

Fava, Daniel S.; Steffen, Martin; Stolz, Volker

Operational semantics of a weak memory model with channel synchronization. (English)

Zbl 1417.68093

J. Log. Algebr. Methods Program. 103, 1-30 (2019).

Summary: There exists a multitude of weak memory models supporting various types of relaxations and synchronization primitives. On one hand, such models must be lax enough to account for hardware and compiler optimizations; on the other, the more lax the model, the harder it is to understand and program for. Though the right balance is up for debate, a memory model should provide what is known as the *SC-DRF guarantee*, meaning that data-race free programs behave in a sequentially consistent manner.

We present a weak memory model for a calculus inspired by the Go programming language. Thus, different from previous approaches, we focus on buffered channel communication as the sole synchronization primitive. Our formalization is operational, which allows us to prove the SC-DRF guarantee using a standard simulation technique. Contrasting against an axiomatic semantics, where the notion of a program is abstracted away as a graph with memory events as nodes, we believe our operational semantics and simulation proof can be clearer and easier to understand. Finally, we provide a concrete implementation in \mathbb{K} , a rewrite-based executable semantic framework, and derive an interpreter for the proposed language.

MSC:

68Q55 Semantics in the theory of computing

Keywords:

operational semantics; weak memory model; data-race freedom guarantee; channel communication

Software:

CakeML; K Prover

Full Text: DOI

References:

- [1] Adve, S. V.; Gharachorloo, K., Shared Memory Consistency Models: A Tutorial, (1995), Research Report 95/7, Digital WRL
- [2] Adve, S. V.; Hill, M. D., Weak ordering — a new definition, SIGARCH Comput. Archit. News, 18, 3a, 2-14, (1990)
- [3] Alglave, J.; Maranget, L.; Tautschnig, M., Herding Cats: modelling, simulation, testing, and data-mining for weak memory, ACM Trans. Program. Lang. Syst., 36, 2, (2014)
- [4] Alrahman, Y. A.; Andric, M.; Beggiato, A.; Lluch-Lafuente, A., Can we efficiently check concurrent programs under relaxed memory models in Maude?, (Escobar, S., Rewriting Logic and Its Applications - 10th International Workshop. Rewriting Logic and Its Applications - 10th International Workshop, WRLA 2014. Rewriting Logic and Its Applications - 10th International Workshop, WRLA 2014, Lecture Notes in Computer Science, vol. 8663, (2014), Springer Verlag), 21-41, Revised Selected Papers · Zbl 1367.68057
- [5] Andrews, G. R., Foundations of Multithreaded, Parallel, and Distributed Programming, (2000), Addison-Wesley
- [6] Aspinall, D.; Ševčík, J., Java memory model examples: good, bad and ugly, (Proc. of VAMP 7, (2007))
- [7] Batty, M.; Memarian, K.; Nienhuis, K.; Pichon-Pharabod, J.; Sewell, P., The problem of programming language concurrency semantics, (Vitek, J., Programming Languages and Systems: 24th European Symposium on Programming. Programming Languages and Systems: 24th European Symposium on Programming, ESOP 2015. Programming Languages and Systems: 24th European Symposium on Programming, ESOP 2015, Lecture Notes in Computer Science, vol. 9032, (2015), Springer Verlag), 283-307
- [8] Batty, M.; Owens, S.; Sarkar, S.; Weber, T., Mathematizing C++ concurrency, (Proceedings of POPL '11, (2011), ACM), 55-66 · Zbl 1284.68165
- [9] Becker, Programming Languages — C++, (2011), ISO/IEC 14882:2001
- [10] Boehm, H.-J.; Adve, S. V., Foundations of the C++ concurrency memory model, (ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), (2008), ACM), 68-78
- [11] Boehm, H.-J.; Demsky, B., Outlawing ghosts: avoiding out-of-thin-air results, (Proceedings of the Workshop on Memory

Systems Performance and Correctness. Proceedings of the Workshop on Memory Systems Performance and Correctness, MSPC '14, (2014), ACM: ACM New York, NY, USA), 7, 6 pp

- [12] Boudol, G.; Petri, G., Relaxed memory models: an operational approach, (Proceedings of POPL '09, (2009), ACM), 392-403 · [Zbl 1315.68173](#)
- [13] Collier, W. W., Reasoning about Parallel Architectures, (1992), Prentice Hall · [Zbl 0759.68009](#)
- [14] Demange, D.; Laporte, V.; Zhao, L.; Jagannathan, S.; Pichardie, D.; Vitek, J., Plan B: a buffered memory model for Java, (Proceedings of POPL '13, (2013), ACM), 329-342 · [Zbl 1301.68098](#)
- [15] Donovan, A. A.A.; Kernighan, B. W., The Go Programming Language, (2015), Addison-Wesley
- [16] Fava, D., Operational semantics of a weak memory model with channel synchronization, (2017)
- [17] Fava, D.; Steffen, M.; Stolz, V., Operational semantics of a weak memory model with channel synchronization, (Havelund, K.; Peleska, J.; Roscoe, B.; de Vink, E., FM. FM, Lecture Notes in Computer Science, vol. 10951, (2018), Springer Verlag), 1-19
- [18] Flanagan, C.; Freund, S. N., Adversarial memory for detecting destructive races, (Zorn, B.; Aiken, A., ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), (2010), ACM), 244-254
- [19] Go language specification, the Go programming language specification, (2016)
- [20] The Go memory model, (2014), version of May 31, 2014, covering Go version 1.9.1
- [21] Guerraoui, R.; Henzinger, T. A.; Singh, V., Software transactional memory on relaxed memory models, (Bouajjani, A.; Maler, O., Proceedings of CAV '09. Proceedings of CAV '09, Lecture Notes in Computer Science, vol. 5643, (2009), Springer Verlag), 321-336 · [Zbl 1242.68162](#)
- [22] Hoare, C. A.R., Communicating sequential processes, *Commun. ACM*, 21, 8, 666-677, (1978) · [Zbl 0383.68028](#)
- [23] Inc, S. I.; Weaver, D. L., The SPARC Architecture Manual, (1994), Prentice-Hall
- [24] Jagadeesan, R.; Pitcher, C.; Riely, J., Generative operational semantics for relaxed memory models, (Gordon, A. D., Programming Languages and Systems. Programming Languages and Systems, Lecture Notes in Computer Science, vol. 6012, (2010), Springer Verlag), 307-326 · [Zbl 1260.68050](#)
- [25] Jones, G.; Goldsmith, M., Programming in occam2, (1988), Prentice-Hall International: Prentice-Hall International Hemel Hempstead
- [26] K framework, The K framework, (2017), available at
- [27] Kang, J.; Hur, C.; Lahav, O.; Vafeiadis, V.; Dreyer, D., A promising semantics for relaxed-memory concurrency, (Castagna, G.; Gordon, A. D., Proceedings of POPL '17, (2017), ACM), 175-189, (2017) · [Zbl 1380.68103](#)
- [28] Kumar, R.; Myreen, M. O.; Norrish, M.; Owens, S., CakeML: a verified implementation of ML, (The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20-21, 2014, (2014)), 179-192 · [Zbl 1284.68405](#)
- [29] Lamport, L., Time, clocks, and the ordering of events in a distributed system, *Commun. ACM*, 21, 7, 558-565, (1978) · [Zbl 0378.68027](#)
- [30] Lamport, L., How to make a multiprocessor computer that correctly executes multiprocess programs, *IEEE Trans. Comput.*, C-28, 9, 690-691, (1979) · [Zbl 0419.68045](#)
- [31] Lange, J.; Ng, N.; Toninho, B.; Yoshida, N., Fencing off Go: liveness and safety for channel-based programming, (Castagna, G.; Gordon, A. D., Proceedings of POPL '17, (2017), ACM), 748-761 · [Zbl 1380.68298](#)
- [32] Lochbihler, A., Making the Java memory model safe, *ACM Trans. Program. Lang. Syst.*, 35, 4, (2013), 65 pp
- [33] Manson, J.; Pugh, W.; Adve, S. V., The Java memory model, (Proceedings of POPL '05, (2005), ACM), 378-391 · [Zbl 1369.68079](#)
- [34] Maranget, L.; Sarkar, S.; Sewell, P., A Tutorial Introduction to the ARM and POWER Relaxed Memory Models (Version 120), (2012)
- [35] Milner, R., An algebraic definition of simulation between programs, (Proceedings of the Second International Joint Conference on Artificial Intelligence, (1971), William Kaufmann), 481-489
- [36] Milner, R.; Parrow, J.; Walker, D., A calculus of mobile processes, part I/II, *Inf. Comput.*, 100, 1-77, (1992)
- [37] Palamidessi, C., Comparing the expressive power of the synchronous and the asynchronous π -calculus, (Proceedings of POPL '97, (1997), ACM), 256-265
- [38] Peters, K.; Nestmann, U., Is it a “good” encoding of mixed choice?, (Proceedings of the International Conference on Foundations of Software Science and Computation Structures (FoSSaCS '12). Proceedings of the International Conference on Foundations of Software Science and Computation Structures (FoSSaCS '12), Lecture Notes in Computer Science, vol. 7213, (2012), Springer Verlag), 210-224 · [Zbl 1352.68190](#)
- [39] Pichon-Pharabod, J.; Sewell, P., A concurrency-semantics for relaxed atomics that permits optimisation and avoids thin-air executions, (Proceedings of POPL '16, (2016), ACM), 622-633 · [Zbl 1347.68091](#)
- [40] Pugh, W., Fixing the Java memory model, (Proceedings of the ACM Java Grande Conference, (1999)), 89-98
- [41] Roşu, G.; Şerbănuţă, T. F., An overview of the K semantic framework, *J. Log. Algebraic Methods Program.*, 79, 6, 397-434, (2010) · [Zbl 1214.68188](#)
- [42] Sabry, A.; Felleisen, M., Reasoning about programs in continuation-passing style, (Clinger, W., Conference on LISP and

Functional Programming. Conference on LISP and Functional Programming, San Francisco, California, (1992), ACM), 288-298

- [43] Steffen, M., A small-step semantics of a concurrent calculus with goroutines and deferred functions, (Ábrahám, E.; Huisman, M.; Johnsen, E. B., Theory and Practice of Formal Methods. Essays Dedicated to Frank de Boer on the Occasion of His 60th Birthday (Festschrift). Theory and Practice of Formal Methods. Essays Dedicated to Frank de Boer on the Occasion of His 60th Birthday (Festschrift), Lecture Notes in Computer Science, vol. 9660, (2016), Springer Verlag), 393-406
- [44] Tandler, J. M.; Dodson, J. S.; Fields, J. S.; Le, H. Q.; Sinharoy, B., POWER4 system microarchitecture, IBM J. Res. Dev., 46, 1, 5-25, (2002)
- [45] Zhang, Y.; Feng, X., An operational happens-before memory model, Front. Comput. Sci., 10, 1, 54-81, (2016) · [Zbl 1403.68028](#)

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.