Colored intersection types: a bridge between higher-order model-checking and linear logic

Charles Grellois and Paul-André Melliès

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

The model-checking problem for higher-order recursive programs, expressed as *higher-order recursion schemes* (HORS), and where properties are specified in *monadic second-order logic* (MSO) has received much attention since it was proven decidable by Ong ten years ago. Every HORS may be understood as a simply-typed λ-term $\mathcal{G}$ with fixpoint operators Y whose free variables $a, b, c \ldots \in \Sigma$ are of order at most one. Following the principles of a Church encoding, these variables provide the tree constructors of a ranked alphabet Σ, so that the normalization of the recursion scheme $\mathcal{G}$ produces a typically infinite *value tree* $\langle \mathcal{G} \rangle$ over this ranked alphabet.

In order to check whether a given MSO formula φ holds at the root of such a value tree $\langle \mathcal{G} \rangle$, a convenient and traditional approach is to run an equivalent automaton $\mathcal{A}_\varphi$ over it. In the specific case of MSO logic, the corresponding notion of automaton is provided by *alternating parity tree automata* (APT), a kind of non-deterministic top-down tree automaton enriched with *alternation* and *coloring*. Every run of such an automaton may be understood as a syntactic proof-search of the validity of the formula φ over the value tree $\langle \mathcal{G} \rangle$. A typical transition over a binary symbol $a \in \Sigma$ is of the following shape:

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

When reading the symbol a in the state q_0, the automaton $\mathcal{A}_\varphi$ can either (1) drop the left subtree of a, and explore the right subtree with state q_2, or (2) 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 transitions of an alternating tree automaton $\mathcal{A}$ can be reflected by giving to the symbol a the following refined intersection type:

$$a \;:\; (\emptyset \to q_2 \to q_0) \wedge ((q_1 \wedge q_2) \to q_0 \to q_0) \qquad (1)$$

Using intersection types in this way, Kobayashi constructs a type system where a higher-order recursion scheme $\mathcal{G}$ is typed by a state q_0 of the automaton $\mathcal{A}$ iff its value tree $\langle \mathcal{G} \rangle$ is recognized from that state q_0. In order to recover the full expressive power of MSO logic, one needs to adapt this correspondence theorem to alternating *parity* automata (APT), and thus to integrate colors in the intersection type system. Recall that every state q of such an APT is assigned a color $\Omega(q) \in \mathbb{N}$. This additional information is devised so that a run-tree of the APT $\mathcal{A}_\varphi$ over the value tree $\langle \mathcal{G} \rangle$ proves the validity of the associated MSO formula φ 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 [6, 7], we establish a tight and somewhat unexpected connection between higher-order model-checking and linear logic, starting from a modal reformulation of Kobayashi and Ong's work. In particular, we show that their original type system 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 (1) associated to the transition of an APT may be colored (or modalised) in the following way:

$$a \;:\; (\emptyset \rightarrow \Box_{c_2}\ q_2 \rightarrow q_0) \wedge ((\Box_{c_1}\ q_1 \wedge \Box_{c_2}\ q_2) \rightarrow \Box_{c_0}\ q_0 \rightarrow q_0) \tag{2}$$

where $\Box_c$ describes a family of modal operators, indexed by colors $c \in \mathbb{N}$. 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 $!A$. In the relational semantics of linear logic, the exponential modality ! is interpreted as a *finite multiset* construction, so that the model keeps track of the number of times an argument is called by the function. The relational semantics is called *quantitative* for that reason. We translate in [5] the intersection type system originally devised by Kobayashi (restricted to the simply-typed λ-calculus) into an equivalent intersection type system where intersection is non-idempotent. Adapting a correspondence developped by Bucciarelli and Ehrhard [1, 2] between indexed linear logic and the relational semantics of linear logic, we establish that the resulting intersection type system computes the relational semantics of simply-typed λ-terms. At this stage, there remains to extend the correspondence to the simply-typed λ-calculus with a fixpoint operator Y. One conceptual difficulty is that the traditional interpretation of $!A$ in the relational semantics of linear logic is biased towards an inductive (rather than coinductive) interpretation of the fixpoint operator Y. Technically speaking, this comes from the fact that the multisets in $!A$ are *finite*. For that reason, we develop an alternative relational semantics of linear logic where the exponential modality noted $A \mapsto \lightning A$ is interpreted as the set $\mathcal{M}_{\leq\omega}(A)$ of *finite-or-countable* multisets of elements of A, see [6] for details. This alternative and "infinitary" relational interpretation of linear logic enables us to establish a clean correspondence between (1) the coinductive intersection type system originally constructed by Kobayashi (2) the run-trees of an alternating tree automaton with coinductive acceptance condition (3) our "infinitary" variant of the traditional relational semantics of linear logic. Put all together, these results provide a semantic account of higher-order model-checking where the acceptance condition of the underlying alternating tree automaton $\mathcal{A}$ is restricted however to the purely coinductive case.

At this stage, there thus remained to capture the full power of the MSO logic. To that purpose, we incorporated the family $\Box_c$ of modal operators mentioned earlier to our infinitary relational semantics of linear logic. The key idea is that this extra coloring information living at the level of the intersection type system reduces once reformulated at the level of the relational semantics into 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 the alternating parity tree automaton $\mathcal{A}_\varphi$ associated to the MSO-formula φ. 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 $\lightning$ in the original 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 colored intersection typing (2) has the following interpretation $[\![a]\!]$ of the symbol $a \in \Sigma$ as semantic counterpart:

$$[\![a]\!] \quad = \quad \{ \quad ([], ([(c_2, q_2)], q_0)) \quad , \quad ([(c_1, q_1), (c_2, q_2)], ([(c_0, q_0)], q_0)) \quad \} \tag{3}$$

Note that the interpretation of the symbol a of the alternating parity tree automaton $\mathcal{A}_\varphi$ is a subset of the interpretation of $o \to o \to o$, where o is interpreted as the set Q in our colored relational semantics:

$$[\![a]\!] \quad \subseteq \quad \lightning \,\square\, o \otimes \lightning \,\square\, o \multimap o \quad = \quad (\mathcal{M}_{\leq\omega}(Col \times Q))^2 \times Q$$

We then defined an inductive-coinductive fixpoint operator Y, based on the principles of alternating parity tree automata: the fixpoint operator iterates finitely in the scope of an odd color, and infinitely when the color is even, see [6] for details. This interpretation of the fixpoint operator Y based on parity may be also formulated at the level of intersection types: it corresponds in that setting to the introduction of a fixpoint rule, together with an appropriate notion of winning derivation tree formulated by the authors in [7]. Finally, we prove that a recursion scheme $\mathcal{G}$ produces a tree accepted from q by $\mathcal{A}_\varphi$ if and only if its colored relational semantics contains q – or alternatively, if and only if there is a winning derivation typing $\mathcal{G}$ with q.

This connection with linear logic leads us to a new proof of the decidability of the "selection problem" established by Carayol and Serre [3]. Our semantic proof of decidability [4] 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 in the work by Terui [9]. It should be noted that the decidability of the "selection problem" implies in particular the decidability result for MSO formulas established by Ong [8] 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 [4, 5, 6, 7] between higher-order model-checking and linear logic[1].

References

[1] Antonio Bucciarelli and Thomas Ehrhard. On phase semantics and denotational semantics in multiplicative-additive linear logic. *Ann. Pure Appl. Logic*, 102(3):247–282, 2000.

[2] Antonio Bucciarelli and Thomas Ehrhard. On phase semantics and denotational semantics: the exponentials. *Ann. Pure Appl. Logic*, 109(3):205–241, 2001.

[3] Arnaud Carayol and Olivier Serre. Collapsible pushdown automata and labeled recursion schemes: Equivalence, safety and effective selection. In *LICS*, 2012.

[4] Charles Grellois and Paul-André Melliès. Finitary semantics of linear logic and higher-order model-checking. submitted, `http://arxiv.org/abs/1502.05147`, 2015.

[5] 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.

[6] Charles Grellois and Paul-André Melliès. An infinitary model of linear logic. In Andrew M. Pitts, editor, *FoSSaCS 2015*, volume 9034 of *LNCS*, 2015.

[7] Charles Grellois and Paul-André Melliès. Relational semantics of linear logic and higher-order model-checking. submitted, `http://arxiv.org/abs/1501.04789`, 2015.

[8] C.-H. Luke Ong. On model-checking trees generated by higher-order recursion schemes. In *LICS*, pages 81–90. IEEE Computer Society, 2006.

[9] Kazushige Terui. Semantic evaluation, intersection types and complexity of simply typed lambda calculus. In *RTA*, 2012.

[1] Although the papers mentioned here [4, 5, 6, 7] 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.