Combining Monitoring with Run-time Assertion Checking

Gouw, C.P.T. de

Bibliography

[1] A. E. Abdallah, C. B. Jones, and J. W. Sanders, editors. *Communicating Sequential Processes: The First 25 Years, Symposium on the Occasion of 25 Years of CSP, London, UK, July 7-8, 2004, Revised Invited Papers*, volume 3525 of *Lecture Notes in Computer Science*. Springer, 2005.

[2] G. A. Agha. Actors: A model of concurrent computation in distributed systems. *MIT Press, Cambridge, MA, USA*, 1990.

[3] C. Allan, P. Avgustinov, A. S. Christensen, L. J. Hendren, S. Kuzins, O. Lhoták, O. de Moor, D. Sereni, G. Sittampalam, and J. Tibble. Adding trace matching with free variables to AspectJ. In *OOPSLA*, pages 345–364, 2005.

[4] K. R. Apt, F. S. de Boer, E.-R. Olderog, and S. de Gouw. Verification of Object-Oriented programs: A transformational approach. *J. Comput. Syst. Sci.*, 78(3):823–852, 2012.

[5] C. Artho, D. Drusinsky, A. Goldberg, K. Havelund, M. R. Lowry, C. S. Pasareanu, G. Rosu, and W. Visser. Experiments with test case generation and runtime analysis. In *Abstract State Machines*, pages 87–107, 2003.

[6] J. W. Backus. The syntax and semantics of the proposed international algebraic language of the Zurich ACM-GAMM conference. In *IFIP Congress*, pages 125–131, 1959.

[7] J. C. M. Baeten, T. Basten, and M. A. Reniers. *Process Algebra: Equational Theories of Communicating Processes.*

Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[8] D. Bartetzko, C. Fischer, M. Möller, and H. Wehrheim. Jass - Java with assertions. *Electr. Notes Theor. Comput. Sci.*, 55(2), 2001.

[9] A. Bauer, M. Leucker, and C. Schallhart. Comparing LTL semantics for runtime verification. *J. Log. Comput.*, 20(3):651–674, 2010.

[10] B. Beckert, R. Hähnle, and P. H. Schmitt, editors. *Verification of Object-Oriented Software: The KeY Approach*, volume 4334 of *Lecture Notes in Computer Science*. Springer-Verlag, 2007.

[11] J. Bengtsson, K. G. Larsen, F. Larsson, P. Pettersson, and W. Yi. Uppaal — a tool suite for automatic verification of real–time systems. In *Proc. of Workshop on Verification and Control of Hybrid Systems III*, number 1066 in Lecture Notes in Computer Science, pages 232–243. Springer–Verlag, Oct. 1995.

[12] J. Berdine, C. Calcagno, and P. W. O'Hearn. Smallfoot: Modular automatic assertion checking with Separation Logic. In *FMCO*, pages 115–137, 2005.

[13] J. Berdine, B. Cook, and S. Ishtiaq. SLAyer: Memory safety for systems-level code. In *CAV*, pages 178–183, 2011.

[14] J. A. Bergstra and J. W. Klop. $\mathrm{Act}_{\mathrm{tau}}$: A universal axiom system for process specification. In *Algebraic Methods*, pages 447–463, 1987.

[15] Y. Bertot, P. Castran, G. Huet, and C. Paulin-Mohring. *Interactive theorem proving and program development : Coq'Art : the calculus of inductive constructions*. Texts in theoretical computer science. Springer, Berlin, New York, 2004.

[16] M. Brörkens and M. Möller. Dynamic event generation for runtime checking using the JDI. *Electr. Notes Theor. Comput. Sci.*, 70(4), 2002.

[17] L. Burdy, Y. Cheon, D. R. Cok, M. D. Ernst, J. R. Kiniry, G. T. Leavens, K. R. M. Leino, and E. Poll. An overview of JML tools and applications. *International Journal on Software Tools for Technology Transfer*, 7(3):212–232, 2005.

[18] C. Calcagno, H. Yang, and P. W. O'Hearn. Computability and complexity results for a spatial assertion language for data structures. In *FSTTCS*, pages 108–119, 2001.

[19] P. Chalin, P. R. James, and G. Karabotsos. JML4: Towards an industrial grade IVE for java and next generation research platform for JML. In *VSTTE*, pages 70–83, 2008.

[20] F. Chen and G. Rosu. MOP: an efficient and generic runtime verification framework. In *OOPSLA*, pages 569–588, 2007.

[21] Y. Cheon and A. Perumandla. Specifying and checking method call sequences of Java programs. *Software Quality Journal*, 15(1):7–25, 2007.

[22] A. Cimatti, E. M. Clarke, F. Giunchiglia, and M. Roveri. Nusmv: A new symbolic model verifier. In *CAV*, pages 495–499, 1999.

[23] D. Clarke, M. Helvensteijn, and I. Schaefer. Abstract delta modeling. In *GPCE*, pages 13–22, 2010.

[24] E. M. Clarke and E. A. Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Logic of Programs*, pages 52–71, 1981.

[25] E. M. Clarke, O. Grumberg, and D. Peled. *Model checking*. MIT Press, 2001.

[26] C. Colombo, G. J. Pace, and G. Schneider. LARVA — safer monitoring of real-time java programs (tool paper). In *SEFM*, pages 33–37, 2009.

[27] W. Damm and D. Harel. LSCs: Breathing life into message sequence charts. *Formal Methods in System Design*, 19(1):45–80, 2001.

[28] F. S. de Boer, M. M. Bonsangue, M. Steffen, and E. Ábrahám. A fully abstract semantics for UML components. In *FMCO*, pages 49–69, 2004.

[29] F. S. de Boer, S. de Gouw, E. B. Johnsen, A. Kohn, and P. Y. H. Wong. Run-time assertion checking of data- and protocol-oriented properties of Java programs: An industrial case study. *Transactions on Aspect-Oriented Software Development*, 11 (to appear), 2013.

[30] F. S. de Boer, S. de Gouw, and J. Vinju. Prototyping a tool environment for run-time assertion checking in JML with communication histories. In *Proceedings of the 12th Workshop on Formal Techniques for Java-Like Programs*, FTFJP '10, pages 6:1–6:7, New York, NY, USA, 2010. ACM.

[31] F. S. de Boer, S. de Gouw, and P. Y. H. Wong. Run-time verification of coboxes. In *SEFM*, 2013.

[32] S. de Gouw and F. S. de Boer. Run-time verification of black-box components using behavioral specifications: An experience report on tool development. In *FACS*, 2012.

[33] S. de Gouw, F. S. de Boer, W. Ahrendt, and R. Bubel. Weak arithmetic completeness of Object-Oriented first-order assertion networks. In *SOFSEM*, pages 207–219, 2013.

[34] S. de Gouw, F. S. de Boer, E. B. Johnsen, and P. Y. H. Wong. Run-time checking of data- and protocol-oriented properties of Java programs: an industrial case study. In *SAC*, pages 1573–1578, 2013.

[35] D. Distefano and M. J. Parkinson. jStar: towards practical verification for Java. In *OOPSLA*, pages 213–226, 2008.

[36] J.-C. Filliâtre and C. Marché. The Why/Krakatoa/Caduceus platform for deductive program verification. In *CAV*, pages 173–177, 2007.

[37] C. Fischer and H. Wehrheim. Behavioural subtyping relations for Object-Oriented formalisms. In *AMAST*, pages 469–483, 2000.

[38] R. W. Floyd. Assigning meanings to programs. In J. T. Schwartz, editor, *Mathematical Aspects of Computer Science*, volume 19 of *Proceedings of Symposia in Applied Mathematics*, pages 19–32, Providence, Rhode Island, 1967. American Mathematical Society.

[39] D. M. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-Dimensional Modal Logics: Theory and Applications*. Elsevier, 2003.

[40] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns. Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.

[41] D. Grune and C. J. Jacobs. *Parsing Techniques - A Practical Guide (Second Edition)*. Springer-Verlag, 2008.

[42] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, Cambridge, MA, 2000.

[43] G. Hedin. Incremental attribute evaluation with side-effects. In D. Hammer, editor, *Compiler Compilers and High Speed Compilation, 2nd CCHSC Workshop, Berlin GDR, October 10-14, 1988, Proceedings*, volume 371 of *Lecture Notes in Computer Science*, pages 175–189. Springer, 1988.

[44] M. Heisel, W. Reif, and W. Stephan. Implementing verification strategies in the KIV-system. In *CADE*, pages 131–140, 1988.

[45] M. Hennessy. *Algebraic theory of processes*. MIT Press series in the foundations of computing. MIT Press, 1988.

[46] T. A. Henzinger, R. Jhala, R. Majumdar, and G. Sutre. Software verification with BLAST. In *SPIN*, pages 235–239, 2003.

[47] C. A. R. Hoare. An axiomatic basis for computer programming. *Commun. ACM*, 12(10):576–580, 1969.

[48] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.

[49] G. J. Holzmann. The model checker SPIN. *IEEE Trans. Software Eng.*, 23(5):279–295, 1997.

[50] International Telecommunication Union. ITU-T Recommendation Z.120: Message Sequence Chart (MSC). Technical report, ITU, Geneva, 2001.

[51] B. Jacobs, J. Smans, P. Philippaerts, F. Vogels, W. Penninckx, and F. Piessens. VeriFast: a powerful, sound, predictable, fast verifier for C and Java. In *Proceedings of the Third international conference on NASA Formal methods*, NFM'11, pages 41–55, Berlin, Heidelberg, 2011. Springer-Verlag.

[52] A. Jeffrey and J. Rathke. Java Jr: Fully abstract trace semantics for a core Java language. In S. Sagiv, editor, *14th European Symposium on Programming (ESOP'05)*, volume 3444 of *Lecture Notes in Computer Science*, pages 423–438. Springer-Verlag, 2005.

[53] R. Jhala and R. Majumdar. Software model checking. *ACM Comput. Surv.*, 41(4), 2009.

[54] E. B. Johnsen, R. Hähnle, J. Schäfer, R. Schlatte, and M. Steffen. ABS: A core language for abstract behavioral specification. In B. Aichernig, F. S. de Boer, and M. M. Bonsangue, editors, *Proc. 9th International Symposium on Formal Methods for Components and Objects (FMCO 2010)*, volume 6957 of *LNCS*, pages 142–164. Springer-Verlag, 2011.

[55] E. B. Johnsen and O. Owe. An asynchronous communication model for distributed concurrent objects. *Software and System Modeling*, 6(1):35–58, Mar. 2007.

[56] S. C. Kleene. Representation of events in nerve nets and finite automata. *Automata Studies*, 1956.

[57] G. Klein and T. Nipkow. A machine-checked model for a Java-like language, virtual machine, and compiler. *ACM Trans. Prog. Lang. Syst.*, 28(4):619–695, 2006.

[58] P. Klint, T. van der Storm, and J. J. Vinju. Rascal: a domain specific language for source code analysis and manipulation. In A. Walenstein and S. Schupp, editors, *Proceedings of the IEEE International Working Conference on Source Code Analysis and Manipulation (SCAM 2009)*, pages 168–177, 2009.

[59] D. E. Knuth. Semantics of context-free languages. *Mathematical Systems Theory*, 2(2):127–145, 1968.

[60] M. Z. Kwiatkowska, G. Norman, and D. Parker. Prism: Probabilistic symbolic model checker. In *Computer Performance Evaluation / TOOLS*, pages 200–204, 2002.

[61] L. Lee. Fast context-free grammar parsing requires fast boolean matrix multiplication. *J. ACM*, 49(1):1–15, 2002.

[62] X. Li, Z. Liu, and J. He. A formal semantics of UML sequence diagram. In *Australian Software Engineering Conference*, pages 168–177, 2004.

[63] J. C. Martin. *Introduction to Languages and The Theory of Computation*. McGraw-Hil, 2010.

[64] M. Martin, B. Livshits, and M. S. Lam. Finding application errors and security flaws using PQL: a Program Query Language. In *OOPLSLA*, 2005.

[65] B. Meyer. *Object-Oriented Software Construction*. Prentice Hall, 2 edition, 1997.

[66] L. Mikhajlov and E. Sekerinski. A study of the fragile base class problem. In *ECOOP*, 1998.

[67] R. Milner. Fully abstract models of typed λ-calculi. *Theoretical Comput. Sci.*, 4:1–22, 1977.

[68] R. Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer, 1980.

[69] R. Milner. *Communicating and Mobile Systems: The π-Calculus*. Cambridge University Press, New York, NY, USA, 1999.

[70] B. Nobakht, M. M. Bonsangue, F. S. de Boer, and S. de Gouw. Monitoring method call sequences using annotations. In *FACS*, pages 53–70, 2010.

[71] A. Okhotin. Conjunctive and Boolean grammars: The true general case of the context-free grammars. *Computer Science Review*, 9:27–59, 2013.

[72] T. J. Parr and R. W. Quong. Adding semantic and syntactic predicates to LL(k): pred-LL(k). In *In Computational Complexity*, pages 263–277. Springer-Verlag, 1994.

[73] B. C. Pierce. *Types and programming languages*. MIT Press, 2002.

[74] A. Pnueli. The temporal logic of programs. In *Foundations of Computer Science, 1977., 18th Annual Symposium on*, pages 46–57, 1977.

[75] A. Pnueli and A. Zaks. PSL model checking and run-time verification via testers. In *FM*, pages 573–586, 2006.

[76] V. R. Pratt. Semantical considerations on Floyd-Hoare logic. In *FOCS*, pages 109–121, 1976.

[77] J. C. Reynolds. Separation logic: A logic for shared mutable data structures. In *LICS*, pages 55–74, 2002.

[78] D. Sangiorgi and D. Walker. *The Pi-Calculus - a theory of mobile processes*. Cambridge University Press, 2001.

[79] I. Schaefer, L. Bettini, V. Bono, F. Damiani, and N. Tanzarella. Delta-oriented programming of software product lines. In *SPLC*, pages 77–91, 2010.

[80] J. Schäfer and A. Poetzsch-Heffter. JCoBox: Generalizing active objects to concurrent components. In *European Conference on Object-Oriented Programming (ECOOP) (ECOOP'10)*, volume 6183 of *Lecture Notes in Computer Science*, pages 275–299. Springer-Verlag, June 2010.

[81] M. Sipser. *Introduction to the theory of computation*. PWS Publishing Company, 1997.

[82] M. Sirjani, A. Movaghar, A. Shali, and F. S. de Boer. Modeling and verification of reactive systems using Rebeca. *Fundam. Inform.*, 63(4):385–410, 2004.

[83] V. Stolz and F. Huch. Runtime verification of concurrent Haskell programs. *Electr. Notes Theor. Comput. Sci.*, 113:201–216, 2005.

[84] L. G. Valiant. General context-free recognition in less than cubic time. *J. Comput. Syst. Sci.*, 10(2):308–315, 1975.

[85] J. van den Berg and B. Jacobs. The LOOP Compiler for Java and JML. In *TACAS*, pages 299–312, 2001.

[86] P. H. J. van Eijk, C. Vissers, and M. Diaz, editors. *Formal Description Technique Lotos: Results of the Esprit Sedos Project*. Elsevier Science Inc., New York, NY, USA, 1989.

[87] W. Visser, K. Havelund, G. P. Brat, S. Park, and F. Lerda. Model checking programs. *Autom. Softw. Eng.*, 10(2):203–232, 2003.

[88] P. Y. H. Wong, E. Albert, R. Muschevici, J. Proença, J. Schäfer, and R. Schlatte. The ABS tool suite: modelling, executing and analysing distributed adaptable Object-Oriented systems. *STTT*, 14(5):567–588, 2012.