Higher-order model-checking, categorical semantics, and linear logic

Charles Grellois and Paul-André Melliès

Laboratoires LIAFA & PPS
CNRS & Université Paris Diderot
{grellois,mellies}@pps.univ-paris-diderot.fr

The verification of higher-order recursive programs is a challenging issue, for which model-checking techniques have been considered. Programs are abstracted using *higher-order recursion schemes* (HORS); and a recursion scheme $\mathcal{G}$ can be understood as a simply-typed λ-term with fixpoint operators Y, over a set $\Sigma = a, b, c \ldots$ of free variables of order at most one. In the spirit of a Church encoding, the set Σ can be seen as a ranked alphabet, over which the normalization of $\mathcal{G}$ computes a potentially infinite *value tree* $\langle \mathcal{G} \rangle$, typically approximating the set of behaviors of a functional program.

The higher-order model-checking problem then consists in checking whether a *monadic second-order logic* (MSO) formula ϕ holds at the root of $\langle \mathcal{G} \rangle$. A convenient approach is to translate ϕ into an equivalent *alternating parity tree automaton* (APT) $\mathcal{A}_\phi$, a kind of non-deterministic top-down tree automaton enriched with *alternation* and *coloring*. A transition over a binary symbol a is typically of the form

$$\delta(q_0, a) \qquad = \qquad (2, q_2) \quad \vee \quad ((1, q_1) \wedge (1, q_2) \wedge (2, q_0)) \tag{1}$$

This models the alternating behavior of $\mathcal{A}_\phi$: when reading the symbol a in the state q_0, the automaton $\mathcal{A}_\varphi$ can either (i) drop the left subtree of a, and explore the right subtree with state q_2, or (ii) explore twice the left subtree of a in parallel, once with state q_1 and the other time with state q_2, and explore the right subtree of a with state q_0. Kobayashi observed that the transition (1) can be reflected by giving to the symbol a the following refined intersection type:

$$a \ : \ (\emptyset \to q_2 \to q_0) \wedge ((q_1 \wedge q_2) \to q_0 \to q_0) \tag{2}$$

From this observation, and given an APT $\mathcal{A}$, Kobayashi defines an intersection type system where a higher-order recursion scheme $\mathcal{G}$ has type q iff the automaton $\mathcal{A}$ has an execution over its value tree $\langle \mathcal{G} \rangle$ from that state q. The connection of intersection types with linear logic comes from the linear decomposition of the intuitionistic arrow

$$A \Rightarrow B \quad = \quad !A \multimap B$$

which regards a program of type $A \Rightarrow B$ as a program of type $!A \multimap B$ which thus uses its input $!A$ only once in order to compute its output B ; but where the exponential modality "!" enables at the same time the program to discard or to duplicate this single input $!A$. This striking similarity with the alternating behavior of APT led us to investigate the relation between Kobayashi's intersection type system and the relational semantics of linear logic. In the relational model, the exponential modality ! is interpreted as a *finite multiset* construction, which keeps track of the number of times an argument is called by the function. In [4], we translate Kobayashi's intersection type system (restricted to the simply-typed λ-calculus) to a non-idempotent variant. Adapting a correspondence developped by Bucciarelli and Ehrhard [1] between *indexed linear logic* and the relational semantics of linear logic, we then establish that the resulting type system computes the relational semantics of simply-typed λ-terms.

An obstruction to the extension of this correspondence to all HORS comes from the fact that a value tree $\langle \mathcal{G} \rangle$ is typically infinite, and can therefore use countably some of its free variables. For that reason, we developped an alternative relational semantics of linear logic, where the exponential modality noted $A \mapsto \lightning A$ is interpreted as the set of *finite-or-countable* multisets of elements of A, see [5] for details. These semantics feature a coinductive fixpoint operator, which interprets the syntactic recursion operator Y. This alternative and "infinitary" relational interpretation of linear logic enables us to establish a clean correspondence between (i) the coinductive intersection type system originally constructed by Kobayashi (ii) the executions of an alternating tree automaton (iii) our "infinitary" variant of the traditional relational semantics of linear logic.

However, this approach only captures a fragment of MSO logic, since it does not account for the *coloring* ability of APT. Indeed, $\mathcal{A}_\phi$ has a coloring function Ω which assigns to every state $q \in Q$ a color $\Omega(q) \in \mathbb{N}$. This additional information is devised so that an execution of the APT $\mathcal{A}_\varphi$ over the value tree $\langle \mathcal{G} \rangle$ proves the validity of the associated MSO formula φ iff it is *winning* for the parity condition induced by Ω, that is iff for every infinite branch of the run-tree, the greatest color encountered infinitely often is even. Kobayashi and Ong extended the original intersection type system in order to integrate this extra coloring information. In a series of recent papers [5, 6], we show that the type system they obtain can be slightly altered (and in fact improved) in order to disclose the modal nature of colors, and its connection to the exponential modality of linear logic. In our modal reformulation, the refinement type (2) associated to the transition of an APT may be colored (or modalised) in the following way:

$$a \ : \ (\emptyset \to \Box_{c_2}\ q_2 \to q_0) \wedge ((\Box_{c_1}\ q_1 \wedge \Box_{c_2}\ q_2) \to \Box_{c_0}\ q_0 \to q_0) \qquad (3)$$

where $\Box_c$ describes a family of modal operators, indexed by colors $c \in \mathbb{N}$. This coloring information can be incorporated in the infinitary relational semantics using a very simple and elementary comonad, defined as follows:

$$\Box\, A \quad = \quad Col \times A \quad = \quad \&_{c \in Col}\ A$$

where $Col \subseteq \mathbb{N}$ typically denotes the finite set of colors appearing in $\mathcal{A}_\varphi$. The existence of a distributive law $\lambda : \lightning \circ \Box \Rightarrow \Box \circ \lightning$ enables us to compose the comonad $\Box$ with the exponential modality $\lightning$ of our infinitary relational semantics, in order to obtain a new and "colored" exponential modality $A \mapsto \lightning \Box A$. In the resulting infinitary and colored relational model, the semantic counterpart of the colored intersection typing (3) is:

$$[\![a]\!] \quad = \quad \{ \quad ([], ([(c_2, q_2)], q_0)) \quad , \quad ([(c_1, q_1), (c_2, q_2)], ([(c_0, q_0)], q_0)) \quad \} \qquad (4)$$

We then defined an inductive-coinductive fixpoint operator Y, based on the principles of alternating parity tree automata: it behaves inductively in the scope of an odd color, and coinductively when the color is even, see [5] for details. This interpretation of the fixpoint operator Y based on parity may be also formulated at the level of intersection types, see [6]. Finally, we prove that a recursion scheme $\mathcal{G}$ produces a tree $\langle \mathcal{G} \rangle$ over which $\mathcal{A}_\varphi$ has a winning execution from q if and only if its colored relational semantics contains q.

This connection with linear logic leads us to a new proof of the decidability of the "selection problem" established by Carayol and Serre [2]. Our semantic proof of decidability [3] is based on the construction of a finitary and colored semantics of linear logic, adapted this time from the traditional *qualitative* semantics of linear logic based on prime-algebraic lattices and Scott-continuous functions between them — rather than from its alternative *quantitative* relational semantics. Interestingly, this qualitative semantics of linear logic corresponds to an intersection type system with subtyping, formulated in particular by Terui in [8]. It should be noted that the decidability of the "selection problem" implies in particular the decidability result for MSO formulas established by Ong [7] ten years ago. This decidability result gives a strong evidence of the conceptual as well as technical relevance of the connection which we have established and developed [3–6] between higher-order model-checking and linear logic[1].

References

1. Antonio Bucciarelli and Thomas Ehrhard. On phase semantics and denotational semantics: the exponentials. *Ann. Pure Appl. Logic*, 109(3):205–241, 2001.
2. Arnaud Carayol and Olivier Serre. Collapsible pushdown automata and labeled recursion schemes: Equivalence, safety and effective selection. In *LICS*, 2012.
3. Charles Grellois and Paul-André Melliès. Finitary semantics of linear logic and higher-order model-checking. submitted, `http://arxiv.org/abs/1502.05147`, 2015.
4. Charles Grellois and Paul-André Melliès. Indexed linear logic and higher-order model checking. In Jakob Rehof, editor, *ITRS 2014*, volume 177 of *EPTCS*, pages 43–52, 2015.
5. Charles Grellois and Paul-André Melliès. An infinitary model of linear logic. In Andrew M. Pitts, editor, *FoSSaCS 2015*, volume 9034 of *LNCS*, 2015.
6. Charles Grellois and Paul-André Melliès. Relational semantics of linear logic and higher-order model-checking. submitted, `http://arxiv.org/abs/1501.04789`, 2015.
7. C.-H. Luke Ong. On model-checking trees generated by higher-order recursion schemes. In *LICS*, pages 81–90. IEEE Computer Society, 2006.
8. Kazushige Terui. Semantic evaluation, intersection types and complexity of simply typed lambda calculus. In *RTA*, 2012.

[1] Although the papers mentioned here [3–6] will be published this year, the truth is that it took us several years of work to carry out the connection between higher-order model-checking and linear logic described in this brief survey. The idea and the details of the connection were thus exposed in seminar talks and at various stages of development in the past three years.