Software Model Checking in the Multicore Era

CENIIT Project Final Report

Ahmed Rezine, ahmed.rezine@liu.se
www.ahmedrezine.com

September 17, 2018

1 Summary of the Most Important Scientific Results

The project resulted in 17 conference papers and 3 journal articles all of which were accepted to international peer-reviewed venues (`http://www.ahmedrezine.com/ceniit-12-04/publications/`). The project made possible the extension of automatic verification techniques to:

New classes of concurrent programs. Concurrent programs exhibit an intractable number of possible interleavings among their concurrent sequential parts. The project allowed for defining and comparing new abstraction based model-checking techniques that could establish or refute safety properties (e.g., assertion violations, runtime errors, deadlocks) for programs creating an arbitrary but finite number of concurrent threads whose correctness might depend on invariants relating values of shared variables to numbers of threads in given states, on synchronization barriers or on complex constructs such as the Habanero Java phasers [1, 2, 3, 4].

Side-channel attacks. Computer hardware makes use of complex micro-architectural constructions (e.g., caches, speculative execution) to optimize program executions. New security attacks leverage such constructions to retrieve sensitive information (e.g., secret crypto keys, other processes' virtual memories). The project made possible starting a framework [5] to formally assess the vulnerability of existing mainstream software to cache-based attacks. This work has been invited to a special issue to the ACM Transactions on Embedded Computing (TECS). This line of work resulted in a Singapore-based project where the CENIIT project leader co-investigates, with Prof.Sudipta Chattopadhyay from SUTD-Singapore, a project on the "Verification and Validation of Side-channel Freedom project". The Singapore-based project is supported by a highly competitive three-years grant from the Singapore Ministry of Education (MOE).

Highly concurrent data structures. Concurrent libraries are prone to errors because complex fine grained locking or non-blocking schemes are adopted. The project allowed for the participation in the development of automatic verification techniques that handle arbitrary numbers of concurrent threads, without limiting the heap size or the data domains and in the absence of garbage collectors (best paper award [6] at TACAS'13 and its journal version [7]).

Weak memory models. These relaxations on the orders at which instructions appear to different threads result in extremely intricate behaviors. The problem appears for highly specialized code meant to be intensively used by concurrent software. The project allowed for the participation in several contributions that resulted in new approaches for the automatic placement of memory fences and for the removal of undesired additional behaviors [8, 9, 10].

Hybrid transactional memories. Hybrid transactional memories leverage on the versioning capabilities of underlying cache coherence protocols to use the existing parallelism while giving the programmer the illusion of atomicity. The project allowed for adapting and developing approaches to establish strict serializability, liveness and coherence of the underlying protocols [11, 12].

String manipulating programs. Many programs and scripts manipulate strings to deal with usernames, internet links, commands, keywords etc. The project allowed for key contributions in defining decision procedures for a logic that combines word equations over string variables denoting words of arbitrary lengths, together with constraints on the length of words, and on the regular languages to which words belong. This work was implemented and allowed for the verification and debugging of programs that were beyond the capabilities of existing decision procedures [13, 14, 15].

2 Summary of degrees and promotions

The PhD thesis of Yunyun Zhu (defended spring 2018, Uppsala) on "Caches, Transactions and Memories: Models, Coherence and Consistency" was partially supervised within the project. The PhD thesis of Zeinab Ganjei (defense planned autumn 2019, Linköping) on the "Algorithmic Verification of Concurrent Programs" is also supervised within the project. The project leader was promoted to assistant professor on February 1st 2016 and a docent lecture is planned for October 1st 2018. The project leader has been appointed vice-chair (proprefekt) since January 2018 of the Computer and Information Science Department (IDA). He has also been director of undergraduate education (2015-2017) of the Software and Systems (SaS) division of the Computer and Information Science Department (IDA).

3 Relevant thesis works and teaching

The project leader was involved in the supervision and the examination of several relevant Bachelor and Masters projects. Among them, [16] explored using SMT solvers to aid test case generation for constrained feature models, [17] extracted analyzable models from multi-threaded programs, [18] studied the effects of mutation testing on safety critical software, [19] studied how to improve MCDC adequate test sets for safety critical software to be RORG adequate, and [20] studied the applicability of learning based techniques leveraging model-checking tools to improve testing of safety critical software.

The project leader teaches/taught a software verification course (advanced level), a satisfiability modulo theory (SMT) and optimization course (graduate level), and an introduction to automatic verification course (graduate level). He also participates in the teaching of software security (advanced level), Software testing (advanced level), and advanced software engineering (advanced level).

4 Funded persons

The project funded 30% of the project leader's costs (salary, premises, overhead) in addition to the participation to conferences and workshops.

5 Industrial Connections made possible by the project

The project allowed for collaborations with Saab via several student projects [18, 19] and the participation in the Saab led Vinnova NFFP6 project PILOT - Platform Independent Level of Testing. There has also been collaborations with Ericsson via student projects [16].

6 New research group

The project allowed the applicant to supervise two PhD students, one in Uppsala and one in Linköping, to conduct world-class research, to collaborate with national and international researchers and to introduce automatic verification techniques in graduate and undergraduate courses at LiU. In collaboration with the research activities of the ESLAB group at IDA, the CENIIT project will have enabled fostering the beginning of a formal verification group at LiU.

References

[1] G. Delzanno and A. Rezine, "A lightweight regular model checking approach for parameterized systems," *International Journal on Software Tools for Technology Transfer*, vol. 14, pp. 207–222, aug 2012.

[2] Z. Ganjei, A. Rezine, P. Eles, and Z. Peng, "Lazy constrained monotonic abstraction," in *Verification, Model Checking, and Abstract Interpretation - 17th International Conference, VMCAI 2016, St. Petersburg, FL, USA, January 17-19, 2016. Proceedings* (B. Jobstmann and K. R. M. Leino, eds.), vol. 9583 of *Lecture Notes in Computer Science*, pp. 147–165, Springer, 2016.

[3] Z. Ganjei, A. Rezine, P. Eles, and Z. Peng, "Counting dynamically synchronizing processes," *International Journal on Software Tools for Technology Transfer*, vol. 18, pp. 517–534, jan 2016.

[4] Z. Ganjei, A. Rezine, P. Eles, and Z. Peng, "Safety verification of phaser programs," in *2017 Formal Methods in Computer Aided Design, FMCAD 2017, Vienna, Austria, October 2-6, 2017* (D. Stewart and G. Weissenbacher, eds.), pp. 68–75, IEEE, 2017.

[5] S. Chattopadhyay, M. Beck, A. Rezine, and A. Zeller, "Quantifying the information leak in cache attacks via symbolic execution," in *Proceedings of the 15th ACM-IEEE International Conference on Formal Methods and Models for System Design, MEMOCODE 2017, Vienna, Austria, September 29 - October 02, 2017* (J. Talpin, P. Derler, and K. Schneider, eds.), pp. 25–35, ACM, 2017.

[6] P. A. Abdulla, F. Haziza, L. Holík, B. Jonsson, and A. Rezine, "An integrated specification and verification technique for highly concurrent data structures," in *Tools and Algorithms for the Construction and Analysis of Systems - 19th International Conference, TACAS 2013, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2013, Rome, Italy, March 16-24, 2013. Proceedings* (N. Piterman and S. A. Smolka, eds.), vol. 7795 of *Lecture Notes in Computer Science*, pp. 324–338, Springer, 2013.

[7] P. A. Abdulla, F. Haziza, L. Holík, B. Jonsson, and A. Rezine, "An integrated specification and verification technique for highly concurrent data structures for highly concurrent data structures," *International Journal on Software Tools for Technology Transfer*, vol. 19, pp. 549–563, mar 2017.

[8] P. A. Abdulla, M. F. Atig, Y. Chen, C. Leonardsson, and A. Rezine, "Counter-example guided fence insertion under TSO," in *Tools and Algorithms for the Construction and Analysis of Systems - 18th International Conference, TACAS 2012, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2012, Tallinn, Estonia, March 24 - April 1, 2012. Proceedings* (C. Flanagan and B. König, eds.), vol. 7214 of *Lecture Notes in Computer Science*, pp. 204–219, Springer, 2012.

[9] P. A. Abdulla, M. F. Atig, Y. Chen, C. Leonardsson, and A. Rezine, "Automatic fence insertion in integer programs via predicate abstraction," in *Static Analysis - 19th International Symposium, SAS 2012, Deauville, France, September 11-13, 2012. Proceedings* (A. Miné and

D. Schmidt, eds.), vol. 7460 of *Lecture Notes in Computer Science*, pp. 164–180, Springer, 2012.

[10] P. A. Abdulla, M. F. Atig, Y. Chen, C. Leonardsson, and A. Rezine, “Memorax, a precise and sound tool for automatic fence insertion under TSO,” in *Tools and Algorithms for the Construction and Analysis of Systems - 19th International Conference, TACAS 2013, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2013, Rome, Italy, March 16-24, 2013. Proceedings* (N. Piterman and S. A. Smolka, eds.), vol. 7795 of *Lecture Notes in Computer Science*, pp. 530–536, Springer, 2013.

[11] P. A. Abdulla, S. Dwarkadas, A. Rezine, A. Shriraman, and Y. Zhu, “Verifying safety and liveness for the flextm hybrid transactional memory,” in *Design, Automation and Test in Europe, DATE 13, Grenoble, France, March 18-22, 2013* (E. Macii, ed.), pp. 785–790, EDA Consortium San Jose, CA, USA / ACM DL, 2013.

[12] P. A. Abdulla, M. F. Atig, Z. Ganjei, A. Rezine, and Y. Zhu, “Verification of cache coherence protocols wrt. trace filters,” in *Formal Methods in Computer-Aided Design, FMCAD 2015, Austin, Texas, USA, September 27-30, 2015.* (R. Kaivola and T. Wahl, eds.), pp. 9–16, IEEE, 2015.

[13] P. A. Abdulla, M. F. Atig, Y. Chen, L. Holík, A. Rezine, P. Rümmer, and J. Stenman, “String constraints for verification,” in *Computer Aided Verification - 26th International Conference, CAV 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 18-22, 2014. Proceedings* (A. Biere and R. Bloem, eds.), vol. 8559 of *Lecture Notes in Computer Science*, pp. 150–166, Springer, 2014.

[14] P. A. Abdulla, M. F. Atig, Y. Chen, L. Holík, A. Rezine, P. Rümmer, and J. Stenman, “Norn: An SMT solver for string constraints,” in *Computer Aided Verification - 27th International Conference, CAV 2015, San Francisco, CA, USA, July 18-24, 2015, Proceedings, Part I* (D. Kroening and C. S. Pasareanu, eds.), vol. 9206 of *Lecture Notes in Computer Science*, pp. 462–469, Springer, 2015.

[15] P. A. Abdulla, M. F. Atig, Y. Chen, B. P. Diep, L. Holík, A. Rezine, and P. Rümmer, “Flatten and conquer: a framework for efficient analysis of string constraints,” in *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2017, Barcelona, Spain, June 18-23, 2017* (A. Cohen and M. T. Vechev, eds.), pp. 602–617, ACM, 2017.

[16] P. Borek, “Smt aided test case generation for constrained feature models,” 2014.

[17] A. Karetsos, “Extracting analyzable models from multi-threaded programs,” 2015.

[18] C. Nylén, “Improving mcdc adequate test sets for safety critical software to be rorg adequate,” 2015.

[19] R. Johnsson and N. Svensson, “Effects of mutation testing on safety critical software,” 2017.

[20] S. Stenlund, “Testing safety critical avionics software using lbtest,” 2016.