Formal Methods for Conformance Testing: Theory Can Be Practical

Ed Brinksma

Chair of Formal Methods and Tools,
Faculty of Computer Science, University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands
`brinksma@cs.utwente.nl`

Extended Abstract

Although testing is the most widely used technique to control the quality of software systems, it is a topic that, until relatively recently, has received scant attention from the computer research community. Although some pioneering work was already done a considerable time ago [Cho78,GG83,How78,Mye79], the testing of software systems has never become a mainstream activity of scientific research. The reasons that are given to explain this situation usually include arguments to the effect that testing as a technique is inferior to verification – *testing can show only the presence of errors, not their absence* – and that we should therefore concentrate on developing theory and tools for the latter. It has also been frequently said that testing is by its very nature a non-formal activity, where formal methods and related tools are at best of little use.

The first argument is incorrect in the sense that it gives an incomplete picture of the situation. Testing is inferior to verification if the verification model can be assumed to be correct and if its complexity can be handled correctly by the person and or tool involved in the verification task. If these conditions are not fulfilled, which is frequently the case, then testing is often the only available technique to increase the confidence in the correctness of a system. In this talk we will show that the second argument is flawed as well. It is based on the identification of testing with robustness testing, where it is precisely the objective to find out how the system behaves under unspecified circumstances. This excludes the important activity of *conformance testing*, which tries to test the extent to which system behaviour conforms to its specification. It is precisely in this area where formal methods and tools can help to derive tests systematically from specifications, which is a great improvement over laborious, error-prone and costly manual test derivation.

In our talk we show how the process algebraic testing theory due to De Nicola and Hennessy [DNH84,DeN87], originally conceived out of semantic considerations, may be used to obtain principles for test derivation. We will give an overview of the evolution of these ideas over the past ten years or so, starting with the conformance testing theory of simple synchronously communicating reactive systems [Bri88,Lan90] and leading to realistic systems that involve sophisticated asynchronous message passing mechanisms [Tre96,HT97]. Written

accounts can be found in [BHT97,He98]. We discuss how such ideas have been used to obtain modern test derivation tools, such as TVEDA and TGV [Pha94, CGPT96,FJJV96], and the tool set that is currently being developed in the Côte-de-Resyste project [STW96]. The advantage of a test theory that is based on well-established process algebraic theory is that in principle there exists a clear link between testing and verification, which allows the areas to share ideas and algorithms [FJJV96,VT98]. Time allowing, we look at some of the methodological differences and commonalities between model checking techniques and testing, one of the differences being that of *state space coverage*, and an important commonality that of *test property selection*.

In recent years the research into the use of formal methods and tools for testing reactive systems has seen a considerable growth. An overview of different approaches and school of thought can be found in [BPS98], reporting on the first ever Dagstuhl seminar devoted to testing. The formal treatment of conformance testing based on process algebra and/or concurrency theory is certainly not the only viable approach. An important school of thought is the FSM-testing theory grown out of the seminal work of Chow [Cho78], of which a good overview is given in [LY96]. Another interesting formal approach to testing is based on abstract data type theory [Gau95,BGM91].

References

- [BGM91] G. Bernot, M.-C. Gaudel, and B. Marre. Software testing based on formal specifications: a theory and a tool. *Software Engineering Journal*, 1991 (November): 387–405.
- [Bri88] E. Brinksma. A theory for the derivation of tests. In: S. Aggarwal and K. Sabnani, editors, *Protocol Specification, Testing, and Verification VIII*, 63–74, North-Holland, 1988.
- [BHT97] E. Brinksma, L. Heerink, and J. Tretmans. Developments in testing transition systems. In: M. Kim, S. Kang, and K. Hong, editors, *Tenth Int. Workshop on Testing of Communicating Systems*, 143–166, Chapman & Hall, 1997.
- [BPS98] E. Brinksma, J. Peleska, and M. Siegel, editors, *Test Automation for Reactive Systems – Theory and Practice*, Dagstuhl Seminar report 223 (98361), SchloßDagstuhl, Germany, 1998.
- [Cho78] T.S. Chow. Testing software design modeled by finite-state systems. *IEEE Transactions on Software Engineering*, 4(3):178–187, 1978.
- [CGPT96] M. Clatin, R. Groz, M. Phalippou, and R. Thummel. Two approaches linking test generation with verification techniques. In: A. Cavalli and S. Budkowski, editors, *Eighth Int. Workshop on Testing of Communicating Systems*. Chapman & Hall, 1996.
- [DeN87] R. De Nicola. Extensional equivalences for transition systems. *Acta Informatica*, 24:211–237, 1987.
- [DNH84] R. De Nicola and M.C.B. Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34:83–133, 1984.

- [FJJV96] J.-C. Fernandez, C. Jard, T. Jéron, and C. Viho. Using on-the-fly verification techniques for the generation of test suites. In: R. Alur and T.A. Henzinger, editors, *Computer Aided Verification CAV'96*. LNCS 1102, Springer-Verlag, 1996.
- [Gau95] M.-C. Gaudel. Testing can be formal, too. In: P.D. Mosses, M. Nielsen, and M.I. Schwarzbach, editors, *TAPSOFT'95: Theory and Practice of Software Development*, 82–96, LNCS 915, Springer -Verlag, 1995.
- [GG83] J.B. Goodenough and S.L. Gerhardt. Toward a theory of test data selection. *IEEE Transactions on Software Engineering*, 9(2), 1983.
- [He98] L. Heerink. *Ins and Outs in Refusal Testing*. Docoral dissertation, University of Twente, The Netherlands, 1998.
- [HT97] L. Heerink and J. Tretmans. Refusal Testing for classes of transition systems with inputs and outputs. In: T. Mizuno, N. Shiratori, T. Higashino, and A. Togashi, editors, *Formal Description Techniques and Protocol Specification, Testing, and Verification FORTE X/PSTV XVII*, 23–38, Chapman & Hall, 1997.
- [How78] W.E. Howden. Algebraic program testing. *Acta Informatica*, 10:53–66, 1978.
- [Mye79] G.J. Myers. *The Art of Software Testing*. John Wiley & Sons Inc., 1979.
- [Lan90] R. Langerak. A testing theory for LOTOS using deadlock detection. In: E. Brinksma, G. Scollo, and C.A. Vissers, editors, *Protocol Specification, Testing, and Verification IX*, 87–98, North-Holland, 1990.
- [LY96] D. Lee and M. Yannakakis. Principles and methods for testing finite state machines. *Proceedings of the IEEE*. August 1996.
- [Pha94] M. Phalippou. *Relations d'implémentation et hypothèses de test sur des automates à entrées et sorties*. PhD Thesis, Université de Bordeaux I, France, 1994.
- [STW96] Dutch Technology Foundation STW. *Côte-de-Resyste – CONformance TEsting of REactive SYSTEmS*, project TIF.4111. University of Twente, Eindhoven University of Technology, Philips Research, KPN Research, Utrecht, The Netherlands, 1996. URL: <http://fmt.cs.utwente.nl/projects/CdR-html/>.
- [Tre96] J. Tretmans. Test Generation with inputs, outputs, and quiescence. *Software – Concepts and Tools*, 17(3):103–120, 1996.
- [VT98] R.G. de Vries and J. Tretmans. On-the-fly conformance testing using SPIN. In: G. Holzmann, E. Najm, and A. Serhrouchni, editors, *Fourth Workshop on Automata Theoretic Verification with the SPIN Model Checker*, ENST 98 S 002, 115–128, Paris, France, 1998.