ML and Extended Branching VASS

C. Cotton-Barratt
University of Oxford

A. S. Murawski
University of Warwick

C.-H. L. Ong
University of Oxford

RML is a prototypical call-by-value functional language with state [2], which may be viewed as the canonical restriction of Standard ML to ground-type references. This talk will be about the decidability of observational equivalence of *finitary* RML, which is based on finite ground types and uses iteration instead of recursion.

Recall that two terms $\Gamma \vdash M, N : \theta$ are *observationally* (or *contextually*) *equivalent*, written $\Gamma \vdash M \cong N$, if they are interchangeable in all program contexts without causing any observable difference in the computational outcome. Observational equivalence is a compelling notion of program equality, but it is hard to reason about because of the universal quantification over program contexts. Our ultimate goal is to completely classify the decidable fragments of finitary RML, and characterise each fragment by an appropriate class of automata. In the case of finitary Idealized Algol [21] – the call-by-name counterpart of RML – the decidability of observational equivalence depends on the type-theoretic order [17] of the terms. By contrast, the decidability of RML terms is not neatly characterised by order: there are undecidable fragments of terms-in-context of order as low as 2 [16], amidst interesting decidable fragments at each of orders 1 to 4. In this talk we shall present the latest results regarding the observational equivalence problem for finitary RML, which – combined with prior work – yield a decidability classification for closed terms $\vdash M : \theta$.

Earlier work [6], [12] has established decidability for the following cases of θ, which we summarise below, where β is a base type (i.e. unit or int; note that in finitary RML the type int has finitely many values)[1].

Paper	θ
[6]	$\beta \to \cdots \to \beta$
[12]	$(\beta \to \cdots \to \beta) \to \beta$

Note that the first case ranges over all first-order types and the second one concerns the simplest second-order types. Because [6] also establishes undecidability for the second-order type (unit $\to$ unit) $\to$ unit $\to$ unit and the simplest third-order type ((unit $\to$ unit) $\to$ unit) $\to$ unit, as far as closed terms are concerned, the only unclassified cases are second-order types of the shape

$$\underbrace{\beta \to \cdots \to \beta}_{m} \to (\underbrace{\beta \to \cdots \to \beta}_{n}) \to \beta$$

where $m \geq 1$ and $n \geq 2$. These types will be the subject of this talk.

Our main contribution concerns the closed terms of types of the shape $\beta \to (\beta \to \cdots \to \beta) \to \beta$ and relates their observational equivalence problem to the reachability problem for *extended branching vector addition systems with states* (EBVASS) [14], whose decidability status is, to our knowledge, unknown. Our result applies not only to closed terms but also to a fragment $\mathrm{RML}_{\mathrm{EBVASS}}$ [8] in which free variables are subject to certain type constraints. The main result is thus the following.

Theorem 1. *Observational equivalence for the terms-in-context in* $\mathrm{RML}_{\mathrm{EBVASS}}$ *is recursively equivalent to the reachability problem for extended branching vector addition systems.*

Our second result is that the reachability problem for *reset vector addition systems with states* [3] is reducible to the observational equivalence of closed terms of type $\beta \to \beta \to (\beta \to \beta) \to \beta$. It follows that the observational equivalence of closed terms of all of the remaining types, i.e., where $m, n \geq 2$, is undecidable.

In the following, we discuss the key ideas behind the main results. Like the earlier results [6], [12], we appeal to the game semantics of RML [2], [10], which is *fully abstract*, i.e., the equational theory induced by the semantics coincides with observational equivalence. In game semantics [1], [13], player P takes the viewpoint of the term-in-context, and player O takes the viewpoint of the program context or environment. Thus a term-in-context, $\Gamma \vdash M : \theta$ with $\Gamma = x_1 : \theta_1, \cdots, x_n : \theta_n$, is interpreted as a P-strategy $[\![\Gamma \vdash M : \theta]\!]$ in the prearena $[\![\theta_1, \cdots, \theta_n \vdash \theta]\!]$. A play is a sequence of moves, made alternately by O and P, such that each non-initial move has a justification pointer to some earlier move. Thanks to the fully abstract game semantics of RML [2], [10], observational equivalence is characterised by *complete plays*, i.e., $\Gamma \vdash M \cong N$ holds iff the respective P-strategies, $[\![\Gamma \vdash M : \theta]\!]$ and $[\![\Gamma \vdash N : \theta]\!]$, contain the same set of complete plays. Strategies may be viewed as highly constrained processes, and are amenable to automata-theoretic representations. The main technical challenge, however, lies in the encoding of the justification pointers of the plays.

In recent work [5], [6], we considered finitary RML terms of first-order type. To represent the plays in the game semantics of such terms, we need to encode O-pointers (i.e. justification pointers from O-moves), which is tricky, because O-moves are controlled by the environment rather than the term. It turns out that the game semantics of these terms are representable as *nested data class memory automata* (NDCMA) [7], which are a variant of *class memory automata* [4] whose data values exhibit a tree structure, reflecting the tree structure of the

[1] For the sake of clarity, we do not list types with int ref and the corresponding constraints. They are analogous to treating int ref as $\beta \to \beta$.

Type	Automata / Status
$\beta \to \cdots \to \beta$	NDCMA / decidable [6], [11]
$(\beta \to \cdots \to \beta) \to \beta$	VPA / decidable [12]
$\beta \to (\beta \to \cdots \to \beta) \to \beta$	EBVASS (this paper)
$\beta \to \beta \to (\beta \to \beta) \to \beta$	Undecidable (this paper)
$(\beta \to \beta) \to \beta \to \beta$	Undecidable [6]
$((\beta \to \beta) \to \beta) \to \beta$	Undecidable [6]

Fig. 1: Classification of finitary RML closed terms wrt decidability of the observational equivalence problem.

threads in the plays.

Because of the type constraints, a play (in the strategy denotation) of a term in $\mathrm{RML}_{\mathrm{EBVASS}}$ may be viewed as an interleaving of "visibly pushdown" threads, subject to the global well-bracketing condition. In order to model such plays, we introduce *visibly pushdown class memory automata* (VPCMA), which naturally augment class memory automata with a stack and follow a visibly pushdown discipline, but also add data values to the stack so that matching push- and pop-moves must share the same data value. To give a clear representation of the game semantics, we introduce a slight variant of VPCMA with a run-time constraint on the words accepted, called *scoping VPCMA* (SVPCMA). This constraint prevents data values from being read once the stack element that was at the top of the stack when the data value was first read in the run has been popped off the stack. Although these two models are expressively different, they have equivalent emptiness problems.

Unlike in class memory automata (CMA), weakness[2] does not affect the hardness of the emptiness problem for VPCMA, as the stack can be used to check the local acceptance condition. However, like CMA, weakness does help with the closure properties of the languages recognised. The closure properties of these automata are the same as for normal CMA [6]: weak deterministic VPCMA are closed under union, intersection and complementation; similarly for SVPCMA. We show that the complete plays in the game semantics of each $\mathrm{RML}_{\mathrm{EBVASS}}$ term-in-context are representable as a weak deterministic SVPCMA. Thanks to the closure property of SVPCMA, it then follows that $\mathrm{RML}_{\mathrm{EBVASS}}$ observational equivalence is reducible to the emptiness problem for VPCMA.

Finally and most importantly, we show that the emptiness problem for VPCMA (equivalently for SVPCMA) is equivalent to the reachability problem for extended branching VASS (EBVASS) [14], the decidability of which remains an open problem. In particular, reachability in EBVASS is a harder problem than the long-standing open problem of reachability in BVASS (equivalently, provability in multiplicative exponential linear logic) [9], which is known to be non-elementary [15].

In summary, the results complete our programme to give an automata classification of the ML types with respect to the observational equivalence problem for closed terms of finitary RML. Our findings are summarised in Figure 1.

Related work.: Automata over an infinite alphabet (specifically, pushdown register automata) have also been applied to game semantics [19], [20] for a different purpose, namely, to model generation of fresh names in fragments of ML [20] and Java [18]. When extended with name storage, observational equivalence of terms-in-context with types in $\mathrm{RML}_{\mathrm{EBVASS}}$ becomes undecidable [20]; in particular, this is already the case for closed terms of type unit → unit → unit.

[2] *Weak* class memory automata [5], [6] are class memory automata in which the local acceptance condition is dropped.

References

[1] S. Abramsky, R. Jagadeesan, and P. Malacaria. Full abstraction for PCF. *Inf. Comput.*, 163(2):409–470, 2000.

[2] S. Abramsky and G. McCusker. Call-by-value games. In *CSL'97*.

[3] T. Araki and T. Kasami. Some decision problems related to the reachability problem for Petri nets. *TCS*, 3(1):85–104, 1976.

[4] H. Björklund and T. Schwentick. On notions of regularity for data languages. In *FCT'07*.

[5] C. Cotton-Barratt. *Using Class Memory Automata in Algorithmic Game Semantics*. DPhil thesis, University of Oxford, 2017.

[6] C. Cotton-Barratt, D. Hopkins, A. S. Murawski, and C.-H. L. Ong. Fragments of ML decidable by nested data class memory automata. In *FoSSaCS'15*.

[7] C. Cotton-Barratt, A. S. Murawski, and C.-H. L. Ong. Weak and nested class memory automata. In *LATA'15*.

[8] C. Cotton-Barratt, A. S. Murawski, and C.-H. L. Ong. ML and extended branching VASS. In *ESOP'17*.

[9] P. de Groote, B. Guillaume, and S. Salvati. Vector addition tree automata. In *LICS'04*.

[10] K. Honda and N. Yoshida. Game-theoretic analysis of call-by-value computation. *Theor. Comput. Sci.*, 221(1-2):393–456, 1999.

[11] D. Hopkins. *Game Semantics Based Equivalence Checking of Higher-Order Programs*. DPhil thesis, University of Oxford, 2012.

[12] D. Hopkins, A. S. Murawski, and C.-H. L. Ong. A fragment of ML decidable by visibly pushdown automata. In *ICALP'11*.

[13] J. M. E. Hyland and C.-H. Luke Ong. On full abstraction for PCF: I, II, and III. *Inf. Comput.*, 163(2):285–408, 2000.

[14] F. Jacquemard, L. Segoufin, and J. Dimino. FO2 on data trees, data tree automata and branching vector addition systems. *LMCS*, 12(2), 2016.

[15] R. Lazic and S. Schmitz. Nonelementary complexities for branching VASS, MELL, and extensions. *ACM ToCL*, 16(3):20:1–20:30, 2015.

[16] A. S. Murawski. Functions with local state: regularity and undecidability. *TCS*, 338(1/3):315–349, 2005.

[17] A. S. Murawski, C.-H. L. Ong, and I. Walukiewicz. Idealized Algol with ground recursion, and DPDA equivalence. In *ICALP'05*.

[18] A. S. Murawski, S. J. Ramsay, and N. Tzevelekos. Game semantic analysis of equivalence in IMJ. In *ATVA'15*.

[19] A. S. Murawski and N. Tzevelekos. Algorithmic nominal game semantics. In *ESOP'11*.

[20] A. S. Murawski and N. Tzevelekos. Algorithmic games for full ground references. In *ICALP'12*.

[21] J. C. Reynolds. The essence of ALGOL. In *Proceedings of the International Symposium on Algorithmic Languages*. Elsevier Science Inc., 1981.