HIGHER-ORDER MODEL CHECKING MEETS IMPLICIT AUTOMATA: FINITARY COLOURED SEMANTICS OF INFINITARY λ-TERMS

ABHISHEK DE, CHARLES GRELLOIS, LÊ THÀNH DŨNG (TITO) NGUYỄN, AND CÉCILIA PRADIC

The *higher-order model-checking problem* (HOMC) concerns model-checking trees generated by recursion schemes. The second author and Melliès's approached HOMC by evaluation in a finitary semantics of the $\lambda\mathbf{Y}$-calculus (simply typed λ-calculus with fixpoints) derived from the *Scott model of linear logic* augmented with a colouring modality [Gre16]. Since their semantics do not interpret arbitrary infinitary λ-terms (or even Böhm trees), the decidability proof is more indirect than expected. This leads us to state some natural conjectures that would directly imply the decidability of HOMC (some results in this direction are sketched in Melliès's follow-up work [Mel17]).

Implicit automata theory is an up-and-coming research programme led by the third and fourth authors among others that seeks to understand classes of formal languages (such as regular or star-free languages) and functions (e.g. regular or polyregular string-to-string functions) as corresponding to the expressive power of typed λ-calculi [HK96, NP20, Ngu21, PP24]. Our aforementioned results help us to extend this programme to ω-automata over infinite strings and trees.

1. The coinductive Scott semantics of infinitary STλC

Recall that the standard interpretation of the simply typed λ-calculus (STλC) in $\mathbf{FinScottL}_!$ maps any simple type τ (generated by the grammar $\sigma, \tau ::= o \mid \sigma \to \tau$) to a preordered set $[\![\tau]\!]$:

- $[\![o]\!]$ can be arbitrarily chosen – we will always take some finite set of "states" with equality.
- $[\![\sigma \to \tau]\!] = \mathcal{P}([\![\sigma]\!]) \times [\![\tau]\!]$ where $\underbrace{(X, \alpha)}_{\text{denoted later by } X \to \alpha} \leq (Y, \beta)$ when $\underbrace{\forall y \in Y, \exists x \in X.\, y \leq x}_{\text{“}Y \leq X\text{”}}$ and $\alpha \leq \beta$.

The semantics of a term $t : \tau$ is a *downwards-closed subset* $[\![t]\!] \subseteq [\![\tau]\!]$. The definition of $[\![t]\!]$ can be presented by an *intersection type system* (this syntactic exposition avoids talking about the cartesian closed structure of the category $\mathbf{FinScottL}_!$, and will be convenient for later extensions):

$$[\![t]\!] = \{\alpha \in [\![\tau]\!] \mid \varnothing \vdash t : \alpha :: \tau\}$$

where the judgment $\Gamma \vdash t : \alpha :: \tau$ is defined by the syntax-directed inference rules below. Note that on the left of $\vdash$, the assumption $x : X :: \tau$ makes sense when $X \subseteq [\![\tau]\!]$, whereas on the right, $t : \alpha :: \tau$ makes sense when $\alpha \in [\![\tau]\!]$.

$$\text{(Var)}\ \frac{\exists \alpha' \in X.\, \alpha \leq \alpha'}{x : X :: \tau \vdash x : \alpha :: \tau} \qquad \text{(Abs)}\ \frac{\Gamma, x : X :: \sigma \vdash t : \alpha :: \tau}{\Gamma \vdash \lambda x.\, t : X \to \alpha :: \sigma \to \tau}$$

$$\text{(App)}\ \frac{\Gamma \vdash t : X \to \alpha :: \sigma \to \tau \qquad (\forall \beta \in X)\ \Gamma \vdash u : \beta :: \sigma}{\Gamma \vdash t\, u : \alpha :: \tau}$$

to which we add a standard weakening rule. We need to include weakening because later, on infinitary derivations, it will not be clear that all weakenings can be pushed to variables).

For now we have focused on $\mathbf{FinScottL}_!$ as a model of finitary STλC. The intersection type system above gives us two obvious ways of extending it to an *infinitary* λ-term $t : \tau$ (where τ is still a simple type, see e.g. [SW13, §2]): either consider inductive (well-founded) or coinductive (non-well-founded) derivations to define $[\![t]\!] \subseteq [\![\tau]\!]$.

(Abhishek De) University of Birmingham, UK
(Charles Grellois) University of Sheffield, UK
(Lê Thành Dũng (Tito) Nguyễn) École normale supérieure de Lyon, France
(Cécilia Pradic) Swansea University, UK

We can also interpret the finite $\lambda\mathbf{Y}$-calculus by exhibiting a fixpoint operator in the semantics. Note that this is related to the infinitary λ-calculus by the *unfolding* operation defined coinductively by $\mathsf{unfold}(\mathbf{Y}\,t) = t\,(\mathsf{unfold}(\mathbf{Y}\,t))$ and the other obvious other clauses for the other constructors.

Claim 1. For a finite $\lambda\mathbf{Y}$-term $t : \tau$, we have:

$$[\![t]\!] \text{ with } \textit{least} \text{ fixpoint semantics} = [\![\mathsf{unfold}(t)]\!] \text{ with } \textit{inductive} \text{ derivations}$$
$$[\![t]\!] \text{ with } \textit{greatest} \text{ fixpoint semantics} = [\![\mathsf{unfold}(t)]\!] \text{ with } \textit{coinductive} \text{ derivations}$$

We will see later (§4.2) that one can get something in-between these extremal possibilities by considering infinitary λ-terms with *boundaries*. For now, we focus on the coinductive semantics (but note that the inductive semantics is the classical domain-theoretic one, where the denotation of a Böhm tree is the supremum of the denotations of its finite approximants).

The coinductive intersection typing derivations correspond exactly to the run-trees of Melliès's *higher-order automata* [Mel17]. He shows that they indeed define an actual semantics – as in, an invariant of reduction – for infinitary simply typed λ-terms. It is important to note that this is highly non-trivial – it is the main technical contribution of [Mel17].

Theorem 2 (rephrasing of [Mel17, Th. 3])**.** *If t has a strongly convergent (potentially infinite) β-reduction sequence to t' (notation: $t \to_\beta^\infty t'$), then $[\![t]\!] = [\![t']\!]$ for the coinductive semantics.*

The model is a bit limited because these higher-order automata correspond, on Church-encoded trees, to tree automata with trivial acceptance conditions (this can be seen from the shape of the derivations). To go further we need to move to the coloured semantics of [Gre16].

2. Adding colours and the parity condition

Fix a set $\mathrm{Col} = \{0, \dots, k\}$ of *colours*; 0 is a "neutral" colour whereas $1, \dots, k$ are the *priorities* used in parity automata/games of index k. The inductive case in the definition of $[\![-]\!]$ changes to

$$[\![\sigma \to \tau]\!] = \mathcal{P}(\mathrm{Col} \times [\![\sigma]\!]) \times [\![\tau]\!]$$

with $(c, \alpha) \leq (c', \beta) \iff c = c' \wedge \alpha \leq \beta$ and $Y \leq X$, $(X \to \alpha) \leq (Y \to \beta)$ redefined accordingly.

2.1. **The semantics of finitary STλC.** The intersection type system is adapted as follows. The only subtle case is application. Inductive derivations suffice to interpret finite simply typed λ-terms.

$$(\mathrm{Var})\frac{\exists (c', \alpha') \in X.\, c = c' \wedge \alpha \leq \alpha'}{x : X :: \tau \vdash x : \alpha :: \tau} \qquad (\mathrm{Abs})\frac{\Gamma, x : X :: \sigma \vdash t : \alpha :: \tau}{\Gamma \vdash \lambda x.\, t : X \to \alpha :: \sigma \to \tau}$$

$$(\mathrm{App})\frac{\Gamma \vdash t : \{(c_1, \beta_1), \dots, (c_n, \beta_n)\} \to \alpha :: \sigma \to \tau \qquad (\forall i \in \{1, \dots, n\})\ \Gamma_i \vdash u : \beta_i :: \sigma}{\Gamma \cup \uparrow_{c_1} \Gamma_1 \cup \dots \cup \uparrow_{c_n} \Gamma_n \vdash t\,u : \alpha :: \tau}$$

where $\uparrow_c (x_1 : X_1 :: \kappa_1, \dots, x_n : X_n :: \kappa_n) = x_1 :\uparrow_c X_1 :: \kappa_1, \dots, x_n :\uparrow_c X_n :: \kappa_n$ and

$$\uparrow_c X = \{(\underbrace{\max(c, c')}_{\text{meant to fit with a parity condition on priorities}}, \gamma) \mid (c', \gamma) \in X\}$$

2.2. **A fixpoint operator.** Finite $\lambda\mathbf{Y}$-terms are interpreted in [Gre16] using non-well-founded derivations with a *validity criterion* which is a parity condition. To the rules above, one adds:

$$(\mathbf{Y})\frac{\Gamma \vdash t : \{(c_1, \beta_1), \dots, (c_n, \beta_n)\} \to \alpha :: \sigma \to \tau \qquad (\forall i \in \{1, \dots, n\})\ \Gamma_i \vdash \mathbf{Y}\,t : \beta_i :: \sigma}{\Gamma \cup \uparrow_{c_1} \Gamma_1 \cup \dots \cup \uparrow_{c_n} \Gamma_n \vdash \mathbf{Y}\,t : \alpha :: \tau}$$

An edge (labeled with an intersection typing judgment) in a derivation tree is given the colour c_i if it is the premise $\Gamma_i \vdash \mathbf{Y}\,t : \beta_i :: \sigma$ of some $\mathbf{Y}$-rule of the above shape, otherwise it has colour 0. A derivation is *valid* if in all infinite branches, the maximum colour occurring infinitely often is non-zero and even.

2.3. **Towards an interpretation of infinitary λ-terms.** We would like to design a semantics for infinitary λ-terms, such that for every finite $\lambda\mathbf{Y}$-term t,

$$[\![t]\!] \text{ according to the above interpretation} = [\![\mathsf{unfold}(t)]\!] \text{ in the new system}$$

The close match between the (App) and (**Y**) rules above suggest that this must be possible. The only issue is that in the **Y**-less system, we do not have a validity criterion yet. The obvious solution would be to put the colour c_i on the premise $\Gamma_i \vdash u : \beta_i :: \sigma$ of each (App) rule, and keep the parity condition on branches as it is. This seems to correspond to the notion of *higher-order parity automaton* which is sketched but not fully specified in [Mel17].

Claim 3. For t a finite $\lambda\mathbf{Y}$-term, $[\![t]\!]$ using (App+**Y**)-colouring = $[\![\mathsf{unfold}(t)]\!]$ using (App)-colouring.

This is syntactically obvious. The less obvious relationship with Grellois and Melliès' original definition would then be captured by the following conjecture.

Conjecture 4. *For t a finite $\lambda\mathbf{Y}$-term, $[\![t]\!]$ using $(App + \mathbf{Y})$-colouring = $[\![t]\!]$ using only (**Y**)-colouring.*

Put together these two statements would imply that our coloured semantics of the infinitary simply typed λ-calculus indeed extends Grellois and Melliès's coloured model of the $\lambda\mathbf{Y}$-calculus. But we would need to make sure that it is a true semantics.

Conjecture 5. *This coloured semantics of infinitary λ-terms is invariant under infinite reductions.*

In Section 4, we shall explain our strategy to tackle the above conjecture. Before that, let us see some of its consequences.

2.4. **Higher-order model checking made easier?** Since *the coloured semantics of any finite $\lambda\mathbf{Y}$-term can be computed by solving a parity game*, the following lemma entails the decidability of HOMC. Let $\mathtt{Tree}_\Sigma$ be the type of Church-encoded trees over some ranked alphabet Σ. Write $[\![-]\!]_{Q,k}$ for the coloured semantics whose parameters are $[\![o]\!] = Q$ and $\mathrm{Col} = \{0, \dots, k\}$.

Lemma 6. *Let $t : \mathtt{Tree}_\Sigma$ be a finite $\lambda\mathbf{Y}$-term, whose Böhm tree b does not contain unproductive divergence. Given any alternating parity automaton $\mathcal{A}$ with states Q and index k over Σ, one can determine from $[\![t]\!]_{Q,k}$ whether $\mathcal{A}$ accepts the infinite ranked tree described by b.*

In [Gre16], a special case of this lemma (for $\lambda\mathbf{Y}$-terms in a certain form coming from recursion schemes) is proved by rather indirect means, going through a coloured relational semantics with countable multiplicities and then through games on graphs. We now observe that Lemma 6 would also follow, more directly, from Conjecture 5.

Proof. First, from the shape of the intersection typing rules, one can see that typing derivations for a Böhm tree $b : \mathtt{Tree}_\Sigma$ closely correspond to runs of alternating parity automata over the corresponding ranked trees, which suffices to show that:

Claim 7. From $[\![b]\!]$ one can determine whether b is accepted by such an automaton (with set of states and index corresponding to the parameters $[\![o]\!]$ and Col in the semantics).

Assuming Conjecture 5, we would have $[\![t]\!] = [\![\mathsf{unfold}(t)]\!] = [\![b]\!]$ where b is the Böhm tree of t, since $\mathsf{unfold}(t) \to_\beta^\infty b$. □

Since Conjecture 5 is non-trivial, this would probably not be the easiest proof of the decidability of HOMC. But it has the advantage of being natural: Conjecture 5 is of obvious independent interest anyway, as witnessed by another application that we give in the next section.

3. Application to implicit automata: reflection of regular languages

Assume that Conjecture 5 is true. Then we get this result in the spirit of [NP20]:

Corollary 8. *Let $f : \{\text{trees over } \Sigma\} \to \{\text{trees over } \Gamma\}$ be a total function defined by some infinitary simply typed λ-term $t : \mathtt{Tree}_\Sigma[\tau] \to \mathtt{Tree}_\Gamma$. (As usual, $\sigma[\tau] = \sigma\{o := \tau\}$.) Then, for any regular language L of infinite trees over Γ, the inverse image $f^{-1}(L)$ is also regular.*

To be clear, f is defined by $t\,\overline{T}[\tau] \to_\beta^\infty \overline{f(T)}$ where $\overline{(\cdot)}$ is the Church encoding.

Proof sketch. By a semantic evaluation argument akin to [HK96, Theorem 3.4]. Consider an alternating parity automaton $\mathcal{A}$ recognising L, with states Q and index k. Then by Claim 7, $[\![\overline{f(T)}]\!]_{Q,k}$ suffices to determine whether $f(T) \in L$. We have

$$\begin{aligned}
[\![\overline{f(T)}]\!]_{Q,k} &= [\![t\,\overline{T}[\tau]]\!]_{Q,k} && \text{by Conjecture 5}\\
&= [\![t]\!]_{Q,k}([\![\overline{T}[\tau]]\!]_{Q,k}) && \text{semantic interpretation of application}\\
&= [\![t]\!]_{Q,k}([\![\overline{T}]\!]_{[\![\tau]\!]_{Q,k},k})
\end{aligned}$$

with a subtlety: the semantics $[\![-]\!]_{[\![\tau]\!]_{Q,k},k}$ is defined with base case

$$[\![o]\!]_{[\![\tau]\!]_{Q,k},k} = [\![\tau]\!]_{Q,k} \quad \textit{as preordered sets}, \text{ not just as sets}$$

so we step out of the setting we were working in until now where the preorder on $[\![o]\!]$ was equality. Nevertheless, even with a non-trivial preorder on $[\![o]\!]$, the finite family

$$\mathcal{A} \text{ an automaton with states } [\![\tau]\!]_{Q,k} \text{ and index } k \mapsto \text{does } A \text{ accept } T?$$

of regular predicates on T should entirely determine $[\![\overline{T}]\!]_{[\![\tau]\!]_{Q,k},k}$ (a sort of converse to Claim 7), and thus $[\![\overline{f(T)}]\!]_{Q,k}$ according to the previous computation, which suffices to conclude. □

4. A syntactic reduction from coloured to uncoloured?

In this section we propose some ideas that might lead to a proof of Conjecture 5 without redoing all the work that went into Theorem 2. We fix $[\![o]\!]$ and the number k of colours.

4.1. **The comonadic translation for finitary STλC.** First, we relate the colourless and coloured semantics by means of a syntactic translation. For finitary STλC, this "comonadic translation" was derived in [Mel17, §VIII] as the interpretation function into a syntactic model, obtained by considerations of categorical semantics.[1] (See also [Wal19] for a variant.)

A purely syntactic account of the translation may be given as follows. On types, we have $\widehat{o} = o$ and $\widehat{\sigma \to \tau} = \underbrace{\widehat{\sigma} \to \cdots \to \widehat{\sigma}}_{k+1 \text{ times}} \to \widehat{\tau}$. We then translate terms and their typing derivations as follows:

$$\frac{}{\Gamma,\, x : \tau \vdash x : \tau} \rightsquigarrow \frac{}{\widehat{\Gamma},\, x^{(0)} : \widehat{\tau},\, \ldots,\, x^{(k)} : \widehat{\tau} \vdash x^{(0)} : \widehat{\tau}}$$

$$\frac{\Gamma,\, x : \sigma \vdash t : \tau}{\Gamma \vdash \lambda x.\, t : \sigma \to \tau} \rightsquigarrow \frac{\widehat{\Gamma},\, x^{(0)} : \widehat{\sigma},\, \ldots,\, x^{(k)} : \widehat{\sigma} \vdash \widehat{t} : \widehat{\tau}}{\widehat{\Gamma} \vdash \widehat{\lambda x.\, t} = \lambda x^{(0)}.\, \ldots\, \lambda x^{(k)}.\, \widehat{t} : \widehat{\sigma \to \tau}}$$

$$\frac{\Gamma \vdash t : \sigma \to \tau \quad \Gamma \vdash u : \sigma}{\Gamma \vdash t\,u : \tau} \rightsquigarrow \frac{\dfrac{\widehat{\Gamma} \vdash \widehat{t} : \widehat{\sigma \to \tau} \quad \widehat{\Gamma} \vdash \uparrow_0 \widehat{u} : \widehat{\sigma}}{\vdots} \ldots \quad \widehat{\Gamma} \vdash \uparrow_k \widehat{u} : \widehat{\sigma}}{\widehat{\Gamma} \vdash \widehat{t\,u} = \widehat{t}\,(\uparrow_0 \widehat{u})\ \ldots\ (\uparrow_k \widehat{u}) : \widehat{\tau}}$$

where $\uparrow_c \widehat{u} = \widehat{u}[x^{(i)} := x^{(\max(c,i))} \mid x \in \mathrm{dom}(\Gamma),\, i = 0, \ldots, k]$.

Observe that the translation of type derivations is well-defined because from a type derivation for $\widehat{\Gamma} \vdash \widehat{u} : \widehat{\sigma}$, one can get a derivation for $\widehat{\Gamma} \vdash \uparrow_c \widehat{u} : \widehat{\sigma}$ – indeed, the operation $\uparrow_c$ substitutes for each variable a variable that is given the same type by $\widehat{\Gamma}$.

Claim 9. For every simple type τ, there is a canonical isomorphism $\varphi_\tau : [\![\tau]\!]_{\text{coloured}} \xrightarrow{\sim} [\![\widehat{\tau}]\!]_{\text{colourless}}$ of preordered sets such that for every finite λ-term $t : \tau,$, we have $\varphi([\![t]\!]_{\text{coloured}}) = [\![\widehat{t}]\!]_{\text{colourless}}$.

This should be basically true by definition (thus, proof by mechanical induction).

[1] More precisely, this syntactic model can be defined as the coKleisli category for the linear exponential comonad $\Box A = A \times \cdots \times A$ ($k+1$ times) over the syntactic category of STλC (i.e. the initial cartesian closed category).

4.2. **A validity condition for infinitary λ-terms with boundaries.** We would like to extend this translation to infinitary λ-terms in such a way as to get the expected generalisation of Claim 9. In order to do so, we first need to recall a refinement of Theorem 2 sketched in [Mel17, §VII].

Definition 10. A *boundary* B of an infinitary λ-term is a subset of its set of infinite branches.

Now, consider a non-well-founded intersection type derivation D in the colourless system of Section 1 for an infinitary λ-term t. Notice that every branch (downwards path from the root) of D maps to a branch of t – indeed D is a labeled "thick subtree"[2] of t. (Not just an ordinary labeled subtree, because the application rule may "explore the argument subterm 0, 1 or several times".) Thus, a boundary B of t induces a validity criterion: every infinite branch of D must map to an element of B. We get the inductive/LFP semantics with the empty boundary, and the coinductive/GFP semantics with the full boundary.

In general, this defines the $\mathbf{ScottL}_!$ semantics $[\![(t,B)]\!]$ of a *λ-term with boundary* (t,B). Terms with boundaries can be treated as computational objects: if $t \to_\beta^\infty t'$ then the potentially infinite sequence of reductions canonically transports B to a boundary B' on t' [Mel17, Proposition 6]; we say in this case that "$(t,B) \to_\beta^\infty (t',B')$". Theorem 2 then extends to:

Theorem 11 ([Mel17, §VII])**.** *If* $(t,B) \to_\beta^\infty (t',B')$ *then* $[\![(t,B)]\!] = [\![(t',B')]\!]$.

4.3. **The comonadic translation with boundary.** By treating coinductively the rules of §4.1, we get a translation $t \mapsto \widehat{t}$ between infinitary λ-terms. In order to factor the coloured semantics of the infinitary STλC through this translation, we also need to endow the resulting terms $\widehat{t}$ with a well-chosen boundary. (Somewhat surprisingly, this has *not* been already done in [Mel17].) This boundary will encode a parity condition:

- First, we define from an infinitary λ-term t a node colouring on $\widehat{t}$.
  - In $\widehat{x}$, the only node $x^{(0)}$ is given the neutral colour 0.
  - In $\widehat{\lambda x.\, t} = \lambda x^{(0)}. \ldots \lambda x^{(k)}.\, \widehat{t}$, the λ-nodes have colour 0, and the other nodes keep the colour they had in $\widehat{t}$.
  - In $\widehat{t\, u} = \widehat{t}\,(\uparrow_0 \widehat{u}) \ldots (\uparrow_k \widehat{u})$, the root of each $\uparrow_c \widehat{u}$ is given the colour c, overriding the colour that it had in $\widehat{u}$ (necessarily 0, by construction). The other nodes in the subterms $\widehat{t}, \uparrow_0 \widehat{u}, \ldots, \uparrow_k \widehat{u}$ keep their colours from $\widehat{t}, \widehat{u}$; the new application nodes have colour 0.
- Then we say that the boundary $\widehat{\partial}(t)$ consists of all infinite branches on which the maximum colour occurring infinitely often is even and non-zero.

The required property is expected to straightforward to prove:

Conjecture 12. *For every infinitary simply typed λ-term* $t : \tau$*, we have* $\varphi_\tau([\![t]\!]_{coloured}) = [\![(\widehat{t}, \widehat{\partial}(t))]\!]$ *using the canonical isomorphism* $\varphi_\tau : [\![\tau]\!]_{coloured} \xrightarrow{\sim} [\![\widehat{\tau}]\!]_{colourless}$ *from Claim 9.*

We would also like the comonadic translation to be well-behaved with respect to infinitary reductions, and this might be more tricky:

Conjecture 13. *If* $t \to_\beta^\infty t'$*, then* $(\widehat{t}, \widehat{\partial}(t)) \to_\beta^\infty (\widehat{t'}, \widehat{\partial}(t'))$.

Let us conclude with the following observation:

Theorem 11 ∧ Conjecture 12 ∧ Conjecture 13 ⟹ Conjecture 5.

Acknowledgments. We would like to thank Denis Kuperberg for stimulating discussions about implicit automata on ω-words.

The first author is supported by a UKRI Future Leaders Fellowship, 'Structure vs Invariants in Proofs', project reference MR/S035540/1. The third author is supported by the LabEx MILyon, operated by the French National Research Agency (ANR-10-LABX-0070).

[2] In the vocabulary of French game semanticists, cf. e.g. [BC21].

References

[BC21] Lison Blondeau-Patissier and Pierre Clairambault. Positional injectivity for innocent strategies. In Naoki Kobayashi, editor, *6th International Conference on Formal Structures for Computation and Deduction, FSCD 2021, July 17-24, 2021, Buenos Aires, Argentina (Virtual Conference)*, volume 195 of *LIPIcs*, pages 17:1–17:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.FSCD.2021.17.

[Gre16] Charles Grellois. *Semantics of linear logic and higher-order model-checking*. PhD thesis, Université Paris 7, April 2016. URL: https://tel.archives-ouvertes.fr/tel-01311150/.

[HK96] Gerd G. Hillebrand and Paris C. Kanellakis. On the Expressive Power of Simply Typed and Let-Polymorphic Lambda Calculi. In *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science*, pages 253–263. IEEE Computer Society, 1996. doi:10.1109/LICS.1996.561337.

[Mel17] Paul-André Melliès. Higher-order parity automata. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12, Reykjavik, Iceland, June 2017. IEEE. doi:10.1109/LICS.2017.8005077.

[Ngu21] Lê Thành Dũng Nguyễn. *Implicit automata in linear logic and categorical transducer theory*. PhD thesis, Université Paris XIII (Sorbonne Paris Nord), December 2021. URL: https://theses.hal.science/tel-04132636.

[NP20] Lê Thành Dũng Nguyễn and Cécilia Pradic. Implicit automata in typed λ-calculi I: aperiodicity in a non-commutative logic. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 135:1–135:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.135.

[PP24] Cécilia Pradic and Ian Price. Implicit automata in λ-calculi iii: affine planar string-to-string functions, 2024. To appear in the proceedings of MFPS'24. arXiv:2404.03985.

[SW13] Sylvain Salvati and Igor Walukiewicz. Evaluation is MSOL-compatible. In Anil Seth and Nisheeth K. Vishnoi, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2013)*, volume 24 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 103–114, Dagstuhl, Germany, 2013. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.FSTTCS.2013.103.

[Wal19] Igor Walukiewicz. LambdaY-calculus with priorities. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*, pages 1–13. IEEE, 2019. doi:10.1109/LICS.2019.8785674.