Nonlinear Polynomials, Interpolants and Invariant Generation for System Analysis*

Deepak Kapur[1]

Dept. of Computer Science, University of New Mexico, USA, kapur@cs.unm.edu

Abstract. Invariant properties at various program locations play a critical role in enhancing confidence in the reliability of software, hardware and hybrid systems. While there has recently been considerable interest in researching heuristics for generating loops invariants, almost all developments have focused on generating invariants typically handled using SMT solvers including propositional formulas, difference and octagonal formulas and linear formulas. While we have been investigating methods based on symbolic computation algorithms including Gröbner basis and approximate quantifier elimination for over a decade (see [1, 7, 8, 11, 4, 5, 9, 10, 6, 13, 12, 2, 3] for some of our papers), the SMT and CAV community have only recently started considering nonlinear polynomial invariants since many programs, especially linear filters, hybrid systems, and other applications, need nonlinear invariants for analysis of their behavior.

We present an overview of our research with a focus on our most recent work on nonlinear invariant [2] and nonlinear interpolant generation [3] from the perspective of their role in software and hybrid system analysis. Our approach is in sharp contrast to some recent approaches in which nonlinear polynomials are approximated using linear inequalities and symbolic-numeric techniques. We will outline problems and challenges faced by the SMT community for incorporate nonlinear reasoning in their tools as well as issues we encounter in our research.

References

1. David Cyrluk and Deepak Kapur. Reasoning about nonlinear inequality constraints: a multi-level approach. In *Proc. of Image Understanding Workshop*, pages 904–914, 1989.
2. S. Falke and D. Kapur. When is a formula a loop invariant? In *Logic, Rewriting and Concurrency: Essays dedicated to Jose Meseguer on the Occasion of His* 65^{th} *Birthday*, LNCS 9200, pages 264–286. Springer-Verlag, 2015.
3. Ting Gan, Liyun Dai, Bican Xia, Naijun Zhan, Deepak Kapur, and Mingshuai Chen. Interpolant synthesis for quadratic polynomial inequalities and combination with euf. In *Proceedings, International Joint Conference on Auomated Reasoning (IJCAR) 2016*, Lecture Notes in Computer Science, pages 195–212. Springer, 2016.

* This research has been partially supported by the NSF awards CCF-1248069 and DMS-1217054 as well as a grant for Visiting Professorship for Senior International Scientists, the Chinese Academy of Sciences.

4. D. Kapur. Automatically Generating Loop Invariants using Quantifier Elimination. Technical report, Department of Computer Science, University of New Mexico, Albuquerque, NM, USA, 2003.
5. D. Kapur. A quantifier-elimination based heuristic for automatically generating inductive assertions for programs. *Journal of Systems Science and Complexity*, 19(3):307–330, 2006.
6. D. Kapur. Elimination techniques for program analysis. In *Proceedings of Harald Ganzinger Memorial Workshop, 2006*. Springer, 2012.
7. Deepak Kapur and R. K. Shyamasundar. Synthesizing controllers for hybrid systems. In *Proceedings of Hybrid and Real-Time Systems, International Workshop. HART'97*, Lecture Notes in Computer Science, pages 361–375, Grenoble, France, March 1997. Springer.
8. E. Rodríguez-Carbonell and D. Kapur. Automatic Generation of Polynomial Loop Invariants: Algebraic Foundations. *Intl. Symp. on Symbolic and Algebraic Computation (ISSAC)*, pages 266–273, July 2004.
9. E. Rodríguez-Carbonell and D. Kapur. Automatic generation of polynomial invariants of bounded degree using abstract interpretation. *J. of Science of Programming*, 64(1):54–75, 2007.
10. E. Rodríguez-Carbonell and D. Kapur. Generating all polynomial invariants in simple loops. *J. of Symbolic Computation*, 42(4):443–476, 2007.
11. E. Rodríguez-Carbonell and D. Kapur. An Abstract Interpretation Approach for Automatic Generation of Polynomial Invariants. *11th Symposium on Static Analysis (SAS)*, pages 280–295, August 2004.
12. H. Zhao, N. Zhan, and D. Kapur. Synthesizing switching controllers for hybrid systems by generating invariants. In *Proc. Festchrift Symp. in honor of He Jifeng*, pages 354–373, 2013.
13. Hengjun Zhao, Naijun Zhan, Deepak Kapur, and Kim G. Larsen. A "hybrid" approach for synthesizing optimal controllers of hybrid systems: A case study of the oil pump industrial example. In Dimitra Giannakopoulou and Dominique Méry, editors, *FM 2012: Formal Methods - 18th International Symposium, Paris, France, August 27-31, 2012. Proceedings*, volume 7436 of *Lecture Notes in Computer Science*, pages 471–485. Springer, 2012.