Hierarchies of Infinite Structures Generated by Pushdown Automata and Recursion Schemes

C.-H.L. Ong

Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford OX1 3QD, England
users.comlab.ox.ac.uk/luke.ong/

Abstract. Higher-order recursion schemes and higher-order pushdown automata are closely related methods for generating infinite hierarchies of infinite structures. Subsuming well-known classes of models of computation, these rich hierarchies (of word languages, trees, and graphs respectively) have excellent model-checking properties. In this extended abstract, we survey recent expressivity and decidability results about these infinite structures.

A class of infinite-state systems of fundamental importance to software verification are *pushdown automata*. It is an old idea that first-order imperative programs with recursive procedures can be accurately modelled by pushdown automata (see e.g. [1] for a precise account). Viewed abstractly, a pushdown automaton (PDA) is a finitely-presentable infinite-state transition graph in which a state (vertex) is a reachable configuration that carries a finite but unbounded amount of information, namely, the contents of the stack. Müller and Schupp [2] have proved that the monadic second-order (MSO) theory of the transition graph of a pushdown automaton is decidable, and the automata-theoretic technique using two-way alternating automata [3] provides a direct (elementary) model-checking decision procedure for temporal properties.

1 Pushdown Automata and Safe Recursion Schemes

There is in fact an infinite hierarchy of *higher-order pushdown automata*. First introduced by Maslov [4,5] as accepting devices for word languages, order-0 and order-1 pushdown automata are, by definition, the finite-state and standard pushdown automata respectively. An order-2 PDA has an order-2 stack, which is a stack of order-1 (i.e. ordinary) stacks; in addition to the usual (first-order actions) *push* and *pop*, it has an order-2 $push_2$ that duplicates the top 1-stack, and an order-2 pop_2 that pops the entire top 1-stack. As n varies over the natural numbers, the languages accepted by order-n PDA form an infinite hierarchy. In *op. cit.* Maslov showed that the hierarchy can be defined equivalently as languages generated by *higher-order indexed grammars*, generalising indexed grammars in the sense of Aho [6]. Yet another characterisation of Maslov's hierarchy was given by Damm and Goerdt [7,8]: they studied *higher-order recursion*

L. Kučera and A. Kučera (Eds.): MFCS 2007, LNCS 4708, pp. 15–21, 2007.

schemes that satisfy the constraint of *derived types*, and showed that the word languages generated by order-n such schemes coincide with those accepted by order-n PDA. The low orders of the Maslov hierarchy are well-known – orders 0, 1 and 2 are respectively the regular, context-free and indexed languages, though little is known about languages at higher orders.

Higher-order PDA as a generating device for (possibly infinite) ranked trees was first studied by Knapik, Niwiński and Urzyczyn in a TLCA'01 paper [9]. As in the case of word languages, an infinite hierarchy of trees are thus defined. Lower orders of the pushdown hierarchy are well-known classes of trees: orders 0, 1 and 2 are respectively the regular [10], algebraic [11] and hyperalgebraic trees [9]. In a follow-up paper in FoSSaCS'02 [12] Knapik *et al.* considered another method for generating such trees, namely, by higher-order (deterministic) recursion schemes that satisfy the constraint of *safety*[1]. A major result in that work is the equi-expressivity of the two methods as generators of trees; another is that all trees in the pushdown hierarchy have decidable monadic second-order (MSO) theories.

In a MFCS'02 paper, Caucal [14] introduced a tree hierarchy and a graph hierarchy, which are defined by mutual induction using a pair of transformations, one from trees to graphs, and the other from graphs to trees. The order-0 graphs are by definition the finite graphs. Inductively the order-n trees are defined as the unravelling of the order-n graphs; and the order-$(n+1)$ graphs are defined as the *inverse rational mappings* of the order-n trees. Since these transformations preserve the decidability of MSO theories, all infinite structures belonging to the two hierarchies have decidable MSO theories. Caucal's hierarchies turn out to be closely related to higher-order PDA: Carayol and Wöhrle [15] have shown that the order-n graphs in Caucal's graph hierarchy are exactly the ε-closure of the configuration graphs of order-n pushdown systems.

To summarise, the hierarchies of infinite structures generated by higher-order pushdown automata are of considerable interest to infinite-state verification:

1. *Expressivity*: They are rich and very general; as tabulated below, the lower orders of the hierarchies are well-known classes of models of computation

	Pushdown (= Safe Recursion Scheme) Hierarchies of		
Order	*Word Languages*	*Trees*	*Graphs*
0	regular	regular	finite
1	context-free	algebraic [11]	prefix-recognisable [16]
2	indexed [6]	hyper-algebraic [9]	
⋮	⋮	⋮	⋮

   though little is known about the structures at higher orders.

[1] As a syntactic constraint, *safety* is equivalent to Damm's *derived types* [7]; see de Miranda's thesis [13] for a proof.

2. *Robustness*: Remarkably, order-n pushdown automata are equi-expressive with order-n recursion schemes that satisfy the syntactic constraint of safety, as generators of word languages [7,8] and of trees [12] respectively.
3. *Excellent model-checking properties*: The infinite structures that are generated have decidable MSO properties. The criterion of MSO decidability is appropriate because the MSO logic is commonly regarded as the gold standard of specification languages for model checking: standard temporal logics such as LTL, ETL, CTL, CTL*, and even modal mu-calculus can all be embedded in MSO; moreover any obvious extension of the logic would break decidability.

2 Recursion Schemes and Collapsible Pushdown Automata

Several questions arise naturally from the preceding equi-expressivity and decidability results.

1. Syntactically awkward, the *safety* constraint [12] seems unnatural. Is it really necessary for MSO decidability? Precisely, do trees that are generated by (arbitrary) recursion schemes have decidable MSO theories?
2. Can the expressivity of (arbitrary) recursion schemes be characterised by an appropriate class of automata (that contains the higher-order pushdown automata)?
3. Does safety constrain expressivity? I.e. is there a tree or a graph that is generated by an unsafe, but *not* by any safe, recursion scheme?

Recent work has provided answers to the first two of these questions, and a partial answer to the third. Using new ideas and techniques from *innocent game semantics* [17], we have proved [18]:

Theorem 1 (MSO decidability). *The modal mu-calculus model checking problem for ranked trees generated by order-n recursion schemes is n-EXPTIME complete, for each $n \geq 0$. Hence these trees have decidable MSO theories.*

To our knowledge, the hierarchy of trees generated by (arbitrary) recursion schemes is the largest, generically-defined class of ranked trees that have decidable MSO theories, subsuming earlier results such as [10,11] and also [12,19,20,21] etc. A novel ingredient in the proof of Theorem 1 is a certain *transference principle* from the tree $[\![G]\!]$ generated by the recursion scheme G – the *value tree* – to an auxiliary *computation tree* $\lambda(G)$, which is in essence an infinite λ-term obtained by unfolding the recursion scheme *ad infinitum*. The transference relies on a strong correspondence theorem between *paths* in the value tree and what we call *traversals* in the computation tree, established using innocent game semantics [17]. This allows us to prove that a given alternating parity tree automaton (APT) has an accepting run-tree over the value tree iff it has an accepting traversal-tree over the computation tree. The second ingredient is the simulation of an accepting traversal-tree by a certain set of annotated paths over the computation tree.

Higher-order recursion schemes are essentially closed terms of the simply-typed lambda calculus with general recursion, generated from uninterpreted first-order function symbols. A fundamental question in higher-type recursion is to characterise the expressivity of higher-order recursion schemes in terms of a class of automata. Thus the results of Damm and Goerdt [8], and of Knapik *et al.* [12], may be viewed as partial answers of the question. An exact correspondence with recursion schemes has never been established before.

Another partial answer was recently obtained by Knapik, Niwiński, Urzyczyn and Walukiewicz. In an ICALP'05 paper [19], they proved that order-2 homogeneously typed (but not necessarily safe) recursion schemes are equi-expressive with a variant class of order-2 pushdown automata called *panic automata*. In recent joint work [22] with Hague, Murawski and Serre, we have given a complete answer to the question. We introduce a new kind of higher-order pushdown automata (which generalises *pushdown automata with links* [23], or equivalently panic automata, to all finite orders), called *collapsible pushdown automata* (CPDA), in which every symbol in the stack has a link to a (necessarily lower-ordered) stack situated somewhere below it. In addition to the higher-order stack operations $push_i$ and pop_i, CPDA have an important operation called *collapse*, whose effect is to "collapse" a stack s to the prefix as indicated by the link from the top_1-symbol of s. A major result of [22] is the following:

Theorem 2 (Equi-expressivity). *For each $n \geq 0$, order-n recursion schemes and order-n collapsible pushdown automata define the same trees.*

Thus order-n CPDA may be viewed as a machine characterisation of order-n recursively-defined lambda-terms, and hence also of order-n innocent strategies (since innocent strategies are a universal model of higher-order recursion schemes). In one direction of the proof, we show that for every (tree-generating) order-n pushdown automaton, there is an order-n recursion scheme that generates the same tree. In the other direction, we introduce an algorithm (as implemented in the tool HOG [24]) translating an order-n recursion scheme G to an order-n CPDA A_G that computes exactly the traversals over the computation tree $\lambda(G)$ and hence paths in the value tree $[\![G]\!]$.

The Equi-Expressivity Theorem has a number of useful consequences. It allows us to translate decision problems on trees generated by recursion schemes to equivalent problems on CPDA and *vice versa*. Chief among them is the Modal Mu-Calculus Model-Checking Problem (equivalently the Alternating Parity Tree Automaton Acceptance Problem); another is the Monadic Second-Order (MSO) Model-Checking Problem. We observe that these problems – concerning infinite structures generated by recursion schemes – reduce to the problem of solving a parity game played on a *collapsible pushdown graph* i.e. the configuration graph of a corresponding collapsible pushdown system (CPDS).

The transfer of techniques goes both ways. Another result in our work [22] is a self-contained (without recourse to game semantics) proof of the solvability of parity games on collapsible pushdown graphs by generalising *standard* techniques in the field:

Theorem 3 (Solvability). *For each $n \geq 0$, solvability of parity games over the configuration graphs of order-n collapsible pushdown systems is n-EXPTIME complete.*

The Theorem subsumes a number of well-known results in the literature [25,26,19]. By appealing to the Equi-Expressivity Theorem, we obtain new proofs for the decidability (and optimal complexity) of model-checking problems of trees generated by recursion schemes as studied in [18].

Finally, in contrast to higher-order pushdown graphs (which do have decidable MSO theories [14]), we show in [22] that the MSO theories of collapsible pushdown graphs are undecidable. Hence collapsible pushdown graphs are, to our knowledge, the first example of a natural class of finitely-presentable graphs that have undecidable MSO theories while enjoying decidable modal mu-calculus theories.

3 Practical Relevance to Semantics and Verification

Recursion schemes are an old and influential formalism for the semantical analysis of both imperative and functional programs [27,7]. Indeed one of the first models of "Algol-like languages" (i.e. higher order procedural languages) was derived from the pushdown hierarchy of word languages (see Damm's monograph [7]). As indicated by the recent flurry of results [12,14,19,18,22], the hierarchies of (collapsible) pushdown automata and recursion schemes are highly relevant to infinite-state verification. In the light of the mediating algorithmic game semantics [17,28], it follows from the strong correspondence (Theorem 2) between recursion schemes and collapsible pushdown automata, that the collapsible pushdown hierarchies are accurate models of computation that underpin the computer-aided verification of higher-order procedural languages (such as Ocaml, Haskell, F#, etc.) — a challenging direction for software verification.

Acknowledgements. Some of the results cited above are based on (as yet unpublished) joint work [22] with Matthew Hague, Andrzej Murwaski and Olivier Serre. A full account will be published elsewhere.

References

1. Jones, N.D., Muchnick, S.S.: Complexity of finite memory programs with recursion. Journal of the Association for Computing Machinery 25, 312–321 (1978)
2. Muller, D.E., Schupp, P.E.: The theory of ends, pushdown automata, and second-order logic. Theoretical Computer Science 37, 51–75 (1985)
3. Kupferman, O., Vardi, M.Y.: An automata-theoretic approach to reasoning about infinite-state systems. In: Emerson, E.A., Sistla, A.P. (eds.) CAV 2000. LNCS, vol. 1855, Springer, Heidelberg (2000)
4. Maslov, A.N.: The hierarchy of indexed languages of an arbitrary level. Soviet mathematics Doklady 15, 1170–1174 (1974)
5. Maslov, A.N.: Multilevel stack automata. Problems of Information Transmission 12, 38–43 (1976)

6. Aho, A.: Indexed grammars - an extension of context-free grammars. J. ACM 15, 647–671 (1968)
7. Damm, W.: The IO- and OI-hierarchy. Theoretical Computer Science 20, 95–207 (1982)
8. Damm, W., Goerdt, A.: An automata-theoretical characterization of the OI-hierarchy. Information and Control 71, 1–32 (1986)
9. Knapik, T., Niwiński, D., Urzyczyn, P.: Deciding monadic theories of hyperalgebraic trees. In: Abramsky, S. (ed.) TLCA 2001. LNCS, vol. 2044, pp. 253–267. Springer, Heidelberg (2001)
10. Rabin, M.O.: Decidability of second-order theories and automata on infinite trees. Transactions of the American Mathematical Society 141, 1–35 (1969)
11. Courcelle, B.: The monadic second-order logic of graphs IX: machines and their behaviours. Theoretical Computer Science 151, 125–162 (1995)
12. Knapik, T., Niwiński, D., Urzyczyn, P.: Higher-order pushdown trees are easy. In: Nielsen, M., Engberg, U. (eds.) ETAPS 2002 and FOSSACS 2002. LNCS, vol. 2303, pp. 205–222. Springer, Heidelberg (2002)
13. de Miranda, J.: Structures generated by higher-order grammars and the safety constraint. PhD thesis, University of Oxford (2006)
14. Caucal, D.: On infinite terms having a decidable monadic theory. In: Diks, K., Rytter, W. (eds.) MFCS 2002. LNCS, vol. 2420, pp. 165–176. Springer, Heidelberg (2002)
15. Carayol, A., Wöhrle, S.: The Caucal hierarchy of infinite graphs in terms of logic and higher-order pushdown automata. In: Pandya, P.K., Radhakrishnan, J. (eds.) FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science. LNCS, vol. 2914, pp. 112–123. Springer, Heidelberg (2003)
16. Caucal, D.: On infinite transition graphs having a decidable monadic theory. In: Meyer auf der Heide, F., Monien, B. (eds.) ICALP 1996. LNCS, vol. 1099, pp. 194–205. Springer, Heidelberg (1996)
17. Hyland, J.M.E., Ong, C.H.L.: On Full Abstraction for PCF: I. Models, observables and the full abstraction problem, II. Dialogue games and innocent strategies, III. A fully abstract and universal game model. Information and Computation 163, 285–408 (2000)
18. Ong, C.H.L.: On model-checking trees generated by higher-order recursion schemes. In: Proceedings of the 21st Annual IEEE Symposium on Logic in Computer Science (LICS'06), pp. 81–90. IEEE Computer Society Press, Los Alamitos (2006)
19. Knapik, T., Niwiński, 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)
20. Aehlig, K., Miranda, J.G.d., 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)
21. Aehlig, K.: A finite semantics for simply-typed lambda terms for infinite runs of automata. In: Ésik, Z. (ed.) CSL 2006. LNCS, vol. 4207, pp. 104–118. Springer, Heidelberg (2006)
22. Hague, M., Murawski, A.S., Ong, C.H.L., Serre, O.: Collapsible pushdown automata and recursion schemes. Technical report, Oxford University Computing Laboratory, p. 56 (Preprint, 2007), downloadable from `users.comlab.ox.ac.uk/luke.ong/publications/cpda-long.pdf`

23. Aehlig, K., de Miranda, J.G., Ong, C.H.L.: Safety is not a restriction at level 2 for string languages. In: Sassone, V. (ed.) FOSSACS 2005. LNCS, vol. 3441, pp. 490–501. Springer, Heidelberg (2005)
24. Blum, W.: A tool for constructing structures generated by higher-order recursion schemes and collapsible pushdown automata (2007), web.comlab.ox.ac.uk/oucl/work/william.blum/
25. Walukiewicz, I.: Pushdown processes: games and model-checking. Information and Computation 157, 234–263 (2001)
26. Cachat, T.: Higher order pushdown automata, the Caucal hierarchy of graphs and parity games. In: Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J. (eds.) ICALP 2003. LNCS, vol. 2719, pp. 556–569. Springer, Heidelberg (2003)
27. Nivat, M.: On the interpretation of recursive polyadic program schemes. Symp. Math. XV, 255–281 (1975)
28. Murawski, A.S., Ong, C.H.L., Walukiewicz, I.: Idealized Algol with ground recursion and DPDA equivalence. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 917–929. Springer, Heidelberg (2005)