edition. March 2008T.-H. Pham, A.-H. Truong, and N.-T. Truong. Computing Heap Space Cost of Java Card Applets. In A. Demaille, T. Cao, and B. Ho, editors, Contributions to the 2008 IEEE International Conference on Research, Innovation and Vision for the Future in Computing & Communication Technologies, pages 190-196, University of Science - Vietnam National University, Ho Chi Minh City, July 2008F. Spoto. JULIA: A Generic Static Analyser for the Java Bytecode. In Proc. of the 7th Workshop on Formal Techniques for Java-like Programs, FTfJP'2005, Glasgow, Scotland, July 2005Truong, H., Bezem, M., Finding Resource Bounds in the Presence of Explicit Deallocation (2005) LNCS, 3722, pp. 227-241. , D. V. Hung and M. Wirsing, editors, Proc. of ICTAC 2005, of, Springer
11. Unnikrishnan, L., Stoller, S.D., Liu, Y.A., Optimized Live Heap Bound Analysis (2003) VMCAI 2003: Proc. of the 4th International Conference on Verification, Model Checking, and Abstract Interpretation, pp. 70-85. , London, UK, Springer
12. Vasconcelos, P.B., Hammond, K., Inferring Cost Equations for Recursive, Polymorphic and Higher-Order Functional Programs (2003) LNCS, 3145, pp. 86-101. , P. Trinder, G. Michaelson, and R. Pe?a, editors, 15th International Workshop, IFL, Edinburgh, UK, September 8-11, of, Springer, 2004

13. Venners, B., (2000) Inside the Java Virtual Machine, , McGraw-Hill, Inc, New York, NY, USA