Interaction Graphs: Full Linear Logic

Thomas Seiller

Interaction Graphs: Full Linear Logic*

Thomas Seiller[1]

1 **Proofs, Programs, Systems – UMR 7126 CNRS – Paris 7 University**

Abstract

Interaction graphs were introduced as a general, uniform, construction of dynamic models of linear logic, encompassing all *Geometry of Interaction* (GoI) constructions introduced so far. This series of work was inspired from Girard's hyperfinite GoI, and develops a quantitative approach that should be understood as a dynamic version of weighted relational models. Until now, the interaction graphs framework has been shown to deal with exponentials for the constrained system ELL (Elementary Linear Logic) while keeping its quantitative aspect. Adapting older constructions by Girard, one can clearly define "full" exponentials, but at the cost of these quantitative features. We show here that allowing interpretations of proofs to use continuous (yet finite in a measure-theoretic sense) sets of states, as opposed to earlier Interaction Graphs constructions were these sets of states were discrete (and finite), provides a model for full linear logic with second order quantification.

Keywords and phrases Interaction Graphs; Linear Logic; Geometry of Interaction; Quantitative Semantics; Measurable Dynamics

1 Introduction

This work deals with so-called dynamical models of proof theory, such as game semantics and geometry of interaction, as well as with quantitative models of computation. It extends previous work providing a uniform construction of quantitative dynamical models of (fragments of) linear logic to full linear logic with second-order quantification.

Geometry of Interaction. A Geometry of Interaction (GoI) construction, i.e. a construction that fulfills the GoI research program [18], is in a first approximation a representation of linear logic proofs that accounts for the dynamics of cut-elimination. Contrarily to denotational semantics, a proof π and its normalized form π' are not represented by the same object, but they remain related through a semantic interpretation of the cut-elimination called the *execution* Ex: $\mathrm{Ex}(\pi) = \pi'$. A GoI construction hence represents both the proofs and their normalization; it is in some ways an untyped variant of game semantics [22, 1].

The further aim of geometry of interaction is to reconstruct logical operations from such a dynamical representation of proofs. The objects of study in a GoI construction are actually a generalization of the notion of proof – sometimes called paraproofs. This point of view allows a reconstruction of logic as a description of how paraproofs interact, in the same spirit as classical realizability [23, 24]: a program is of type $\mathtt{nat} \to \mathtt{nat}$ because it produces a natural number when given a natural number as an argument. As in game semantics and classical reliazability, one can however describe a necessary condition for being the interpretation of a proof, and defines *winning paraproofs* as those objects satisfying it.

In spite of their seemingly deep abstraction, the GoI constructions offer a mathematical model which is very close to actual computing. As an illustration of this fact, let us mention the *Geometry of Synthesis* program initiated by Ghica [11, 12, 13, 14]. This research

* This work was partially supported by the ANR 12 JS02 006 01 project COQUAS and the ANR-10-BLAN-0213 Logoi.

24th EACSL Annual Conference on Computer Science Logic.

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

program, inspired by geometry of interaction, aims at obtaining logical synthesis methods for VLSI (Very Large Systems Integration) designs.

Quantitative Semantics. Quantitative semantics find its origins in Girard's work on functors's models for lambda-calculus [16]. This work, which predates its seminal work on linear logic [15] and actually inspired it, exhibits for the first time a decomposition of the semantic interpretation of lambda-terms as Taylor series. These series capture a number of informations about the time, space, resource consumption of the programs it represents. Quantitative semantics are therefore more involved than so-called *qualitative* semantics, since they mirror more informations about the programs that are interpreted. Recently, quantitative semantics has been used to give denotational semantics for various algebraic extensions of lambda calculus such as probabilistic [7] or differential lambda calculi [9]. Work by Laird, Manzonetto, McCusker and Pagani [25] provides a uniform account of several denotational models accouting for quantitative notions, using a refinement of the relational model.

Interaction Graphs. Interaction graphs were first introduced by the author [29] as a combinatorial approach to Girard's hyperfinite Geometry of Interaction [21], restricted to the multiplicative fragment of linear logic. An extension capable to deal with additives connectives was then defined [30] and shown to abstract not only the (additive fragment of the) hyperfinite GoI model but all previously introduced GoI constructions as well. Both papers proposed a model construction in the spirit of Girard's GoI construction where proofs were interpreted by graphs instead of infinite operators. Dealing with exponentials however needs one to consider infinite objects. This is why a third paper [32] showed how the construction on graphs can be applied when working with a generalisation of graphs named *graphings*. Graphings are in some sense *geometric realisations* of graphs on a measured space $\mathbf{X}$. This allows not only to consider infinite graphs (which can be used to define exponentials in the same way as the original GoI constructions), but also graphs acting on continuous, thus infinite, finite-measure spaces. This general construction on graphings was shown [32] to improve on Girard's hyperfinite GoI [21] since it allows a satisfactory treatment of second-order quantification. Lastly, a fourth paper [31] showed how the consideration of graphings can be used to define "quantitative" exponential connectives for Elementary Linear Logic [20], a fragment of linear logic that captures elementary time computation [8].

Unbounded Exponentials and Quantitative Aspects. The author's work on Interaction Graphs should be understood as a *dynamic counterpart* of weighed relational models, i.e. its relation to standard dynamical models (geometry of interaction, game semantics) is comparable to weighted relational models' relations with standar relational models. Indeed, it provides a uniform construction of models which not only captures all of Girard's GoI models, but also extends them: while Girard's constructions can be understood as interpreting proofs as graphs[1], we here interpret proofs as *weighted graphs*, i.e. graphs with weighted edges[2]. Furthermore, interaction graphs models can reflect these quantitative informations at the level of types since the latter are built from an *orthogonality relation* which can take into account those weights. Indeed, the orthogonality relation is defined through a measurement of cycles [32] by means of an integral over a finite-measured space – the *support*

[1] Girard interprets proofs as partial isometries acting on a Hilbert space $\mathbb{H}$ which, by considering the right basis for $\mathbb{H}$ correspond to graphs.

[2] Actually, the most general models are build around the lesser known notion of weighted *graphing*. However, thinking about graphings as graphs should provide the reader with good intuitions.

of the cycle. In the simplest cases one measures a cycle π of suppport $\mathrm{supp}(\pi)$ and weight $\omega(\pi) \in \Omega$, along a measurable map $m : \Omega \to \bar{\mathbf{R}}_+$, by the following integral:

$$\int_{\mathrm{supp}(\pi)} m(\omega(\pi)) \tag{1}$$

Since Interaction Graphs provide a generalization of Girard's constructions, one could easily adapt the interpretation of exponential connectives from Girard's first constructions [17, 19] to obtain a model of full linear logic. This adaptation would extend to Danos' interpretation of pure lambda-calculus in GoI [6]. However, this interpretation of exponential connectives corresponds to defining $!a$ as a (countable) infinite family of copies of a. Thus, even if a is represented by a graphing acting on a space of finite measure, its exponentiated version $!a$ acts on a space of infinite measure. This fact hinders the quantitative aspects of our model since it creates cycles π whose support $\mathrm{supp}(\pi)$ are spaces of infinite measure. As a consequence, the integral defining the orthogonality relation (Equation 1) diverges as soon as the weight is not mapped to 0, i.e. as soon as $m(\omega(\pi)) \neq 0$. The resulting model is therefore no longer capable of depicting quantitative informations.

Contributions and Related Work. We define, in the framework of interaction graphs, exponential connectives for full linear logic in a way that preserves the quantitative aspects of the construction. We define exponential connectives along the same lines as in our work on (bounded) exponentials [31], avoiding the involvement of infinite-measure sets. With this definition of exponential connectives, one would however expect only a restriction of linear logic, such as ELL. To bypass this restriction, we relax the notion of states. Indeed, the interpretation of proofs in interaction graphs makes use of so-called *thick graphs* – or *thick graphings* in the general framework –, which can be understood as graphs with states. While previous work considered only finite sets of states, we loosen this definition to allow for infinite yet finite-measure (actually continuous) sets of states. This modification impacts slightly on the basic notions notions and constructions considered in previous work [32], for which we consider adequate generalisations. These changes, however, do not raise any technical difficulties. The resulting model is then shown to model digging and dereliction in addition to the principles of Elementary Linear Logic, thus interpreting full linear logic.

This paper is the second GoI-style construction for full linear logic (including additives) and second order quantification. Indeed, Girard's so-called GoI3 construction [19] already provided such a model. However, as explained above, Girard's treatment of exponential connectives prevents from any generalization accounting for quantitative informations. The main contribution of this paper therefore lies in its quantitative aspects. Moreover, we are able to pinpoint the computational principles (represented as measurable maps) that are essential to interpret digging and dereliction, i.e. we exhibit a single map – the *exchange* `xch` – which turns a model of ELL into a model of LL. Lastly, we provide a full soundness result and discuss the issue of the representation of cut-elimination in the model.

Apart from geometry of interaction constructions, related work include quantitative realizability [4, 3] – which provides characterizations of computational complexity classes – and quantitative game semantics for linear logic [5], although the latter does not deal with additives and quantifiers and seems more limited in its possible quantitative features. In particular, we believe our construction to be a generalization of quantitative realizability, allowing for characterisations of a larger family computational complexity classes [34].

2 Interaction Graphs

We start by a discussion meant to give intuitions about the basic principles at work in the interaction graphs models. We illustrate those principles by explaining the notion of *thick and sliced graphs* [31]. This discussion is quite informal in that we do not provide explicit and complete definitions of the objects and operations considered, to avoid overloading the reader with non-essential definitions. Indeed, the actual model uses thick and sliced *graphings*, a generalisation needed to accomodate both exponentials and quantifiers. Before providing the formal definition of those at the end of the section, we discuss the generalisation to continuous sets of states.

2.1 Thick and Sliced Graphs

The term "graph" will stand for "directed weighted graphs", i.e. directed graphs with a weight function from the set of edges to a monoid[3] of weights Ω. Given a graph (or later, a graphing) G, we will always denote E^G its set of edges, S^G and D^G its support and dialect, t^G and s^G its target and source maps, and ω^G its weight map.

The notion of *thick* graphs corresponds to considering graphs with a set of states – called a dialect. A graph G with dialect D^G is nothing more than a graph whose set of vertices is of the form $V^G = S^G \times D^G$ – the set S^G is called its *support*. The set D^G then acts as a set of states when considering two graphs *in interaction* through the notion of execution. In interaction graphs the execution of programs – or equivalently the cut-elimination procedure in logic – is represented as the computation of a graph of (alternating) paths. If G and H are two graphs with dialects D^G and D^H respectively, an alternating path between G and H is a finite sequence of edges $e_0 e_1 \dots e_k$ and a sequence of triples $(s_i, g_i, h_i)_{i=0}^{k+1}$ such that:

(Alternation) $e_i \in E^G$ if and only if $e_{i+1} \in E^H$;

(States) if $e_i \in E^G$ then $s^G(e_i) = (s_i, g_i)$, $t^G(e_i) = (s_{i+1}, g_{i+1})$ and $h_i = h_{i+1}$;

For two thick graphs G, H, the shared vertices represent a *cut*; the result of the elimination of this cut is called the *execution* of G and H. It is defined as the thick graph $G :: H$, of dialect $D^G \times D^H$, whose edges are exactly the alternating paths between G and H whose source and target lie outside of the cut. This is reminiscent of game semantics' *composition and hiding*: composition corresponds here to the computation of all alternating paths, while hiding corresponds to the restriction to those paths starting and ending outside the cut.

Now, *thick and sliced graphs* are simply finite formal weighted sums of graphs $\sum_{i \in I^G} \alpha_i^G G_i$ where the coefficients α_i^G are real numbers and the graphs G_i all share the same sets of vertices. This notion is crucial for treating additive connectives [30]. The notion of execution is then extended "by linearity" (although the sums are not linear combinations), letting:

$$\left(\sum_{i \in I^G} \alpha_i^G G_i\right) :: \left(\sum_{i \in I^H} \alpha_i^H H_i\right) = \sum_{(i,j) \in I^G \times I^H} \alpha_i^G \alpha_j^H G_i :: H_j$$

2.2 Continuous Dialects

Graphings are in some sense *geometric realisations* of graphs on a measured space $\mathbf{X}$. Specifically, a graphing G is defined as a graph such that for each edge $e \in E^G$, $s^G(e)$ and $t^G(e)$ are measurable subsets of $\mathbf{X}$, and there is a measurable map $\phi_e^G : s^G(e) \to t^G(e)$ which *realises*

[3] As we consider paths in the following, the structure of monoid is essential as it allows to define the weight of a path as the product of the weights of the edges that it is composed of.

e. As for graphs, one can define *thick and sliced graphings* by first defining graphings with a dialect – thick graphings, then consider formal weighted sums of those. It is natural, while working with graphings, to consider (finite) discrete probability spaces as dialects; a thick graphing of dialect $\mathbf{D}$ is then simply a graphing over the measured space $\mathbf{X} \times \mathbf{D}$.

The purpose of the work is to extend this definition to allow for *continuous dialects*, i.e. continuous dialects, instead of discrete ones. We will show how to define in this setting the interpretation of second order linear logic without hindering the "quantitative" features of the interaction graphs construction. This however comes with a small drawback in the form of a small complexification of the framework, which we now explain.

We did not dwell on this point earlier, but thick graphs (graphs with dialects) are considered *up to* renaming of their dialect; a thick graph G which is a dialect-renaming of a thick graph F is called a *variant* of F. To define correctly this notion of variant one needs to consider bijections between the dialects. However, when considering graphings and replacing the dialects with possibly continuous probability spaces, we face a problem when considering the two probability spaces $\mathbf{k} = \{1, \dots, k\}$ with discrete measure and $[\mathbf{0}, \mathbf{1}]$ with Lebesgue measure. Indeed, any thick graphing G with dialect $\mathbf{k}$ has a variant H with dialect $[\mathbf{0}, \mathbf{1}]$: each element $i \in \mathbf{k}$ is represented by the interval $I_i = [i/(k+1), (i+1)/(k+1)]$, and an edge of source (v, i) and target (v', j) realised by a map $\phi : (v, i) \to (v', j)$ in G is realised in H by $\phi_1 \times T_{i,j}$ where $T_{i,j}$ is the translation $x \mapsto x + (j - i)/(k + 1)$ and ϕ_1 is the map $v \to v'$ underlying[4] ϕ. This cannot be formalised through a notion of bijection (here it would amount to consider *Borel automorphisms*) between $\mathbf{k}$ and $[\mathbf{0}, \mathbf{1}]$. To avoid these troubles, we will therefore consider all our dialects to be isomorphic to $[\mathbf{0}, \mathbf{1}]$ with its Lebesgue measure; as explained above, we do not lose any graphings in the process since a graphing with discrete dialect always has a variant with $[\mathbf{0}, \mathbf{1}]$ as dialect.

The second change from earlier work [32] is that we need to consider an extension of the notion of *microcosm*. A microcosm $\mathfrak{m}$ is a monoid of measurable maps $\mathbf{X} \to \mathbf{X}$ used to consider "restrictions" of the model to $\mathfrak{m}$-graphings: graphings whose realizers can be extended to an element of $\mathfrak{m}$. The original notion of microcosm did not need to incorporate the dialect since the latter was discrete, and therefore the measurable maps realizing an edge in a thick graphing were nothing more than measurable maps from $\mathbf{X}$ to $\mathbf{X}$. Now that we allow for continuous dialects, one can consider realisers of edges that do not simply arise from[5] a map $\mathbf{X} \to \mathbf{X}$. The following definition therefore adapts (in fact extends) the previously considered notion of microcosm to incorporate this change.

▶ **Definition 1** (Microcosm)**.** Let $\mathbf{X}$ be a measured space. A *microcosm* is a monoid (for the composition of functions) of measurable maps[6] $\mathbf{X} \times [\mathbf{0}, \mathbf{1}] \to \mathbf{X} \times [\mathbf{0}, \mathbf{1}]$.

2.3 Graphings and Exponential-Free Linear Logic

This section is meant to recall the main results of previous work [32], to which we refer the reader for a complete picture. We first define weighted (thick) graphings, a generalization of the homonymous notion considered by Adams [2] and later by Gaboriau [10].

▶ **Definition 2** (Graphing)**.** Let $\mathfrak{m}$ be a microcosm, Ω a monoid of weights, V^G a measurable

[4] Since $\mathbf{k}$ is discrete, any measurable map $\phi : (v, i) \to (v, j)$ is defined from a measurable map $\phi_1 : v \to v'$ by $\phi(x, i) = (\phi_1(x), j)$.

[5] As an example, one can consider the *exchange* map defined below (Theorem 10) and which is needed to interpret both digging and dereliction.

[6] For technical reasons discussed in earlier work [32], these should be Borel-preserving and non-singular.

subset of $\mathbf{X}$ and $\mathbf{D}^G$ a probability space isomorphic to $[\mathbf{0},\mathbf{1}]$. A thick Ω-weighted $\mathfrak{m}$-graphing G of carrier V^G and dialect $\mathbf{D}^G$ is given by:

- a set of edges E^G;
- for each edge $e \in E^G$, source $s^G(e)$ and target $t^G(e)$ measurable subsets of $V^G \times D^G$;
- for each edge $e \in E^G$, a *realiser* $\phi_e^G \in \mathfrak{m}$ such that $\phi_e^G(s^G(e)) = t^G(e)$;
- for each edge $e \in E^G$, a *weight* $\omega^G(e)$.

A graphing G is *dialect-free* if it does not make use of its dialect, i.e. if for all edge e, $\phi_e^G = \tilde{\phi}_e^G \times \mathrm{Id}_{\mathbf{D}^G}$, with $\tilde{\phi}_e^G : \mathbf{X} \to \mathbf{X}$.

▶ Notations. Let B be a Borel automorphism of $\mathbf{X} \times [\mathbf{0},\mathbf{1}]$. We denote $B(A)$ the graphing whose edges are $B^{-1} \circ \phi \circ B$; up to the automorphism between $\mathbf{D}^A$ and $[\mathbf{0},\mathbf{1}]$. When B is a Borel automorphism of $\mathbf{X}$, we abusively denote by $B(A)$ the graphing $B \times \mathrm{Id}_{[\mathbf{0},\mathbf{1}]}(A)$. We also denote by $A \times \mathrm{Id}_{[\mathbf{0},\mathbf{1}]}$ the graphing of dialect $\mathbf{D}^A \times [\mathbf{0},\mathbf{1}]$ whose edges are realised as $\phi_e \times \mathrm{Id}_{[\mathbf{0},\mathbf{1}]}$.

▶ **Definition 3** (Variants). Let F and G be graphings. If there exists a Borel automorphism $\phi : [\mathbf{0},\mathbf{1}] \to [\mathbf{0},\mathbf{1}]$ such that $F = \mathrm{Id}_{\mathbf{X}} \times \phi(G)$, we say that F and G are variants.

Morally, graphings are sort of graphs which offer richer combinatorics since two vertices might have a non-trivial intersection without being equal. In particular, when considering paths, one should be careful about the domains: a path in a graphing G is a sequence of edges $\pi = e_1, e_2, \dots, e_k$ in E^G such that $s^G(e_{i+1}) \cap t^G(e_i)$ is of strictly positive measure[7]. This path is then naturally realised as the composite $\phi_\pi^G = \phi_{e_k}^G \circ \cdots \circ \phi_{e_1}^G$, and is considered with its *maximal domain* $s^G(\pi)$, i.e. the set of all x such that $\phi_{e_i}^G \circ \cdots \circ \phi_{e_1}^G(x) \in s^G(e_{i+1})$, and its codomain $t^G(\pi) = \phi_\pi^G(s^G(\pi))$. The weight of π is obviously defined as $\omega^G(\pi) = \omega^G(e_k)\omega^G(e_{k-1}) \dots \omega^G(e_1)$ using the composition law of Ω.

We can then define *alternating path* between thick graphings graphings as in the case of graphs. From this, one defines the *execution* between thick graphings, the semantic counterpart to the cut-elimination procedure. We here define execution only in the specific case when the support of one graphing is included in the support of the other: this represents an *application*, i.e. a modus ponens; the general case of the cut can be consulted in our earlier paper [32] or deduced from this specific case[8].

▶ **Definition 4** (Execution). Let F and G be graphings with $V^G = V^F \uplus V$. Their *execution* $F :: G$ is the graphing of all alternating paths between F and G whose domain and codomain are restricted to V.

▶ **Example 5.** We consider two one-edge graphings (without dialects or weights to be concise) G and H illustrated on the left-hand side of Figure 1. The edge of G has source the segment $[0,2]$, target the segment $[4,6]$ and is realised by the map $x \mapsto 6-x$. The edge of H has source the segment $[5,6]$, target the segment $[8,9]$ and is realised by the map $x \mapsto x+3$. The *cut* is represented by the segment $[5,6]$. The execution of G and H, illustrated on the right-hand side of Figure 1, is composed of two paths: the restriction of the edge of G to the segment $[1,2]$, and the composition of the two edges.

---

[7] This is a crude approximation, since a path of length 3 might satisfy these conditions but such that $s^G(e_3) \cap \phi_{e_2} \circ \phi_{e_1}(s^G(e_1))$ is of null measure. The right conditions are defined in our earlier work introducing the general framework of interaction graphs and graphings [32].

[8] Noticing that $(A \multimap B) \otimes (B \multimap C) \multimap (B \multimap B) \multimap (A \multimap C)$ (using the semi-distributivity law twice), one can rewrite a cut between $f \in A \multimap B$ and $g \in B \multimap C$ as an application of $f \otimes g \in (B \multimap B) \multimap (A \multimap C)$ with the axiom $a \in B \multimap B$.

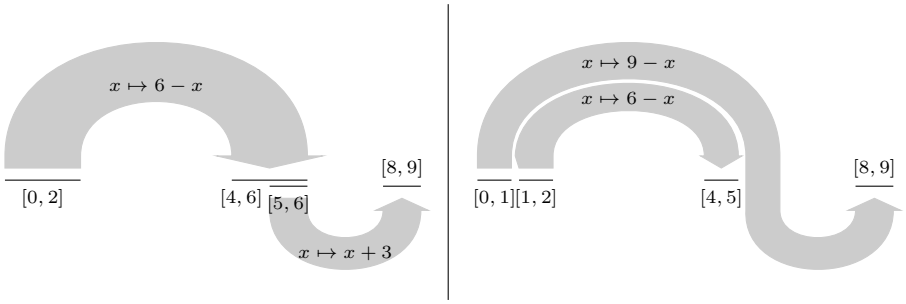


Figure 1 Example of an execution between two graphings.

Based on the notion of alternating cycle – defined easily from the notion of alternating paths, one defines a measurement $[\![\cdot, \cdot]\!]_m$ of couples of graphings and taking values in $\bar{\mathbf{R}}_+$. This measurement is parametrized[9] by the choice of a measurable map $m : \Omega \to \bar{\mathbf{R}}_+$. It is a quite involved work to define and study, so we refer the interested reader to our previous paper [32]. In the specific case of graphs – which are graphings over a discrete space – this measurement is simply equals the sum, over the set of alternating cycles π, of $m(\omega(\pi))$ where m is any map $\Omega \to \bar{\mathbf{R}}_+$. This notion of measurement is extended to couples (a, A) where a is a real number (potentially infinite) and A a graphing; the consideration of this additional real number – the wager – finds its reasons in technical details that are explained in previous papers [29, 30]. The resulting couples, called projects, are used to interpret proofs.

▶ **Definition 6** (Project). A project is a couple $\mathfrak{a} = (a, A)$ where $a \in \bar{\mathbf{R}}_+$ and A is a formal weighted sum of graphings $A = \sum_{i \in I^A} \alpha_i^A A_i$.

From the measurement, one defines a notion of orthogonality that accounts for linear negation. This orthogonality relation is used to define *conducts*, specific sets of projects which will interpret formulas.

▶ **Definition 7** (Orthogonality – Conduct). Two projects $\mathfrak{a} = (a, A)$ and $\mathfrak{b} = (b, B)$ of equal supports are orthogonal, denoted $\mathfrak{a} \perp \mathfrak{b}$, when $\ll \mathfrak{a}, \mathfrak{b} \gg_m \neq 0, \infty$. A *conduct* of support V is the orthogonal of a set T of projects of support v, i.e. $V = T^{\perp} = \{\mathfrak{a} \mid \forall \mathfrak{b} \in T, \mathfrak{a} \perp \mathfrak{b}\}$.

Finally, one can define a category whose objects are conducts and morphisms are projects and which is shown to interpret multiplicative-additive linear logic. We do not detail this construction since it is quite involved. However, let us point out that the resulting model is completely non-degenerate (none of the connectives or constants are identified) and do not satisfy the mix and weakening rules [30].

▶ **Theorem 8** (Seiller [32]). *Let $\mathbf{X}$ be a measured space, $\mathfrak{m}$ a microcosm, Ω a monoid of weights. For all measurable map $m : \Omega \to \bar{\mathbf{R}}_+$, conducts and projects built from Ω-weighted $\mathfrak{m}$-graphings, with the orthogonality defined from the measurement defined from m, form a model of Multiplicative-Additive Linear Logic.*

[9] We won't dwell on this choice of parameter in this paper, in order to avoid unnecessary complications. Although we here mention it for the sake of exactness, it will not play any specific role here. A fine analysis of the models would imply a consideration of specific values of m, but none of the results obtained in this paper depend on the choice of m.

3 The model

To describe the model, we will pick a measured space $\mathbf{X}$ together with a microcosm $\mathfrak{U}_\rho$ which are defined below. The construction we describe will not depend on the choice of Ω and $m : \Omega \to \mathbf{R}_{\geqslant 0}$, and therefore describes a family of quantitative models of second order linear logic.

Although the underlying space used here differs from our earlier work on exponentials [31], both are equivalent up to a Borel automorphism. The presentation was chose to work with here has the advantage of showing more explicitely the dynamics at work, while gaining intuitions from standard work on exponentials. Indeed, we chose to work explicitely with the Hilbert cube $[\mathbf{0},\mathbf{1}]^{\mathbf{N}}$, underlying an intuitive correspondence between *boxes* used to treat exponentials in proof nets and the copies of $[\mathbf{0},\mathbf{1}]$.

▶ **Definition 9** (The space)**.** We define the measured space $\mathbf{X} = \mathbf{R} \times [\mathbf{0},\mathbf{1}]^{\mathbf{N}}$, product of the real line with the Hilbert cube, endowed with its usual Borel algebra and Lebesgue measure.

▶ Notations. We will write elements of $\mathbf{X}$ as couples (a, s), where $a \in \mathbf{R}$ and s is a sequence of elements in $[\mathbf{0},\mathbf{1}]$. We will sometimes write sequences as $s \bullet s'$, i.e. as the concatenation of a finite sequence $s = (x_1, \dots, x_k)$ and a sequence s'; when s contains only one element x we will identify x and (x). When considering elements of the space $\mathbf{X} \times [\mathbf{0},\mathbf{1}]$, we will use a natural extension of this notation, and write them (a, s, e), with $(a, s) \in \mathbf{X}$ and $e \in [\mathbf{0},\mathbf{1}]$.

We now define the microcosm, denoted $\mathfrak{U}_\rho$, that will be used to interpret proofs. We could very well have worked with the biggest microcosm possible (the so-called *macrocosm*) or any microcosm containing $\mathfrak{U}_\rho$. It is however more interesting to point out exactly the principles that are necessary to interpret second-order linear logic.

▶ **Definition 10** (The microcosm)**.** Let ρ be a measure-preserving bijection $[\mathbf{0},\mathbf{1}]^2 \to [\mathbf{0},\mathbf{1}]$. We define the microcosm $\mathfrak{U}_\rho$ as the monoid of measurable[10] maps $\mathbf{X} \to \mathbf{X}$ generated by:

- affine transformations over the real line: $A^\alpha_\lambda : (x, s) \mapsto (\alpha x + \lambda, s)$;
- (finitely supported) permutations over the Hilbert cube: $P_\sigma : (x, s) \mapsto (x, \sigma(s))$;
- the maps $D_\rho : (a, (x, y) \bullet s) \mapsto (a, \rho(x, y) \bullet s)$ and its inverse D^{-1}_ρ.
- the *exchange* $\mathtt{xch} : \mathbf{X} \times [\mathbf{0},\mathbf{1}] \to \mathbf{X} \times [\mathbf{0},\mathbf{1}]$ defined as $(a, x \bullet s, e) \mapsto (a, e \bullet s, x)$

Notice that the exchange `xch` is an example of map that could not arise from a microcosm of maps from $\mathbf{X}$ to itself. This added principle is crucial for the definition of both dereliction and digging. Intuititvely, the microcosm of Definition 10 *without the exchange map* allows for Elementary Linear Logic[11], in the same spirit as our previous work on exponentials [31]; the added principle – the exchange – adds both dereliction and digging simultaneously.

▶ Remark. One actually consider thick and sliced graphings up to a larger equivalence than that of variants. Indeed, we consider that the sliced and thick graphing $\sum_{i=1}^{k} \frac{1}{k} A_i$

[10] We notice that those are all Borel automorphisms, thus in particular Borel-preserving and non-singular.

[11] To be more exact, the microcosm allowing for a model of ELL is the microcosm $\mathfrak{U}_\rho$ without the exchange but with the maps D_σ which permute the family of intervals $\{[(i-1)/k, i/k]\}_{i=1}^k$ in the dialect along a permutation σ of $\{1, \dots, k\}$. Without these maps, one cannot define contraction as one cannot represent *slice-changing edges* [31]; it is not necessary to have all of them, though, as for instance all such D_σ for permutations σ over sets $\{1, \dots, 2^p\}$ are enough. Notice that these maps – in the case $k = 2^p$ – are elements of $\mathfrak{U}_\rho$, defined as $D_\sigma = \mathtt{xch} \circ \rho_{(p)} \circ P_\sigma \circ \rho^{-1}_{(p)} \circ \mathtt{xch}$, where $\rho_{(p)}$ is recursively defined by:

$$\rho_{(0)} = \rho \qquad\qquad \rho_{(p)} = \rho_{(p-1)} \circ (\textstyle\prod_{i=1}^{2^{p-1}} \rho)$$

is equivalent to the *universal*[12] graphing H whose restriction to the part of the dialect $[(i-1)/k, i/k]$ is equal to A_i, modulo the affine transformation $[(i-1)/k, i/k] \to [0,1], x \mapsto (x \times k) - i + 1$.

By Theorem 8, we know that for any choices of Ω and m, the induced model interprets MALL. We will thus concentrate on exponential connectives here and refer the interested reader to earlier papers for the definition of MALL connectives.

4 The Exponentials

We now define the perennisation; it is not defined on all projects, but on the subset of so-called *balanced* projects. These are in particular projects whose dialect is equal to $[\mathbf{0},\mathbf{1}]$ or, by extension, whose graphings are "balanced" sums: $\sum_{i=1}^{k} \alpha_i A_i$ such that $\alpha_i = 1/k$ for all i. This is not a problem since projects interpreting proofs will will all satisfy these conditions. Exponentials are then quite easy to define on balanced projects: from A we construct $!A$ by "pushing" the dialect at the first position in the sequence $[\mathbf{0},\mathbf{1}]^{\mathbf{N}}$. For technical reasons, we also need to create a fresh new copy of $[\mathbf{0},\mathbf{1}]$ for promotion.

▶ **Definition 11.** A project $\mathfrak{a} = (a, A)$ is *balanced* if $a = 0$ and the dialect of A is $[\mathbf{0},\mathbf{1}]$. If E is a set of projects, we write $\mathrm{bal}(E)$ the subset of balanced projects in E.

In order to define exponentials, we will need the following map:

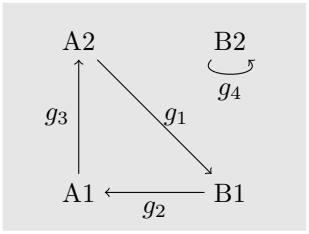
$$B : \left\{ \begin{array}{rcl} \mathbf{X} \times [\mathbf{0},\mathbf{1}] & \to & \mathbf{X} \\ (a,s,d) & \mapsto & (a, d \bullet s) \end{array} \right.$$

This map B will be used to encode the dialect of A in the support of the banged project $!A$. This way, the resulting project $!A$ will contain the exact same information as A, but will be dialect-free. Though it might seem a transparent and useless operation, the fact that the dialect is now part of the support makes the projects $!A$ and A behave quite differently when put into interaction with other projects. Intuitively, while the dialect is something private – e.g. states – the support is not, and some projects might interact with $!A$ non-uniformly w.r.t. the former dialect of A.

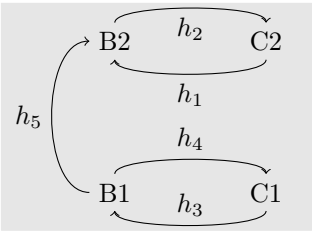
▶ **Example 12.** We consider two graphings, say G and H, both though of[13] as graphings with dialects $\{1,2\}$. Now, suppose that G and H are of type $\mathbf{A}$ and $\mathbf{A} \multimap \mathbf{B}$ respectively. Their execution $F :: G$ is then of type $\mathbf{B}$, and its dialect should be thought of as $\{1,2\} \times \{1,2\}$. We can also consider $!G$ and $!H$, which are of respective types $!\mathbf{A}$ and $!(\mathbf{A} \multimap \mathbf{B})$, and their execution $!G :: !H$. Let us explain why the latter cannot be of type $!\mathbf{B}$. Figure 2 illustrates this situation with examples of graphings G, $!G$, H, $!H$, as well as lists of the edges (alternating paths) of $G :: H$ and $!G :: !H$.

The execution of $!G$ and $!H$ actually produces the graphing defined as follows: compute the execution of G and H *as if they did not have any dialect*, and then take the perennisation of the result. In other words, the only alternating paths computed between G and H are those where the states of G and H are equal: this creates new paths (pictured in red path in Figure 2), this deletes paths (the blue paths), and leaves some of them "unchanged". As a consequence, we cannot prove that $!G :: !H$ is of type $!\mathbf{B}$ since the only graphing we know for sure to belong to this type is $!(G :: H)$.

[12] This is the *smallest* such graphing, i.e. if H' also satisfies this property, then H is included in H'.
[13] Recall that we are actually working with "variants" G^c and H^c whose dialect are $[\mathbf{0},\mathbf{1}]$.



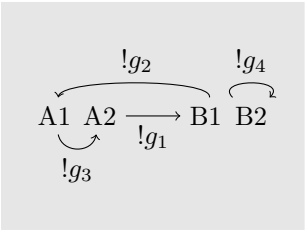
(a) Graphing G



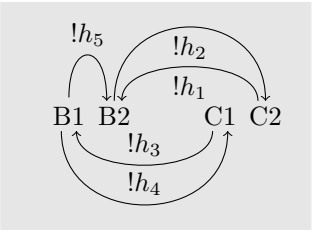
(b) Graphing H

(c) Edges of $F :: G$

g_1h_4	g_1h_2
$h_1g_4h_2$	
h_1g_2	g_3
h_3g_2	$h_3g_4h_4$



(d) Graphing $!G$



(e) Graphing $!H$

(f) Edges of $!F :: !G$

$!g_1!h_4$	
$!h_1!g_4!h_2$	$!g_1!h_5!g_4!h_2$
	$!g_3$
$!h_3!g_2$	

Figure 2 Illustration of Example 12 with graphing seens as graphs.

▶ **Definition 13** (Perennisation)**.** Let $\mathfrak{a} = (0, A)$ be a balanced project. We define its *perennisation* $!\mathfrak{a} = (0, !A)$ by considering the dialect-free graphing $!A = B^2(A \times \mathrm{Id}_{[\mathbf{0},\mathbf{1}]})$.

▶ **Definition 14** (Exponentials)**.** Let $\mathbf{A}$ be a conduct. We define the perennial conduct $!\mathbf{A}$ as the bi-orthogonal closure of the set

$$\sharp\mathbf{A} = \{!\mathfrak{a} \mid \mathfrak{a} \in \mathrm{bal}(\mathbf{A})\}$$

5 A Model of Full Linear Logic

To ensure that we have a sound interpretation of exponential connectives, we need to show that the following principles can be implemented:

- functorial promotion $(!A \otimes !(A \multimap B)) \multimap !B$;
- dereliction $!A \multimap A$;
- digging $!A \multimap !!A$.

The principle of *contraction* $!A \multimap !A \otimes !A$ does not appear in this list as it holds for every possible definition of perennisation[14]. Let us notice moreover that the principle of functorial promotion was already obtained in our earlier work on exponentials [31]. We will however use here a less involved method for defining exponentials and implementing functorial promotion. The principles at work are the same as in our earlier work, but this new implementation – inspired from recent work on complexity [34] – offers a clearer picture.

The change of perspective illustrated in Example 12 is at the heart of the question of implementing functorial promotion. We want to "simulate" the disjointness of dialects. This is done in two steps: first make the encodings (in $!G$ and $!H$) of the dialects of G and H disjoint, by linking $!G$ and $!H$ through the permutation exchanging the two first copies of $[0, 1]$. This corresponds to encoding the dialect of one of the two graphings on the second

[14] As explained in Footnote 11, the microcosm already contains all the needed maps to define contraction.

copy of $[0,1]$ instead of the first. Then we compute the result of this execution, obtaining a graphing which is almost $!(G :: H)$ except for the fact that his dialect is encoded on the two first copies of $[0,1]$ and not only on the first. We then use a specific graphing that will use $\rho : [0,1]^2 \to [0,1]$ to encode this dialect on the first copy only.

▶ **Theorem 15.** *Functorial Promotion holds.*

Proof (Sketch). The proof is much simpler in this setting than in our previous work on exponentials [31]. The principle is however quite the same: we use a first map to ensure the disjointness of the two "public dialects", and then we use a second map that will merge both copies. I.e. we define the maps:

$$\begin{array}{rcccc} \text{twist} & : & (\lambda, (x, \rho(y,z)) \bullet s) & \mapsto & (\lambda, (y, \rho(x,z)) \bullet s) \\ \text{merge} & : & (\lambda, (x, \rho(y,z)) \bullet s) & \mapsto & (\lambda, (\rho(x,y), z) \bullet s) \end{array}$$

To prove the result, we exhibit a project $\mathfrak{prom}$ and show that $\mathfrak{prom} \in !\mathbf{A} \otimes !(\mathbf{A} \multimap \mathbf{B}) \multimap !\mathbf{B}$. For this, we show that for all $!\mathfrak{a} = (0, !A) \in !\mathbf{A}$ and $!\mathfrak{f} = (0, !F) \in !(\mathbf{A} \multimap \mathbf{B})$, we have

$$\mathfrak{prom} :: !\mathfrak{a} :: !\mathfrak{f} = (0, \text{merge}(!A :: \text{twist}(!F))$$

Finally, one easily checks that $\text{merge}(!A :: \text{twist}(!F))$ is equal to $!(F :: A)$. ◀

Both digging and dereliction will work based on the simple idea that a continuous dialect $[\mathbf{0},\mathbf{1}]$ can be exchanged with a copy of $[\mathbf{0},\mathbf{1}]$ appearing in the Hilbert cube. This is exactly the computational principle encapsulated in the exchange map $\mathtt{xch}$. This implies that the *potential infinite* of (finite) dialects – i.e. the fact that a dialect can be any finite set, without bounds on its cardinality – can be managed within the projects themselves, something that could not be done in earlier constructions.

▶ **Theorem 16.** *Digging holds.*

Proof (Sketch). As for the proof of Theorem 15, we exhibit an element $\mathfrak{dig} \in !\mathbf{A} \multimap !!\mathbf{A}$. We show that, for all $!\mathfrak{a} = (0, !A)$ in $!\mathbf{A}$, one can compute $\mathfrak{dig} :: !\mathfrak{a} = (0, \text{push}(!A))$, where:

$$\text{push} : (\lambda, (x, \rho(y, \rho(z,w))) \bullet s, e) \mapsto (\lambda; (e, z, x, y) \bullet s, w)$$

It is clear that $\text{push}(!A)$ is equal to $!!A$, since the dialect of $!A$ (although $!A$ is dialect-free, not all elements of $!\mathbf{A}$ are, and therefore this is important) is encoded in the first copy of $[\mathbf{0},\mathbf{1}]$, while the second copy remains unused (this is because the second copy of $[\mathbf{0},\mathbf{1}]$ in $!A$ is unused[15]). ◀

The dereliction consists in "reconstructing" a dialect from a banged project. This can be performed using the same kind of tricks, i.e. using a continuous dialect.

▶ **Theorem 17.** *Dereliction holds.*

Proof (Sketch). Again, we exhibit an element $\mathfrak{der} \in !\mathbf{A} \multimap \mathbf{A}$. For this, we show that for all $!\mathfrak{a} \in !\mathbf{A}$, one can compute $\mathfrak{der} :: !\mathfrak{a} = (0, \text{raise}(!A))$ where

$$\text{raise} : (\lambda, (x,y) \bullet s, e) \mapsto (\lambda, s, \rho(x, \rho(y,e)))$$

Again, one easily checks that $\text{raise}(!A)$ is equal to A. ◀

[15] As this is not the case for elements of $\mathbf{A}$, this explains why digging is not a co-dereliction.

$$\frac{\Delta, N \Vdash \Gamma; \Theta}{\Delta, !N \Vdash \Gamma; \Theta}\ \mathrm{der}^{pol} \qquad \frac{\Delta \Vdash \Gamma, B; \Theta}{\Delta, !B^{\perp} \Vdash \Gamma; \Theta}\ \mathrm{der} \qquad \frac{\Delta, !!N \Vdash \Gamma; \Theta}{\Delta, !N \Vdash \Gamma; \Theta}\ \mathrm{dig}$$

Figure 3 Additional Rules for Dereliction and Digging

Notice that the maps used to interpret dereliction and digging are not the only ones that satisfy the right properties, e.g. if one replaces raise by the map $\mathrm{raise}^{(2)}(\lambda, (x, y) \bullet s, e) \mapsto (\lambda, s, \rho(\rho(x, y), e))$, we still have $\mathrm{raise}^{(2)}$. The exact expressions are however important when to ensure that the execution soundly represents cut-elimination.

6 Interpretation of proofs

We first recall the notion of winning projects [31]. Winning projects are the equivalent of game semantics' winning strategies or classical realizability's proof-like terms. In particular, all interpretations of proofs will be winning projects.

▶ **Definition 18.** A project $\mathfrak{a} = (a, A)$ is *winning* if it is balanced and if A is a disjoint union of transpositions, i.e. each edge e in A has a reverse edge e^* with $\phi^A_{e^*} = (\phi^A_e)^{-1}$ and the sources of edges are pairwise disjoint.

We now recall the basics of the proof system for which we define the interpretation of proofs. We are working with three different kinds of formulas, *positive*, *negative* and *neutral*. The technical reasons behind this are explained in our work on ELL [31]. Intuitively, neutral formulas correspond to the fragment of linear logic which does not allow for structural rules, negative formulas are those created from a perennisation while positive formulas are duals of negative formulas. They are defined inductively through the following grammar (neutral formulas are denoted by B which stands for *behavior* [31]):

$$\begin{aligned} B &:= X \mid X^{\perp} \mid \mathbf{0} \mid \mathbf{T} \mid B \otimes B \mid B \parr B \mid B \oplus B \mid B \,\&\, B \mid \forall X\ B \mid \exists X\ B \mid B \otimes N \mid B \parr P \\ N &:= \mathbf{1} \mid !B \mid !N \mid N \otimes N \mid N \,\&\, N \mid N \oplus N \mid N \parr P \mid \forall X\ N \mid \exists X\ N \\ P &:= \bot \mid ?B \mid ?P \mid P \parr P \mid P \,\&\, P \mid P \oplus P \mid N \otimes P \mid \forall X\ P \mid \exists X\ P \end{aligned}$$

▶ **Definition 19.** A sequent $\Delta \Vdash \Gamma; \Theta$ is such that Δ, Θ contain only negative formulas, Θ containing at most one formula and Γ containing only neutrals.

▶ **Definition 20** (The System $\mathrm{LL}_{\mathrm{pol}}$)**.** A proof in the system $\mathrm{LL}_{\mathrm{pol}}$ is a derivation tree constructed from the derivation rules of $\mathrm{ELL}_{\mathrm{pol}}$ [31], which are nothing more than polarised variants of elementary linear logic sequent calculus rules – presented with functorial promotion, extended with the rules in Figure 3.

One can then extend the inductive interpretation of proofs defined for $\mathrm{ELL}_{\mathrm{pol}}$ in earlier work [31] by interpreting the additional rules as follows: the interpretation $\|\pi\|$ of a proof π obtained from a proof π' by using a dereliction rule (resp. a digging rule) is defined as the execution of $\|\pi'\|$ with the project $\mathfrak{der}$ (resp. $\mathfrak{dig}$).

▶ **Theorem 21.** *For every proof π of a sequent $\Delta \Vdash \Gamma; \Theta$ in $\mathrm{LL}_{\mathrm{pol}}$, the interpretation $\|\pi\|$ is a winning project in $\|\Delta \Vdash \Gamma; \Theta\|$*

Proof. The proof is uninteresting and follows exactly the proof of the same result for the restricted system $\mathrm{ELL}_{\mathrm{pol}}$. The additional cases of dereliction and digging rules are completely transparent since the projects exhibited in the proofs of Theorem 16 and 17 are clearly winning projects. ◀

Notice that it is an open question whether the exponential isomorphism between $!(A \& B)$ and $!A \otimes !B$ did hold in the ELL model[16]. In the model we just described, however, this isomorphism holds; one just has to write down the usual derivation which can be interpreted soundly in the model:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Vdash A, A^{\perp};}{!A \Vdash A;}\,\text{der}}{!A, !B \Vdash A;}\,\text{weak} \qquad \dfrac{\dfrac{\dfrac{\Vdash B, B^{\perp};}{!B \Vdash B;}\,\text{der}}{!A, !B \Vdash B;}\,\text{weak}}{!A, !B \Vdash; A \& B}\,\&}{!!A, !!B \Vdash; !(A \& B)}\,!^{pol}}{!A, !!B \Vdash; !(A \& B)}\,\text{dig}}{!A, !B \Vdash; !(A \& B)}\,\text{dig}}{\Vdash; (!A \otimes !B) \multimap !(A \& B)}$$

This implies not only that that two conducts $!A \otimes !B$ and $!(A \& B)$ are isomorphic, but that they are equal. Indeed, the inclusion $!(A \& B) \subseteq !A \otimes !B$ can be proved as in our earlier paper [31]. Moreover, the interpretation $\|\pi\|$ of the above derivation can be shown to satisfy $\|\pi\| :: (!\mathfrak{a} \otimes !\mathfrak{b}) = !(\mathfrak{a} \& \mathfrak{b})$, where $\mathfrak{a} \& \mathfrak{b}$ is the usual construction of the & rule between $\mathfrak{a}$ and $\mathfrak{b}$, yielding the converse inclusion.

▶ **Theorem 22.** *For any conducts* $\mathbf{A}$ *and* $\mathbf{B}$ *in the model,* $!(A \& B) = !A \otimes !B$.

What about cut-elimination? It is known that GoI does not represent cut-elimination exactly, i.e. it is not always the case that if π' is the normal form of π, then $\|\pi'\| = \mathrm{Ex}(\|\pi\|)$. It was shown that cut-elimination for MLL is soundly represented by execution [29], but there is a mismatch even in the exponential-free fragment because of additive cuts. This issue is discussed in details in previous work [30], where we solve this problem by considering a notion of *observational equivalence* $\cong$. Indeed, we showed that, even if $\|\pi'\| \neq \mathrm{Ex}(\|\pi\|)$ in presence of an additive cut, $\|\pi'\|$ and $\mathrm{Ex}(\|\pi\|)$ are observationally equivalent.

▶ **Theorem 23** (Seiller [30])**.** *If* π' *is obtained from* π *by applying a step of cut-elimination* $(\&/\oplus)$ *then* $\|\pi\| \cong \|\pi'\|$.

We now consider the exponential connectives. We will consider a promotion rule cut against the following rules: dereliction, digging, and contraction. We consider for this a proof π of $\Vdash A^{\perp}, B$, or a proof π of $A \Vdash; B$ as both promotion rules are treated similarly, and its interpretation $\|\pi\| \in \mathbf{A} \multimap \mathbf{B}$. Applying a promotion rule (polarised or not) to π yields a proof ρ whose interpretation is $\|\rho\| = !\|\pi\| :: \mathfrak{prom}$. Then, given a proof π' to which we apply one of the three structural rules above to obtain a proof ρ', we consider the interpretations of the proof ν obtained by a cut between π and ρ, and the interpretation of the proof ν' obtained by applying a step of the cut-elimination procedure on ν. It turns out that those are equal, i.e. $\|\nu\| = \|\nu'\|$.

▶ **Theorem 24.** *If* π' *is obtained from* π *by applying a step of cut-elimination among (promotion/dereliction), (promotion/digging) or (promotion/contraction), then* $\|\pi\| = \|\pi'\|$.

[16] This is discussed in our earlier paper [31], but can be understood as follows: in the non-affine sequent calculus for ELL (presented with the functorial promotion rule) one cannot prove the implication $(!A \otimes !B) \multimap !(A \& B)$.

So execution computes these elimination steps on the nose. What about the last step, namely (promotion/weakening)? In that case, we are faced a problem similar to what happens for additive cuts in MALL. As execution is a completely local procedure, it cannot erase a whole proof at once. Thus, this elimination step is not soundly represented by the execution. However, one can show the following weaker result.

▶ **Theorem 25.** *If π' is obtained from π by applying a step of cut-elimination among (promotion/weakening), then $\|\pi\| \cong \|\pi'\|$.*

However, these results are not enough to entail that cut-elimination is soundly represented by execution *up to observational equivalence*. In particular, one would need to show that perennisation and observational equivalence interact properly, i.e. one would hope for a result stating that, given two balanced projects $\mathfrak{a}$ and $\mathfrak{b}$, $\mathfrak{a} \cong \mathfrak{b}$ if and only if $!\mathfrak{a} \cong !\mathfrak{b}$. One can prove that $\mathfrak{a} \ncong \mathfrak{b}$ implies $!\mathfrak{a} \ncong !\mathfrak{b}$, however the converse implication is still an open question.

7 Conclusion

Girard's so-called "GoI3" model [19] already provided an interpretation of full linear logic. However, we managed to do so in a quantitative-flavoured framework. Beyond the results presented here, the adaptation of the interaction graphs framework to deal with continuous dialects results in a more mature and complete construction. Indeed, the possibly to manage the maps at the level of the dialect through the microcosm opens new possibilities in terms of computational complexity. Indeed, while the weight monoid and the measurement of weights seem to be related to different computational paradigms and can be used, for instance, for representing probabilistic computation [34], the microcosm can be used to restrict the computational principles allowed in the model and characterise in this way various complexity classes [34, 33]. All the characterisations obtained considered variants of exponentials satisfying at least the contraction principle. In the more general construction explained here, we are now able to consider models of exponentials that do not satisfy this principle. In this line of work, it would be interesting to understand if one can adapt Mazza and Terui's work on parsimonious lambda-calculus [27, 28, 26], and obtain an interaction graph model for it.

References

1 Samson Abramsky and Radha Jagadeesan. Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic*, 59(2):543–574, 1994.

2 Scot Adams. Trees and amenable equivalence relations. *Ergodic Theory and Dynamical Systems*, 10:1–14, 1990.

3 Aloïs Brunel. Quantitative classical realizability. *Information and Computation*, 2014.

4 Ugo dal Lago and Martin Hofmann. Realizability models and implicit complexity. *Theoretical Computer Science*, 412:2029–2047, 2011.

5 Ugo Dal Lago and Olivier Laurent. Quantitative game semantics for linear logic. In Michael Kaminski and Simone Martini, editors, *Computer Science Logic*, volume 5213 of *Lecture Notes in Computer Science*, pages 230–245. Springer, 2008.

6 Vincent Danos. *La Logique Linéaire Appliquée à l'Étude de Divers Processus de Normalisation (principalement du λ-calcul)*. PhD thesis, Paris VII University, 1990.

7 Vincent Danos and Thomas Ehrhard. Probabilistic coherence spaces as a model of higher-order probabilistic computation. *Information and Computation*, 2011.

8 Vincent Danos and Jean-Baptiste Joinet. Linear logic & elementary time. *Information and Computation*, 183(1):123–137, 2003.

9 Thomas Ehrhard. Finiteness spaces. *Mathematical Structures in Computer Science*, 15(4):615–646, 2005.

10 Damien Gaboriau. Coût des relations d'équivalence et des groupes. *Inventiones Mathematicae*, 139:41–98, 2000.

11 Dan R. Ghica. Geometry of synthesis: a structured approach to vlsi design. In Martin Hofmann and Matthias Felleisen, editors, *POPL*, pages 363–375. ACM, 2007.

12 Dan R. Ghica and Alex Smith. Geometry of synthesis II: From games to delay-insensitive circuits. *Electr. Notes Theor. Comput. Sci.*, 265:301–324, 2010.

13 Dan R. Ghica and Alex Smith. Geometry of synthesis III: resource management through type inference. In T. Ball and M. Sagiv, editors, *POPL*, pages 345–356. ACM, 2011.

14 Dan R. Ghica, Alex Smith, and Satnam Singh. Geometry of synthesis IV: compiling affine recursion into static hardware. In M. Chakravarty, Z. Hu, and O. Danvy, ed., *ICFP*. 2011.

15 Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–102, 1987.

16 Jean-Yves Girard. Normal functors, power series and λ-calculus. *Annals of Pure and Applied Logic*, 37(2):129–177, February 1988.

17 Jean-Yves Girard. Geometry of interaction I: Interpretation of system F. In *In Proc. Logic Colloquium 88*, 1989.

18 Jean-Yves Girard. Towards a geometry of interaction. In *Proceedings of the AMS Conference on Categories, Logic and Computer Science*, 1989.

19 Jean-Yves Girard. Geometry of interaction III: Accommodating the additives. In *Advances in Linear Logic*, number 222 in Lecture Notes Series, pages 329–389. CUP, 1995.

20 Jean-Yves Girard. Light linear logic. In *Selected Papers from the International Workshop on Logical and Computational Complexity*, LCC '94, pages 145–176, London, UK, UK, 1995. Springer-Verlag.

21 Jean-Yves Girard. Geometry of interaction V: Logic in the hyperfinite factor. *Theoretical Computer Science*, 412:1860–1883, 2011.

22 John Martin Elliott Hyland and C.-H. Luke Ong. On full abstraction for PCF: I, II, and III. *Information and Computation*, 163(2):285–408, 2000.

23 Jean-Louis Krivine. Typed lambda-calculus in classical zermelo-fraenkel set theory. *Archive for Mathematical Logic*, 40(3):189–205, 2001.

24 Jean-Louis Krivine. Realizability in classical logic. *Panoramas et synthèses*, 27, 2009.

25 Jim Laird, Guy McCusker, Giulio Manzonetto, and Michele Pagani. Weighted relational models of typed lambda-calculi. In *Proceedings of the 28th Annual IEEE Symposium on Logic in Computer Science, LICS 2013, June 25-28, 2013, New Orleans, USA*, 2013.

26 Damiano Mazza. Simple affine types and logspace.

27 Damiano Mazza. Non-uniform polytime computation in the infinitary affine lambda-calculus. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Proceedings of ICALP*, volume 8573 of *LNCS*, pages 305–317. Springer, 2014.

28 Damiano Mazza and Kazushige Terui. Parsimonious types and non-uniform computation. Submitted to ICALP 2015.

29 Thomas Seiller. Interaction graphs: Multiplicatives. *Annals of Pure and Applied Logic*, 163:1808–1837, December 2012.

30 Thomas Seiller. Interaction graphs: Additives. *Accepted for publication in Annals of Pure and Applied Logic*, 2014.

31 Thomas Seiller. Interaction graphs: Exponentials. *Submitted*, 2014.

32 Thomas Seiller. Interaction graphs: Graphings. *Submitted*, 2014.

33 Thomas Seiller. Interaction graphs and complexity I. In preparation, 2015.

34 Thomas Seiller. Towards a *Complexity-through-Realizability* theory. *Submitted*, 2015.