Automata, Logic and Games for the λ-Calculus*

C.-H. Luke Ong

University of Oxford

Automata, logic and games provide the mathematical theory that underpins the model checking of reactive systems:

- *automata* on infinite words and trees as models of computation for state-based systems,
- *logical systems* such as temporal and modal logics for specifying correctness properties, and
- *two-person games* as a mathematical model of the interactions between a system and its environment.

An elegant and fundamental result in the theory of automata, logic and games [3] is the correspondence between alternating parity tree automata (APT), the modal mu-calculus L_μ, and parity games, which are standard formalisms for algorithmic reasoning about trees. On the one hand, the mu-calculus model-checking problem and the PARITY decision problem (Does Verifier have a winning strategy in a given parity game?) are interreducible [17, 19]. On the other, modal mu-calculus and alternating parity tree automata are recursively equivalent for defining tree languages [2, 8, 19].

Research in model checking has traditionally concerned itself with the verification of properties of "ground type objects" such as words or trees. A recent trend in algorithmic verification is *higher-order model checking* [4, 9, 10], which is the model checking of infinite trees generated by the $\lambda\mathbf{Y}$-calculus (simply-typed lambda calculus extended with fixpoint operators) [13, 15] or, equivalently, recursion schemes. Higher-order model checking has been applied with some success to the verification of higher-type functional programs [5, 7, 14, 11]. In this model checking approach, the verification problem is reduced to the model checking of Böhm trees, which are the computation trees of functional programs.

Our work is motivated by the question: *what is the automata-logic-games correspondence for (higher-type) Böhm trees?* Simply-typed Böhm trees are ordered ranked trees extended with binders in the form of lambda-abstractions; which may be viewed as higher-order functions on trees. Indeed, one may well ask if such a correspondence is plausible, since Clairambault and Murawski [1] have considered a monadic second order logic over a class of binding structures and shown the model checking problem to be undecidable. However, we develop just such a correspondence for Böhm trees. In the following we discuss the main ideas.

The starting point is recent work by Tsukada and Ong [18] on the model checking problem for higher-type (possibly infinite) Böhm trees. They introduced

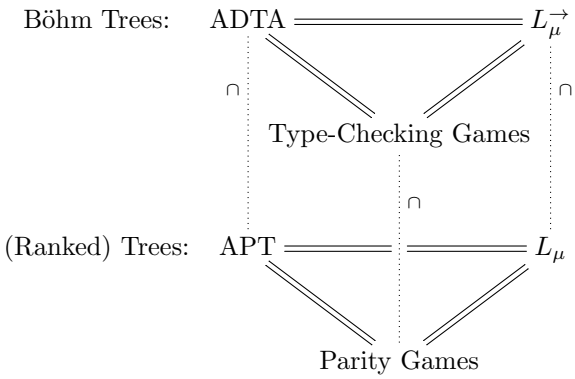
* Abstract of an invited talk presented at ICLA 2017.

a notion of type [6] for Böhm trees: a Böhm tree u is said to have type σ if Verifier has a winning strategy in the corresponding *type-checking game*. Tsukada and Ong showed that the type checking of $\lambda\mathbf{Y}$-definable Böhm trees is decidable. This type-checking game is the games component of the new trio for higher types.

The automata component of the trio is *alternating dependency tree automata* (ADTA), which was first introduced by Stirling [16] for finite binding trees, in order to characterise solution sets of the Higher-Order Matching Problem. We extend ADTA to infinite binding trees with ω-regular winning conditions. ADTA are closed under union, intersection and complementation. The emptiness problem for nondeterministic dependency tree automata is decidable, but undecidable for alternating dependency tree automata [12]. A key result is that types and ADTA are effectively equivalent[1] for defining languages of Böhm trees. As a corollary, the ADTA acceptance problem for $\lambda\mathbf{Y}$-definable Böhm trees is decidable.

The logic for describing the corresponding correctness properties is *higher-type mu-calculus* $L_\mu^\rightarrow$, which extends ordinary modal mu-calculus with predicates for detecting variables, and corresponding constructs for detecting λ-abstractions. There is a characterisation of the set-theoretic semantics of $L_\mu^\rightarrow$ by a model checking game. Furthermore $L_\mu^\rightarrow$ and ADTA are recursively equivalent for defining languages of Böhm trees.

Thus there is an exact automata-logic-games correspondence for Böhm trees at higher types, which naturally extends the classical correspondence for ordinary trees, as illustrated by the following diagram.



[1] We suppress a delicate distinction between types and a subsystem of *parity permissive types*. There is a corresponding distinction between ADTA and a subclass of *parity permissive ADTA*. The expressive equivalence result holds both generally and when restricted to the parity permissive subsystems.

Acknowledgements. This is based on joint work with Matthew Hague, Steven Ramsay, and Takeshi Tsukada, partially funded by EPSRC UK. Part of the work was done while the authors were visiting the Institute for Mathematical Sciences, National University of Singapore in 2016. The visit was partially supported by the Institute.

References

1. P. Clairambault and A. S. Murawski. Böhm trees as higher-order recursive schemes. In *Proceedings of IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2013)*, volume 24 of *LIPIcs*, pages 91–102. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.
2. E. A. Emerson and C. S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 1-4 October 1991*, pages 368–377, 1991.
3. E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.
4. T. Knapik, D. Niwiński, and P. Urzyczyn. Higher-order pushdown trees are easy. In *FoSSaCS 2002*, pages 205–222, 2002.
5. N. Kobayashi. Model Checking Higher-Order Programs. *Journal of the ACM*, 60(3):1–62, June 2013.
6. N. Kobayashi and C.-H. L. Ong. A type system equivalent to the modal mu-calculus model checking of higher-order recursion schemes. In *Proceedings of the 24th Annual IEEE Symposium on Logic in Computer Science, LICS 2009, 11-14 August 2009, Los Angeles, CA, USA*, pages 179–188, 2009.
7. N. Kobayashi, R. Sato, and H. Unno. Predicate abstraction and CEGAR for higher-order model checking. In M. W. Hall and D. A. Padua, editors, *PLDI*, pages 222–233. ACM, 2011.
8. O. Kupferman, M. Y. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the ACM*, 47(2):312–360, mar 2000.
9. C.-H. L. Ong. On model-checking trees generated by higher-order recursion schemes. In *Proceedings of 21th IEEE Symposium on Logic in Computer Science (LICS 2006)*, pages 81–90. IEEE Computer Society, 2006.
10. C.-H. L. Ong. Higher-order model checking: An overview. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 1–15, 2015.
11. C.-H. L. Ong and S. J. Ramsay. Verifying higher-order functional programs with pattern-matching algebraic data types. In *POPL 2011*, volume 46, pages 587–598, Jan. 2011.
12. C.-H. L. Ong and N. Tzevelekos. Functional Reachability. *2009 24th Annual IEEE Symposium on Logic In Computer Science (LICS 2009)*, pages 286–295, Aug. 2009.
13. R. A. Platek. *Foundations of Recursion Theory.* PhD thesis, Standford University, 1966.
14. S. J. Ramsay, R. P. Neatherway, and C.-H. L. Ong. A type-directed abstraction refinement approach to higher-order model checking. In *The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20-21, 2014*, pages 61–72. ACM, 2014.

15. D. S. Scott. A type-theoretical alternative to iswim, cuch, OWHY. *Theor. Comput. Sci.*, 121(1&2):411–440, 1993.
16. C. Stirling. Dependency tree automata. In *FoSSaCS*, volume 5504 of *LNCS*, pages 92–106. Springer, 2009.
17. R. S. Streett and E. A. Emerson. An automata theoretic decision procedure for the propositional mu-calculus. *Inf. Comput.*, 81(3):249–264, 1989.
18. T. Tsukada and C.-H. L. Ong. Compositional higher-order model checking via ω-regular games over böhm trees. In *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, pages 78:1–78:10, 2014.
19. I. Walukiewicz. Pushdown processes: Games and model-checking. *Inf. Comput.*, 164(2):234–263, 2001.