Finite semantics of polymorphism, complexity and the power of type fixpoints

Lê Thành Dũng Nguyễn, Thomas Seiller, Paolo Pistone, Lorenzo Tortora de Falco

Finite semantics of polymorphism, complexity and the power of type fixpoints

Lê Thành Dũng Nguyễn and Thomas Seiller
LIPN, UMR CNRS 7030,
Université Paris 13, Sorbonne Paris Cité
Villetaneuse, France
Email: {lethanhdung.nguyen,seiller}@lipn.fr

Paolo Pistone
Wilhelm Schickard Institut
Universität Tübingen
Tübingen, Germany
Email: paolo.pistone@uni-tuebingen.de

Lorenzo Tortora de Falco
Dipartimento di Matematica e Fisica
Università Roma Tre
Rome, Italy
Email: tortora@uniroma3.it

***Abstract*—Many applications of denotational semantics, such as higher-order model checking or the complexity of normalization, rely on finite semantics for monomorphic type systems. We present two constructions of finite semantics for second-order Multiplicative-Additive Linear Logic (MALL_2) and study their properties. We apply this to understand the gap in expressive power between MALL_2 and its extension with type fixpoints, and to obtain an implicit characterization of regular languages in Elementary Linear Logic. Furthermore, some semantic results established here lay the groundwork for a sequel paper [1] proposing a new approach to sub-polynomial implicit complexity.**

I. Introduction

Polymorphism is a central topic in theoretical computer science since the sixties. A breakthrough in its logical understanding was its analysis by means of second order quantifiers, that is the introduction of System F at the beginning of the seventies. Some years later, this considerable success led Jean-Yves Girard to develop a denotational semantics for System F [2], to get a deeper understanding of its computational features. Indeed, the general goal of denotational semantics is to give a "mathematical" counterpart to syntactic devices such as proofs and programs, thus bringing to the fore their essential properties. Sometimes this eventually results in improvements of the syntax: Linear Logic itself [3] arose precisely from the denotational model introduced in [2].

But denotational semantics is not just a matter of increasing our understanding of programming languages, it also has direct algorithmic applications. Let us mention:

- in the simply-typed lambda calculus ($\mathrm{ST}\lambda$), the *semantic evaluation* technique for complexity bounds, see Terui's paper [4] and references therein;
- in $\mathrm{ST}\lambda$ extended with a fixed-point combinator, the semantic approach to *higher-order model checking* (HOMC) advocated by Salvati and Walukiewicz [5], [6] (see also [7], [8]).

The following little-known theorem illustrates both kinds of applications. Indeed, it is an implicit complexity result and, at the same time an instance of the correspondence between Church encodings and automata that HOMC generalizes to infinite trees.

Theorem I.1 (Hillebrand & Kanellakis [9]). *The languages decided by* $\mathrm{ST}\lambda$ *terms from Church-encoded binary strings to Church booleans*[1] *are exactly the* regular *languages.*

The main idea of this result to build a deterministic finite automaton (DFA) computing the denotation of its input string. Crucially, this relies on the existence of a *finite semantics* for $\mathrm{ST}\lambda$ – such as the category of finite sets – which will provide the states of the DFA. In general, this finiteness property, or finer cardinality bounds, are key to these applications.

This theorem also holds when replacing $\mathrm{ST}\lambda$ by propositional linear logic, which also admits finite semantics. In fact, Terui's solution to the complexity of $\mathrm{ST}\lambda$ normalization at fixed order [4] relies on such a semantics. As for HOMC, Grellois and Melliès have developed an approach relying on models of linear logic [10], [11].

However, all of this concerns only *monomorphic* type systems, for a simple reason: equality is definable on the type of System F natural numbers, so its denotation in any non-trivial model must be infinite.

A. Contributions

a) Polymorphism, linearity, and type fixpoints: Actually, second-order quantification is not the only culprit here: one can also blame the *non-linearity* of the System F integers. What we show in this paper is that a semantics for a purely linear language with impredicative polymorphism can be finite:

Theorem I.2. *Second-order Multiplicative-Additive Linear Logic* (MALL_2) *admits finite semantics.*

This mere existence already allows us to clarify the difference in expressive power between MALL_2 and $\mu\mathrm{MALL}$, i.e. MALL with type fixpoints [12]. While $\mu\mathrm{MALL}$ can be translated in second-order LL with exponentials, it was argued informally that a translation to MALL_2 could not exist [12, §2.3]; since $\mu\mathrm{MALL}$, which can encode infinite data types, does not admit non-trivial finite semantics, we can now provide a clear proof of this impossibility (Theorem III.1).

The power of type fixpoints also appears in implicit complexity, in Baillot's characterization of polynomial time [13] in

[1] That is, $\mathrm{ST}\lambda$ terms of type $((A \to A) \to (A \to A) \to (A \to A)) \to (o \to o \to o)$, where o is a base type and A may be chosen depending on the language, but is independent of the input string.

second-order Elementary Linear Logic[2] (ELL_2) with recursive types. An open question was whether the same result holds without type fixpoints; using Theorem I.2, we show that we get instead a characterization of *regular languages* (Theorem III.4). Our proof is directly inspired by that of Theorem I.1 by Hillebrand and Kanellakis, unsurprisingly since inputs are Church-encoded in both cases.

b) Witness-erasing semantics of polymorphism: In models of MALL_2, the main obstacle to finiteness is the existential quantifier. Indeed, while existential-free MALL_2 formulae have finitely many cut-free proofs thanks to linearity, existential variables may have witnesses of arbitrary size. To prove Theorem I.2, the goal is therefore to "erase" these witnesses in the interpretation, in order to compress the proof to some bounded data depending only on the proven formula. Following the programming language point of view on existential types as abstract data types, this will mean remembering just enough information to determine their interaction with the generic (universally typed) programs which might use them.

Typically, all the proofs

$$\dfrac{\dfrac{\cdots}{\vdash A}}{\vdash \exists X.X}\ \exists$$

should be collapsed – recall that $\exists X.X$ is the impredicative encoding of the additive unit $\top$.

Our first implementation of this idea is purely syntactic: it consists in taking an *observational quotient* of the syntax. Although the model thus obtained is indeed finite (Proposition II.16) and has the advantages of simplicity and canonicity, we will see that it is not effective in several ways (Theorems II.17 and II.18), which may impede applications. Still, we believe that the finiteness proof for this quotient conveys important intuitions.

To get an effective model, we turn to Girard's semantics of polymorphism in *coherence spaces* [2], [3]. It is more concrete than its formulation using category-theoretic machinery (normal functors) could suggest, and as it turns out, Girard's definitions already lead to finite and computable denotations of MALL_2 types (Theorems IV.16 and IV.18). We prove this by singling out a notion of *finite degree* which is preserved by MALL_2 connectives and ensures finiteness, and sketch a combinatorial presentation. By the way, the witness-erasing character of this model – indeed, the non-trivial computational contents of its existential introduction, which subsumes the cut rule – was already noted in [3, p. 57] as being "key to a semantic approach to computation".

c) Sub-polynomial implicit complexity: Although we do not detail an application of the effectiveness of coherence spaces here, we lay the groundwork (Proposition IV.19) for a sequel paper [1] which will apply it to a conjectural implicit characterization of *deterministic logarithmic space* with a few distinctive features. As we shall discuss in the present paper, trying to go beyond regular languages in ELL_2 naturally leads to changing the representation of inputs, and this conjecture arises directly from pursuing this line of thought by taking inspiration from by Hillebrand's PhD thesis [14].

By performing semantic evaluation in coherence spaces, the sequel manages to establish a sub-polynomial upper bound – though not yet deterministic logspace soundness. We believe this approach to implicit complexity to be rather novel, as will be explained in [1]: unlike in previous works on characterizing logarithmic space via substructural logics [15], [16], [17], it seems that soundness cannot be established by variants of the Geometry of Interaction interpretation of the multiplicative-exponential fragment of linear logic. Let us stress, then, that this novelty is only made possible thanks to the semantic investigations undertaken here.

Let us also mention that our result on regular languages in ELL_2 betrays the original spirit of light logics [18] which consisted in bounding the complexity of normalization "geometrically", independently of types. Here, while geometry still plays an important structuring role, our fine-grained analysis requires to take into account the influence of types through semantics.

d) Finite models and parametricity: We finally compare the two models presented with two properties arising from parametricity [19], a well-known approach to polymorphism. Parametricity is a desirable property from our viewpoint as it provides a "smallness" condition on the interpretation of quantifiers. In particular, *dinaturality*, proposed as a categorical formalisation of parametricity since [20], provides a third example of witness-erasing semantics, since the interpretation of quantifers as *ends/coends* leads to identify proofs with different witnesses. However, the coherent model is not dinatural (Proposition V.6) and the construction of finite parametric models of MALL_2 seems a non-trivial task, also as a consequence of Proposition V.9, which shows that compact closed models do not satisfy the *constancy* property, another property of parametric models introduced in [21].

B. Notations and definitions

We recall the syntax of MALL_2. Given a set of type variables $\mathcal{V}$, formulas of MLL_2 are defined by the following grammar:

$$F, G := 1 \mid \bot \mid X \mid F \otimes G \mid F \parr G \mid \forall X\ F \mid \exists X\ F,$$

where $X \in \mathcal{V}$. Formulas of MALL_2 are defined from the following extension of the latter:

$$F, G := \cdots \mid 0 \mid \top \mid F \oplus G \mid F \,\&\, G$$

We will also consider here the fragments:

- MALL_0 of quantifier-free MALL_2 formulas;
- MALL_0^- of unit-free MALL_0 formulas;
- MLL_0 and MLL_0^- the analogous restrictions of MLL_2.

II. Observational equivalence

In this section, we will explain the first finite semantics of MALL_2 we obtained. The idea behind this model is quite intuitive, yet the proofs are quite involved. The model is here obtained as a quotient on the syntax of MALL_2 proofs with respect to some *observational equivalence*.

[2]In fact, Baillot works with Elementary *Affine* Logic, but unlike type fixpoints, weakening makes no difference as to complexity. See also Laurent's notes "Polynomial Time in Untyped Elementary Linear Logic".

A. *Witness-oblivious observational equivalence*

As explained in the introduction, a finite semantics, if it exists, has to perform some identifications between proofs using different witnesses. This will be expressed here as an observational equivalence. The idea of the quotient we consider is to restrict to observations – i.e. tests we allow to be applied to proofs – which are not allowed to access existential witnesses. This is done by restricting observations for proofs of A to proofs of formulas $A \multimap B$ where B is quantifier-free.

Definition II.1. Let A be a MALL_2 type, and π, π' be proofs of A. We say that π and π' are MALL_0^--equivalent, noted $\pi \sim_0 \pi'$, when for all proof ρ of type $A \multimap B$ with B a MALL_0^- type, the normal forms of $\mathrm{cut}(\pi, \rho)$ and $\mathrm{cut}(\pi', \rho)$ are equal.

One can then check that this equivalence defines a congruence. Moreover, it does not identify too many proofs, i.e. the resulting quotient is still a computationally relevant model.

Lemma II.2. *The MALL_0^--equivalence of proofs defines a congruence on the syntactic model of MALL_2, which coincides with syntactic equality on MALL_0^- types.*

What is left then is to prove the finiteness of the quotient. We will sketch in the next section the proof for the multiplicative fragment, which uses the notion of *proof nets* for MLL_2, and we will explain how the result extends to MALL_2.

B. *Proof nets*

We here recall very quickly the definition of proof nets.

Definition II.3. A MLL_2 *proof structure* is a directed edge-labeled hypergraph with edge labels in the set $\{\mathrm{ax}, \mathrm{cut}, \otimes, \parr, \forall, \exists\}$ subject to the following constraints:

- edges labeled $\otimes, \parr$ have two sources and one target;
- edges labeled $\forall, \exists$ have one source and one target;
- edges labeled ax have no source and two targets;
- edges labeled cut have two sources and no target;

A structure is *cut-free* when the label cut is not used. Vertices which are not the source of an edge are called *conclusions*.

Definition II.4. One can inductively define a natural translation $\mathtt{d}^+(\_)$ of MLL_2 sequent calculus proofs to MLL_2 proof structures. Figure 1 shows how rules are interpreted by local operations on hypergraphs (dashed boxes represent translations of proofs of the premises of the rule).

A *proof net* is[3] a proof structure which is the translation of some sequent calculus proof through $\mathtt{d}^+(\_)$.

Cut-elimination on proof structures (and proof nets) is defined from simple local rewriting rules [3], shown in Figure 2.

C. *Finiteness of the quotient for* MLL_2

Definition II.5 (Public Tree)**.** Let $\mathcal{R}$ be a cut-free proof net, coming from a proof of $\vdash A_1, \ldots, A_n$. An edge e in $\mathcal{R}$ is *public* if the connective it introduces appears in the syntax tree of the formula. The edge e is *private* when it is not public, i.e. when it is an axiom or when the connective it introduces belongs to the syntax tree of an existential witness.

Remark II.6. This notion of public/private part of the proof net is essential for the following proof. Its consideration, and its definition, deeply rests on our choice to work with proof nets instead of sequent calculus. For instance, the notion is much less natural in sequent calculus, as the public part need not be a subtree of the derivation. For instance, let us consider a proof of the form

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\vdots \\ \vdash A^\perp \otimes B^\perp, E\end{matrix} \qquad \begin{matrix}\vdots \\ \vdash C \otimes D, F\end{matrix}}{\vdash A^\perp \otimes B^\perp, C \otimes D, E \otimes F}\otimes}{\vdash C \otimes D, (A^\perp \otimes B^\perp) \parr (E \otimes F)}\parr}{\vdash C \otimes D, \exists X.((A^\perp \otimes B^\perp) \parr X)}\exists$$

Then, while the last two rules are public, the previous one is private (because it introduces a connective which do not appear in the conclusion). However public rules could appear above the subtree shown here, introducting the public $\otimes$ connectives in the formulas $A^\perp \otimes B^\perp$ and $C \otimes D$. Moreover, commutations of the private $\otimes$ rule is not allowed as all connectives involved are positive (i.e. not reversible).

Figure 3 illustrates this, with the public part in light gray and the private part in dark gray. There is another important point here: the private part alone[4] also forms a proof structure, whose conclusions correspond to occurrences of atoms of X, as indicated by the labels in Figure 3.

Definition II.7. Consider $\mathcal{R}$ and $\mathcal{Q}$ two proof structures. For any injection ι from the set of conclusions of $\mathcal{R}$ to that of $\mathcal{Q}$, we denote by $[\mathcal{R}, \mathcal{Q}]_\iota$ the vertex-labeled MLL_2 proof structure obtained by adding cuts between every conclusion of $\mathcal{R}$ with the corresponding conclusion of $\mathcal{Q}$ through ι.

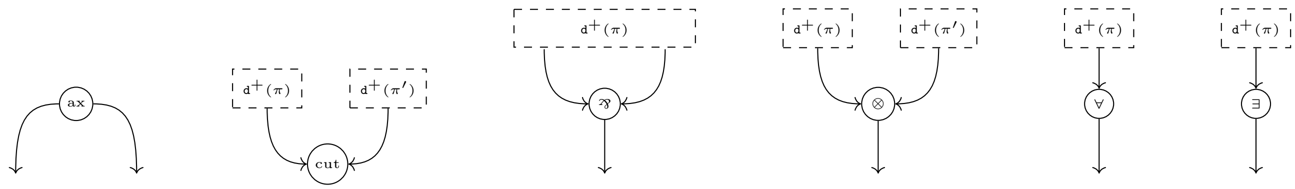
Let π and ρ be two proof nets representing proofs of $\vdash A$ and $\vdash A^\perp, B$ respectively. We may assume that in these proofs, each public connective was introduced by its introduction rule, and not by an axiom rule (else, apply an "η-expansion" procedure). Consider the proof net obtained by adding a cut between the conclusions A and $A^\perp$. It should be clear that cut-elimination can always be considered as following the strategy consisting in first eliminating the cuts whose premises are edges in the public tree of π and of ρ. After this initial *public* step of elimination, the resulting edge-labeled proof structure has the form shown in Figure 4, i.e. it is of the form $[\mathcal{R}, \mathcal{Q}]_\iota$ for proof structures $\mathcal{R}$ and $\mathcal{Q}$.

Since $\mathcal{R}$ is the private part of π, as mentioned before its conclusion correspond to the atoms of A. Similarly, the conclusions of $\mathcal{Q}$ in the image of ι correspond to the atoms of $A^\perp$, and ι sends atoms in A to their duals in $A^\perp$.

Remark II.8. For the reader familiar with proof nets: here we work with untyped proof structures, there is no labeling

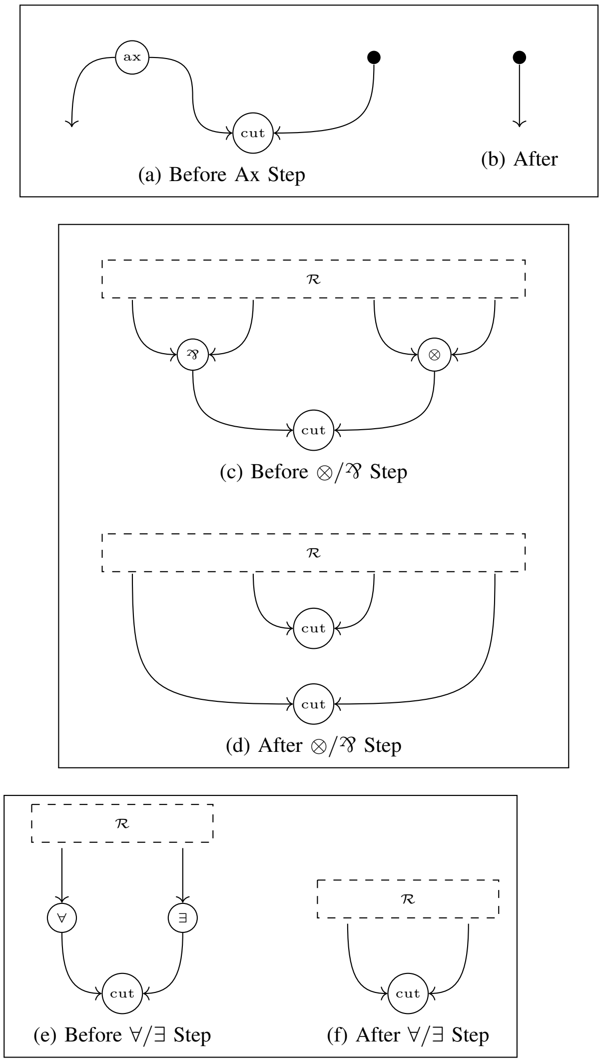
[3]There are also combinatorial characterizations, known as *correctness criteria*, which can serve as the definition of proof nets [3], [22], [23].

[4]That is, the subhypergraph induced by the private edges.



(a) Axiom Rule (b) Cut Rule (c) ⅋ Rule (d) ⊗ Rule (e) ∀ Rule (f) ∃ Rule

Fig. 1: Translation $\mathtt{d}^{+}(\_)$.



(a) Before Ax Step (b) After

(c) Before ⊗/⅋ Step

(d) After ⊗/⅋ Step

(e) Before ∀/∃ Step (f) After ∀/∃ Step

Fig. 2: Cut Elimination Steps.

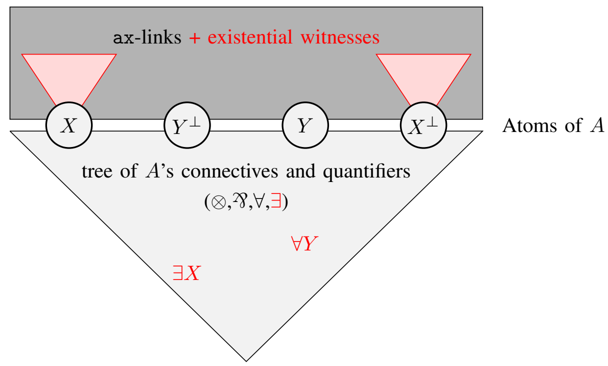
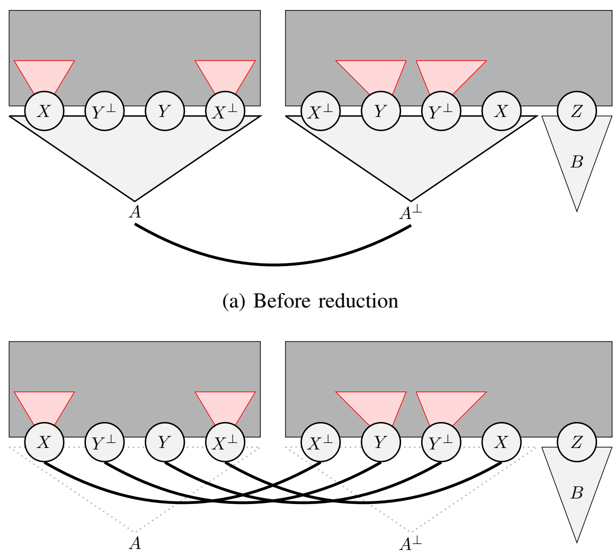


Fig. 3: Shape of a MLL_2 proof net coming from a proof of A.



(a) Before reduction

(b) After reduction

Fig. 4: Public reduction

by formulas. The labeling of conclusions by atoms we use is in fact a subtly different notion. (Think about the substitutions performed during cut-elimination of typed proof nets). Untyped reduction of proof structures is in general delicate, see for example [24].

We will now define a big step reduction on the intermediate structures of the form $[\mathcal{R}, \mathcal{Q}]_\iota$. The result will then follow from the fact that this reduction terminates to a cut-free net.

Definition II.9. Let $\mathcal{R}$ be a cut-free proof structure and suppose we have enumerated its conclusions with natural numbers. Then:

- $\mathbf{del}[i, j](\mathcal{R})$ is defined when there exists an ax edge of $\mathcal{R}$ whose targets are the conclusions numbered i and j, and its value is the proof structure obtained from $\mathcal{R}$ by erasing this axiom and the conclusions i and j;
- if the proof structure obtained by connecting the conclusions numbered i and j of $\mathcal{R}$ by means of a cut-link

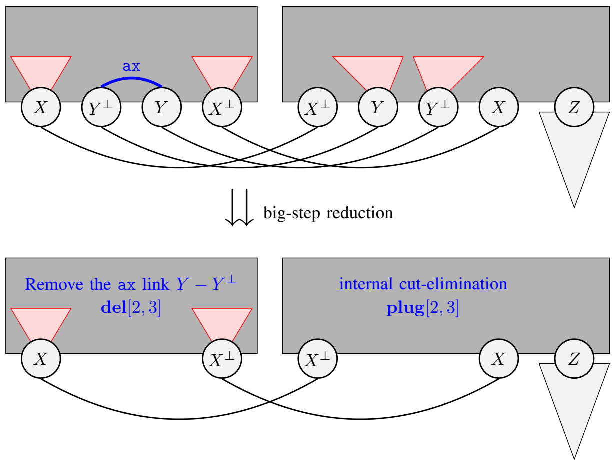


Fig. 5: The reduction rule $\rightrightarrows$.

normalizes, then $\mathbf{plug}[i,j](\mathcal{R})$ is defined as its normal form.

Definition II.10 (Big-step reduction $\rightrightarrows$)**.** Let A be an MLL_2 formula. Let $\mathcal{R}$ and $\mathcal{Q}$ be two proof structures. Let ι be an injection from the conclusions of $\mathcal{R}$ – labeled by occurrences of atoms of A – to those of $\mathcal{Q}$ – the image of ι being labeled by the dual atoms in $A^\perp$. (We do not require all atoms of A to appear as labels.) If

- $\mathbf{del}[i,j](\mathcal{R})$ is defined (resp. $\mathbf{del}[\iota(i),\iota(j)](\mathcal{Q})$);
- the labels of the conclusions i and j (resp. $\iota(i)$ and $\iota(j)$) of $\mathcal{R}$ (resp. $\mathcal{Q}$) are dual *universal* atoms of A (resp. $A^\perp$);

we define $[\mathcal{R},\mathcal{Q}]_\iota \rightrightarrows [\mathbf{del}[i,j](\mathcal{R}), \mathbf{plug}[\iota(i),\iota(j)](\mathcal{Q})]_{\iota'}$ (resp. $[\mathcal{R},\mathcal{Q}]_\iota \rightrightarrows [\mathbf{plug}[i,j](\mathcal{R}), \mathbf{del}[\iota(i),\iota(j)](\mathcal{Q})]_{\iota'}$), where ι' is the restriction of ι on indices different from i, j.

Lemma II.11 (Adequacy)**.** *If $[\mathcal{R},\mathcal{Q}]_\iota \rightrightarrows [\mathcal{R}',\mathcal{Q}']_{\iota'}$, then $[\mathcal{R},\mathcal{Q}]_\iota \to^* [\mathcal{R}',\mathcal{Q}']_{\iota'}$ where $\to$ is the usual cut-elimination of proof nets.*

Lemma II.12 (Progress)**.** *Let π be a cut-free proof net of A and ρ a proof net of $\vdash A^\perp, B$. Let $[\mathcal{R},\mathcal{Q}]_\iota$ the result of their public reduction, and $[\mathcal{R},\mathcal{Q}]_\iota \rightrightarrows^* [\mathcal{R}',\mathcal{Q}']_{\iota'}$.*

If there is a cut-link in $[\mathcal{R}',\mathcal{Q}']_{\iota'}$ then there is a redex for the rewriting rule $\rightrightarrows$, i.e. there is an axiom link in $[\mathcal{R}',\mathcal{Q}']_{\iota'}$ whose targets are conclusions of either $\mathcal{R}'$ or $\mathcal{Q}'$ corresponding to dual universal atoms of respectively A or $A^\perp$.

Proof. In appendix, Section A. $\square$

Corollary II.13. *If $[\mathcal{R},\mathcal{Q}]$ is the result of a the public reduction of π against ρ, then if a $\rightrightarrows$-reduction starting from $[\mathcal{R},\mathcal{Q}]$ terminates on some proof structure, then this result is the normal form of* $\mathbf{cut}(\pi,\rho)$ *for usual cut-elimination.*

While adequacy is easy to check on the definition of proof structure reduction, progress is combinatorially tricky, and its proof is relegated to the appendix. As $\rightrightarrows$ reduces the number of conclusions of the proof structures involved, we also have:

Proposition II.14 (Bounded strong normalization)**.** *With the notations of Lemma II.12, any $\rightrightarrows$-reduction sequence starting from $[\mathcal{R},\mathcal{Q}]_\iota$ and ending in a cut-free net has length $N/2$, where N is the number of occurrences of atoms in A.*

Definition II.15. Let A be a MLL_2 type with N occurrences of atoms. We define the alphabet Σ_N as the set of all symbols $\mathbf{del}[i,j](\_)$ and $\mathbf{plug}[i,j](\_)$ for $1 \leqslant i < j \leqslant N$. It is of size $N(N-1)$, thus bounded by N^2.

Let π be a proof and $\mathcal{R}$ its private proof structure. A word $w \in (\Sigma_N)^{N/2}$ is *good for* π if the sequence $\mathcal{R}_0, \mathcal{R}_1, \ldots, \mathcal{R}_{N/2}$, where $\mathcal{R}_0 = \mathcal{R}$ and $\mathcal{R}_{i+1} = w_i(\mathcal{R}_i)$, is well-defined and at each step the operation w_i applies to atoms which are labeled by dual formulas. The set of all words good for π is written $\alpha(\pi)$ and called the *strategy* of π.

Proposition II.16. *The data of $\alpha(\pi)$ is enough to determine the equivalence class of π for MLL_0^- equivalence.*

D. *Extension for additives*

The above result holds for MLL_2. We now explain how it extends to additives. A proof of MALL_2 can be considered as a set of MALL_2 proofs, usually called *slices* (see e.g. [25, §3.1]), defined by forgetting the derivations of one of the premises of $\&$ rules. Such slices translate to proof structures extended with edges labeled $\oplus_l, \oplus_r, \&_l, \&_r$ having exactly one source and one target. These extended proof structures possess a cut-elimination procedure, where pairs $\oplus_i/\&_j$ reduce analogously to $\forall/\exists$ when $i = j$, and the cut raises an error if $i \neq j$.

The proof sketched above can then be followed to define a notion of strategy for these extended proof structures: a strategy is still a sequence of elements of Σ^N good for the proof in the sense that the sequence of applications of the operations of the sequence is well-defined *and do not raise any errors*. A strategy for a proof of MALL_2 is then defined naturally as the union of the strategies of the slices.

- the notion of strategy is enough to characterise the equivalence class of the proof, because it allows to compute the linkings of the Hughes and van Glabbeek proof net of the result [26];
- the number of possible strategies is finite: even though the number of slices is not bounded by the type (i.e. proofs of a given MALL_2 formula have arbitrarily large numbers of slices in general),

Another proof would be to use the injectivity for MALL_0^- of the finite semantics in coherence spaces considered later in this paper, but that would be much less enlightening.

E. *Non-effectiveness of the observational quotient*

We will here explain some limitations of the above model, that is its non-effectivity. While the mere existence of a finite model of MALL_2 will be enough to provide the results of the next section, an effective model would (and will) allow for more precise results. Non-effectivity is related to the following theorem of Lafont and Scedrov.

Theorem II.17 (Lafont & Scedrov [27])**.** *Provability in second-order Multiplicative Linear Logic is undecidable.*

This means that even though the set $A/\sim_A$ is finite, one cannot enumerate it: indeed, its non-emptiness is equivalent to the provability of A.

In the presence of additives, there is a second source of non-effectiveness: equivalence classes cannot be represented in a way that allow for equality testing. The following theorem is obtained (cf. Section B) using the Kanovich–Lafont MALL_2 encoding of two-counter machines [28, Theorem 3] and adapting the proof that provability of the formula A_M encoding a two-counter machine M implies that M is accepting.

Theorem II.18. *The equivalence $\sim_0^-$ is undecidable.*

III. Type fixpoints and complexity

A. MALL_2 *versus* μMALL *and regular languages*

One direct result stemming from the existence of a finite semantics for MALL_2 is the following theorem.

Theorem III.1. *There exists no faithful translation from* μMALL *to* MALL_2.

Proof. Let us suppose there existed such a translation and consider the type $\mu X.1 \oplus X$. Since this type contains an infinite number of non-equivalent proofs which can all be distinguished by definable predicates $(\mu X.1 \oplus X) \multimap 1 \oplus 1$, its semantics should be infinite, leading to a contradiction. □

To provide a more specific comparison and understand the expressiveness added by type fixpoints, we study the set of predicates computed within Girard's Elementary Linear Logic (ELL) [18].

Elementary Linear Logic corresponds to removing from usual Linear Logic the principles of *digging* $!A \multimap !!A$ and *dereliction* $!A \multimap A$. As these are the only principles governing exponential connectives that change the number of '!' in formulas, the proofs in ELL have a natural stratification, i.e. a given connective has a fixed depth – the number of '!' modalities it is in the scope of – which does not change during normalisation. As a consequence, depth of proofs (i.e. the maximal depth of the connectives in the proof) coincide with a notion of complexity.

Notation III.2. We consider the stratified Church encoding of strings: $\texttt{Str} = \forall X.\texttt{Str}[X]$, where $\texttt{Str}[X] = !(X \multimap X) \multimap !(X \multimap X) \multimap !(X \multimap X)$. We also define $\texttt{Bool} = 1 \oplus 1$.

It is known that ELL characterises Kalmar's class of elementary functions from natural numbers to natural numbers [29]. A later result by Baillot [13] shows[5] that the type $!\texttt{Str} \multimap !^{k+2}\texttt{Bool}$ in μELL_2 corresponds exactly to the class $k-\mathbf{Exptime}$ of predicates computable in time bounded by[6] $2 \uparrow^k (p(n))$, where p is a polynomial and n is the size of the input. In particular, the following theorem mentioned in the introduction is a consequence of Baillot's general result.

[5]Baillot works with Elementary *Affine* Logic, but as said before, weakening makes no difference here.

[6]We use Knuth's up-arrow notation [30] for iterated exponentials: $2 \uparrow^{k+1} (n) = 2^{2\uparrow^k(n)}$, and $2 \uparrow^0 (n) = n$.

Theorem III.3. *The type* $!\texttt{Str} \multimap !^2\texttt{Bool}$ *in* μELL_2 *characterises the class of predicates computable in* polynomial time.

Note also that further developments building upon this result [31], [32] all make use of type fixpoints. We here show that forbidding type fixpoints puts a huge restriction on the computable functions representable in the system. Indeed, we will show the following theorem.

Theorem III.4. *The type* $!\texttt{Str} \multimap !^2\texttt{Bool}$ *in* ELL_2 *characterises the class of* regular languages.

Proving extensional completeness, i.e. that any regular language can be computed by an element of type $!\texttt{Str} \multimap !^2\texttt{Bool}$ in ELL_2, is a quite easy task. Indeed, any deterministic finite automaton with set of states Q and transition function δ can be encoded as a proof of[7] $!\texttt{Str} \multimap !^2\texttt{Bool}$: states are represented as $\oplus_{q\in Q}1$, and then one simply takes an argument s in $\texttt{Str}$ and gives it as arguments the functions $\delta(\_, 0)$ and $\delta(\_, 1)$.

B. *Soundness for regular languages in* ELL_2

The proof of soundness uses the known method of *semantic evaluation*, and the theorem generalises Hillebrand and Kanellakis' [9] result characterising the languages computable by simply typed lambda-terms. Using the finiteness of the semantics, we show that any function computable by a program of type $!\texttt{Str} \multimap !^2\texttt{Bool}$ can be recognized by a finite automaton[8].

The first step in the proof of soundness is to understand the form of proofs of $!\texttt{Str} \multimap !^2\texttt{Bool}$ in ELL_2.

Lemma III.5. *Up to commutations, a proof π of* $!\texttt{Str} \multimap !^2\texttt{Bool}$ *is of the form:*

$$\dfrac{\dfrac{\dfrac{\dfrac{\hat{\pi}}{\vdash \texttt{Str}[A_1]^\perp, \ldots, \texttt{Str}[A_n]^\perp, !\texttt{Bool}}}{\vdash \texttt{Str}^\perp, \ldots, \texttt{Str}^\perp, !\texttt{Bool}}}{\vdash ?\texttt{Str}^\perp, \ldots, ?\texttt{Str}^\perp, !!\texttt{Bool}}\,!}{\dfrac{\vdash ?\texttt{Str}^\perp, !!\texttt{Bool}}{\vdash !\texttt{Str} \multimap !!\texttt{Bool}}\,\parr}$$

This defines a map $\hat{(\_)}$ *from proofs of* $!\texttt{Str} \multimap !!\texttt{Bool}$ *to proofs of* $\texttt{Str}[A_1]^\perp, \ldots, \texttt{Str}[A_n]^\perp, !\texttt{Bool}$.

See appendix section C-A for the proof. Now, a crucial observation is that *the A_i in the previous lemma can be taken in* MALL_2 *w.l.o.g.* This is where the *stratification* property of ELL_2 plays a key role.

Lemma III.6. *For any proof π of* $!\texttt{Str} \multimap !^2\texttt{Bool}$*, there is a proof π' whose witnesses $A_1, \ldots, A_n$ are in* MALL_2*, and which decides the same language as π.*

See appendix section C-B for the proof. Thus, from now on, we assume that $A_1, \ldots, A_n$ are MALL_2 types. We are now ready to prove soundness by using any finite semantics $[\![-]\!]$ of MALL_2.

[7]In fact it comes from $\texttt{Str} \multimap !\texttt{Bool}$ by functorial promotion.

[8]We will actually use a classical characterization of regular languages in terms of monoid morphisms, which is still much closer to finite automata than to regular expressions.

Definition III.7. Let $w \in \{0,1\}^*$ and A be a MALL_2 type. We define the map $||w||_A : \mathrm{End}([\![A]\!])^2 \to \mathrm{End}([\![A]\!])$ $||w||_A(f_0, f_1) = f_{w_1} \circ \ldots \circ f_{w_n}$.

Proposition III.8. *Let $w \in \{0,1\}^*$ and $\bar{w}$: $\mathtt{Str}$ be its encoding. For any MALL_2 type A and $f_0, f_1 : A \multimap A$, $\bar{w}(!f_0, !f_1)$ normalizes into some $!g$ with $g : A \multimap A$, and $||w||_A([\![f_0]\!], [\![f_1]\!]) = [\![g]\!]$.*

Lemma III.9. *The normal form of the following proof:*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\hat{\pi}}{\vdash \mathtt{Str}[A_1]^\perp,\ldots,\mathtt{Str}[A_n]^\perp, !\mathtt{Bool}}}{\vdash \mathtt{Str}^\perp,\ldots,\mathtt{Str}^\perp,!\mathtt{Bool}}}{\vdash ?\mathtt{Str}^\perp,\ldots,?\mathtt{Str}^\perp,!!\mathtt{Bool}}\,!}{\vdash ?\mathtt{Str}^\perp,!!\mathtt{Bool}} \qquad \dfrac{\bar{w}}{\vdash !\mathtt{Str}}\,!}{\vdash !!\mathtt{Bool}}\,\mathit{cut}$$

is completely determined by the functions $||w||_{A_i}$.

This uses crucially the fact that our non-trivial finite semantics $[\![-]\!]$ can distinguish between the two elements of $\mathtt{Bool}$.

Proposition III.10. *For all MALL_2 types A and all $f_0, f_1, g \in \mathrm{End}([\![A]\!])$, the following language is regular:*

$$\mathcal{L} = \{w \in \{0,1\}^* \mid ||w||_A(f_0, f_1) = g\}$$

Proof. Let $\phi : \{0,1\}^* \to \mathrm{End}([\![A]\!])$ be defined as $\phi(w) = ||w||_A(f_0, f_1)$. We have $\mathcal{L} = \phi^{-1}(\{g\})$. Since ϕ is a monoid morphism of finite codomain, $\mathcal{L}$ is regular. □

The finiteness of the $\mathrm{End}([\![A_i]\!])$ is ensured by the existence of a finite semantics of MALL_2. This ends the proof of soundness, given that the set of regular languages is closed under boolean operations.

C. Overcoming the expressivity barrier

Analyzing the above proof reveals that fundamentally, what restricts the computational power is a conjunction of two facts:

1) we know in advance the $A_1, \ldots, A_n$ with which the input $\mathtt{Str}$ will be instantiated, that is, the types of iterations;
2) these A_i are morally finite data types, since they admit finite semantics.

This makes it impossible to iterate over, say, the configurations of a Turing machine, since their size depends on the input and the type A_i cannot "grow" to accomodate data of variable size.

If we stay at depth 2 in ELL_2, there is no way of avoiding the second fact (see the proof of Lemma III.6), so if we want to retrieve something closer to Baillot's theorem – by which we mean a larger complexity class than regular languages – without resorting to type fixpoints, we should try to circumvent the first obstacle. That means that the A_i should vary with the input. Thus, we are led to consider *existential* input types so that an input can provide a witness upon which A_i will depend.

This is indeed the topic of the sequel paper [1] which will study such an input type inspired by finite model theory, following Hillebrand's thesis [14]. To conclude this section, let us stress that the sequel will depend crucially upon an *effective* finite semantics. For our purpose here, finiteness is enough to prove the existence of a finite automaton, and thus the regularity of a language, even without computing this automaton; but in general, performing evaluation within a model requires complexity bounds on the operations supported by the semantics. The next section investigates precisely such an effective semantics, and it will conclude with a space complexity bound on instantiation.

Remark III.11. In fact there is a third fact which plays a role in bridling the complexity: the shape of the type $\mathtt{Str}$ which codes sequential iterations (but the same could be said of Church encodings of free algebras – with such inputs one characterizes regular tree languages).

For instance, let us consider as inputs circuits made of $\mathtt{true}$ constants and $\mathtt{nand}$ gates, renpresented by the type

$$\forall X.!X \multimap !(X \otimes X \multimap X) \multimap !(X \multimap X \otimes X) \multimap !X$$

where $X \multimap X \otimes X$ is a duplication gate used to represent fan-out. Then instantiating this with $X = \mathtt{Bool}$ and the obvious evaluation maps gives us an encoding of the circuit value problem, which is P-complete.

Although this input type seems morally less legitimate than Church encodings, it is hard to pinpoint precisely why it should be rejected.

IV. Coherence spaces

In this section, we will detail a second finite semantics for MALL_2. Indeed, after finding the syntactic model based on observational equivalence, the authors realised a finite semantics of MALL_2 had been lying around all along. Indeed, Girard's coherence spaces model can be restricted to a finite model when considering the exponential-free second order fragment of linear logic. While around for more than thirty years, the authors are not aware of anyone noticing this fact beforehand.

A. Variable types as normal functors

Recall that a *coherence space* is an undirected (reflexive) graph, i.e. a pair $X = (|X|, \frown_X)$ of a set $|X|$ – the *web* of X – and a symmetric and reflexive relation $\frown_X \subseteq |X| \times |X|$ – its *coherence relation*. We refer to [33] for the definition of operations $\otimes, \parr, \&, \oplus, (-)^\perp$ on coherence spaces, and for the notion of *linear map* between coherence spaces. This defines a category CohL of coherence spaces and linear maps which is $*$-autonomous with finite products and coproducts, i.e. a model of MALL_0.

In order to interpret second-order quantification, we want to represent types parameterized by free type variables as functors. This stumbles on the fact that while the binary connectives are *covariant* bifunctors on CohL, linear negation is a *contravariant* endofunctor. In dinatural models of polymorphism, this would lead to considering multivariant functors (cf. Section V). Instead, we work in a "category of embeddings" (see [34]) to make negation covariant.

Definition IV.1. An *embedding* of a coherence space X into a space Y is an injection $f : |X| \to |Y|$ such that $x \frown_X x' \Leftrightarrow f(x) \frown_Y f(x')$.

The category CohI has as objects the coherence spaces, and as morphisms the embeddings.

Proposition IV.2. $(-)^\perp$ *is a* covariant *endofunctor of* CohI.

Proof. If X is an induced subgraph of Y, then its complement $X^\perp$ is an induced subgraph of $Y^\perp$. □

Girard interprets second-order types by functors which are not only covariant but *normal* – this is the fundamental notion of this section, which we present now.

Definition IV.3. A functor is *normal* if it preserves filtered colimits and finite pullbacks.

The name comes from Girard's *normal form theorem*:

Theorem IV.4. *Let* $F : \mathsf{CohI}^n \to \mathsf{CohI}$ *be a functor,* $|F|$ *be the covariant presheaf obtained by taking the web, and* $\mathsf{El}(|F|)$ *be its category of elements.*

F *is normal if and only if, for any coherence space* X *and point* $x \in |F(X)|$*, the slice category* $\mathsf{El}(|F|)/(X, x)$ *admits a* finite initial *object* (X', x').

In this case, (X', x') *is initial in its own slice category. We call an object of* $\mathsf{El}(|F|)$ *enjoying this property a* normal form.

Remark IV.5. Girard's normal functors correspond to Kock's *finitary polynomial functors*. See the discussion in [35, §1.18–1.21] for a survey of related notions of functors and variants of the normal form theorem.

Definition IV.6. Let F be a normal functor. We define $\text{NF}(F)$ to be its set of isomorphism classes[9] of normal forms.

Now, types with n variables will be interpreted as normal functors $F : \mathsf{CohI}^n \to \mathsf{CohI}$. Then a normal form of F is a tuple $(X_1, \ldots, X_n, x)$ with $x \in |F(X_1, \ldots, X_n)|$; initiality means that these X_i are "minimal" for ensuring the existence of the point x.

Notation IV.7. We shall use the notation $\langle X_1, \ldots, X_n \vdash x\rangle$ for normal forms $(X_1, \ldots, X_n, x) \in \text{NF}(F)$.

We now introduce a notion of *degree* of a normal functor, which will witness the finiteness of the interpretation of MALL (and later of MALL_2).

Definition IV.8. Let $F : \mathsf{CohI}^n \to \mathsf{CohI}$ be a normal functor. We define the degree $\deg F$ as:

$$\sup\{\text{card}(|X_i|) \mid \langle X_1, \ldots, X_n \vdash x\rangle \in \text{NF}(F), i \in \{1, \ldots, n\}\}.$$

We say F is *finite* if it preserves finiteness of cardinality and is of finite degree.

Note that a normal functor may have finite but unbounded normal forms, so that its degree is in fact infinite. Typically, this is the case for the exponential modalities, which explains why the model is not finite for full second-order linear logic. In fact, the degree characterises a notion of *polynomial growth* (at least in the case of functors of one variable).

Theorem IV.9. *Let* $F : \mathsf{CohI} \to \mathsf{CohI}$ *be a normal functor. There exists* $d \in \mathbf{N}$ *s.t.* $\text{card}(|F(X)|) = \mathcal{O}(\text{card}(|X|^d))$ *if and only if* F *is a finite functor. In that case,* $\deg F$ *is the smallest such* d.

Proof. In appendix, Section D-A. □

B. Interpretation of MALL_2 *types*

We describe here how MALL_2 formulae are translated into finite normal functors. The description of the "morphisms of variable types" in the model is postponed to the next subsection.

The multiplicative and additive connectives of linear logic are then interpreted in a natural way: we apply the usual coherence space operations pointwise on functors.

Proposition IV.10. *If* F *and* G *normal functors in* $\mathsf{CohI}^n \to \mathsf{CohI}$*, then* $\deg F^\perp = \deg F$*,* $\deg F \otimes G = \deg F + \deg G$*, and* $\deg F \oplus G = \max\{\deg F, \deg G\}$.

Proof. The case of negation is straightforward as $|F(X_1, \ldots, X_n)| = |F^\perp(X_1, \ldots, X_n)|$. For the $\oplus$ case, one easily checks that $\text{NF}(F \oplus G) \cong \text{NF}(F) \uplus \text{NF}(G)$.

Only the $\otimes$ case needs to be carefully checked. Consider $\langle \vec{Z} \vdash (x, y)\rangle \in \text{NF}(F \otimes G)$, and the corresponding normal forms $\langle \vec{X} \vdash x\rangle \in \text{NF}(F)$ and $\langle \vec{Y} \vdash y\rangle \in \text{NF}(G)$. Then one can show, from minimality of $\vec{Z}$, that[10] $\vec{X} \cup \vec{Y} \subseteq \vec{Z}$. Moreover, since $x \in F(\vec{Z})$ and $y \in G(\vec{Z})$, we have $\vec{X} \cup \vec{Y} \supseteq \vec{Z}$. Thus $\vec{X} \cup \vec{Y} = \vec{Z}$ and $\deg(F \otimes G) \leqslant \deg F + \deg G$. The converse inequality is obtained by noticing that if $\langle \vec{X} \vdash x\rangle \in \text{NF}(F)$ and $\langle \vec{Y} \vdash y\rangle \in \text{NF}(G)$, then $\langle \vec{X} \oplus \vec{Y} \vdash (x, y)\rangle \in \text{NF}(F \otimes G)$. □

Next, we interpret quantifiers using normal forms.

Definition IV.11. Let $F : \mathsf{CohI} \to \mathsf{CohI}$ be a normal functor. We endow $\text{NF}(F)$ with a non-reflexive coherence relation: $\langle X \vdash x\rangle$ and $\langle Y \vdash y\rangle$ are strictly incoherent exactly when there are embeddings $\iota_X : X \to Z$ and $\iota_Y : X \to Z$ such that $F(\iota_X)(x)$ and $F(\iota_Y)(y)$ are strictly incoherent in $F(Z)$.

The trace $\text{Tr}(F)$ is defined as the coherence space made of the self-coherent normal forms of F, equipped with the coherence relation above.

Definition IV.12. Let F be a normal functor $F : \mathsf{CohI}^{n+1} \to \mathsf{CohI}$ which is the interpretation of a proof π with variables $X_1, \ldots, X_n, X_{n+1}$. The interpretation of the proof π' obtained by introducing a universal quantifier $\forall X_{n+1}$ is interpreted by $\text{Tr}_X(F)$, the functor which maps a family $(Y_i)_{i=1,;n}$ to $\text{Tr}(F(Y_1, \ldots, Y_n, \_))$.

Proposition IV.13. *For any normal functor* $F : \mathsf{CohI}^{n+1} \to \mathsf{CohI}$*,* $\deg \text{Tr}_X(F) \leqslant \deg F$.

[9]For isomorphisms in $\mathsf{El}(|F|)$.

[10]For legibility purposes, we assume that F preserves inclusions, and work with inclusions instead of embeddings; also, all operations are applied componentwise on the n-tuples $\vec{X}$, $\vec{Y}$, $\vec{Z}$.

Proof. Suppose $\langle Y_1, \ldots, Y_n \vdash \langle X_1 \vdash x \rangle\rangle \in \text{NF}(\text{Tr}_X(F))$. It suffices to check that $\langle Y_1, \ldots, Y_n, X_1 \vdash x\rangle \in \text{NF}(F)$. ☐

Theorem IV.14. *Finite normal functors are closed under* MALL_2 *connectives.*

Proof. The only non-trivial observation remaining is that if F is finite, then $\text{Tr}_X(F)$ preserves finiteness of cardinality. This reduces to the one-variable case : if $F : \mathsf{Cohl} \to \mathsf{Cohl}$ is a finite normal functor, then $\text{Tr}(F)$ is a finite coherence space. Actually one can show that $\text{NF}(F)$ is finite: for any $\langle X \vdash x\rangle$, $\text{card}(|X|) \leq \deg(F)$ so there are finitely many choices for X (up to isomorphism), and for each X there are finitely many possible x because F preserves finiteness of cardinality. ☐

Corollary IV.15. *All* MALL_2 *formulae are interpreted in the coherence space model as finite normal functors.*

C. Finitely many morphisms

Essentially, we have just concluded the proof that the coherence space model is finite. Nevertheless, to understand why that is the case, we have to say a few words about the interpretation of proofs with free type variables – that is, about our notion of morphisms between normal functors. Indeed, the right notion of finite semantics for our applications is the finiteness of hom-sets. By monoidal closure, it suffices to consider[11] the sets $\text{Hom}(1, F)$ for $F : \mathsf{Cohl}^n \to \mathsf{Cohl}$.

When $n = 0$, F is a closed type and this should reduce to the propositional case, i.e. the model CohL of MALL_0. Then $\text{Hom}(1, F)$ is the set of linear maps from 1 to X, which are the same as *cliques* (sets of pairwise coherent points) in X. Generalizing to $n > 0$, $\text{Hom}(1, F)$ consists of *variable cliques*, i.e. families of cliques $c_{X_1,\ldots,X_n}$ in each $F(X_1, \ldots, X_n)$, satisfying a uniformity condition called the *mutilation property* [2]. The point of this definition is that the variable cliques of F are in bijection with the cliques of its universal closure $\text{Tr}_{X_1,\ldots,X_n}(F)$.

If F is a finite normal functor, $\text{Tr}_{X_1,\ldots,X_n}(F)$ also is, and such a functor with zero arguments is none other than a finite coherence space, which contains finitely many cliques, hence:

Theorem IV.16. *Coherence spaces provide a finite semantics for* MALL_2.

D. An effective combinatorial description

Actually, the points of $\text{Tr}_{X_1,\ldots,X_n}(F)$ – and more generally, of $\text{NF}(F)$ – may be described in a very concrete syntactic way when F is MALL_2-definable. Our exposition here is inspired by the description of normal functors over the category of sets and injections in [36, §IV.5].

The idea is to see the webs $|X_i|$ in a normal form $\langle X_1, \ldots, X_n \vdash x\rangle$ as sets of *bound variables*, $\langle \cdot \vdash \cdot \rangle$ being a binder. Recall that these normal forms are considered up to isomorphism in a category of elements $\mathsf{El}(|F|)$; these isomorphisms should be understood as α-renamings. The initiality condition on normal forms means that all the variables in the $|X_i|$ appear free in x – otherwise, one could take a smaller X'_i. Note that the coherence spaces X_i specify not only which variables are bound, but also the coherence relation between them.

In turn, this x is a syntax tree with binders – indeed the interpretation of quantifiers uses (unary) normal forms. The grammar of terms is as follows:

$$x ::= a \in \text{Var} \mid (x, x) \mid \text{inl}(x) \mid \text{inr}(x) \mid \langle X \vdash x\rangle$$

where $|X| \subset \text{Var}$. The functorial action of a MALL_2-definable functor F on embeddings then corresponds to substituting the free variables of x – indeed an embedding $\iota_i : X_i \to Y_i$ is an assignment of variables.

The shape of the term is in fact heavily constrained by the MALL_2 formula which F interprets. With this point of view, one sees that $\deg(F)$ is the maximum number of leaves which a syntax tree in $\text{NF}(F)$ can have.

With such a concrete description it becomes easier to see how one can compute operations on these variable types and cliques. For instance:

Proposition IV.17. *For any* MALL_2*-definable functor* F*, the non-reflexive coherence relation on* $\text{NF}(F)$ *is decidable.*

Indeed, the definition of coherence on $\text{NF}(F)$ involves a quantification over all spaces in which the normal form(s) involved may embed, but this quantification can be bounded as remarked in [2, Remark C.3] – this was by the way Girard's motivation for restricting the model of *qualitative domains* to its binary case, i.e. coherence spaces. As a corollary:

Theorem IV.18. *For any* MALL_2*-definable functor* F*,* $\text{Tr}_{X_1,\ldots,X_n}(F)$ *is computable. Thus, one can enumerate the variable cliques of* F.

This is in stark contrast to the undecidability of MALL_2 provability which prevents any quotient of the syntax from having this property. Note also that equality is decidable in the coherence space model – just compare syntax trees! – unlike the case of the observational quotient.

E. Logarithmic space instantiation

We conclude our treatment of coherence spaces by establishing an effectiveness result with a finer complexity bound. Although it is simple, it will play a crucial role in the sequel to this paper [1]. Indeed, as mentioned at the end of section III, the sequel will consider existential input types; thus, it will need to perform instantiations of polymorphic programs at the witness provided by the input.

Proposition IV.19. *Fix any* MALL_2 *one-parameter variable type* $F : \mathsf{Cohl} \to \mathsf{Cohl}$*, and any variable clique* θ *in* $\text{Tr}(F)$*. The function* $X \mapsto \theta_X$*, which instantiates* θ *to a clique in* $F(X)$*, can be computed in* logarithmic space*.*

Proof. In appendix, Section D-B. ☐

[11] That said, one can describe directly $\text{Hom}(F, G)$ by writing down a version of the mutilation property for "variable linear maps". It turns out to be highly non-trivial – and overlooked in Girard's papers – that this property composes, though a proof is known by experts and considered folklore (it is often credited to Eugenio Moggi).

V. Finite models and parametricity

For applications in complexity, it is desirable that a MALL_2 model be finite and effective. As we have seen, both the observational quotient and the coherence space model are finite, but only the second one is effective. Another property which is often desirable at both theoretical and applicative level is *parametricity*. The founding idea of parametric polymorphism [19] is that a proof of $\forall X\ A$ should behave "in the same way" on all its possible instantiations $A[B/X]$. Hence parametricity can be seen as a "smallness" condition on the interpretation of quantifiers.

In this section we compare our previous models with two fundamental properties of parametric models, namely *dinaturality* [20] and *constancy* [21]. Dinaturality is a well-investigated categorical notion which was proposed as a formalisation of parametricity and provides a third example of a witness-erasing semantics: in dinatural models quantifiers correspond to *ends/coends*, i.e. universal wedges/co-wedges. This interpretation leads then to identify proofs with different witnesses.

Constancy is a property which holds in all dinatural (and more generally, parametric) models of System F and expresses a weaker "smallness" condition for quantifiers. We show that, while the coherence spaces model is not dinatural, it has the constancy property. We also show an essential incompatibility between constancy and compact closed models of MALL_2.

A. Linear fibrations

In order to compare different models, we need to introduce a general notion of MALL_2 model. We essentially follow the definition in [37]. We let a *linear category* be a *-autonomous category with finite products and co-products.

Definition V.1. A *linear fibration* is a pair (L, S), with S a category with finite products and a distinguished object Ω, and L an indexed category over S[12] such that:

- Ω is a *generic object*: for all object I of S, $Obj(\mathsf{L}(I)) \simeq \mathsf{S}(I,\Omega)$ and for all $f \in \mathsf{S}(I,J)$, the functor $f^\sharp := \mathsf{L}(f)$ acts by pre-composition;
- the fiber categories $\mathsf{L}(I)$ are linear;
- for all $I, J \in Obj(\mathsf{S})$, the projection functor $\pi^\sharp_{I,J}$ – where $\pi_{I,J} \in \mathsf{S}(I\times J, I)$ is the projection arrow – has indexed right and left adjoints $\Pi_{I,J}$ and $\Sigma_{I,J}$, respectively.

Linear fibrations are the linear counterpart of $\lambda 2$-*fibrations* [38], the standard notion of model for System F.

The syntactic model of MALL_2 yields a linear fibration by letting S be the category with natural numbers as objects (with the sum as product) and m-tuples of n-ary MALL_2 types as arrows $n \to m$. Then $\mathsf{L}(n)$ is the syntactic category generated by proofs of n-ary types (considered up to $\beta\eta$-equivalence). The model arising from the observational quotient also yields a linear fibration, defined by quotienting the fiber categories $\mathsf{L}(n)$. In both cases one easily verifies that $\forall$ and $\exists$ are adjoint

[12]That is, L is a pseudo-functor from S to the 2-category of categories Cat.

$$\begin{array}{ccc} C & \xrightarrow{\delta_X} & FXX \\ \downarrow{\scriptstyle \delta_Y} & & \downarrow{\scriptstyle CXf} \\ FYY & \xrightarrow{CfY} & FXY \end{array} \qquad \begin{array}{ccc} FYX & \xrightarrow{CfX} & FXX \\ \downarrow{\scriptstyle CYf} & & \downarrow{\scriptstyle \omega_X} \\ FYY & \xrightarrow{\omega_Y} & D \end{array}$$

Fig. 6: Wedge and co-wedge conditions

to the "projection functor" consisting in adding a dummy variable to a type.

The coherence spaces model yields a linear fibration by letting S be the category with natural numbers as objects and m-tuples of finite normal functors as arrows. In that case, $\mathsf{L}(n)$ is the category of variable cliques between n-ary finite normal functors. A standard property of the trace is that the, given normal functors F, G of respective arities n and $n+1$, the variable cliques between $F'(\vec{X}, Y) = F(\vec{X})$ and $G(\vec{X}, Y)$ are in bijection with those between $F(\vec{X})$ and $\mathrm{Tr}_Y(G)$, i.e. $\mathrm{Tr}_Y(G)$ is right adjoint to the projection functor.

B. Dinaturality

The interpretation of proofs as dinatural transformations was originally proposed as a categorical formalization of parametricity in [20]. At the basis of this construction is the remark (see section IV) that logical types do not correspond to usual covariant functors, but to functors of mixed variance. For such functors one needs to generalize the usual notion of natural transformation.

Definition V.2. Given functors $F, G : \mathbb{C}^{op} \times \mathbb{C} \to \mathbb{D}$, a *dinatural transformation* from F to G is a family of arrows $\theta_X \in \mathbb{D}(FX, GX)$ such that for all $X, Y \in \mathbb{C}$ and $f \in \mathbb{C}(X,Y)$, $GXf \circ \theta_X \circ FfX = GfY \circ \theta_Y \circ FYf$.

Dinatural transformations are in general less well-behaved than usual natural transformations. In particular, they need not compose, hence they do not form a category. However, it is well-known that MLL_0 proofs provide a category of composable dinatural transformations [39].

Given a functor $F : \mathbb{C}^{op} \times \mathbb{C} \to \mathbb{D}$, a *wedge for F* is a pair (C, δ) of an object $C \in \mathbb{D}$ and a dinatural transformation[13] $\delta : C \Rightarrow F$. Dually, a *co-wedge for F* is an object $D \in \mathbb{D}$ together with a dinatural transformation $\omega : F \Rightarrow D$ (fig. 6).

The dinatural interpretation of second order quantifiers is by means of *universal wedges/co-wedges* (also called *ends* and *coends*, indicated as $\int_X F$ and $\int^X F$).

Remark V.3. The universality of $\delta : \int_X F \to F$ is equivalent to the existence of a dinatural isomorphism between the set of arrows from 1 to F and the set of dinatural transformations from 1 to $\int_X F$, i.e. to the fact that $\int_X$ is right-adjoint to a projection functor, as in linear fibrations.

Since dinatural transformations need not compose, a *dinatural model* is usually defined by restricting to a class of composable dinatural transformations and considering *relativized ends*

[13]We here denote by C the constant functor equal to an object C.

$$
\dfrac{\dfrac{\dfrac{\overset{\pi}{\vdash C, B^\perp} \qquad \vdash B, B^\perp}{\vdash C, B^\perp, B \otimes B^\perp}}{\vdash C \parr B^\perp, B \otimes B^\perp}}{\vdash C \parr B^\perp, \exists X(X \otimes X^\perp)}
\qquad
\dfrac{\dfrac{\dfrac{\vdash C, C^\perp \qquad \overset{\pi}{\vdash C, B^\perp}}{\vdash C, B^\perp, C \otimes C^\perp}}{\vdash C \parr B^\perp, C \otimes C^\perp}}{\vdash C \parr B^\perp, \exists X(X \otimes X^\perp)}
$$

Fig. 7: Proofs identified by a co-wedge condition

$$
\begin{array}{ccc}
B \otimes C^\perp & \xrightarrow{B \otimes \pi^\perp} & B \otimes B^\perp \\
{\scriptstyle \pi \otimes C^\perp}\downarrow & & \downarrow{\scriptstyle \omega_B} \\
C \otimes C^\perp & \xrightarrow{\omega_C} & \exists X(X \otimes X^\perp)
\end{array}
$$

Fig. 8: Co-wedge condition for the proofs in fig. 7

and coends, that is, wedges and co-wedges which are universal among the class of dinaturals considered (see e.g.[20]). In this situation, by remark V.3, dinaturals allow to define models in the sense recalled in the previous subsection.

By considering C and D in fig. 6 as $\forall XF$ and $\exists XF$ respectively, one can see that the wedge/co-wedge conditions lead to the identification of proofs with different universal or existential witnesses. E.g., if π is a proof of $\vdash C, B^\perp$, any dinatural model identifies the proofs π_1, π_2 in fig. 7 , their equivalence following from the co-wedge condition (fig. 8).

Remark V.4. The two proofs π_1, π_2 are also identified by the observational quotient. Indeed, if π' is a proof of $(C^\perp \otimes B) \otimes \forall X(X \multimap X), P$, for some MALL_0 type B, then the proof net associated with π' must contain an ax edge with target the two dual occurrences of X. The reduction algorithm described in section II applied to the proof nets associated with $\mathtt{cut}(\pi_1, \pi')$ and $\mathtt{cut}(\pi_2, \pi')$ then leads to two configurations as in fig. 9 (where $\mathcal{R}$ indicates the proof net associated with π), which converge on the same proof net. In other words, $\exists X(X \otimes X^\perp)$ satisfies the co-wedge condition. We do not yet know whether all quantified formulas in the observational model satisfy wedge/co-wedge conditions.

C. Dinaturality and the coherence spaces model

To compare the coherent model and dinaturality, we first observe that a normal functor $F : (\mathsf{CohL}^{\mathrm{op}} \times \mathsf{CohL})^n \to \mathsf{CohL}$ induces a normal functor $\overline{F} : \mathsf{Cohl}^n \to \mathsf{Cohl}$, obtained by removing the contravariance. We can then compare the sets $\mathsf{din}(\mathsf{CohL})(1, F)$ and $\mathrm{Tr}_X(\overline{F})$. The first result is then that the trace is not an end.

Proposition V.5. $\mathrm{Tr}_X(\overline{F \multimap G}) \neq \int_X F \multimap G$.

Proof. If $\mathrm{Tr}_X(\overline{F \multimap G})$ were an end, then its cliques would be in bijection with the natural transformations from F to G. But $\mathrm{Tr}_X(X \otimes X \multimap X)$ contains the intersection function $i = \{\langle\{x\} \vdash ((x,x),x)\rangle\}$, which is not a natural transformation from $X \otimes X$ to X (understood as functors from CohL to CohL). If it were, then all linear functions would satisfy $f(c \cap d) = f(c) \cap f(d)$, even when the cliques c and d are incoherent. $\square$

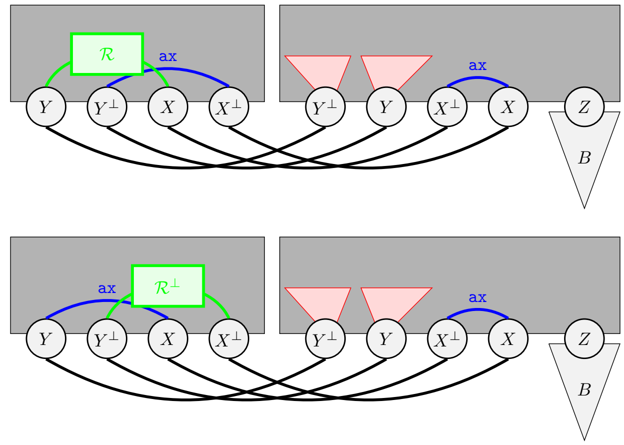


Fig. 9: Big step reduction for $\mathtt{cut}(\pi_1, \pi')$ and $\mathtt{cut}(\pi_2, \pi')$.

We now show that the trace is not even a wedge.

Proposition V.6. *$\mathrm{Tr}_X(\overline{F})$ is not a wedge for F.*

Proof. In appendix, Section E $\square$

It's worth mentioning that two dinatural variants of the coherence spaces model are sketched in [40].

D. The constancy property

We now consider a second property investigated in connection with parametricity, related to *axiom C* in [21]. The latter states that, if A is a type in which X does not occur free and π is a proof of $\forall XA$, then for all types B, B', the instantiations $\pi[B/X]$ and $\pi[B'/X]$ are equal. This means intuitively that a polymorphic function whose output type does not depend on the input type variable X does not itself depend on X.

Axiom C is independent of System F, but consistent with it. Following [21], Axiom C for MALL_2 can be restated as a *constancy* property for linear fibrations:

Definition V.7 (constancy)**.** A linear fibration has the *constancy property* if all projection functors are full.

In other words, a linear fibration (L, S) has the constancy property when for all objects I, J in its base category, there is an isomorphism between $\mathsf{S}(I \times J)(\_ \circ \pi^\sharp_{I,J}, \_ \circ \pi^\sharp_{I,J})$ and $\mathsf{S}(I)(\_, \_)$. Indeed, from the isomorphism between $\mathsf{S}(I \times J)(F \circ \pi^\sharp_J, G)$ and $\mathsf{S}(I)(F, \Pi_J.G)$ it follows that the fibration has the constancy property exactly when the arrows from F to $\Pi_{I,J}(G \circ \pi^\sharp_{I,J})$ are in bijection with those from F to G.

It is well-known that dinatural models have the constancy property, as the two models considered in this paper.

Lemma V.8. *The coherence space model and the observational model have the constancy property.*

Proof. For the observational model this is clear. In the coherence spaces model: there exists, for all n-ary normal functor F, an isomorphism between $\text{NF}(F)$ and $\text{NF}(F')$, where F' is the functor $F'(X_1, \ldots, X_n, X_{n+1}) = F(X_1, \ldots, X_n)$, given by $(X_1, \ldots, X_n, x) \in \text{NF}(F)$ iff $(X_1, \ldots, X_n, 0, x) \in \text{NF}(F)$ where 0 is the empty coherence space. $\square$

We conclude by showing that compact closed MALL_2 models cannot have this property. This implies, in particular, that there is no compact closed dinatural MALL_2 model.

Proposition V.9. *No compact closed fibration has the constancy property. In particular, no such model is dinatural.*

Proof. Suppose p is such a model. Let us consider the unit and counit arrows $\eta_X : 1 \Rightarrow X \otimes X^*$ and $\epsilon_X : X \otimes X^* \Rightarrow 1$ and let $\theta_X : 1 \Rightarrow 1$ their composition. The instantiation of θ_X at 0 is the composition of a terminal and an initial arrow (i.e. a zero arrow). The instantiation of θ_X at 1 is the composition of two identities, hence the identity arrow from 1 to 1. $\square$

Remark V.10. The composition problem in proposition V.9 had already been noticed in previous attempts to define a variant of the coherence space model in the (compact closed) category of relations (see [36]).

Remark V.11. Dinatural transformations have been employed to prove full-completeness results for MLL_0. For instance, in [41] it is shown that a large class of compact closed categories (including the categories FinVec of finitely dimensional vector spaces and the category FinRel of finite sets and relations) satisfy a property called *feeble full completeness*, which roughly implies that for all MALL_0 type A, the dinaturals from 1 to $[\![A]\!]$ are linear combinations of the interpretation of MALL_0 proof structures.

We call a linear fibration compact closed when all fiber categories are (fiberwise) compact closed. By exploiting remark V.3 and the compact closed structure, from feeble full completeness it can be deduced that several categories (including FinVec and FinRel) are closed under MALL_2-connectives. Hence in particular all ends and coends of MALL_2-definable functors exist (and are thus finite). However, no linear compact closed fibration is a dinatural model, by prop. V.9. This means that in linear compact closed categories there is a payoff between composability of dinaturals and constancy.

A dinatural model of MLL_2 over the category of topological vector spaces (which is not compact closed) is suggested in [42]. As a similar full-completeness property holds in this category, it might be interesting to investigate whether this model yields a finite quotient of MALL_2 proofs.

VI. Conclusion

We showed the finiteness of two different natural semantics of MALL_2– an observational quotient of the syntax, and the historical model of coherence spaces and normal functors – and studied to which extent these models were parametric. This finiteness was applied to understand the difference in computational power between MALL_2 and μMALL, which manifests quantitatively as the gap between regular languages and polynomial time as our results on ELL_2 show.

In the sequel [1], we will pursue this complexity-theoretic line of research by making good use of the effectiveness of the coherence space model. We mention below some other possibilities for further work.

A. Dinatural and dynamic semantics of MALL_2

The last section raised the open problem of building a finite dinatural model of MALL_2. The last section raised the open problem of building a finite dinatural model of MALL_2. While the existence of such a model is suggested by several full completeness results based on dinaturality, proposition V.9 suggests to consider non-degenerate models of MALL_0 (e.g. arising from the *glueing* approach [41]) as a starting point.

The finiteness of the coherence space model suggests a way to look for a finite *game semantics* for MALL_2. Indeed, the points of the web of (hyper)coherence spaces can be seen in some ways as external positions of games [43], [44], [45], [46] in the monomorphic case. The idea would then be to extend this correspondence to second-order types. An encouraging fact is that recent game models of parametric polymorphism [47], [48] seem to be witness-erasing.

In a similar vein, our proof of finiteness of the observational quotient clearly involves a kind of dialogue – though it is hard to find structure in it, e.g. determine how the shape of the formula influences the order of "moves". A dynamic semantics could eventually be provided by finding a notion of "strategy" which composes well, instead of our ad-hoc definition. Current work-in-progress suggests that this dialogue could be formulated as a least fixpoint, hinting at a possible wave-style Geometry of Interaction model [49].

B. Open problems on ELL_2 complexity

We saw that Baillot's characterization of the $k-\mathbf{Exptime}$ hierarchy in μELL_2 does not work for $k = 0$ in ELL_2: in this case it actually captures regular languages. Naturally, we believe that ELL_2 is also less expressive than μELL_2 for $k > 0$, but for now we only know that the complexity class obtained in ELL_2 lies between $(k-1)-\mathbf{Exptime}$ and $k-\mathbf{Exptime}$.

A related question is the complexity of normalizing a proof of $!^k\texttt{Bool}$ in ELL, analogously to normalization at fixed order in the simply typed λ-calculus [4]. We know that it is:

- for $k = 0$: $\mathbf{P}$-complete;
- for $k = 1$: $\mathbf{Pspace}$-hard, in $\mathbf{Exptime}$;
- for $k \geq 2$: $(k-1)-\mathbf{Exptime}$-hard, in $k-\mathbf{Exptime}$.

Acknowledgment

The starting point of this paper was a discussion with Damiano Mazza about complexity in the simply typed λ-calculus. The authors would also like to thank him for discussions about the coherence space model of polymorphism.

Lê Thành Dũng Nguyễn is grateful to Thomas Ehrhard for his knowledge of the folklore on coherence spaces, Paul-André Melliès and Kazushige Terui for many references on the applications of semantics, and especially Pierre Pradic for too many things to list. Thomas Seiller wishes to thank Seng Beng Goh for the discussions about normal functors.

L. T. D. Nguyễn and T. Seiller were partially supported by the ANR project Elica (ANR-14-CE25-0005). T. Seiller was partially supported by the CNRS INS2I JCJC grant BiGRE.

REFERENCES

[1] L. T. D. Nguyễn and P. Pradic, "From normal functors to subpolynomial queries," 2019, in preparation. Rough draft available at https://nguyentito.eu/subpoly.pdf.

[2] J.-Y. Girard, "The system F of variable types, fifteen years later," *Theoretical Computer Science*, vol. 45, pp. 159–192, Jan. 1986.

[3] ——, "Linear logic," *Theoretical Computer Science*, vol. 50, no. 1, pp. 1–101, Jan. 1987.

[4] K. Terui, "Semantic Evaluation, Intersection Types and Complexity of Simply Typed Lambda Calculus," in *RTA'12*, 2012.

[5] I. Walukiewicz and S. Salvati, "Using models to model-check recursive schemes," *Logical Methods in Computer Science*, vol. Volume 11, Issue 2, Jun. 2015.

[6] ——, "Typing weak MSOL properties," *Logical Methods in Computer Science*, vol. Volume 13, Issue 1, Mar. 2017.

[7] K. Aehlig, "A Finite Semantics of Simply-Typed Lambda Terms for Infinite Runs of Automata," *Logical Methods in Computer Science*, vol. 3, no. 3, Jul. 2007.

[8] M. Hofmann and J. Ledent, "A cartesian-closed category for higher-order model checking," in *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. Reykjavik, Iceland: IEEE, Jun. 2017, pp. 1–12.

[9] G. G. Hillebrand and P. C. Kanellakis, "On the Expressive Power of Simply Typed and Let-Polymorphic Lambda Calculi," in *LICS'96*, 1996.

[10] C. Grellois and P.-A. Melliès, "Relational Semantics of Linear Logic and Higher-order Model Checking," in *24th EACSL Annual Conference on Computer Science Logic (CSL 2015)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), S. Kreutzer, Ed., vol. 41. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015, pp. 260–276.

[11] C. Grellois, "Semantics of linear logic and higher-order model-checking," Ph.D. dissertation, Université Paris Diderot, Apr. 2016.

[12] D. Baelde, "Least and Greatest Fixed Points in Linear Logic," *ACM Transactions on Computational Logic*, vol. 13, no. 1, pp. 1–44, Jan. 2012.

[13] P. Baillot, "On the expressivity of elementary linear logic: Characterizing Ptime and an exponential time hierarchy," *Information and Computation*, vol. 241, pp. 3–31, Apr. 2015.

[14] G. G. Hillebrand, "Finite Model Theory in the Simply Typed Lambda Calculus," Ph.D. dissertation, Brown University, Providence, RI, USA, 1994.

[15] U. Schöpp, "Stratified Bounded Affine Logic for Logarithmic Space," in *22nd Annual IEEE Symposium on Logic in Computer Science (LICS 2007)*, Jul. 2007, pp. 411–420.

[16] U. Dal Lago and U. Schöpp, "Computation by interaction for space-bounded functional programming," *Information and Computation*, vol. 248, pp. 150–194, Jun. 2016.

[17] D. Mazza, "Simple Parsimonious Types and Logarithmic Space," in *24th EACSL Annual Conference on Computer Science Logic (CSL 2015)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), S. Kreutzer, Ed., vol. 41. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015, pp. 24–40.

[18] J.-Y. Girard, "Light Linear Logic," *Information and Computation*, vol. 143, no. 2, pp. 175–204, Jun. 1998.

[19] Q. Ma and J. C. Reynolds, "Types, abstraction, and parametric polymorphism. II," in *Mathematical foundations of programming semantics (Pittsburgh, PA, 1991)*, ser. Lecture Notes in Comput. Sci. Springer, Berlin, 1992, vol. 598, pp. 1–40.

[20] E. S. Bainbridge, P. J. Freyd, A. Scedrov, and P. J. Scott, "Functorial polymorphism," *Theor. Comput. Sci.*, vol. 70, no. 1, pp. 35–64, Jan. 1990. [Online]. Available: http://dx.doi.org/10.1016/0304-3975(90)90151-7

[21] G. Longo, K. Milsted, and S. Soloviev, "The genericity theorem and parametricity in the polymorphic lambda-calculus," *Theor. Comput. Sci.*, vol. 121, no. 1-2, pp. 323–349, Dec. 1993. [Online]. Available: http://dx.doi.org/10.1016/0304-3975(93)90093-9

[22] J.-Y. Girard, "Quantifiers in linear logic II," in *Nuovi problemi della logica e della filosofia della scienza*, Corsi and Sambin, Eds. Bologna: CLUEB, 1991, pp. 79–90.

[23] L. Tortora de Falco, "Additives of linear logic and normalization- part I: a (restricted) church-rosser property," *Theoretical Computer Science*, vol. 294/3, pp. 489–524, 2003.

[24] M. Pagani and L. Tortora de Falco, "Strong normalization property for second order linear logic," *Theoretical Computer Science*, vol. 441/2, pp. 410–444, 2010.

[25] H. G. Mairson and K. Terui, "On the Computational Complexity of Cut-Elimination in Linear Logic," in *Theoretical Computer Science*, ser. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Oct. 2003, pp. 23–36.

[26] D. J. Hughes and R. J. Van Glabbeek, "Proof nets for unit-free multiplicative-additive linear logic," *ACM Transactions on Computational Logic (TOCL)*, vol. 6, no. 4, pp. 784–842, 2005.

[27] Y. Lafont and A. Scedrov, "The Undecidability of Second Order Multiplicative Linear Logic," *Information and Computation*, vol. 125, no. 1, pp. 46–51, Feb. 1996.

[28] Y. Lafont, "The undecidability of second order linear logic without exponentials," *The Journal of Symbolic Logic*, vol. 61, no. 02, pp. 541–548, Jun. 1996.

[29] V. Danos and J.-B. Joinet, "Linear logic and elementary time," *Information and Computation*, vol. 183, no. 1, pp. 123–137, May 2003.

[30] D. E. Knuth, "Mathematics and computer science: Coping with finiteness," *Science*, vol. 194, no. 4271, pp. 1235–1242, 1976. [Online]. Available: http://science.sciencemag.org/content/194/4271/1235

[31] P. Baillot, E. De Benedetti, and S. Ronchi Della Rocca, "Characterizing polynomial and exponential complexity classes in elementary lambda-calculus," *Information and Computation*, vol. 261, pp. 55–77, Aug. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0890540118300774

[32] P. Baillot and A. Ghyselen, "Combining Linear Logic and Size Types for Implicit Complexity," in *27th EACSL Annual Conference on Computer Science Logic (CSL 2018)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), D. Ghica and A. Jung, Eds., vol. 119. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018, pp. 9:1–9:21. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2018/9676

[33] J.-Y. Girard, "Linear logic: its syntax and semantics," in *Advances in Linear Logic*, ser. London Mathematical Society Lecture Notes, J.-Y. Girard, Y. Lafont, and L. Regnier, Eds., vol. 222. Cambridge University Press, 1995.

[34] T. Coquand, "Categories of embeddings," *Theoretical Computer Science*, vol. 68, no. 3, pp. 221–237, Nov. 1989.

[35] N. Gambino and J. Kock, "Polynomial functors and polynomial monads," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 154, no. 01, pp. 153–192, Jan. 2013, arXiv: 0906.4931.

[36] A. Bac-Bruasse, "Logique linéaire indexée du second ordre," Ph.D. dissertation, Université Aix-Marseille II, Dec. 2001.

[37] R. A. G. Seely, "Polymorphic linear logic and topos models," *C. R. Math. Rep. Acad. Sci. Canada*, vol. 12, no. 1, pp. 21–26, 1990.

[38] ——, "Categorical semantics for higher order polymorphic lambda calculus," *J. Symbolic Logic*, vol. 52, no. 4, pp. 969–989, 1987. [Online]. Available: https://doi.org/10.2307/2273831

[39] R. Blute, "Linear logic, coherence and dinaturality," *Theoret. Comput. Sci.*, vol. 115, no. 1, pp. 3–41, 1993, 4th Summer Conference on Category Theory and Computer Science (Paris, 1991). [Online]. Available: https://doi.org/10.1016/0304-3975(93)90053-V

[40] P. J. Freyd, J. . Girard, A. Scedrov, and P. J. Scott, "Semantic parametricity in polymorphic lambda calculus," in *[1988] Proceedings. Third Annual Symposium on Logic in Computer Science*, July 1988, pp. 274–279.

[41] A. Schalk and H. P. Steele, "Constructing fully complete models of multiplicative linear logic," *Log. Methods Comput. Sci.*, vol. 11, no. 3, pp. 3:6, 72, 2015. [Online]. Available: https://doi.org/10.2168/LMCS-11(3:6)2015

[42] R. F. Blute and P. J. Scott, "Linear Läuchli semantics," *Ann. Pure Appl. Logic*, vol. 77, no. 2, pp. 101–142, 1996. [Online]. Available: https://doi.org/10.1016/0168-0072(95)00017-8

[43] T. Ehrhard, "Parallel and serial hypercoherences," *Theoretical Computer Science*, vol. 247, no. 1, pp. 39–81, Sep. 2000.

[44] P. Boudes, "Projecting Games on Hypercoherences," in *Automata, Languages and Programming*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, J. Díaz, J. Karhumäki, A. Lepistö, and D. Sannella, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, vol. 3142, pp. 257–268.

[45] P.-A. Melliès, "Sequential algorithms and strongly stable functions," *Theoretical Computer Science*, vol. 343, no. 1, pp. 237–281, Oct. 2005.
[46] ——, "On dialogue games and coherent strategies," in *Computer Science Logic 2013 (CSL 2013)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), S. R. D. Rocca, Ed., vol. 23. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2013, pp. 540–562.
[47] J. Laird, "Game Semantics for a Polymorphic Programming Language," *Journal of the ACM*, vol. 60, no. 4, pp. 29:1–29:27, Sep. 2013.
[48] G. Jaber and N. Tzevelekos, "A Trace Semantics for System F Parametric Polymorphism," in *Foundations of Software Science and Computation Structures*, C. Baier and U. Dal Lago, Eds. Springer International Publishing, 2018, vol. 10803, pp. 20–38.
[49] S. Abramsky, "Retracing some paths in process algebra," in *CONCUR '96: Concurrency Theory*, ser. Lecture Notes in Computer Science, U. Montanari and V. Sassone, Eds. Springer Berlin Heidelberg, 1996, pp. 1–17.
[50] K. Terui, "Proof nets and boolean circuits," in *19th IEEE Symposium on Logic in Computer Science (LICS 2004), 14-17 July 2004, Turku, Finland, Proceedings*, 2004, pp. 182–191. [Online]. Available: https://doi.org/10.1109/LICS.2004.1319612

APPENDIX A
PROOF OF THE PROGRESS LEMMA (LEMMA II.12)

Let $\rho : (A \vdash B)$ and $\pi : A$.

First, we suppose without loss of generality that all variables in A are quantified – if not, we replace A by $\forall X_1 \dots X_n . A$ and modify π and ρ to obtain proofs π' and ρ' of $\vdash \forall X_1 \dots X_n . A$ and $\vdash (\exists X_1 \dots X_n . A^\perp), B$ such that $\mathbf{cut}(\pi', \rho')$ reduces to $\mathbf{cut}(\pi, \rho)$.

To better understand the cut-elimination between π and ρ, we decorate them with annotations of sorts: modalities $\Box_X$ (with dual $\Box_{X^\perp}$) indexed by type variable, whose $\Box_X$-intro rule in sequent calculus

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, \Box_X A}$$

commutes with all other rules. These $\Box_X$-intros will correspond to unary edges (one source, one target) in proof structures.

Proposition A.1. *MLL_2 sequent calculus extended with these modalities and the obvious cut-elimination rule is strongly normalizing. Correspondingly, proof nets for MLL_2 + modalities, using a $\Box_X / \Box_{X^\perp}$ reduction rule analogous to $\forall/\exists$, strongly normalize.*

Proof. Straightforward extension of the MLL_2 case. □

The following definition and subsequent proposition are meant to be applied to proof nets Π which are reducts of $\mathbf{cut}(\pi, \rho)$.

Definition A.2. The proof structure $\Pi^\Box$ is defined by replacing within π, for all $\exists X$ in A, the witness T_X^Π by $\Box_X(T_X^\Pi)$, and therefore adding an edge $\Box_X$ (resp. $\Box_{X^\perp}$) to the root of every subtree of conclusion T_X^Π (resp. $(T_X^\Pi)^\perp$) c corresponding to an occurrence of X (resp. $X^\perp$) in A.

If $\mathcal{R}$ is the private part of a proof net for $\vdash A$, we define in the same way $\mathcal{R}^\Box$ by adding modalities $\Box_X$ for the $\exists X$. Notice these $\exists X$ do not appear in a formula in $\mathcal{R}$, as in general the corresponding $\exists$ edges disappeared during the public reduction.

Proposition A.3. *If Π is a MLL_2 proof net coming from a proof of A, then $\Pi^\Box$ is also a proof net, coming from the previous proof to which $\Box_X$-intro rules have been added, leaving the conclusion A unaffected.*

Proof. The essential remark is that for a given $\exists X$, the same changes are made to all occurrences of T_X^Π concerned with the $\exists$-intro, and the dual changes are made to all concerned occurrences of $(T_X^\Pi)^\perp$. □

The operation $(-)^\Box$ satisfies some commutations:

Proposition A.4. *If $\Pi \to \Pi'$ then $\Pi^\Box \to^+ \Pi'^\Box$.*

If the public reduction of $\mathbf{cut}(\pi, \rho)$ yields $[\mathcal{R}, \mathcal{Q}]_\iota$, then that of $\mathbf{cut}(\pi, \rho)^\Box = \mathbf{cut}(\pi^\Box, \rho^\Box)$ yields $[\mathcal{R}^\Box, \mathcal{Q}^\Box]_\iota$.

If $[\mathcal{R}, \mathcal{Q}]_\iota \rightrightarrows [\mathcal{R}', \mathcal{Q}']_{\iota'}$ then $[\mathcal{R}^\Box, \mathcal{Q}^\Box]_\iota \rightrightarrows [\mathcal{R}'^\Box, \mathcal{Q}'^\Box]_{\iota'} = [\mathcal{R}', \mathcal{Q}']_{\iota'}^\Box$.

Let us now call a *non-axiomatic cut* a cut edge such that neither of its sources are targets of ax edges. We will show the absence of non-axiomatic cuts during the big-step reduction; this is the key combinatorial property which will be exploited to show the existence of a redex.

Proposition A.5. *A reduct of $\mathbf{cut}(\pi, \rho)^\Box$ of the form $[\mathcal{R}'^\Box, \mathcal{Q}'^\Box]_{\iota'}$ contains no non-axiomatic cuts. This applies in particular to any proof net $\Pi^\Box$ such that $[\mathcal{R}^\Box, \mathcal{Q}^\Box]_\iota \rightrightarrows^* \Pi$, where $[\mathcal{R}^\Box, \mathcal{Q}^\Box]_\iota$ is the result of the public reduction of $\mathbf{cut}(\pi, \rho)$.*

Proof. The two sources of a non-axiomatic cut edge would have to be targets of respectively a $\Box_X$ and a $\Box_Y$, where X is an existentially quantified variable in A and Y an existentially quantified variable in $A^\perp$. Now recall that since $\mathbf{cut}(\pi, \rho)^\Box$ is a proof net, so is $[\mathcal{R}'^\Box, \mathcal{Q}'^\Box]_{\iota'}$, therefore the latter comes from a sequent calculus proof. This proof would thus contain a cut

$$\frac{\vdash \Gamma, \Box_X F \qquad \vdash \Box_Y G, \Delta}{\vdash \Gamma, \Delta}$$

This would force $\Box_X F$ and $\Box_Y G$ to be dual. But since A and $A^\perp$ do not share existential variables, this is never the case. □

Proposition A.6. *With the notations of the previous proposition, $[\mathcal{R}'^\Box, \mathcal{Q}'^\Box]_{\iota'}$ contains an ax edge whose targets are both sources of cut edges.*

Proof. The result is obvious if all cut edges have sources which are targets of ax edges.

Otherwise, consider a non-ax edge whose target is source of a cut edge. At some point in the cut elimination procedure, this edge must disappear. Indeed, if it were not the case, it would introduce a connective appearing in conclusion of $\mathbf{cut}(\rho, \pi)$. However such an edge must necessarily be a $\Box_X$ and such modalities do not appear in the conclusion.

So cut-elimination will make a non-axiomatic cut involving this $\Box_X$ edge appear at some point – by the previous proposition, there is no such cut in $[\mathcal{R}'^\Box, \mathcal{Q}'^\Box]_{\iota'}$, so a new redex (for usual cut-elimination) must be created. Since an ax/cut reduction step creates a new cut-elimination redex only when the other target of the axiom is also the source of a cut edge, we get our conclusion. ☐

We are almost there: indeed a redex for $\rightrightarrows$ in $[\mathcal{R}, \mathcal{Q}]_\iota$ is an axiom link:

- whose two targets i and j are sources of cut edges in $[\mathcal{R}, \mathcal{Q}]_\iota$;
- such that the atoms in A (if i, j are in $\mathcal{R}$) or in $A^\perp$ (if they are in $\mathcal{Q}$) are labeled by dual formulas.

The above proposition gives us an axiom satisfying the first condition. The second one follows from a quick reasoning on the edges facing this axiom on the other side of each cut.

Appendix B
Undecidability proof for observational equivalence (Theorem II.18)

The proof uses the Kanovich–Lafont encoding of two-counter machines [28, Theorem 3]. Given a two-counter machine M, we denote by A_M its encoding and by A_M the universal closure of A'_M.

We prove that the following are equivalent:

i) The two-counter machine M is accepting (recall that this acceptance problem is undecidable).
ii) A_M is provable.
iii) The two proofs of $(A_M \& 1) \multimap 1 \oplus 1$ obtained from $1 \vdash 1 \oplus 1$ are non-equivalent for $\sim_0^-$.
iv) There exists a MALL_0^- context Γ such that $\Gamma \vdash A_M$.

Let us first prove (i) ⇒ (ii) ⇒ (iii) ⇒ (iv):

- (i) ⇒ (ii): Lafont showed how to turn an accepting run of M into a proof of A'_M. From this one deduces a proof of A_M by $\forall$-intro.
- (ii) ⇒ (iii): if $\pi : A_M$, then the test $\rho = \lambda f.\, f\,\langle \pi, 1\rangle$ ($\langle -, - \rangle$: additive pairing) distinguishes those two proofs.
- (iii) ⇒ (iv): by analysis on the possible last rules of a proof of $(A_M \& 1) \multimap 1 \oplus 1 \vdash \Delta$ with Δ a MALL_0^- sequent.

We are now left to prove (iv) ⇒ (i), which is the delicate point. For this, we adapt Lafont's proof of (ii) ⇒ (i) which uses phase semantics, interpreting the formula A_M in a free commutative monoid $\mathcal{M}$ equipped with a pole $\perp$ as a phase space. Here we will take the same monoid, but change the pole into $\perp\!\!\!\perp = \perp \cup \{1\}$.

One verifies easily $\perp\!\!\!\perp^{\perp\!\!\!\perp} = \{1\}$, so that both $\perp\!\!\!\perp$ and $\perp\!\!\!\perp^{\perp\!\!\!\perp}$ are true in this phase model. (A fact is true if it contains 1.) Moreover the set $\{\perp\!\!\!\perp, \perp\!\!\!\perp^{\perp\!\!\!\perp}\}$ is closed under $\otimes, \oplus, (-)^\perp$, so interpreting all atoms of Γ as $\perp\!\!\!\perp$, we get that all formulas of Γ are true. Since $\Gamma \vdash A_M$ is provable, that means A_M is also true in $(\mathcal{M}, \perp\!\!\!\perp)$.

Since A_M is the universal closure of A'_M, any assignment of facts to atoms makes A'_M true in $(\mathcal{M}, \perp\!\!\!\perp)$. Lafont shows that in the phase space $(\mathcal{M}, \perp)$, there is an assignment for which the truth of A'_M entails the acceptance of M. We need to adapt this argument to replace $\perp$ with $\perp\!\!\!\perp$.

A tedious part of Lafont's proof consists in checking that a certain number of formulae of the form $A \multimap B$, where A and B are either atoms, a $\otimes$ of atoms or an $\oplus$ of atoms, are true in $(\mathcal{M}, \perp)$ when these atoms are interpreted as certain generators of the free commutative monoid $\mathcal{M}$. Using $X^{\perp\perp} \otimes Y^{\perp\perp} = (X \cdot Y)^{\perp\perp}$ and similarly for $\oplus$ with $\cup$, this reduces to showing that some set independent of the choice $\perp / \perp\!\!\!\perp$ is included in the biorthogonal of some other set which is also independent of the pole. Since Lafont proved that this inclusion holds with $\perp$, it also holds for $\perp\!\!\!\perp$, using the fact that $S^{\perp\!\!\!\perp} = S^\perp$ for any set $S \subset \mathcal{M}$ such that $1 \notin S$. (This is because, since $\mathcal{M}$ is free, for $x \neq 1$, $xy \neq 1$ and so $xy \in \perp\!\!\!\perp \Leftrightarrow xy \in \perp$.) So these formulas $A \multimap B$ are also true in $(\mathcal{M}, \perp\!\!\!\perp)$ with the same assignment of generators to atoms.

We also have to check that $\forall X. (X \& 1) \multimap X \otimes X$ is true in $(\mathcal{M}, \perp\!\!\!\perp)$; this is the case since $\perp\!\!\!\perp^{\perp\!\!\!\perp} = \{1\}$ contains only idempotent elements (cf. the lemma in [28]). After that, the conclusion follows for exactly the same reason as in [28].

Appendix C
Omitted proofs in section III

A. *Proof of Lemma III.5*

A proof of $\vdash\ ?\mathtt{Str}^\perp, \ldots, ?\mathtt{Str}^\perp, !!\mathtt{Bool}$ necessarily ends either with a structural rule or a promotion. From this and the invertibility of $⅋$, one obtains that the proof, up to commutation, ends with the following sequence of rules:

$$\dfrac{\dfrac{\dfrac{\vdash \mathtt{Str}^\perp, \ldots, \mathtt{Str}^\perp, !\mathtt{Bool}}{\vdash ?\mathtt{Str}^\perp, \ldots, ?\mathtt{Str}^\perp, !!\mathtt{Bool}}\ !}{\vdash ?\mathtt{Str}^\perp, !!\mathtt{Bool}}}{\vdash !\mathtt{Str} \multimap !!\mathtt{Bool}}\ ⅋$$

Now, recall that $\mathtt{Str}^\perp = \exists X. \mathtt{Str}[X]^\perp$. Moreover, notice the introduction rule for $\exists$ commutes with all other rules except promotion and the introduction of $\forall$. We only need to show that the introduction rule for all $\exists$ connectives of the occurrences of $\mathtt{Str}^\perp$ are not followed by a promotion or a $\forall$ introduction. Ruling out promotion is easy, as all formulas in the conclusion sequent of a promotion rule have exponential as principal connectives. Moreover, it is possible to rule out $\forall$ introductions as folllws. If a $\forall$ introduction rule appears in the proof, the $\forall$ connective it introduces is part of an existential witness. But all variables existentially quantified appear under the scope of an exponential connective, and therefore can only precede a promotion rule.

B. *Proof of Lemma III.6*

We define the *collapse at depth 2* of a formula as follows: all subformulas of the form $!A$ (resp. $?A$) at depth 2, i.e. in the scope of two other nested $!/?$ modalities, are replaced by 1 (resp. $\perp$). Note that the collapse at depth 2 of $!\mathtt{Str}$ and $!!\mathtt{Bool}$ are themselves.

This operation extends to proofs: any functorial promotion of conclusion $\vdash\ ?B_1, \ldots, ?B_m, !C$ is replaced by the only proof of $\vdash \perp, \ldots, \perp, 1$, while contractions and weakenings are replaced by cuts with $1 \vdash 1 \otimes 1$ and $\vdash 1$. Note that this collapse is the identity on cut-free proofs of $!!\mathtt{Bool}$ and $!\mathtt{Str}$.

One may then check that collapse at depth 2 is compatible with cut-elimination, which means that one can replace π by its collapse at depth 2 and still recognize the same language. Then the A_i are replaced by their "collapse at depth 0" which are MALL_2 formulas.

Appendix D
Omitted proofs in section IV

A. Proof of Proposition IV.9

Note that $\mathrm{Card}(|F(X)|) = O(\mathrm{Card}(|X|)^d)$ implies that F sends finite spaces to finite spaces. Thus, it suffices to prove that for stable functors preserving finiteness,

$$\deg(F) = \inf\{d \in \mathbb{N} \mid \mathrm{Card}(|F(X)|) = O(\mathrm{Card}(|X|)^d)\}$$

We decompose this equality into two inequalities.

($\leq$) Let $\langle X^0 \vdash x \rangle \in \mathrm{NF}(F)$ and $d = \mathrm{Card}(|X^0|)$. Let $[n] = 1 \& \ldots \& 1$ (n times). For all n, there are n^d embeddings $X^0 \hookrightarrow X^0 \otimes [n]$ which are the identity on the first component. If for two such embeddings ι and ι', $F(\iota)(x) = F(\iota')(x)$, then by uniqueness of the normal form ι and ι' are isomorphic in the slice category $\mathsf{CohI}/(X^0 \otimes [n])$; in the commuting triangle $\iota = \rho \circ \iota'$, ρ can only be the identity, since our considered family of embeddings differ only on the component $[n]$, and so $\iota = \iota'$. Thus, $\iota \mapsto F(\iota)(x)$ is injective over the n^d embeddings we consider, and therefore $\mathrm{Card}(|F(X^0 \otimes [n])|) \geq n^d$ while $\mathrm{Card}(|X^0 \otimes [n]|) = dn$.

($\geq$) We assume $d = \deg(F) < \infty$ (when $\deg(F) = \infty$, the inequality is true for trivial reasons). This entails that $\mathrm{NF}(F)$ is finite: there are finitely many coherence spaces of cardinality $\leq d$, and their images by F are all finite since F preserves finiteness. Now let X be a finite coherence space of cardinality n; each point $x \in |F(X)|$ has a normal form, so $\mathrm{Card}(|F(X)|)$ can be bounded by summing over $\langle X^0 \vdash x \rangle \in \mathrm{NF}(F)$ the number of possible embeddings of X^0 in X. This number is at most $O(n^{\mathrm{Card}(X^0)})$ and by definition $\mathrm{Card}(X^0) \leq d$. In the end, using $\mathrm{Card}(\mathrm{NF}(F)) = O(1)$, we get $\mathrm{Card}(|F(X)|) = O(n^d)$.

B. Proof of Proposition IV.19

We first need to recall how instantiation is defined in the coherence space model. Let X be a coherence space, then

$$\theta_X = \{F(\iota)(y) \mid \langle Y \vdash y \rangle \in \theta, \iota : Y \to X \text{ embedding}\}$$

For each point $x \in \theta_X$, there is a unique (ι, y) such that $F(\iota)(y) = x$; indeed, it is the initial object of the slice category $\mathsf{El}(|F|)/(X, x)$. So we have a bijection

$$\theta_X \cong \{(\iota, y) \mid \langle Y \vdash y \rangle \in \theta, \iota : Y \to X \text{ embedding}\}$$

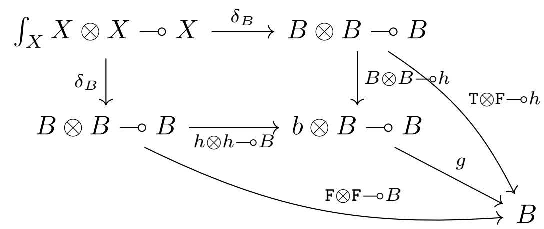
This bijection is computable in logarithmic space: it is just a matter of performing substitutions on terms of size $O(1)$ (since F is fixed), although the ι's are not of constant size (because of the representation of elements of $|X|$). Therefore the problem is reduced to enumerating the right-hand side without repetitions.

There are finitely many $\langle Y \vdash y \rangle$ in θ. For each of them, each injection $Y \to X$ can be represented by $\mathrm{card}(|Y|) \leq \deg(F) = O(1)$ elements of $|X|$: this takes $O(\log \mathrm{card}(|X|))$ space. Thus, all the injections can be enumerated in logarithmic space, and for each injection, whether it is an embedding can be determined in logarithmic space (using the coherence relation on $\mathrm{Tr}(F)$ which may be precomputed independently of the input.)

Appendix E
Proof of Proposition V.6

Consider $\mathbb{B} = (X \otimes X) \multimap (X \otimes X)$ and its two proofs $\mathtt{T}, \mathtt{F}$. It is known [50] that for all function $f : \{0,1\}^n \to \{0,1\}$ there exists a MLL_0 formula A such that f is represented by a MLL_0 proof of $\mathbb{B}[A/X]^n \multimap \mathbb{B}[A/X]$. Let then A be such that there exists a MLL_0 proof $\pi_0 : \mathbb{B}[A/X] \otimes \mathbb{B}[A/X] \multimap \mathbb{B}[A/X]$ representing the constant 0 function. We now consider the two distinct MLL_2 proofs π_1, π_2 of $\forall X (X \otimes X \multimap X) \multimap \mathbb{B}[A/X]$, corresponding to the λ-terms $\lambda x.x\mathtt{FF}$ and $\pi_2 = \lambda x.h(x\mathtt{TF})$.

Suppose $\mathrm{Tr}_X(X \otimes X \multimap X)$ is a wedge. We show then that $[\![\pi_1]\!]_{\mathsf{CohI}} = [\![\pi_2]\!]_{\mathsf{CohI}}$. Let $\pi_3 = \lambda x.x\mathtt{TF}$ of conclusion $(\mathbb{B}[A/X] \otimes \mathbb{B}[A/X] \multimap \mathbb{B}[A/X]) \multimap \mathbb{B}[A/X]$. Let B be the interpretation of $\mathbb{B}[A/X]$, $h := [\![\pi_0]\!]_{\mathsf{CohI}}$, $g := [\![\pi_3]\!]_{\mathsf{CohI}}$ and $F = X \otimes X \multimap X$. Then we have $[\![\pi_1]\!]_{\mathsf{CohI}} = [\![\pi_3]\!]_{\mathsf{CohI}} \circ FhB$ and $[\![\pi_2]\!]_{\mathsf{CohI}} = [\![\pi_3]\!]_{\mathsf{CohI}} \circ FBh$, so we can conclude $[\![\pi_1]\!]_{\mathsf{M}} = [\![\pi_2]\!]_{\mathsf{CohI}}$ by the wedge condition below:



since $[\![\pi_1]\!]_{\mathsf{CohI}} = (\mathtt{F} \otimes \mathtt{F} \otimes b) \circ \delta_b$ and $[\![\pi_2]\!]_{\mathsf{CohI}} = (\mathtt{F} \otimes \mathtt{T} \otimes h) \circ \delta_b$.

However, $[\![\pi_1]\!]_{\mathsf{CohI}} \neq [\![\pi_2]\!]_{\mathsf{CohI}}$: since $\mathrm{Tr}_X(X \otimes X \multimap X)$ contains the intersection function i, the interpretation of π_1 is non-empty while the interpretation of π_2 is empty.