Program Certification by Higher-Order Model Checking

Naoki Kobayashi

The University of Tokyo

Abstract. Model checking of higher-order recursion schemes or (collapsible) higher-order pushdown automata (higher-order model checking, for short) is a generalization of finite state and pushdown model checking, which has been extensively studied in the last decade [1–11, 15–17]. Higher-order recursion schemes are essentially terms of the simply-typed λ-calculus with recursion and tree constructors; therefore, they serve as good models for higher-order functional programs. Indeed, various verification problems for higher-order functional programs can be easily reduced to higher-order model checking, and automated verification tools have been developed based on the reduction [9, 12–14, 18].

In the talk, I will first provide a brief introduction to higher-order model checking and its applications to higher-order program verification. I will then discuss higher-order model checking from the viewpoint of certificates. In particular, I plan to discuss the following questions: (i) How can we certify the result of program verification based on higher-order model checking? (ii) Why does higher-order model checking work at all, despite its extremely high worst-case complexity?

References

1. Aehlig, K.: A finite semantics of simply-typed lambda terms for infinite runs of automata. Logical Methods in Computer Science 3(3) (2007)
2. Broadbent, C.H., Carayol, A., Hague, M., Serre, O.: A Saturation Method for Collapsible Pushdown Systems. In: Czumaj, A., Mehlhorn, K., Pitts, A., Wattenhofer, R. (eds.) ICALP 2012, Part II. LNCS, vol. 7392, pp. 165–176. Springer, Heidelberg (2012)
3. Broadbent, C.H., Carayol, A., Ong, C.-H.L., Serre, O.: Recursion schemes and logical reflection. In: Proceedings ofo LICS 2010, pp. 120–129. IEEE Computer Society Press (2010)
4. Carayol, A., Serre, O.: Collapsible pushdown automata and labeled recursion schemes: Equivalence, safety and effective selection. In: Proceedings of LICS 2012. IEEE Computer Society Press (2012)
5. Hague, M., Murawski, A., Ong, C.-H.L., Serre, O.: Collapsible pushdown automata and recursion schemes. In: Proceedings of 23rd Annual IEEE Symposium on Logic in Computer Science, pp. 452–461. IEEE Computer Society (2008)
6. Hague, M., Ong, C.-H.L.: Symbolic backwards-reachability analysis for higher-order pushdown systems. Logical Methods in Computer Science 4(4) (2008)
7. Knapik, T., Niwiński, D., Urzyczyn, P.: Higher-Order Pushdown Trees Are Easy. In: Nielsen, M., Engberg, U. (eds.) FOSSACS 2002. LNCS, vol. 2303, pp. 205–222. Springer, Heidelberg (2002)

C. Hawblitzel and D. Miller (Eds.): CPP 2012, LNCS 7679, pp. 9–10, 2012.

8. Kobayashi, N.: Model-checking higher-order functions. In: Proceedings of PPDP 2009, pp. 25–36. ACM Press (2009)
9. Kobayashi, N.: Types and higher-order recursion schemes for verification of higher-order programs. In: Proceedings of ACM SIGPLAN/SIGACT Symposium on Principles of Programming Languages (POPL), pp. 416–428 (2009)
10. Kobayashi, N.: A Practical Linear Time Algorithm for Trivial Automata Model Checking of Higher-Order Recursion Schemes. In: Hofmann, M. (ed.) FOSSACS 2011. LNCS, vol. 6604, pp. 260–274. Springer, Heidelberg (2011)
11. Kobayashi, N., Ong, C.-H.L.: A type system equivalent to the modal mu-calculus model checking of higher-order recursion schemes. In: Proceedings of LICS 2009, pp. 179–188. IEEE Computer Society Press (2009)
12. Kobayashi, N., Sato, R., Unno, H.: Predicate abstraction and CEGAR for higher-order model checking. In: Proceedings of ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), pp. 222–233 (2011)
13. Kobayashi, N., Tabuchi, N., Unno, H.: Higher-order multi-parameter tree transducers and recursion schemes for program verification. In: Proceedings of ACM SIGPLAN/SIGACT Symposium on Principles of Programming Languages (POPL), pp. 495–508 (2010)
14. Lester, M.M., Neatherway, R.P., Ong, C.-H.L., Ramsay, S.J.: Model checking liveness properties of higher-order functional programs. In: Proceedings of ML Workshop 2011 (2011)
15. Neatherway, R.P., Ramsay, S.J., Ong, C.-H.L.: A traversal-based algorithm for higher-order model checking. In: ACM SIGPLAN International Conference on Functional Programming (ICFP 2012), pp. 353–364 (2012)
16. Ong, C.-H.L.: On model-checking trees generated by higher-order recursion schemes. In: LICS 2006, pp. 81–90. IEEE Computer Society Press (2006)
17. Ong, C.-H.L.: Models of higher-order computation: Recursive schemes and collapsible pushdown automata. In: Logics and Languages for Reliability and Security, pp. 263–299. IOS Press (2010)
18. Ong, C.-H.L., Ramsay, S.: Verifying higher-order programs with pattern-matching algebraic data types. In: Proceedings of ACM SIGPLAN/SIGACT Symposium on Principles of Programming Languages (POPL), pp. 587–598 (2011)