von Oheimb, David

Hoare logic for Java in Isabelle/HOL. (English) Zbl 0997.68019

Concurrency Comput. Pract. Exp. 13, No. 13, 1173-1214 (2001).

Summary: This article presents a Hoare-style calculus for a substantial subset of Java Card, which we call Java^{light}. In particular, the language includes side-effecting expressions, mutual recursion, dynamic method binding, full exception handling, and static class initialization.

The Hoare logic of partial correctness is proved not only sound (w.r.t. our operational semantics of Java^{light} described in detail elsewhere) but even complete. It is the first logic for an object-oriented language that is provably complete. The completeness proof uses a refinement of the most general formula approach. The proof of soundness gives new insights into the role of type safety. Further by-products of this work are a new general methodology for handling side-effecting expressions and their results, the discovery of the strongest possible rule of consequence, and a flexible call rule for mutual recursion. We also give a small but non-trivial application example.

All definitions and proofs have been done formally with the interactive theorem prover Isabelle/HOL. This guarantees not only rigorous definitions, but also gives maximal confidence in the results obtained.

MSC:

68N15 Theory of programming languages

Keywords:

Hoare-style calculus; Java Card

Software:

Isabelle; Isabelle/HOL; JML

Full Text: DOI

References:

[1] The Java Language Specification. Addison-Wesley, 1996. · Zbl 0865.68001

[2] Proving Java type soundness. Formal Syntax and Semantics of Java (Lecture Notes in Computer Science, vol. 1523), (ed.). Springer: Berlin, 1999; 83-118. · doi:10.1007/3-540-48737-9_3

[3] Describing the semantics of Java and proving type soundness. Formal Syntax and Semantics of Java (Lecture Notes in Computer Science, vol. 1523), (ed.). Springer: Berlin, 1999; 41-82. · doi:10.1007/3-540-48737-9_2

[4] et al. Project LOOP. http://www.cs.kun.nl/?bart/LOOP/ [1998].

[5] Reasoning about Java classes (preliminary report). Proceedings of the ACM Symposium on Object-Oriented Programming: Systems, Languages and Applications, 1998; 329-340.

[6] Java program verification via a Hoare logic with abrupt termination. Fundamental Approaches to Software Engineering (Lecture Notes in Computer Science, vol. 1783). Springer: Berlin, 2000; 284-303. · doi:10.1007/3-540-46428-X_20

[7] Java program verification in Higher-order logic with PVS and Isabelle. PhD Thesis, University of Nijmegen, 2001. http://www-sop.inria.fr/oasis/personnel/Marieke.Huisman/thesis.ps.gz.

[8] Project Oasis: Java semantics. http://www-sop.inria.fr/oasis/java/java_sem.html [1998].

[9] A formal executable semantics for Java. Proceedings of the OOPSLA'98 Workshop on Formal Underpinnings of Java, 1998.

[10] Checking Java programs via guarded commands. Formal Techniques for Java Programs, Jacobs B, Leavens GT, Müller P, Poetzsch-Heffter A (eds.). Technical Report 251, Fernuniversität Hagen. http://gatekeeper.dec.com/pub/DEC/SRC/technical-notes/abstracts/src-tn-1999-002.html [1999].

[11] A logic of object-oriented programs. Technical Report SRC-161, Compaq SRC, 1998. · Zbl 0913.68025

[12] Implementing a program logic of objects in a higher-order logic theorem prover. Theorem Proving in Higher Order Logics: 13th International Conference, TPHOLs 2000 (Lecture Notes in Computer Science, vol. 1869), (eds.). Springer: Berlin, 2000; 267-282. · Zbl 0974.68185 · doi:10.1007/3-540-44659-1_17

[13] Subtyping, modular specification, and modular verification for applicative object-oriented programs. Technical Report 92-28d,

Department of Computer Science, Iowa State University, 1992; revised 1994.

[14] Preliminary design of JML: A behavioral interface specification language for Java. Technical Report 98-06l, Department of Computer Science, Iowa State University, 1998.http://www.cs.iastate.edu/leavens/JML.html.

[15] A logic for the Java Modeling Language JML. Technical Report CSI-R0018, CSI.http://www.cs.kun.nl/csi/reports/info/CSI-R0018.html [2000].

[16] A programming logic for sequential Java. Programming Languages and Systems (ESOP '99) (Lecture Notes in Computer Science, vol. 1576>), (ed.). Springer: Berlin, 1999; 162-176. · doi:10.1007/3-540-49099-X_11

[17] Universes: A type system for controlling representation exposure. Programming Languages and Fundamentals of Programming, Poetzsch-Heffter A, Meyer J (eds.). Technical Report 263, Fernuniversität Hagen. http://www.informatik.fernuni-hagen.de/pi5/publications.html [1999].

[18] A proof theory for a sequential version of POOL. Unpublished.

[19] A WP-calculus for OO. Foundations of Software Science and Computation Structures (Lecture Notes in Computer Science, vol. 1578>). Springer: Berlin, 1999; 135-149. · doi:10.1007/3-540-49019-1_10

[20] Project Bali. http://isabelle.in.tum.de/Bali/ [1998].

[21] Hoare logic and VDM: Machine-checked soundness and completeness proofs. PhD Thesis, ECS-LFCS-98-392, LFCS, 1998.

[22] Analyzing Java in Isabelle/HOL: Formalization, Type Safety and Hoare Logic. PhD Thesis, Technische Universität München. http://www4.in.tum.de/?oheimb/diss/ [2001].

[23] Java Card technology. Sun Microsystems. http://java.sun.com/products/javacard/ [1999].

[24] Axiomatic semantics for Javalight. Formal Techniques for Java Programs, Drossopoulou S, Eisenbach S, Jacobs B, Leavens GT, Müller P, Poetzsch-Heffter A (eds.). Technical Report 269, 5/2000, Fernuniversität Hagen. http://isabelle.in.tum.de/Bali/papers/ECOOP00.html [2000].

[25] Axiomatic semantics for Javalight in Isabelle/HOL. Drossopoulou S, Eisenbach S, Jacobs B, Leavens GT, Müller P, Poetzsch-Heffter A (eds.). Technical Report CSE 00-009, Oregon Graduate Institute. TPHOLs 2000 Supplemental Proceedings. http://isabelle.in.tum.de/Bali/papers/TPHOLs00.html [2000].

[26] Hoare, Communications of the ACM 12 pp 576– (1969) · Zbl 0179.23105 · doi:10.1145/363235.363259

[27] Krzysztof, ACM Transactions on Programming Languages and Systems 3 pp 431– (1981) · Zbl 0471.68006 · doi:10.1145/357146.357150

[28] Isabelle: A Generic Theorem Prover (Lecture Notes in Computer Science, vol. 828), (ed.). Springer: Berlin. http://isabelle.in.tum.de/ [1994]. · Zbl 0825.68059 · doi:10.1007/BFb0030541

[29] Church, Journal of Symbolic Logic 5 pp 56– (1940) · Zbl 0023.28901 · doi:10.2307/2266170

[30] Edinburgh LCF: a Mechanised Logic of Computation (Lecture Notes in Computer Science, vol. 78). Springer: Berlin, 1979. · Zbl 0421.68039 · doi:10.1007/3-540-09724-4

[31] Proof General. http://www.proofgeneral.arg/ [1999].

[32] (eds.). Introduction to HOL: A Theorem-proving Environment for Higher Order Logic. Cambridge University Press: Cambridge, UK, 1993. · Zbl 0779.68007

[33] Isabelle's logics: HOL. Isabelle: A Generic Theorem Prover (Lecture Notes in Computer Science, vol. 828). Springer: Berlin. http://isabelle.in.tum.de/doc/logics-HOL.pdf [1994].

[34] The Isabelle/HOL library. http://isabelle.in.tum.de/library/HOL/.

[35] Isabelle/HOL. The Tutorial. http://isabelle.in.tum.de/doc/tutorial.pdf [2001].

[36] Machine-checking the Java specification: Proving type-safety. Formal Syntax and Semantics of Java (Lecture Notes in Computer Science, vol. 1523), (ed.). Springer: Berlin.http://isabelle.in.tum.de/Bali/papers/Springer98.html [1999]. · doi:10.1007/3-540-48737-9_4

[37] Strachey, Higher-Order and Symbolic Computation 13 pp 11– (2000) · Zbl 0949.68510 · doi:10.1023/A:1010000313106

[38] Parnas, Communications of the ACM 15 pp 330– (1972) · doi:10.1145/355602.361309

[39] Experience with embedding hardware description languages in HOL. Theorem Provers in Circuit Design, (eds.). North-Holland/Elsevier, 1992; 129-156.

[40] Cook, SIAM Journal on Computing 7 pp 70– (1978) · Zbl 0374.68009 · doi:10.1137/0207005

[41] A system of proof rules for the correctness of iterative programs?some notational and organisational suggestions. Unpublished, 1982.

[42] A formulation of Hoare logic suitable for Isar. http://isabelle.in.tum.de/library/HOL/Isar_examples/Hoare.html [2000].

[43] Systematic Program Development Using VDM (International Series in Computer Science) (2nd edn). Prentice-Hall: Englewood Cliffs, NJ, 1990.

[44] Auxiliary variables and recursive procedures. Theory and Practice of Software Development (Lecture Notes in Computer Science, vol. 1214). Springer: Berlin, 1997; 697-711. · doi:10.1007/BFb0030635

[45] Homeier, The Computer Journal 38 pp 131– (1995) · Zbl 05479913 · doi:10.1093/comjnl/38.2.131

[46] Kowaltowski, Acta Informatica 7 pp 357– (1977) · Zbl 0325.68010 · doi:10.1007/BF00289468

[47] Boehm, ACM Transactions on Programming Languages and Systems 7 pp 637– (1981) · Zbl 0575.68011 · doi:10.1145/4472.4474

[48] Toward reliable modular programs. PhD Thesis, California Institute of Technology, 1995; Technical Report CS-TR-95-03.

[49] Semantik und Verifikations. Lecture notes, in German, 1997.

[50] Hoare logic for mutual recursion and local variables. Foundations of Software Technology and Theoretical Computer Science (Lecture Notes in Computer Science, vol. 1738), (eds.). Springer: Berlin. http://isabelle.in.tum.de/Bali/papers/FSTTCS99.html [1999]. · doi:10.1007/3-540-46691-6_13

[51] Mechanical verification of mutually recursive procedures. Proceedings of the 13th International Conference on Automated Deduction (Lecture Notes in Computer Science, vol. 1104), (eds.). Springer: Berlin, 1996; 201-215.

[52] Modular specification and verification of object-oriented programs. PhD Thesis, FernUniversität Hagen, 2001. To appear. · Zbl 0974.68035

[53] A complete axiomatic system for proving assertions about recursive and non-recursive programs. Technical Report 75, Department of Computer Science, University of Toronto, 1975.

[54] Handling mutual recursion. Personal communication, April, 1999.

[55] An architecture for interactive program provers. TACAS00, Tools and Algorithms for the Construction and Analysis of Systems (Lecture Notes in Computer Science, vol. 1785), (eds.). Springer: Berlin, 2000; 63-77. · doi:10.1007/3-540-46419-0_6

[56] et al. Project Verificard. http://www.cs.kun.nl/VerifiCard/ [2001].