

Pun, Ka I; Steffen, Martin; Stolz, Volker

Deadlock checking by data race detection. (English) Zbl 1371.68199
J. Log. Algebr. Methods Program. 83, No. 5-6, 400-426 (2014).

Summary: Deadlocks are a common problem in programs with lock-based concurrency and are hard to avoid or even to detect. One way for deadlock prevention is to statically analyse the program code to spot sources of potential deadlocks.

We reduce the problem of deadlock checking to data race checking, another prominent concurrency-related error for which good (static) checking tools exist. The transformation uses a type and effect-based static analysis, which analyses the data flow in connection with lock handling to find out control-points which are potentially part of a deadlock. These control-points are instrumented appropriately with additional shared variables, i.e., race variables injected for the purpose of the race analysis. To avoid overly many false positives for deadlock cycles of length longer than two, the instrumentation is refined by adding “gate locks”. The type and effect system, and the transformation are formally given. We prove our analysis sound using a simple, concurrent calculus with re-entrant locks.

MSC:

[68Q85](#) Models and methods for concurrent and distributed computing (process algebras, bisimulation, transition nets, etc.) Cited in 2 Documents

Keywords:

[deadlock detection](#); [race detection](#); [type and effect system](#); [concurrency](#); [formal method](#)

Software:

[Cyclone](#)

Full Text: [DOI](#)

References:

- [1] Abadi, M.; Flanagan, C.; Freund, S. N., Types for safe locking: static race detection for Java, *ACM Trans. Program. Lang. Syst.*, 28, 2, 207-255, (2006)
- [2] Agarwal, R.; Sasturkar, A.; Stoller, S. D., Type discovery for parameterized race-free Java, (Sept. 2004), Dept. of Computer Science, SUNY Stony Brooks, NY, Technical report DAR-04-16
- [3] Agarwal, R.; Stoller, S. D., Type inference for parameterized race-free Java, (Steffen, B.; Levi, G., Proceedings of the 5th International Conference on Verification, Model Checking, and Abstract Interpretation, VMCAI, 2004, Lect. Notes Comput. Sci., vol. 2937, (2004), Springer-Verlag)
- [4] Agarwal, R.; Wang, L.; Stoller, S. D., Detecting potential deadlocks with state analysis and run-time monitoring, (Ur, S.; Bin, E.; Wolfsthal, Y., Proceedings of the Haifa Verification Conference 2005, Lect. Notes Comput. Sci., vol. 3875, (2006), Springer-Verlag), 191-207
- [5] Amtoft, T.; Nielson, H. R.; Nielson, F., Type and effect systems: behaviours for concurrency, (1999), Imperial College Press
- [6] Beckman, N. E., A survey of methods for preventing race conditions, (May 2006), available at
- [7] Birrell, A. D., An introduction to programming with threads, (1989), Corporation Research Center, Research report 35, Digital equipment
- [8] Boudol, G., A deadlock-free semantics for shared memory concurrency, (ICTAC'09, Lect. Notes Comput. Sci., vol. 5684, (2009), Springer-Verlag), 140-154 · [Zbl 1250.68207](#)
- [9] Boyapati, C.; Lee, R.; Rinard, M., Ownership types for safe programming: preventing data races and deadlocks, *Object Oriented Programming: Systems, Languages, and Applications (OOPSLA) '02*, Seattle, USA, SIGPLAN Not., 37, 211-230, (Nov. 2002)
- [10] Boyapati, C.; Rinard, M., A parameterized type system for race-free Java programs, (Object Oriented Programming: Systems, Languages, and Applications (OOPSLA) '01, (2001), ACM)
- [11] Coffman, E. G.; Elphick, M.; Shoshani, A., System deadlocks, *ACM Comput. Surv.*, 3, 2, 67-78, (June 1971)
- [12] Corbett, J., Evaluating deadlock detection methods for concurrent software, *IEEE Trans. Softw. Eng.*, 22, 3, 161-180, (Mar.

1996)

- [13] Dijkstra, E. W., Cooperating sequential processes, (1965), Technological University Eindhoven, reprinted in [19]
- [14] Flanagan, C.; Abadi, M., Object types against races, (Baeten, J. C.; Mauw, S., Proceedings of CONCUR '99, Lect. Notes Comput. Sci., vol. 1664, (Aug. 1999), Springer-Verlag), 288-303
- [15] Flanagan, C.; Abadi, M., Types for safe locking, (Swierstra, S., Programming Languages and Systems, Lect. Notes Comput. Sci., vol. 1576, (1999), Springer), 91-108
- [16] Flanagan, C.; Freund, S., Type-based race detection for Java, (Proceedings of PLDI'00, ACM SIGPLAN Conference on ACM Conference on Programming Language Design and Implementation (PLDI), (2000)), 219-232
- [17] Flanagan, C.; Freund, S., Type inference against races, (Proceedings of SAS '04, Lect. Notes Comput. Sci., vol. 3148, (2004), Springer-Verlag), 116-132 · [Zbl 1104.68399](#)
- [18] Flanagan, C.; Sabry, A.; Duba, B. F.; Felleisen, M., The essence of compiling with continuations, ACM Conference on Programming Language Design and Implementation (PLDI), SIGPLAN Not., 28, 6, (June 1993)
- [19] Gerakios, P.; Papaspyrou, N.; Sagonas, K., A type system for unstructured locking that guarantees deadlock freedom without imposing a lock ordering, (Beresford, A. R.; Gay, S. J., Pre-Proceedings of the Workshop on Programming Language Approaches to Concurrent and Communication-Centric Software (PLACES 2009), EPTCS, vol. 17, (2009)), 79-93
- [20] Gerakios, P.; Papaspyrou, N.; Sagonas, K., A type and effect system for deadlock avoidance in low-level languages, (TLDI'11, (2011), ACM)
- [21] Grossman, D., Type-safe multithreading in cyclone, (TLDI'03: Types in Language Design and Implementation, (2003), ACM), 13-25
- [22] Havelund, K., Using runtime analysis to guide model checking of Java programs, (Havelund, K.; Penix, J.; Visser, W., Proceedings of the Spin 2000 Workshop in Model Checking of Software, (2000)) · [Zbl 0976.68576](#)
- [23] Mossin, C., Flow analysis of typed higher-order programs, (1997), DIKU, University of Copenhagen Denmark, PhD thesis, Technical report DIKU-TR-97/1 · [Zbl 0891.68025](#)
- [24] Naik, M.; Aiken, A.; Whaley, J., Effective static race detection for Java, (ACM Conference on Programming Language Design and Implementation (PLDI), Ottawa, Ontario, Canada, (June 2006), ACM), 308-319
- [25] Naik, M.; Park, C.-S.; Sen, K.; Gay, D., Effective static deadlock detection, (31st International Conference on Software Engineering (ICSE 09), (2009), IEEE)
- [26] Pun, K. I.; Steffen, M.; Stolz, V., Behaviour inference for deadlock checking, (July 2012), University of Oslo, Dept. of Informatics, Technical report 416
- [27] Pun, K. I.; Steffen, M.; Stolz, V., Deadlock checking by a behavioral effect system for lock handling, J. Log. Algebr. Program., 81, 3, 331-354, (Mar. 2012), A preliminary version was published as University of Oslo, Dept. of Computer Science Technical Report 404, March 2011
- [28] Pun, K. I.; Steffen, M.; Stolz, V., Deadlock checking by data race detection, (Proc. of the 5th IPM International Conference on Fundamentals of Software Engineering (FSEN'13), Lect. Notes Comput. Sci., vol. 8161, (2013), Springer-Verlag) · [Zbl 1434.68112](#)
- [29] Rinard, M., Analysis of multithreaded programs, (Cousot, P., Proceedings of the 8th International Static Analysis Symposium, SAS '01, Lect. Notes Comput. Sci., vol. 2126, (2001), Springer-Verlag), 1-19 · [Zbl 1005.68896](#)
- [30] Rose, J.; Swamy, N.; Hicks, M., Dynamic inference of polymorphic lock types, Sci. Comput. Program., 58, 3, 366-383, (2005) · [Zbl 1105.68014](#)
- [31] Sasturkar, A.; Agarwal, R.; Wang, L.; Stoller, S., Automated type-based analysis of data races and atomicity, (Ferrante, J.; Padua, D. A.; Wexelblat, R. L., PPOPP'05, (2005), ACM), 83-94
- [32] Seidl, H.; Vojdani, V., Region analysis for race detection, (Palsberg, J.; Su, Z., Proceedings of SAS '09, Lect. Notes Comput. Sci., vol. 5673, (2009), Springer-Verlag), 171-187 · [Zbl 1248.68152](#)
- [33] Suenaga, K., Type-based deadlock-freedom verification for non-block-structured lock primitives and mutable references, (Ramalingam, G., APLAS 2008, Lect. Notes Comput. Sci., vol. 5356, (2008), Springer-Verlag), 155-170
- [34] Talpin, J.-P.; Jouvelot, P., Polymorphic type, region and effect inference, J. Funct. Program., 2, 3, 245-271, (1992) · [Zbl 0817.68099](#)
- [35] Vasconcelos, V.; Martins, F.; Cogumbreiro, T., Type inference for deadlock detection in a multithreaded polymorphic typed assembly language, (Beresford, A. R.; Gay, S. J., Pre-Proceedings of the Workshop on Programming Language Approaches to Concurrent and Communication-Centric Software (PLACES 2009), EPTCS, vol. 17, (2009)), 95-109

This reference list is based on information provided by the publisher or from digital mathematics libraries. Its items are heuristically matched to zbMATH identifiers and may contain data conversion errors. It attempts to reflect the references listed in the original paper as accurately as possible without claiming the completeness or perfect precision of the matching.