Types and Recursion Schemes for Higher-Order Program Verification

Naoki Kobayashi

Tohoku University

Abstract. Higher-order recursion schemes (recursion schemes, for short) are expressive grammars for describing infinite trees. The modal μ-calculus model checking problem for recursion schemes ("Given a recursion scheme G and a modal μ-calculus formula φ, does the tree generated by G satisfy φ?") has been a hot research topic in the theoretical community for recent years [1,2,3,4,5,6,7]. In 2006, it has been shown to be decidable, and n-EXPTIME complete (where n is the order of a recursion scheme) by Ong [5].

The model checking of recursion schemes has recently turned out to be a good basis for verification of higher-order functional programs, just as finite state model checking for programs with while-loops, and pushdown model checking for programs with first-order recursion. First, various program analysis/verification problems such as reachability, flow analysis, and resource usage verification (or equivalently, type-state checking) can be easily transformed into model-checking problems for recursion schemes [8]. Combined with a model checking algorithm for recursion schemes, this yields a sound, complete, and automated verification method for the simply-typed λ-calculus with recursion and finite base types such as booleans. Secondly, despite the extremely high worst-case time complexity (i.e. n-EXPTIME completeness) of the model checking problem for recursion schemes, our type-based model-checking algorithm [9] turned out to run reasonably fast for realistic programs. We have implemented a prototype model checker for recursion schemes TRecS, and are currently working to construct a software model checker for a subset of ML on top of it.

The talk will summarize our recent results [8,9,10,11] on the model checking of recursion schemes as well as its applications to higher-order program verification, and discuss future perspectives.

References

1. Knapik, T., Niwinski, D., Urzyczyn, P.: Deciding monadic theories of hyperalgebraic trees. In: Abramsky, S. (ed.) TLCA 2001. LNCS, vol. 2044, pp. 253–267. Springer, Heidelberg (2001)
2. Knapik, T., Niwinski, 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)
3. Aehlig, K., de Miranda, J.G., Ong, C.-H.L.: The monadic second order theory of trees given by arbitrary level-two recursion schemes is decidable. In: Urzyczyn, P. (ed.) TLCA 2005. LNCS, vol. 3461, pp. 39–54. Springer, Heidelberg (2005)

Z. Hu (Ed.): APLAS 2009, LNCS 5904, pp. 2–3, 2009.

4. Knapik, T., Niwinski, D., Urzyczyn, P., Walukiewicz, I.: Unsafe grammars and panic automata. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 1450–1461. Springer, Heidelberg (2005)
5. Ong, C.-H.L.: On model-checking trees generated by higher-order recursion schemes. In: LICS 2006, pp. 81–90. IEEE Computer Society Press, Los Alamitos (2006)
6. Aehlig, K.: A finite semantics of simply-typed lambda terms for infinite runs of automata. Logical Methods in Computer Science 3(3) (2007)
7. 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, Los Alamitos (2008)
8. Kobayashi, N.: Types and higher-order recursion schemes for verification of higher-order programs. In: Proceedings of POPL 2009, pp. 416–428 (2009)
9. Kobayashi, N.: Model-checking higher-order functions. In: Proceedings of PPDP 2009. ACM Press, New York (2009)
10. Kobayashi, N., Ong, C.-H.L.: Complexity of model checking recursion schemes for fragments of the modal mu-calculus. In: Albers, S., et al. (eds.) ICALP 2009. LNCS, vol. 5556, pp. 223–234. Springer, Heidelberg (2009)
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, Los Alamitos (2009)