Quantitative Semantics of the Lambda Calculus: Some Generalisations of the Relational Model

C.-H. Luke Ong
University of Oxford

***Abstract*—We present an overview of some recent work on the quantitative semantics of the λ-calculus. Our starting point is the fundamental degenerate model of linear logic, the *relational model*. We show that three quantitative semantics of the simply-typed λ-calculus are equivalent: the relational semantics, HO/N game semantics, and the Taylor expansion semantics. We then consider two recent generalisations of the relational model: first, *$\mathcal{R}$-weighted relational models* where $\mathcal{R}$ is a complete commutative semiring, as studied by Laird et al.; secondly, *generalised species of structures*, as introduced by Fiore et al. In each case, we briefly discuss some applications to quantitative analysis of higher-order programs.**

I. An overview of quantitative models

Denotational semantics (or Scott-Strachey semantics) [73] is an approach to formalising the meanings of programs by mapping them into some abstract domain of mathematical objects. A central tenet of denotational semantics is that the semantics should be defined compositionally. The theory of equality induced by the denotational semantics gives a natural notion of program equivalence: two programs are equal if they have the same denotation. On the other hand, according to *operational semantics*, the meaning of a program is the behaviour of some (typically abstract) machine when running it. There is a compelling notion of program equivalence based on operational semantics: two program phrases are *observationally equivalent* just if they compute the same outcome when one is replaced by the other in all possible program contexts. A central problem in the semantics of programming languages is to construct, for a given class of languages, a denotational semantics that agrees with the operational semantics. The strongest such goodness-of-fit criterion is *full abstraction* [62, 70]: the coincidence of the denotational equality of programs with observational equivalence.

Though the basis of a successful programming language theory, the notion of observational equivalence, and the related developments in denotational and operational semantics, have tended to ignore *quantitative* notions such as time, space and energy as computational resource, or in the case of nondeterministic and probabilistic computation, such quantities as the probability of a successful computation, and the expected termination time.

A major step in quantitative semantics was Girard's *linear logic* [39]. A refinement of classical and intuitionistic logic, linear logic emphasises the rôle of formulas as *resources*. The use of linear logic as an organisational principle in semantics of computation has the advantage of immediately revealing information about resource usage. Unsurprisingly models of linear logic are quantitative. This is already evident in the simplest degenerate model, the *relational model* [13, 39], in which multisets are used to record the number of times a resource is used.

Around the time of the introduction of linear logic [39], Girard also proposed the *normal functor semantics* of the λ-calculus [40]. In this semantics, a term is interpreted as a formal power series with set-valued coefficients, i.e., as a functor $X \to \mathbf{Set}$, where the set X is the denotation of a data type. In that work, Girard was "mainly concerned with a quantitative approach: not only to say when f takes the value $\cdots$ at argument $\cdots$, but also how many times it does" [40, p. 172]. His original intuitions came from linear algebra: data types are interpreted as vector spaces; a resource of type A is a vector $\sum_{a \in \mathrm{base}(A)} m_a \cdot a$ where the coefficient m_a gives the multiplicity of the atomic datum a of type A in the resource. Programs are interpreted as power series or analytic functions; programs that use their input exactly once then correspond to linear functions. The idea is that, by choosing the appropriate coefficients, we should be able to use such a semantics to analyse programs, not only *qualitatively*, with respect to "what they can do", but also *quantitatively*, with respect to "in how many steps", or "in how many different ways", or "with what probability".

Girard's work [39, 40] engendered a rich vein of research, initially in models of linear logic. In his normal functor model, the scalars are sets. Ehrhard's *Köthe sequence spaces* [23] and *finiteness spaces* [24] recast Girard's intuitions on actual vector spaces over fields (the former over $\mathbb{R}$ and $\mathbb{C}$, and the latter over any field). In the *weighted relational models* of Lamarche [56] and Laird et al. [51, 55], the scalars are elements of any continuous commutative semiring, or, more generally, any complete commutative semiring. In the quantitative model of a higher-order quantum programming language studied by Pagani et al. [67], the scalars are completely positive maps of a finite dimension. Girard has suggested how his coherence spaces model [38, 39] of linear logic can be refined to give an account of probabilistic computation. Building on Girard's ideas, Danos, Ehrhard et al. [18, 31] have analysed *probabilistic coherence spaces* as a model of higher-order probabilistic computation.

In (idealised) quantitative semantics, programs are analytic functions, which are infinitely differentiable. This motivates the question of understanding differentiation as a program-

978-1-5090-3018-7/17/$31.00 ©2017 IEEE

ming construct in a higher-order setting; and it led Ehrhard and Regnier to introduce the *differential* λ*-calculus* [27], a differential calculus for higher-order functions, and an accompanying *differential linear logic* [29]. In a follow-up paper [30], Ehrhard and Regnier introduced the *Taylor expansion* of a λ-term as a formal sum (with rational coefficients) of terms of the *resource calculus* [30, 69], which may be viewed as linear approximants of the λ-term. Models of the differential λ-calculus (and resource λ-calculus) [6, 14] are naturally quantitative. Many models of linear logic give rise to models of the differential λ-calculus / linear logic. Particularly attractive is the category $\mathbf{CVS}$ of *convenient vector spaces* (c^∞-complete locally convex topological vector spaces) and bornological linear maps [7], in the sense of Frölicher and Kriegl [36]. The category $\mathbf{CVS}$ supports a linear exponential comonad for which the Kleisli category is the category of smooth maps on convenient vector spaces.

There are proposals, from the programming language community, of denotational and operational semantics that carry quantitative information. *Game semantics* [1, 44, 63] is a denotational semantics with a strong operational flavour that typically captures more intensional information about the computation (such as the number of times an input argument is evaluated) than the more abstract Scott-Strachey style denotational semantics. Exploiting the quantitative nature of *HO/N game model* [44], Férée [32] has recently proposed a definition of complexity for higher-order functions, as well as a class of polynomial time computable higher-order functions. Sand's *theory of improvement* [71, 72] gives an operational account of cost based on a refined notion of program equivalence. Inspired by Sand's ideas, Ghica has given a game semantics, called *slot games* [37], which is induced by a notion of observation formalised in Sand's theory of improvement.

Outline of the paper

In this survey paper, we start from the relational model, and consider two ways to generalise it.

In Section II we introduce $\mathbf{MRel}$, the Kleisli category of the finite-multiset comonad on the category $\mathbf{Rel}$ of sets and relations. We then study three quantitative semantics of the simply-typed λ-calculus: (i) relational semantics (ii) HO/N game semantics, and (iii) Taylor expansion semantics. We show that they are equivalent in an appropriate sense.

In Section III we consider the first generalisation of the relational model, namely, the *weighted relational models*, where the weights are elements of any continuous commutative semiring, and, more generally, any complete commutative semiring. We briefly discuss some applications of these models to quantitative analysis of nondeterministic higher-order computation.

Section IV concerns the second generalisation of the relational model, which is in a 2-categorical direction. We introduce the bicategory $\mathbf{Prof}$ of profunctors, and the cartesian closed bicategory $\mathbf{ESP}$ of *generalised species of structures*. We analyse the $\mathbf{ESP}$-semantics of the nondeterministic $\lambda\mathbf{Y}$-calculus, and show that it coincides with the *rigid* Taylor expansion semantics. Rigid Taylor expansion of a λ-term is a version of Taylor expansion that uses linear approximants in the form of *rigid* resource terms, which are list-based (as opposed to bag-based), and considered modulo an isomorphism action.

II. The relational model and its various faces

The *relational model* [40] is the simplest degenerate model of the linear logic. It underlies most denotational models of (differential) linear logic [27, 39], and serves to motivate the general constructions of quantitative models in Sections III and IV. The relational semantics of the λ-calculus is equivalent to type assignment in a system of commutative, associative and nonidempotent, intersection refinement types. We relate the relational semantics to HO/N game semantics on the one hand, and to Taylor expansion semantics on the other.

A. The relational model $\mathbf{MRel}$

We start from the category $\mathbf{Rel}$ of sets and relations. Given sets A and B, $\mathbf{Rel}(A,B) := \mathcal{P}(A \times B)$. The identities are the diagonal relations: $\mathsf{id}_A = \{(a,a) \mid a \in A\}$; and the composite of $s \in \mathbf{Rel}(A,B)$ and $t \in \mathbf{Rel}(B,C)$ is just relational composition:

$$(s\,;t) := \{(a,c) \in A \times C \mid \exists b \in B\,.\,(a,b) \in s \wedge (b,c) \in t\}.$$

Given sets A and B, their tensor product $A \otimes B = A \times B$ is the cartesian product, with unit $1 = \{*\}$, an arbitrary singleton set. $\mathbf{Rel}$ is $*$-autonomous: the linear function space $A \multimap B = A \times B$, with the natural bijection $\mathbf{Rel}(C \otimes A, B) \cong \mathbf{Rel}(C, A \multimap B)$ being induced by the cartesian product associativity isomorphism. The categorical product $A_1 \sqcap A_2$ is the disjoint union, and the terminal object $\top = \emptyset$. The dualising object $\bot = 1$, and so, $\mathbf{Rel}$ is compact closed.

Notation: We represent a finite multiset m over a set A as an unordered list $[a_1, \cdots, a_n]$, and say that n is its cardinality. We write the union of multisets m and m' as $m + m'$, and $\mathcal{M}_{\mathrm{fin}}(A)$ for the set of finite multisets over A.

The finite-multiset construction, $\mathcal{M}_{\mathrm{fin}}$, is a comonad on $\mathbf{Rel}$, acting on morphisms $s \in \mathbf{Rel}(A,B)$ as $\mathcal{M}_{\mathrm{fin}}(s) := \{([a_1,\ldots,a_n],[b_1,\ldots,b_n]) \mid \forall i \le n.(a_i,b_i) \in s\}$, with unit $\mathsf{der}_A := \{([a],a) \mid a \in A\} : \mathcal{M}_{\mathrm{fin}}(A) \to A$ and multiplication $\mathsf{dig}_A := \{(\theta_1 + \cdots + \theta_n, [\theta_1,\ldots,\theta_n]) \mid \forall i \le n\,.\,\theta_i \in \mathcal{M}_{\mathrm{fin}}(A)\} : \mathcal{M}_{\mathrm{fin}}(A) \to \mathcal{M}_{\mathrm{fin}}(\mathcal{M}_{\mathrm{fin}}(A))$.

We define the category $\mathbf{MRel}$ as the Kleisli category of the $\mathcal{M}_{\mathrm{fin}}$ comonad. It is useful to give a direct description of $\mathbf{MRel}$.

- The objects of $\mathbf{MRel}$ are sets.
- A morphism from A to B is a relation from $\mathcal{M}_{\mathrm{fin}}(A)$ to B. I.e. $\mathbf{MRel}(A,B) := \mathcal{P}(\mathcal{M}_{\mathrm{fin}}(A) \times B)$.

The identity map of A is the relation $\mathsf{id}_A := \{([a],a) \mid a \in A\}$. The composite of $s \in \mathbf{MRel}(A,B)$ and $t \in \mathbf{MRel}(B,C)$ is

$$(s\,;t) := \{(m,c) \mid \exists (m_1,b_1),\ldots,(m_k,b_k) \in s\,.\; m = m_1 + \cdots + m_k \;\wedge\; ([b_1,\ldots,b_k],c) \in t\}.$$

$\mathbf{MRel}$ is cartesian closed. The products in $\mathbf{Rel}$ give the products in $\mathbf{MRel}$. It is convenient to regard the canonical

bijection $\mathcal{M}_{\mathrm{fin}}(A_1) \times \mathcal{M}_{\mathrm{fin}}(A_2) \cong \mathcal{M}_{\mathrm{fin}}(A_1 + A_2)$ as equality. Thus we will still write (m_1, m_2) for the corresponding element of $\mathcal{M}_{\mathrm{fin}}(A_1 + A_2)$. Given sets A and B, the exponential object B^A is $\mathcal{M}_{\mathrm{fin}}(A) \times B$, and the evaluation morphism $\mathsf{ev}_{A,B} \in \mathbf{MRel}(B^A \sqcap A, B)$ is

$$\mathsf{ev}_{A,B} := \{(([[(m,b)],m),b) \mid m \in \mathcal{M}_{\mathrm{fin}}(A), b \in B\}$$

Given $s \in \mathbf{MRel}(C \sqcap A, B)$, its exponential transpose

$$\lambda(s) := \{(m,(m',b)) \mid ((m,m'),b) \in s\} \in \mathbf{MRel}(C, B^A).$$

MRel is a *differential* λ*-category* [6, 14]: the homsets are endowed with the semi-additive structure $(\mathbf{MRel}(A,B), +, \emptyset)$. Given $s \in \mathbf{MRel}(A,B)$, we define its *derivative* $D(s) \in \mathbf{MRel}(A \sqcap A, B)$ as

$$D(s) := \{(([a],m),b) \mid (m + [a], b) \in s\}.$$

B. Relational semantics as refinement type assignment

Simple types are defined by the grammar: $A, B ::= \mathsf{o} \mid A \to A$, where o is the unique atomic type. The *relational semantics* of a λ-term-in-context $\Gamma \vdash M : A$, written $[\![\Gamma \vdash M : A]\!]^{\mathrm{MRel}}$, is determined by the cartesian closed structure of **MRel**, once the interpretation of the atomic type o is given. Let $\mathcal{X}$ be a fixed set which is assumed to be countably infinite. (The assumption is not necessary for the relational semantics to be well-defined; see Remark 8.) Define $[\![A]\!]^{\mathrm{MRel}}$ inductively by

$$[\![\mathsf{o}]\!]^{\mathrm{MRel}} := \mathcal{X}$$
$$[\![A \to B]\!]^{\mathrm{MRel}} := \mathcal{M}_{\mathrm{fin}}([\![A]\!]^{\mathrm{MRel}}) \times [\![B]\!]^{\mathrm{MRel}}.$$

Let $\Gamma = x_1 : A_1, \ldots, x_n : A_n$. The relational semantics $[\![\Gamma \vdash M : A]\!]^{\mathrm{MRel}}$ is then a subset of

$$\left(\mathcal{M}_{\mathrm{fin}}([\![A_1]\!]^{\mathrm{MRel}}) \times \cdots \times \mathcal{M}_{\mathrm{fin}}([\![A_n]\!]^{\mathrm{MRel}})\right) \times [\![A]\!]^{\mathrm{MRel}}.$$

Refinement types: A simple but important observation is that the relational semantics $[\![A]\!]^{\mathrm{MRel}}$ of a simple type A can be seen as the set of all nonidempotent intersection types that refine A [12, 20]. Let α and β be elements of $\mathcal{X}$. Then an element of $[\![\mathsf{o}]\!]^{\mathrm{MRel}}$ is an atomic type $\alpha \in \mathcal{X}$. Given a sequence $a_1, \ldots, a_n$ of elements in $[\![A]\!]^{\mathrm{MRel}}$, we write $a_1 \wedge \cdots \wedge a_n$ to mean the multiset $[a_1, \ldots, a_n] \in \mathcal{M}_{\mathrm{fin}}([\![A]\!]^{\mathrm{MRel}})$. It follows that the intersection connective $\wedge$ is commutative, associative and nonidempotent, as in Kfoury's treatment [49] and, more recently, de Carvalho's [20]. An element of $[\![A \to B]\!]^{\mathrm{MRel}} = \mathcal{M}_{\mathrm{fin}}([\![A]\!]^{\mathrm{MRel}}) \times [\![B]\!]^{\mathrm{MRel}}$ is then a pair (θ, a), which we write as $\theta \multimap a$.

Thus, given a simple type A, $[\![A]\!]^{\mathrm{MRel}}$ can be seen as the set of intersection types defined by the grammar

$$a, b ::= \alpha \mid \theta \multimap a \qquad \theta ::= a_1 \wedge \cdots \wedge a_n \quad (n \geq 0)$$

that refines A (written $a \lhd A$), where the *refinement relation* $a \lhd A$ and $\theta \lhd\, !A$ is defined by the following rules.

$$\frac{}{\alpha \lhd \mathsf{o}} \qquad \frac{\theta \lhd\, !A \qquad b \lhd B}{(\theta \multimap b) \lhd (A \to B)} \qquad \frac{\forall i \leq n.\ a_i \lhd A}{a_1 \wedge \cdots \wedge a_n \lhd\, !A}$$

We write $\top$ for the empty intersection; $\multimap$ associates to the right, and $\wedge$ takes precedence over $\multimap$. In case $a \lhd A$, we say a is a *refinement type* of A; similarly in case $\theta \lhd\, !A$, we say θ is a *refinement intersection* of A.

$$\frac{}{x : a \vdash x : a} \qquad \frac{\Delta_0 \vdash M : \theta \multimap b \qquad \Delta_1 \vdash N : \theta}{\Delta_0, \Delta_1 \vdash M\, N : b}$$

$$\frac{\Delta, x : a_1, \ldots, x : a_n \vdash M : b \qquad x \notin \mathrm{dom}(\Delta)}{\Delta \vdash \lambda x.M : (a_1 \wedge \cdots \wedge a_n) \multimap b}$$

$$\frac{\forall i \leq n.\ \Delta_i \vdash M : a_i}{\Delta_1, \ldots, \Delta_n \vdash M : a_1 \wedge \cdots \wedge a_n}$$

Fig. 1. Rules of the type system corresponding to the relational semantics

A *refinement type judgement* is a triple of the form $\Delta \vdash M : b$ where Δ is an *environment*, which is defined to be a finite multiset of type bindings of the form $x : a$ such that a is a refinement of the simple type of x. Given environments Δ_1 and Δ_2, we write Δ_1, Δ_2 for their multiset union, and define $\mathrm{dom}(\Delta) := \{x \mid \exists a.\ (x : a) \in \Delta\}$, the *domain* of Δ. Figure 1 lists the typing rules.

The refinement type system is equivalent to the relational semantics in the following sense. Take a λ-term-in-context $\Gamma \vdash M : A$ with $\Gamma = x_1 : A_1, \cdots, x_n : A_n$, and a refinement-type environment Δ such that $\mathrm{dom}(\Delta) \subseteq \mathrm{dom}(\Gamma)$. Suppose $\Delta = [x_1 : a_{11}, \cdots, x_1 : a_{1r_1}] + \cdots + [x_n : a_{n1}, \cdots, x_n : a_{nr_n}]$; write $\overline{\Delta} = (a_{11} \wedge \cdots \wedge a_{1r_1}, \cdots, a_{n1} \wedge \cdots \wedge a_{nr_n})$. Then we have:

Theorem 1. $\Delta \vdash M : a$ *iff* $(\overline{\Delta}, a) \in [\![\Gamma \vdash M : A]\!]^{\mathrm{MRel}}$.

Thus the relational semantics and refinement type assignment are equivalent, and we shall use them interchangeably.

C. Playful types are the inhabited refinement types

First a quick review of the syntax of the *resource* λ*-calculus* [11, 25, 30]. *Resource terms* and *bags* are given by the grammar:

$$M, N := x \mid \lambda x.M \mid M\, P$$
$$P, Q := [M_1, \ldots, M_n] \quad (n \geq 0).$$

We call M a *term*, and call P, which is a finite multiset of terms, a *bag*. Since a bag is a multiset, it is identified with a permutation of its elements; as is standard, α-equivalent terms (respectively bags) are identified. Henceforth we shall only consider *simply-typed* resource terms and bag (which means that all terms in a bag must have the same simple type). A typed term is β-normal if it does not have a subterm of the form $(\lambda x.M)P$; it is η-long if every application and variable in the term is fully applied. A resource term is *normal* if it is β-normal and η-long.

We define a refinement type assignment system for resource λ-terms. The typing rules are listed in Fig. 2 (except for the last, they are the same as the rules in Fig. 1). We make the same assumptions about environments Δ as in the system for the λ-terms. It follows that every provable judgement respects the simple types of the terms and bags.

$$\frac{}{x : a \vdash x : a} \qquad \frac{\Delta_1 \vdash M : \theta \multimap b \qquad \Delta_2 \vdash P : \theta}{\Delta_1, \Delta_2 \vdash M\, P : b}$$

$$\frac{\Delta, x : a_1, x : a_2, \ldots, x : a_n \vdash M : b \qquad x \notin \mathrm{dom}(\Delta)}{\Delta \vdash \lambda x.M : (a_1 \wedge \cdots \wedge a_n) \multimap b}$$

$$\frac{\forall i \leq n. \Delta_i \vdash M_i : a_i}{\Delta_1, \ldots, \Delta_n \vdash [M_1, \ldots, M_n] : a_1 \wedge \cdots \wedge a_n}$$

Fig. 2. Rules of the refinement type system for resource λ-terms

Say that a refinement type $a \lhd A$ is *inhabited* if $\vdash M : a$, for some (closed) resource term M of simple type A. A natural question is to characterise the inhabited refinement types.

Every refinement type (intersection, respectively) defines a finite unordered $\mathcal{X}$-node-labelled tree (forest, respectively) as follows.

- If, for each $i \leq n$, T_i is the tree defined by a_i, then the forest defined by the intersection $a_1 \wedge \cdots \wedge a_n$ is the disjoint union of (labelled) forests $\{T_1\} + \cdots + \{T_n\}$.
- If, for each $i \leq n$, F_i is the forest defined by θ_i, then the tree defined by the type $\theta_1 \multimap \cdots \multimap \theta_n \multimap \alpha$ is the tree whose root node is labelled α, and the set of the root node's child-subtrees is $F_1 + \cdots + F_n$.

Given a refinement type a, let us call the tree thus defined $\mathcal{T}_a$.

Say that a refinement type a is *involutive* if each atomic type in a has exactly one contravariant and one covariant occurrence. (Nodes of the tree $\mathcal{T}_a$ inherit their polarity from a: nodes at levels $0, 2, 4, \cdots$ are covariant; the other nodes are contravariant.) An involutive refinement type a induces a directed graph $\mathcal{G}_a$ whose nodes are those of $\mathcal{T}_a$, and whose edges are given by:

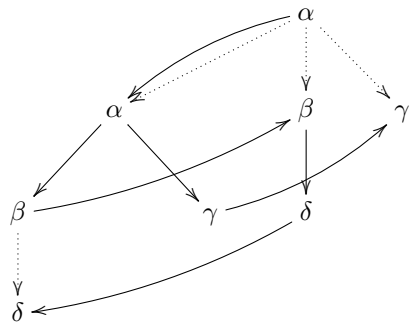
(1) edges of $\mathcal{T}_a$ from contravariant nodes, and
(2) edges linking a covariant node to the other node in $\mathcal{G}_a$ labelled by the same atomic type.

Definition 2 ([78])**.** An involutive refinement type a is *playful* if the induced graph $\mathcal{G}_a$ is acyclic; and for each edge (v, v') of the tree $\mathcal{T}_a$, there is a path in $\mathcal{G}_a$ from v to v'.

Example 3. Consider the (playful) refinement type $a \lhd A$ where $A = ((\mathsf{o} \to \mathsf{o}) \to \mathsf{o}) \to \mathsf{o}$ and

$$a \;=\; \left(\begin{array}{r} \big((\delta \multimap \beta) \wedge (\top \multimap \gamma)\big) \multimap \alpha \\ \wedge \\ (\top \multimap \delta) \multimap \beta \\ \wedge \\ \top \multimap \gamma \end{array} \right) \multimap \alpha.$$

The induced tree $\mathcal{T}_a$ (with edges indicated by straight arrows, either solid or dotted) and graph $\mathcal{G}_a$ (with edges as solid arrows, either straight or curved) are depicted as follows.



Theorem 4. *A refinement type is inhabited iff it is playful.*

A proof can be found in [78].

Remark 5. The original intuition behind the playfulness condition came from (P-visibility in) HO/N game semantics. However an involutive refinement type induces a proof structure, and playfulness is closely connected to the correctness criterion [19, 39] of multiplicative linear logic.

D. $\mathbf{MRel}$*, HO/N games and resource* λ*-calculus*

We show that the following quantitative semantics of the simply-typed λ-calculus are equivalent [78]:

1) Relational semantics (or refinement type assignment)
2) HO/N game semantics [44, 63]
3) Taylor expansion semantics [30]

Take a λ-term M. In the relational semantics, M is interpreted as a collection of refinement types. In HO/N game semantics, M is interpreted as a collection of plays. In the Taylor expansion semantics, M is interpreted as a collection of resource terms in normal form.

Underpining the equivalence of the three semantics is the following technical result [78]:

Theorem 6. *There exists a family of bijections parametrised by simple types* A

$$\varpi_A : \{\, \textit{plays of arena } A \,\}/\!\sim \;\to \left\{ \begin{array}{l} \textit{normal resource} \\ \textit{terms of type } A \end{array} \right\}$$

that preserves composition,, where $\sim$ *is the alternating homotopy relation of Melliès [59, 77]. Via the bijections, the game semantics of a* λ*-term coincides with its Taylor expansion followed by normalisation.*

As a consequence, we can define a functor from the category of HO/N games to the Kleisli category $\mathbf{MRel}$.

Corollary 7. *The category of HO/N games is isomorphic to a sub-cartesian closed category of the Kleisli category* $\mathbf{MRel}$.

Remark 8. Laird et al. [54] developed a method to construct a differential category [6] from a symmetric monoidal category, and then reconstructed a known category of games from a new category of *exhausting games*. It follows from this construction that there is a functor from a category of games to the relational model. Compared to our work, this functor can be

seen as the "colourless" version (i.e. the set of atomic types $\mathcal{X}$ is singleton) of the functor of Corollary 7.

In the rest of this section, we explain the bijective correspondence between the following collections (and direct the reader to [78] for the other details):

1) *playful* refinement types
2) ($\sim$-equivalence classes of) plays
3) normal resource terms

(Because of the length restriction, we will assume basic knowledge of HO/N games, and refer the reader to [44, 77].) A rough and informal sketch of the idea is depicted in Figure 3. A resource term in normal form can be written as a tree, as in the middle of Figure 3. Each node is labelled by λ-abstraction followed by a head variable.[1] The edges express the relationship between functions and arguments: the child of a node is an argument of the head variable of the parent. Then we line up the nodes of the tree in such a way that every node is located to the left of its children. The resulting sequence of λ-abstractions and variables is equipped with leftward pointers: the pointer from an abstraction (solid lines in Figure 3) comes from the parent-child relation in the tree, and the pointer from a variable (dotted lines) is determined by the binder-bindee relation. The sequence with pointers is reminiscent of a play in the HO/N game model; indeed one can construct a play from it by replacing λ-abstraction and variables with O- and P-moves, respectively, in an appropriate way. (The structure that is obtained—as shown on the rightmost of Figure 3—is called *traversal*, as studied in [64, 65].)

A resource term generates a set of plays because the process of lining up is nondeterministic. The main theorem states that the set of plays generated by a resource term is a $\sim$-equivalence class. Moreover every play is generated by a resource term.

The idea should now be intuitively clear. However, to define the bijection ϖ_A of Theorem 6, it is instructive to use the refinement type assignment (or, equivalently, the relational model) as a bridge, because refinement type assignment systems for both the resource calculus and the game model have already been studied: the relational model is a common tool for studying resource terms [11], and game semantics for an intersection type assignment system has been studied [66].

The definition of the bijection ϖ_A of Theorem 6 is illustrated in Figure 4, and is divided into four steps. The simple type in question is $A = ((\mathsf{o}_{111} \to \mathsf{o}_{11}) \to \mathsf{o}_1) \to \mathsf{o}_\epsilon$. (We use subscripts to distinguish the occurrences of the atomic type o in A.)

Step 1: Colouring a play: We assign a "colour" (written as α, β, γ and δ in Fig. 4) to each occurrence of moves in such a way that

- every pair of consecutive O-P moves have the same colour, and
- different occurrences of O-moves (respectively P-moves) have different colours.

[1] We consider λ-abstraction that can bind a (possibly empty) sequence of variables, although only sequences of length 1 appear in Figure 3.

Thus one needs n colours to annotate a play of length $2n$.

Step 2: Representing a coloured play by a tree: This step simply forgets the sequential structure of the (coloured) play. The resulting structure is a tree whose edge is a justification pointer of the play and whose node is labelled by a pair of a move and a colour. For example, the move and colour of the node named l_7 in Figure 4 is o_{11} and δ. The node named l_i corresponds to the ith move in the original play. This structure is called a *valuated thick subtree* in [9], a *high-level arena* in [66], and a *partitioned position* in [21].

Step 3: Constructing a refinement type: High-level arenas (or, valuated thick subtrees) bijectively correspond to intersection types that refine the simple type. Let us write a_i for the type corresponding to the subtree rooted at l_i. For example

(i) a_8 is an atomic type δ that refines o_{111},
(ii) a_3 is the type $\delta \multimap \beta$ that refines $\mathsf{o}_{111} \multimap \mathsf{o}_{11}$,
(iii) a_5 is the type $\top \multimap \gamma$ that refines $\mathsf{o}_{111} \multimap \mathsf{o}_{11}$, where $\top$ is the empty intersection type, and
(iv) a_2 is the type $((\delta \multimap \beta) \wedge (\top \multimap \gamma)) \multimap \alpha$ that refines $(\mathsf{o}_{111} \multimap \mathsf{o}_{11}) \multimap \mathsf{o}_1$.

The type $a_1 = (a_2 \wedge a_4 \wedge a_6) \multimap \alpha$ is written in Figure 4.

Step 4: Computing the inhabitant: The resource term corresponding to the play is the inhabitant of the refinement type. An inhabitant always exists uniquely (up to α-conversion) for an refinement type constructed in this way.

There are several difficulties in proving bijectivity of the map defined above. In particular, in the last step, the existence and uniqueness of an inhabitant are challenging to establish. Our strategy is to study the map defined by Steps 1-3 and the inverse of Step 4. We characterise the images of those maps, using game semantics, and show their coincidence. Then, given a refinement type in the image, we construct a play and a resource term and prove their uniqueness. See [78] for the details.

Preservation of composition by the bijection is relatively easy to prove by applying known results.

E. Taylor expansion as game semantics

For a term L of the simply-typed *nondeterministic* λ-calculus, we define L^* by:

$$x^* := \{\, x \,\} \qquad (\lambda x.L)^* := \{\, \lambda x.M \mid M \in L^* \,\}$$
$$(L\,L')^* := \{\, M\,[N_1, \ldots, N_k] \mid M \in L^*,\ k \geq 0,\ \forall i \leq k.\, N_i \in (L')^* \,\}$$
$$(L_1 + L_2)^* := L_1^* \cup L_2^*$$

We call L^* the *Taylor expansion of* L. For example, $\big(\lambda f.f(\lambda x.f(\lambda y.y))\big)^*$ contains resource terms $\lambda f.f[]$ and $\lambda f.f[\lambda x'.f[], \lambda x.f[\lambda y.y]]$ (and others). Let $\mathcal{M}$ be a set of resource terms; we write $\mathrm{NF}(\mathcal{M})$ for the set of normal forms of elements of $\mathcal{M}$. We write $[\![\Gamma \vdash M : A]\!]^{\mathcal{G}}$ for the game semantics of a simply-typed term-in-context $\Gamma \vdash M : A$. As an application of Theorem 6, we have the following result [78].

Theorem 9. *For every η-long nondeterministic λ-term-in-context $\Gamma \vdash M : A$, we have $\varpi([\![\Gamma \vdash M : A]\!]^{\mathcal{G}}) = \mathrm{NF}(M^*)$.*

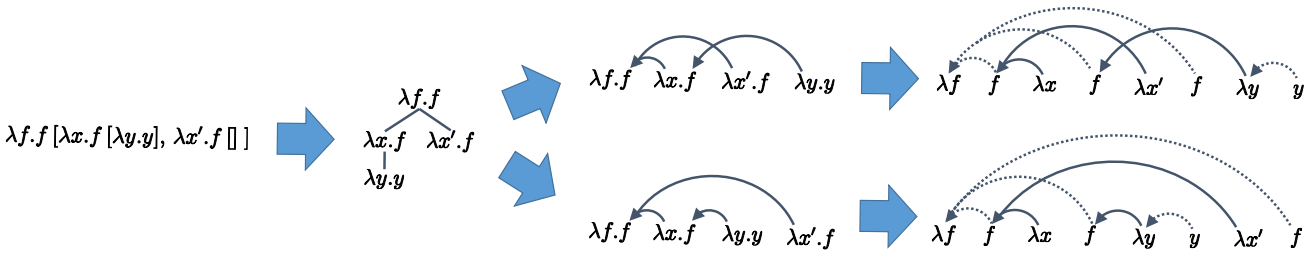


Fig. 3. Idea of the correspondence

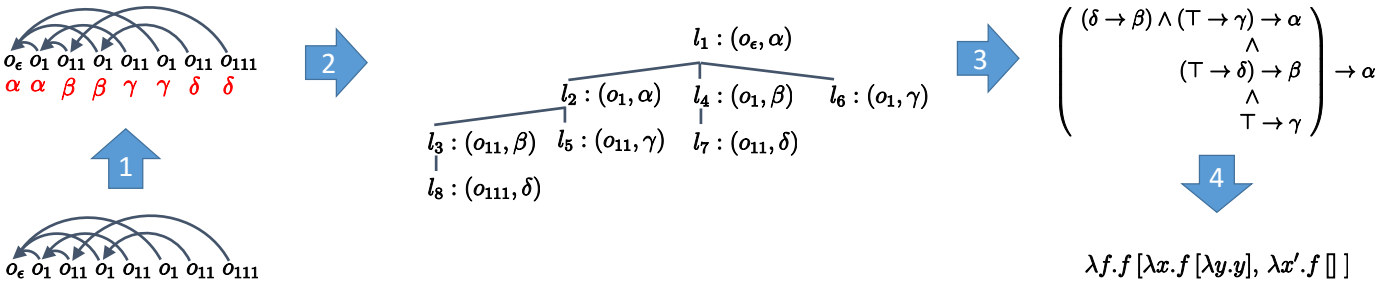


Fig. 4. Illustration of the correspondence (where the simple type, or the arena, is $A = ((\mathsf{o}_{111} \to \mathsf{o}_{11}) \to \mathsf{o}_1) \to \mathsf{o}_\epsilon$.)

III. Weighted relational models

In this section, we generalise the relational model with weights. The category of weighted relations over a complete commutative semiring $\mathcal{R}$ (equivalently, free $\mathcal{R}$-semimodules and linear functions) was introduced as a model of linear logic by Lamarche [56]. This model was further developed by Laird et al. (see [51, 55] among other) and its computational properties analysed via a semantics of $\mathcal{R}$-weighted PCF.

A. Lafont category: a model of intuitionistic linear logic

We begin with a categorical description of models of linear logic as studied in Lafont's PhD thesis [50]. There are more general definitions, but Lafont's simple formulation suits our purpose. For a systematic analysis of categorical semantics of linear logic, see Melliés' monograph [60]. Also relevant is Ehrhard's account [26] from a *differential* linear logic perspective.

Recall that an object A of a symmetric monoidal category (SMC) $(\mathcal{C}, \otimes, 1)$ is a *commutative comonoid* if it is equipped with a *multiplication* morphism $\mathsf{c} : A \to A \otimes A$, and a *unit* morphism $\mathsf{w} : A \to 1$, such that the usual commutativity, associativity, and unit diagrams commute. A *comonoid morphism* f from $(A, \mathsf{c}, \mathsf{w})$ to $(A', \mathsf{c}', \mathsf{w}')$ is given by a morphism $f \in \mathcal{C}(A, A')$ satisfying $f;\mathsf{c}' = \mathsf{c};(\varphi \otimes \varphi)$, and $f;\mathsf{w}' = \mathsf{w}$. Given a SMC $(\mathcal{C}, \otimes, 1)$, the category $\mathbf{Comon}(\mathcal{C})$ has commutative comonoids of $\mathcal{C}$ as objects, and comonoid morphisms as morphisms.

Definition 10 (Lafont category)**.** A symmetric monoidal closed category $(\mathcal{C}, \otimes, \multimap, 1)$ is a *Lafont category* just if

(i) $\mathcal{C}$ has finite products: the SMC $(\mathcal{C}, \times, \top)$ is cartesian,

(ii) $\mathcal{C}$ has *free linear exponentials*: the forgetful functor U from $\mathbf{Comon}(\mathcal{C})$ to $\mathcal{C}$ has a right adjoint !.

Condition (ii) above says $\mathcal{C}$ *has all free commutative comonoids*. That is to say, for every A, there is an object $!A$—the free commutative comonoid generated by A—endowed with a commutative comonoid structure

$$\mathsf{ctr}_A : !A \to !A \otimes !A \qquad \mathsf{wk}_A : !A \to 1$$

and a morphism $\mathsf{der}_A : !A \to A$ satisfying the universal property: for every commutative comonoid B, and every morphism $f : B \to A$, there is a unique comonoid morphism $f^\dagger : B \to !A$ satisfying $f^\dagger ; \mathsf{der}_A = f$. The multiplication (called *contraction*) and unit (call *weakening*) morphisms of $!A$ are named after the standard structural rules in proof theory; and der_A is called *dereliction* in the language of linear logic.

We describe the comonad $(!, \mathsf{der}, \mathsf{dig})$ induced by the (monoidal) adjunction $U \dashv !$ in full. The endofunctor ! maps a morphism $f : A \to B$ to $(\mathsf{der}_A ; f)^\dagger : !A \to !B$. The multiplication morphism, called *digging*, is $\mathsf{dig}_A = (\mathsf{id}_{!A})^\dagger : !A \to !!A$. The unit morphism is just dereliction $\mathsf{der}_A : !A \to A$. We call the induced comonad the *free linear exponential* of the Lafont category. Furthermore, setting the mediating morphisms

$$\mathsf{m}^0_\top : 1 \to !\top$$
$$\mathsf{m}^2_{A,B} : !A \otimes !B \to !(A \times B)$$

as $\mathsf{m}^0_\top = (\top_1)^\dagger$ and $\mathsf{m}^2_{A,B} = \langle \mathsf{der}_A \otimes \mathsf{wk}_B, \mathsf{wk}_A \otimes \mathsf{der}_B \rangle^\dagger$, which are natural isomorphisms, makes

$$(!, \mathsf{m}^0, \mathsf{m}^2) \ : \ (\mathcal{C}, \otimes, 1) \to (\mathcal{C}, \times, \top)$$

a strong symmetric monoidal functor. The natural isomorphisms m^0 and m^2 are called *Seely isomorphisms* [39].

The Kleisli category $\mathcal{C}_!$ over the comonad $(!, \mathsf{der}, \mathsf{dig})$ is automatically cartesian closed. The products in $\mathcal{C}$ give the products in $\mathcal{C}_!$, and the intuitionistic function space $A \Rightarrow B$ is $!A \multimap B$, which is often called the *Girard translation* [39]. The existence of an adjunction $(\cdot) \times A \dashv A \Rightarrow (\cdot)$ is a consequence of the symmetric monoidal closure of $\mathcal{C}$ and the Seely isomorphisms:

$$\begin{aligned}\mathcal{C}_!(C \times A, B) &= \mathcal{C}(!(C \times A), B)\\ &\cong \mathcal{C}(!C \otimes !A, B)\\ &\cong \mathcal{C}(!C, !A \multimap B) = \mathcal{C}_!(C, !A \multimap B)\end{aligned}$$

Remark 11. The linear exponentials of a Lafont category are free constructions, by definition. The relational and weighted relational models considered in Sections II and III are Lafont categories, but there are models of linear logic where several linear exponential modalities may coexist (for example, coherence and hypercoherence spaces models [38, 58], game models [58], and the bicategory of profunctors [16]). To account for this more general setting, Melliés has proposed *new-Lafont category* [60], which is a symmetric monoidal closed category $\mathcal{C}$ with finite products such that the (restriction to $\mathbf{M}$) forgetful functor $U : \mathbf{M} \to \mathcal{C}$ has a right adjoint, where $\mathbf{M}$ is a full subcategory of $\mathbf{Comon}(\mathcal{C})$ closed under tensor and containing the unit comonoid 1.

B. Constructing free linear exponentials from SMC

We discuss a folklore result: a symmetric monoidal category $\mathcal{C}$ has free linear exponentials if it has countable distributive biproducts[2], and all *symmetric* tensor powers. In the following we write $A^{\otimes n}$ for the *nth tensor power* of A, i.e., $A^{\otimes 0} := 1$ and $A^{\otimes (n+1)} := A^{\otimes n} \otimes A$.

Definition 12 (Lafont linear exponential)**.** A family of objects $\{A^{[n]} \mid n \in \omega\}$ of a symmetric monoidal category are the *symmetric tensor powers* of A if

(i) for each $n \geq 0$, the set of $n!$ permutation automorphisms on $A^{\otimes n}$ has an equaliser $\mathsf{eq}_n : A^{[n]} \to A^{\otimes n}$, and
(ii) these equalisers are preserved by the tensor, i.e., for all $m, n \geq 0$

$$\mathsf{eq}_m \otimes \mathsf{eq}_n : A^{[m]} \otimes A^{[n]} \to A^{\otimes m} \otimes A^{\otimes n}$$

is an equaliser for the tensor product of pairs of permutation automorphisms.

We say that an object $!A$ of a SMC with biproducts is the *Lafont linear exponential* of A if it is the biproduct of all the symmetric tensor powers of A, i.e., $!A = \bigoplus_{n \in \omega} A^{[n]}$.

The Lafont linear exponential $!A$ is endowed with a commutative comonoid structure with weakening and contraction morphisms as follows

$$\begin{aligned}\mathsf{wk}_A &= \pi_0 : !A \to 1\\ \mathsf{ctr}_A &= \langle \pi_{m+n} \,;\, \sigma_{m,n} \mid m, n \in \omega \rangle : !A \to !A \otimes !A\end{aligned}$$

[2]Given a SMC with countable biproducts, we say that the tensor *distributes* over biproducts if $(\bigoplus_{i \in I} A_i) \otimes B = \bigoplus_{i \in I} (A_i \otimes B)$ for every countable indexing set I.

where $\sigma_{m,n} : A^{[m+n]} \to A^{[m]} \otimes A^{[n]}$ is the unique morphism satisfying $\mathsf{eq}_{m+n} = \sigma_{m,n} \,;\, (\mathsf{eq}_m \otimes \mathsf{eq}_n)$, and $\pi_n : !A \to A^{[n]}$ is the nth projection.

Proposition 13 ([61])**.** *Let $\mathcal{C}$ be a SMC with countable distributive biproducts. If $\mathcal{C}$ has Lafont linear exponentials, then they are the free linear exponentials.*

C. Constructing Lafont categories from complete semirings

Consider $\mathbf{Rel}$, a simple example of Lafont category. Morphisms from A to B may be viewed as $(A \times B)$-matrices over the Boolean semiring $\mathcal{B}$, and the composite of $s \in \mathbf{Rel}(A, B)$ and $t \in \mathbf{Rel}(B, C)$ is just matrix multiplication: the (a, c)-entry of the composite $(s \,;\, t)$ is the B-indexed sum

$$(s \,;\, t)\,(a, c) := \bigvee_{b \in B} s(a, b) \wedge t(b, c)$$

using the sum (disjunction) and multiplication (conjunction) of the semiring $\mathcal{B}$. A natural way to generalise this construction is to consider as morphisms matrices over semirings that are closed under set-indexed sums. This brings us to the notion of *complete commutative semiring*, and the *free set-indexed biproduct completion* of such a semiring (*qua* one-object category), which gives rise to a Lafont category [55]. The simplest example of the construction is of course $\mathbf{Rel}$, which is itself the free biproduct completion of the Boolean semiring $\mathcal{B}$.

Recall that a (countably) *complete monoid* is a pair $(S, \sum)$ of a set S with a sum operation $\sum$ on (countably) indexed families of elements of S, satisfying

(i) *Partition associativity*: for every partitioning of the indexing set I into $\{I_j \mid j \in J\}$, $\sum_{i \in I} a_i = \sum_{j \in J} \sum_{i \in I_j} a_i$
(ii) *Unary sum*: $\sum_{i \in \{j\}} a_i = a_j$.

We write $\mathbf{0}$ for the sum of the empty family, which is the neutral element of the sum. Every complete monoid is a commutative monoid in the usual sense, with binary sum $a_1 + a_2 := \sum_{i \in \{1,2\}} a_i$.

A *complete semiring* is a tuple $\mathcal{R} = (|\mathcal{R}|, \sum, \cdot, 1)$ where $(|\mathcal{R}|, \sum)$ is a complete monoid, and $(|\mathcal{R}|, \cdot, 1)$ is a monoid which distributes over $\sum$, i.e., $\sum_{i \in I} (a \cdot b_i) = a \cdot \sum_{i \in I} b_i$ and $\sum_{i \in I} (b_i \cdot a) = (\sum_{i \in I} b_i) \cdot a$. We say that $\mathcal{R}$ is *commutative* if $(|\mathcal{R}|, \cdot, 1)$ is a commutative monoid.

Free biproduct completion $\mathcal{R}^\Pi$ of a complete commutative semiring $\mathcal{R}$: Fix a complete commutative semiring $\mathcal{R} = (|\mathcal{R}|, \sum, \cdot, \mathbf{1})$. For convenience we use the *Kronecker notation* δ: given a set A and $a, a' \in A$, define

$$\delta(a, a') := \begin{cases} \mathbf{1} & \text{if } a = a' \\ \mathbf{0} & \text{otherwise} \end{cases}$$

We define the category $\mathcal{R}^\Pi$. The objects of $\mathcal{R}^\Pi$ are sets, and the morphisms from A to B are the $(A \times B)$-matrices over $\mathcal{R}$, i.e., elements of $|\mathcal{R}|^{A \times B}$. The identity over A is the diagonal matrix, $\mathsf{id}_A(a, a') := \delta(a, a')$, for all $a, a' \in A$. The

composite of $s \in \mathcal{R}^\Pi(A,B)$ and $t \in \mathcal{R}^\Pi(B,C)$ is the usual matrix multiplication: for $a \in A$ and $c \in C$

$$(s\,;t)\,(a,c) \;:=\; \sum_{b\in B} s(a,b)\cdot t(b,c).$$

The category $\mathcal{R}^\Pi$ is $*$-autonomous (actually compact closed). The bifunctor $\otimes : \mathcal{R}^\Pi \times \mathcal{R}^\Pi \to \mathcal{R}^\Pi$ acts like the cartesian product: $A \otimes B = A \times B$, and for $s \in \mathcal{R}^\Pi(A,C)$ and $t \in \mathcal{R}^\Pi(B,D)$, $s \otimes t : A \otimes B \to C \otimes D$ is defined as

$$(s \otimes t)((a,b),(c,d)) \;:=\; s(a,c)\cdot t(b,d)$$

The bifunctoriality of tensor follows from the commutativity of $\mathcal{R}$. The unit of the tensor is $1 = \{*\}$, an arbitrary singleton set. $\mathcal{R}^\Pi$ is symmetric monoidal closed: $A \multimap B = A \times B$; and $\mathsf{ev}_{A,B}(((a,b),a'),b') := \delta((a,b),(a',b'))$, and the exponential transpose is defined as $\lambda(s)(c,(a,b)) := s((c,a),b)$ for $s \in \mathcal{C}(C \otimes A, B)$. The object $\bot = 1$ is dualising; it follows that $\mathcal{R}^\Pi$ is compact closed.

By construction $\mathcal{R}^\Pi$ has products given as disjoint union. Since $\mathcal{R}^\Pi$ is compact closed, products are automatically biproducts [43], which we write as $\oplus$. Given a set I of indices, $\bigoplus_{i\in I} A_i := \bigcup_{i\in I}\{i\} \times A_i$, and

$$\pi_j((i,a),a') := \iota_j(a,(i,a)) := \delta((i,a),(j,a'))$$

where π_j is the projection onto A_j, and ι_j the injection from A_j. The terminal object $\top = \emptyset$.

Since $\mathcal{R}^\Pi$ is symmetric monoidal closed and has countable biproducts, it follows from Section III-B that $\mathcal{R}^\Pi$ has free linear exponentials given by the Lafont exponentials. Let us spell out the symmetric tensor powers. Given an object A and $n \in \omega$, $\mathsf{eq}_{A,n} : \mathcal{M}_n(A) \to A^{\otimes n}$ is the equaliser of the $n!$ permutation automorphisms of $A^{\otimes n}$, where $\mathcal{M}_n(A)$ is the set of multisets over A of cardinality n, and $\mathsf{eq}_{A,n}(m,(a_1,\ldots,a_n)) := \delta(m,[a_1,\ldots,a_n])$. These equalisers are preserved by tensor product. We construct the Lafont exponentials accordingly:

$$\begin{aligned} !A &:= \bigoplus_{n\in\omega} \mathcal{M}_n(A) \cong \mathcal{M}_{\mathrm{fin}}(A);\\ \mathsf{der}_A(m,a) &:= \delta(m,[a])\\ \mathsf{ctr}_A(m,(m_1,m_2)) &:= \delta(m,m_1+m_2)\\ \mathsf{wk}_A(m,*) &:= \delta(m,[]). \end{aligned}$$

To summarise:

Proposition 14. *Given a complete commutative semiring $\mathcal{R}$, the free biproduct completion $\mathcal{R}^\Pi$ is a Lafont category, such that each homset is endowed with the structure of a semimodule[3] over $\mathcal{R}$.*

D. Lafont categories from continuous semirings

Continuous semirings [22, 41] are an important class of complete semirings. A *continuous semiring* $\mathcal{R} = (|\mathcal{R}|, +, \cdot, \mathbf{0}, \mathbf{1}, \leq)$ is a semiring equipped with a partial order $\leq$ such that $(|\mathcal{R}|, \leq)$ is a directed-complete partial order (CPO) with $\mathbf{0}$ as the least element, and the operators $+$ and $\cdot$ are continuous. Thanks to directed completeness, we can define *I-indexed sum* over $\mathcal{R}$ as

$$\sum_{r\in I} r \;:=\; \bigvee_{F\subseteq_{\mathrm{fin}} I} \Big(\sum_{r\in F} r\Big)$$

for any subset $I \subseteq |\mathcal{R}|$. It follows that every continuous semiring has a top element ∞, and $p + \infty = \infty$ for all $p \in |\mathcal{R}|$.

Examples: The following semirings, endowed with their natural ordering, are continuous.

1) Boolean: $\mathcal{B} = (\{\mathbf{t},\mathbf{f}\}, \vee, \wedge, \mathbf{f}, \mathbf{t}, \leq)$ where $\mathbf{f} < \mathbf{t}$.
2) Completed numbers: $\mathcal{N} = (\mathbb{N} \cup \{\infty\}, +, \cdot, 0, 1, \leq)$.
3) Completed reals: $\mathcal{P} = (\mathbb{R}^+ \cup \{\infty\}, +, \cdot, 0, 1, \leq)$.

A *continuous semimodule* $(\mathcal{M}, +, \mathbf{0})$ over a continuous semiring $\mathcal{R}$ is a semimodule over $\mathcal{R}$ with a CPO structure such that $\mathbf{0}$ is the least element, and addition and scalar multiplication are continuous.

Proposition 15 ([55])**.** *Given a continuous commutative semiring $\mathcal{R}$, the Kleisli category $\mathcal{R}^\Pi_!$ is a cartesian closed category such that each homset is endowed with the structure of a continuous semimodule over $\mathcal{R}$, and composition is continuous.*

A Lafont category with countable biproducts has a canonical enrichment over the category of countably complete monoids (see e.g. [51]), but it may not be CPO-enriched. However, Laird showed [51] that by constructing a *bifree algebra* (i.e. an initial algebra for which the inverse is a terminal coalgebra) for the free exponential, one can construct fixpoints of morphisms of the Kleisli category without assuming any order-theoretic structure. This is in essence an application of the observation (due to Freyd [35], and Simpson and Plotkin [74]) that uniform fixpoint operators exist (and are unique) for any comonad which is an *algebraically compact functor* [3]. These fixpoints correspond to infinite sums of finitary approximations indexed over nested finite multisets, each representing a unique call-pattern for computation of the fixpoint.

E. Applications to quantitative program analysis

By choosing appropriate continuous commutative semirings $\mathcal{R}$, the Kleisli category $\mathcal{R}^\Pi_!$ can be used to model interesting quantitative operational properties of higher-order computation [55]. Consider the nondeterministic functional language PCF$^{\mathrm{or}}$, which is PCF [70] augmented with nondeterministic branching. The interpretation of PCF$^{\mathrm{or}}$-terms in the CPO-enriched cartesian closed category $\mathcal{R}^\Pi_!$, written $[\![\Gamma \vdash M : A]\!]^{\mathcal{R}}$, is standard. The base type $\mathbf{int}$ is interpreted as the set of natural numbers. The semantics of the PCF-terms is determined by the cartesian closed structure of $\mathcal{R}^\Pi_!$, interpreting the fixpoint operator $\mathbf{Y}_A$ by the least fixpoint construction. For the branching term, we set $[\![\Gamma \vdash M \;\mathtt{or}\; N]\!]^{\mathcal{R}} := [\![\Gamma \vdash M]\!]^{\mathcal{R}} + [\![\Gamma \vdash N]\!]^{\mathcal{R}}$, where $+$ is the sum operation of the corresponding homset.

Theorem 16 ([55])**.** *For all PCF$^{\mathrm{or}}$-program (closed term of type $\mathbf{int}$) M and all $n \geq 0$, $[\![\vdash M]\!]^{\mathcal{N}}(*,n)$ gives the number of reduction sequences from M to the numeral n.*

[3] A *semimodule* over a semiring is like a module over a ring except that it is only required to be a commutative monoid (rather than an abelian group).

Laird et al [55] showed that, by using the appropriate continuous commutative semiring, the weighted relational model can be used to reason about such quantitative properties of PCF$^{\text{or}}$ as may- and must-convergence, and compute such quantities as the probability of convergence, and the minimum and maximum number of reduction steps to convergence.

Weighted relations have shown themselves to be a versatile model for the quantitative analysis of computation. In recent work [52], Laird has given a $\mathcal{R}$-weighted relational model of the *solos calculus* [57], which presents an elegantly economical syntax for describing name mobility in distributed systems. The semantics of a solos term is a matrix over a complete semiring $\mathcal{R}$; it corresponds to the sum in $\mathcal{R}$ of the values associated to the independent reduction paths of the term. The semantics is fully abstract with respect to the sum-of-path evaluation.

Interestingly Laird [53] shows that there is a systematic way to construct accurate quantitative models by transformation from intensional qualitative models (such as games), using the technique of *change of base of an enriched category* [17] This transformation is induced by a monoidal functor from the category of coherence spaces to the category of modules over a complete semiring. Applying this transformation to the game semantics [2] of Idealized Algol (which bears a natural enrichment over the category of coherence spaces), one obtains a $\mathcal{R}$-weighted relational model which is fully abstract.

IV. Profunctors and generalised species of structures

This section is about the second generalisation of the relational model, in a 2-categorical direction:

Relational Model	2-categorical Generalisation
Category **Set**	2-category **Cat**
Category **Rel**	Bicategory **Prof**
Comonad $\mathcal{M}_{\text{fin}}$	Pseudo-comonad $\mathbb{P}$
Category **MRel**	Bicategory **ESP**

We introduce the bicategory **Prof** of profunctors, which corresponds to **Rel**; and the cartesian closed bicategory **ESP** of *generalised species of structures*, which corresponds to **MRel**. We then discuss a recent development [76] of the generalised species of structures as a quantitative model of the nondeterministic $\lambda\mathbf{Y}$-calculus.

A. The bicategory of profunctors

The relational model may be generalised in another way, categorically, to a 2-dimensional level. Given small categories $\mathcal{A}$ and $\mathcal{B}$, a *profunctor* (or *distributors*; see e.g. [4, 5] and [8, § 7.7]) from $\mathcal{A}$ to $\mathcal{B}$ is a functor $f : \mathcal{A} \times \mathcal{B}^{\text{op}} \to \mathbf{Set}$, which is written $f : \mathcal{A} \nrightarrow \mathcal{B}$. Profunctors are a categorical generalisation of relations: a relation $R \subseteq A \times B$ can be seen as a profunctor f_R between sets (*qua* discrete categories) such that $f_R(a, b)$ is either a singleton set or the empty set, depending on whether $(a, b) \in R$. One should view $f : \mathcal{A} \nrightarrow \mathcal{B}$ as a set-valued relation between the categories $\mathcal{A}$ and $\mathcal{B}$. The bicategory **Prof** has small categories as 0-cells; profunctors as 1-cells, and natural transformations between profunctors as 2-cells. The identities are just the hom-functors. Take profunctors $f : \mathcal{A} \nrightarrow \mathcal{B}$ and $g : \mathcal{B} \nrightarrow \mathcal{C}$. A morphism $\varphi \in \mathcal{B}^{\text{op}}(b', b)$ acts on elements in the set $f(a, b')$ by $y \mapsto f(a, \varphi)(y) \in f(a, b)$ and in $g(b, c)$ by $z \mapsto g(\varphi, c)(z) \in g(b', c)$, which we write $y[\varphi]$ and $\{\varphi\}z$, respectively. The composite $f\,;g$ is defined by

$$(f\,;g)(a, c) := \coprod_{b \in \mathcal{B}} \big(f(a, b) \times g(b, c)\big)/\!\sim \qquad (1)$$

where $\sim$ is the least equivalence relation containing $(y, \{\varphi\}z) \sim (y[\varphi], z)$, for every morphism φ in $\mathcal{B}$. Equivalently the composite profunctor $f\,;g : \mathcal{A} \nrightarrow \mathcal{C}$ can be described by a coend formula

$$(f\,;g)(a, c) = \int^{b} f(a, b) \times g(b, c).$$

The 1-cells of **Prof** are functors $F : \mathcal{A} \times \mathcal{B}^{\text{op}} \to \mathbf{Set}$, and so correspond to functors $\lambda(F) : \mathcal{A} \to \widehat{\mathcal{B}}$, writing $\widehat{\mathcal{B}} = [\mathcal{B}^{\text{op}}, \mathbf{Set}]$ for the presheaf category. Because presheaf categories are free colimit completions, profunctors from $\mathcal{A}$ to $\mathcal{B}$ correspond to colimit-preserving functors between presheaf categories from $\widehat{\mathcal{A}}$ to $\widehat{\mathcal{B}}$. The bicategory **Prof** is equivalent to the 2-category **Cocont** whose 0-cells are small categories, whose 1-cells are colimit-preserving functors between the corresponding presheaf categories, and whose 2-cells are the natural transformations between such functors. The bicategory **Prof** has enough structure to model linear logic; it is what may be called a *compact closed bicategory* [16, 48]. With a suitable choice of pseudo-comonad (and there are several such constructions that satisfy the Seely isomorphisms [16, § 9]), **Prof** can be made into a (degenerate) Seely model [60] of linear logic in an appropriate bicategorical sense.

B. Generalised species of structures

We give a direct description of the Kleisli bicategory of a pseudo-comonad on the bicategory **Prof** of profunctors. First a notation. Let $\mathcal{C}$ be a small category, we write $\mathbb{P}\mathcal{C}$ for the free symmetric strict monoidal category on $\mathcal{C}$, whose objects are finite sequences $\langle c_i\rangle_{i\leq n}$ of objects of $\mathcal{C}$, and morphisms from $\langle c_i\rangle_{i\leq n}$ to $\langle d_i\rangle_{i\leq m}$ are tuples $(\sigma, \langle f_i\rangle_{i\leq n})$ where $\sigma : \{1, \cdots, n\} \to \{1, \cdots, m\}$ is a bijection, and $f_i \in \mathcal{C}(c_i, d_{\sigma(i)})$ for all $i \leq n$.

Given small categories $\mathcal{A}$ and $\mathcal{B}$, a $(\mathcal{A}, \mathcal{B})$-*species of structures* [33, 34] is a functor $\mathbb{P}\mathcal{A} \times \mathcal{B}^{\text{op}} \to \mathbf{Set}$, i.e., a profunctor $\mathbb{P}\mathcal{A} \nrightarrow \mathcal{B}$. This gives rise to a bicategory **ESP** whose 0-cells are small categories, 1-cells are $(\mathcal{A}, \mathcal{B})$-species of structures, and 2-cells are natural transformations. The identity species of structures $\mathsf{id}_{\mathcal{A}} : \mathbb{P}\mathcal{A} \nrightarrow \mathcal{A}$ is defined by $\mathsf{id}_{\mathcal{A}}(\theta, a) := \mathbb{P}\mathcal{A}(\langle a\rangle, \theta)$. Given a $(\mathcal{A}, \mathcal{B})$-species of structures $f : \mathbb{P}\mathcal{A} \nrightarrow \mathcal{B}$, we define its *lifting* $f^\sharp : \mathbb{P}\mathcal{A} \nrightarrow \mathbb{P}\mathcal{B}$ as the following coend:

$$f^\sharp(\theta, \langle b_1, \ldots, b_k\rangle) := \int^{(\theta_i)_{i\in[k]} \in (\mathbb{P}\mathcal{A})^k} \Big(\prod_{i\in[k]} f(\theta_i, b_i)\Big) \times \mathbb{P}\mathcal{A}(\theta_1 \cdots \theta_k, \theta)$$

where $\theta_1 \cdots \theta_k$ is the concatenation of lists. Given $f : \mathbb{P}\mathcal{A} \nrightarrow \mathcal{B}$ and $g : \mathbb{P}\mathcal{B} \nrightarrow \mathcal{C}$, their composite $f\,;g : \mathbb{P}\mathcal{A} \nrightarrow \mathcal{C}$ is defined as the profunctorial composite $f^\sharp\,;g$.

Fiore et al. [33, 34] have shown that the bicategory $\mathbf{ESP}$ is cartesian closed, and can be seen as the Kleisli bicategory of a pseudo-comonad $\underline{\mathbb{P}}$ on the bicategory $\mathbf{Prof}$ of profunctors whose action on 0-cells coincides with that of $\mathbb{P}$.

Generalised species of structures are a generalisation of both Girard's *normal functors* [40], which are functors $X \to \mathbf{Set}$, where the set X is the denotation of a type, and Joyal's *combinatorial species* [46, 47], which are functors $\mathbf{P} \to \mathbf{Set}$, where $\mathbf{P}$ is the category of finite cardinals and bijections. There are obvious connections between between these two types of functors, and Hasegawa [42], amongst others, has investigated the connections. There is however an important difference between the two: the domain of (the power series representation of) a normal functor is any set, by which one can interpret a type; whereas a combinatorial species uses $\mathbf{P}$, which has non-trivial (iso)morphisms. Unifying them, generalised species of structures between small categories have domains with enough variations to interpret types and non-trivial (iso)morphisms.

C. $\mathbf{ESP}$-semantics of nondeterministic $\lambda\mathbf{Y}$-calculus

Profunctors are a good semantic basis for quantitative analysis of higher-order computation. Because the bicategory of profunctors is a proof-relevant refinement of the relational model, we can expect the bicategorical framework of generalised species of structures to support a more intensional and precise quantitative analysis. The use of profunctors as a semantic model of computation goes back to the work of Winskel et al. from the 1990s [15, 16, 79]; their motivation was a need of a domain theory for concurrency with a satisfactory account of bisimulation. Also relevant is Hyland's expansive study [45], generalising domain theory from the relational model to give a range of models based on the bicategory of profunctors. More recently Tsukada and Asada [75] gave a profunctorial reformulation of the HO/N games [44] in which plays are graphs: they constructed a pseudofunctor from the category of HO/N games to the bicategory of profunctors, refining a similar result in [78].

As the bicategory $\mathbf{ESP}$ is cartesian closed, it is a model of the λ-calculus. (In fact, it is a differential λ-(bi)category, and so a model of the differential / resource λ-calculus.) But what kind of a quantitative model is it? What is the quantitative property captured by the $\mathbf{ESP}$-semantics? Targeting the simply-typed *nondeterministic* $\lambda\mathbf{Y}$*-calculus*, $\lambda_{\mathrm{nd}}\mathbf{Y}$, these questions have been investigated by Tsukada et al. [76]. For simplicity, simple types are generated from a unique atomic type o, and (raw) $\lambda_{\mathrm{nd}}\mathbf{Y}$*-terms* are defined by:

$$M, N := x \mid \lambda x^A.M \mid M\,M \mid \mathbf{Y}_A\,M \mid M \oplus_A M$$

where $\mathbf{Y}_A$ is the fixpoint operator of type $(A \to A) \to A$ and $M \oplus_A N$ is the nondeterministic branching where M and N are of type A.

Given an interpretation of the atomic type $[\![\mathsf{o}]\!]$ as the category with one object $\star$ and one morphism, the interpretation of the function types is determined by the cartesian closed structure of $\mathbf{ESP}$ (equivalently, compact closure of $\mathbf{Prof}$ via the Girard translation):

$$[\![!A]\!] := \mathbb{P}[\![A]\!]$$
$$[\![A \to B]\!] := [\![!A \multimap B]\!] = \mathbb{P}[\![A]\!]^{\mathrm{op}} \times [\![B]\!].$$

Observe that types are interpreted as groupoids. Similarly, interpreting the nondeterministic branching as the disjoint union, and the fixpoint operator $\mathbf{Y}_A^{\mathbf{ESP}}$ as the least fixpoint construct, the interpretation of a $\lambda_{\mathrm{nd}}\mathbf{Y}$-term-in-context as a generalised species of structures, $[\![M]\!]_{\mathrm{ESP}} : \mathbb{P}[\![\Gamma]\!]^{\mathrm{op}} \times [\![A]\!] \to \mathbf{Set}$, is determined by the cartesian closed structure of $\mathbf{ESP}$.

D. Rigid resource calculus and rigid Taylor expansion

Tsukada et al. [76] have given a characterisation of the ESP-semantics $[\![M]\!]_{\mathrm{ESP}}$ as the *rigid Taylor expansion* of M, denoted $[\![M]\!]_{\mathrm{rTay}}$. Intuitively $[\![M]\!]_{\mathrm{rTay}}$ is the set of ($\sim$-equivalence classes of) linear approximants of M in the form of *rigid resource terms*, where the relation $\sim$ coincides with the equivalence relation (1) in the definition of profunctor composition.

The *rigid resource calculus* is a variant of the resource calculus [10, 30, 69] in which bags of arguments are replaced by lists, written $\langle u_1, \ldots, u_n \rangle$. In the rigid calculus, a permutation of elements in a bag is distinct from but isomorphic to the original bag. Raw terms of the rigid resource calculus are defined by:

$$t, u := x \mid \lambda\vec{x}.t \mid t\,\mu \mid t \oplus \bullet \mid \bullet \oplus t \qquad \mu := \langle u_1, \ldots, u_n \rangle$$

where $\bullet$ is a place holder for the unused part of branching and $\vec{x}$ is a (possibly empty) sequence of variables. Note that (well-typed) rigid resource terms are *linear* by construction, i.e., each variable appears exactly once. A rigid resource term is designed to describe a reduction sequence of a $\lambda_{\mathrm{nd}}\mathbf{Y}$-term it approximates. For example, $t \oplus \bullet$ means that the left-branch should be chosen here; since the right branch is irrelevant in this case, a rigid resource raw term simply ignores it. In contrast to the standard resource calculus, reduction of the rigid resource calculus is *deterministic*, and this is a significant advantage for the quantitative analysis of $\lambda_{\mathrm{nd}}\mathbf{Y}$-calculus.

Fix an infinite sequence $z_1, z_2, \ldots, z_n, \ldots$ of distinct variables, and write $\vec{z}$ for a prefix of this sequence. Let $x_1 : B_1, \ldots, x_m : B_m \vdash M : A$ be a $\lambda_{\mathrm{nd}}\mathbf{Y}$-term in η-long form. Formally

$$[\![M]\!]_{\mathrm{rTay}}(\vec{b}, a) := \{[t]_\sim \mid \vec{z} \lhd \vec{x} \vdash t \lhd M \text{ and } \vec{z} : \vec{b} \vdash t : a\}$$

for $b_i \in \mathrm{Obj}([\![B_i]\!])$, and $a \in \mathrm{Obj}([\![A]\!])$. The judgement, $\vec{z} : \vec{b} \vdash t : a$, is just type assignment judgement, viewing objects of the groupoid $[\![A]\!]$ as the *rigid nonidempotent intersection types* that refine A. Here *rigid* nonidempotent intersection types are just the standard nonidempotent intersection types except that an intersection is not finite multiset but rather finite sequence of types. The judgement, $\vec{z} \lhd \vec{x} \vdash t \lhd M$, says that the rigid

resource term t (with free variables $\vec{z}$) approximates the $\lambda_{\mathrm{nd}}\mathbf{Y}$-term M (with free variables $\vec{x}$).

As mentioned before, a key result of Tsukada et al. [76] is the coincidence of the ESP-semantics with the rigid Taylor expansion semantics.

Theorem 17. *For every term-in-context $\Gamma \vdash M : A$, we have the following natural isomorphism*

$$[\![M]\!]_{ESP} \cong [\![M]\!]_{rTay} \ : \ \mathbb{P}[\![\Gamma]\!]^{\mathrm{op}} \times [\![A]\!] \to \mathbf{Set}. \qquad (2)$$

Recall that each rigid resource term t that approximates a $\lambda_{\mathrm{nd}}\mathbf{Y}$-term (in the sense of the judgement $\vec{z} \lhd \vec{x} \vdash t \lhd M$) corresponds to a evaluation sequence of M. In fact, there is a one-to-one correspondence between equivalence classes $[t]_\sim$ and evaluation sequences of M. The quantitative import of the coincidence in (2) is a compositional method for computing the set of evaluation sequences of $\lambda_{\mathrm{nd}}\mathbf{Y}$-terms. Note that this is an *intensional* (or proof-relevant) version of the quantitative result for $\mathrm{PCF}^{\mathrm{or}}$ in Section III-E.

E. Reasoning about coefficients of the Taylor expansion

The Taylor expansion of a λ-term [30] is a formal sum of (standard) resource terms with real number coefficients. The properties of its support (i.e. terms with non-zero coefficients) has been well-studied, but the coefficients of the Taylor expansion are difficult to reason about. A fundamental property of the Taylor expansion is the *commutation property*, which states that, for a given class $\mathcal{L}$ of λ-terms, the Böhm tree computation $\mathrm{BT}(\cdot)$ commutes with the Taylor expansion $(\cdot)^*$, i.e., $\mathrm{NF}(M^*) = (\mathrm{BT}(M))^*$, for all $M \in \mathcal{L}$. The commutation property was first established by Ehrhard and Regnier [28, 30] for the untyped (deterministic) λ-calculus; it was claimed to hold for System F with weighted branching in [25]. Subsequently Pagani et al. [68] proved that the commutation property is satisfied by weighted nondeterministic strongly normalizing (and so also nondeterministic System F) terms. Using the rigid Taylor expansion semantics, Tsukada et al. [76] have established the commutation property for $\lambda_{\mathrm{nd}}\mathbf{Y}$, although the coefficients may be infinite.

Another application of the quantitative semantics is the calculation of coefficients by a groupoid of isomorphisms and its action on rigid resource terms [76], in a way reminiscent of the *generating series* of combinatorial species. A resource term can be seen as an orbit of the action, and the coefficient is the ratio of the number of elements in the orbit to the number of all isomorphims. From this point of view, the rigid resource calculus may be seen as a developed form of the combinatorial concepts studied in [30].

Acknowledgements

Much of this work was done in collaboration with Takeshi Tsukada, and also with Kazuyuki Asada. The author is grateful to Takeshi Tsukada and Michele Pagani for corrections and comments which greatly improved the manuscript, and to EPSRC for financial support via grant EP/M023974/1.

References

[1] S. Abramsky, R. Jagadeesan, and P. Malacaria, "Full abstraction for PCF," *Inf. Comput.*, vol. 163, no. 2, pp. 409–470, 2000. I

[2] S. Abramsky and G. McCusker, "Linear, sharing and state: a fully abstract semantics for idealized algol with active expressions," in *Algol-like Languages*, P. O'Hearn and R. Tennent, Eds. Birkhauser, 1997. III-E

[3] M. Barr, "Algebraically compact functors," *Journal of Pure and Applied Algebra*, vol. 82, pp. 211–231, 1993. III-D

[4] J. Bénabou, "Les distributeurs," Tech. Rep. 33, 1973. IV-A

[5] ——, "Distributors at work," 2000, course notes, TU Darmstadt. IV-A

[6] R. Blute, J. R. B. Cockett, and R. A. G. Seely, "Differential categories," *MSCS*, vol. 16, no. 6, pp. 1049–1083, 2006. I, II-A, 8

[7] R. Blute, T. Ehrhard, and C. Tasson, "A convenient differential category," *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, vol. LIII-3, pp. 211–232, 2012. I

[8] F. Borceux, *Handbook of Categorical Algebra I*. CUP, 1994. IV-A

[9] P. Boudes, "Thick Subtrees , Games and Experiments," in *TLCA*, 2009, pp. 65–79. II-D

[10] G. Boudol, P. Curien, and C. Lavatelli, "A semantics for lambda calculi with resources," *MSCS*, vol. 9, no. 4, pp. 437–482, 1999. IV-D

[11] A. Bucciarelli, A. Carraro, T. Ehrhard, and G. Manzonetto, "Full abstraction for the resource lambda calculus with tests, through taylor expansion," *LMCS*, vol. 8, pp. 1–44, Oct. 2012. II-C, II-D

[12] A. Bucciarelli and T. Ehrhard, "On phase semantics and denotational semantics: the exponentials," *Ann. Pure Appl. Logic*, vol. 109, no. 3, pp. 205–241, 2001. II-B

[13] A. Bucciarelli, T. Ehrhard, and G. Manzonetto, "A relational model of a parallel and non-deterministic lambda-calculus," in *LFCS*, 2009, pp. 107–121. I

[14] ——, "Categorical models for simply typed resource calculi," *Electr. Notes Theor. Comput. Sci.*, vol. 265, pp. 213–230, 2010. I, II-A

[15] G. L. Cattani, M. P. Fiore, and G. Winskel, "A theory of recursive domains with applications to concurrency," in *LICS*, 1998, pp. 214–225. IV-C

[16] G. L. Cattani and G. Winskel, "Profunctors, open maps and bisimulation," *MSCS*, vol. 15, no. 3, pp. 553–614, 2005. 11, IV-A, IV-C

[17] G. Cruttwell, "Normed spaces and change of base for enriched categories," Ph.D. dissertation, Dalhousie University, 2008. III-E

[18] V. Danos and T. Ehrhard, "Probabilistic coherence spaces as a model of higher-order probabilistic computation," *Inf. Comput.*, vol. 209, no. 6, pp. 966–991, 2011. I

[19] V. Danos and L. Regnier, "The structure of multiplicatives," *Arch. Math. Log.*, vol. 28, no. 3, pp. 181–203, 1989. 5

[20] D. de Carvalho, "Execution time of λ-terms via denotational semantics and intersection types," *Mathematical Structures in Computer Science*, pp. 1–35, 2017. II-B

[21] P. Di Gianantonio and M. Lenisa, "Innocent Game Semantics via Intersection Type Assignment Systems," in *CSL*, vol. 2013, 2013, pp. 231–247. II-D

[22] M. Droste and W. Kuich, "Semirings and formal power series," in *Handbook of Weighted Automata*, M. Droste, W. Kuich, and H. Volger, Eds. Springer-Verlag, 2009. III-D

[23] T. Ehrhard, "On Köthe sequence spaces and linear logic," *MSCS*, vol. 12, no. 5, pp. 579–623, 2002. I

[24] ——, "Finiteness spaces," *MSCS*, vol. 15, no. 4, pp. 615–646, 2005. I

[25] ——, "A finiteness structure on resource terms," in *LICS*, 2010, pp. 402–410. II-C, IV-E

[26] ——, "An introduction to differential linear logic: proof-nets, models and antiderivatives," *Mathematical Structures in Computer Science*, pp. 1–66, 2017. III-A

[27] T. Ehrhard and L. Regnier, "The differential lambda-calculus," *Theor. Comput. Sci.*, vol. 309, no. 1-3, pp. 1–41, 2003. I, II

[28] ——, "Böhm Trees, Krivine's Machine and the Taylor Expansion of Lambda-Terms," in *CiE*, 2006, pp. 186–197. IV-E

[29] ——, "Differential interaction nets," *Theor. Comput. Sci.*, vol. 364, no. 2, pp. 166–195, 2006. I

[30] ——, "Uniformity and the Taylor expansion of ordinary lambda-terms," *Theor. Comput. Sci.*, vol. 403, no. 2-3, pp. 347–372, 2008. I, II-C, 3, IV-D, IV-E

[31] T. Ehrhard, C. Tasson, and M. Pagani, "Probabilistic coherence spaces are fully abstract for probabilistic PCF," in *POPL*, 2014, pp. 309–320. I

[32] H. Férée, "Game semantics approach to higher-order complexity," *J. Comput. Syst. Sci.*, vol. 87, pp. 1–15, 2017. I

[33] M. P. Fiore, "Mathematical models of computational and combinatorial structures," in *FoSSaCS*, 2005, pp. 25–46. IV-B

[34] M. P. Fiore, N. Gambino, J. M. E. Hyland, and G. Winskel, "The cartesian closed bicategory of generalised species of structures," *J. London Math. Soc.*, vol. 77, pp. 203–220, 2008. IV-B

[35] P. Freyd, "Algebraically complete categories," in *Proceedings of Category Theory, Como 1990*, ser. LNM Vol. 1488. Springer, 1990. III-D

[36] A. Frölicher and A. Kriegl, *Linear Spaces and Differentiation Theory*. Wiley, 1988. I

[37] D. R. Ghica, "Slot games: a quantitative model of computation," in *POPL*, 2005, pp. 85–97. I

[38] J. Girard, "The system F of variable types, fifteen years later," *Theor. Comput. Sci.*, vol. 45, no. 2, pp. 159–192, 1986. I, 11

[39] ——, "Linear logic," *Theor. Comput. Sci.*, vol. 50, pp. 1–102, 1987. I, II, 5, III-A

[40] ——, "Normal functors, power series and λ-calculus," *Ann. Pure Appl. Logic*, vol. 37, no. 2, pp. 129–177, 1988. I, II, IV-B

[41] I. Guessarian, *Algebraic Semantics*, ser. Lecture Notes in Computer Science. Springer, 1981, vol. 99. III-D

[42] R. Hasegawa, "Two applications of analytic functors," *Theor. Comput. Sci.*, vol. 272, no. 1-2, pp. 113–175, 2002. IV-B

[43] R. Houston, "Finite products are biproducts in a compact closed category," *Journal of Pure and Applied Algebra*, vol. 212, no. 2, pp. 394 – 400, 2008. III-C

[44] J. M. E. Hyland and C.-H. L. Ong, "On Full Abstraction for PCF: I, II, and III," *Inf. Comput.*, vol. 163, no. 2, pp. 285–408, 2000. I, 2, II-D, IV-C

[45] J. M. E. Hyland, "Some reasons for generalising domain theory," *MSCS*, vol. 20, no. 2, pp. 239–265, 2010. IV-C

[46] A. Joyal, "Une théorie combinatoire des séries formelles," *Adv. Math.*, vol. 42, p. 182, 1981. IV-B

[47] ——, "Foncteurs analytiques et espèces de structures," in *Combinatoire Énumérative*. Springer, 1986, pp. 126–159. IV-B

[48] G. M. Kelly and M. L. Laplaza, "Coherence for compact closed categories," *Journal of Pure and Applied Algebra*, vol. 19, pp. 193–213, 1980. IV-A

[49] A. J. Kfoury, "A linearization of the lambda-calculus and consequences," *J. Log. Comput.*, vol. 10, no. 3, pp. 411–436, 2000. II-B

[50] Y. Lafont, "Logiques, catégories et machines," Ph.D. dissertation, Université Paris 7, 1988. III-A

[51] J. Laird, "Fixed points in quantitative semantics," in *LICS*, 2016. I, III, III-D

[52] ——, "Weighted relational models for mobility," in *FSCD*, 2016, pp. 24:1–24:15. III-E

[53] ——, "From qualitative to quantitative semantics - by change of base," in *FoSSaCS*, 2017, pp. 36–52. III-E

[54] J. Laird, G. Manzonetto, and G. McCusker, "Constructing differential categories and deconstructing categories of games," *Information and Computation*, vol. 222, pp. 247–264, 2013. 8

[55] J. Laird, G. Manzonetto, G. McCusker, and M. Pagani, "Weighted relational models of typed lambda-calculi," in *LICS*, 2013, pp. 301–310. I, III, III-C, 15, III-E, 16, III-E

[56] F. Lamarche, "Quantitative domains and infinitary algebras," *Theor. Comput. Sci.*, vol. 94, no. 1, pp. 37–62, 1992. I, III

[57] C. Laneve and B. Victor, "Solos in concert," *MSCS*, vol. 13, no. 5, pp. 657–683, 2003. III-E

[58] P.-A. Melliès, "Comparing hierarchies of types in models of linear logic," *Inf. Comput.*, vol. 189, no. 2, pp. 202–234, 2004. 11

[59] ——, "Asynchronous games 2: The true concurrency of innocence," *Theor. Comput. Sci.*, vol. 358, no. 2-3, pp. 200–228, 2006. 6

[60] P.-A. Melliès, *Categorical Semantics of Linear Logic*, ser. Panoramas et Synthèses 27, 2009. III-A, 11, IV-A

[61] P.-A. Melliès, N. Tabareau, and C. Tasson, "An explicit formula for the free exponential modality of linear logic," in *ICALP*, 2009, pp. 247–260. 13

[62] R. Milner, "Fully abstract models of typed *lambda*-calculi," *Theor. Comput. Sci.*, vol. 4, no. 1, pp. 1–22, 1977. I

[63] H. Nickau, "Hereditarily sequential functionals," in *LFCS*, 1994, pp. 253–264. I, 2

[64] C.-H. L. Ong, "On model-checking trees generated by higher-order recursion schemes," in *LICS*, 2006, pp. 81–90. II-D

[65] ——, "Normalisation by traversals," *CoRR*, vol. abs/1511.02629, 2015. II-D

[66] C.-H. L. Ong and T. Tsukada, "Two-level game semantics, intersection types, and recursion schemes," in *ICALP*, 2012, pp. 325–336. II-D, II-D

[67] M. Pagani, P. Selinger, and B. Valiron, "Applying quantitative semantics to higher-order quantum computing," in *POPL*, 2014, pp. 647–658. I

[68] M. Pagani, C. Tasson, and L. Vaux, "Strong normalizability as a finiteness structure via the taylor expansion of λ-terms," in *FoSSaCS*, 2016, pp. 408–423. IV-E

[69] M. Pagani and P. Tranquilli, "Parallel reduction in resource lambda-calculus," in *APLAS*, 2009, pp. 226–242. I, IV-D

[70] G. D. Plotkin, "LCF considered as a programming language," *Theor. Comput. Sci.*, vol. 5, no. 3, pp. 223–255, 1977. I, III-E

[71] D. Sands, "Operational theories of improvement in functional languages (extended abstract)," in *Proceedings of Glasgow Workshop on Functional Programming*, 1991, pp. 298–311. I

[72] ——, "Total correctness by local improvement in the transformation of functional programs," *ACM Trans. Program. Lang. Syst.*, vol. 18, no. 2, pp. 175–234, 1996. I

[73] D. Scott and C. Strachey, "Toward a mathematical semantics for computer languages," in *Symposium on Computers and Automata*, J. Fox, Ed., 1971. I

[74] A. K. Simpson and G. D. Plotkin, "Complete axioms for categorical fixed-point operators," in *LICS*, 2000, pp. 30–41. III-D

[75] T. Tsukada and K. Asada, "Strategies in HO/N games as profunctors," 2016, contributed talk, GaLoP Workshop. IV-C

[76] T. Tsukada, K. Asada, and C.-H. L. Ong, "Generalised species of rigid resource terms," in *LICS*, 2017. IV, IV-C, IV-D, IV-E

[77] T. Tsukada and C.-H. L. Ong, "Nondeterminism in game semantics via sheaves," in *LICS*, 2015, pp. 220–231. 6, II-D

[78] ——, "Plays as resource terms via non-idempotent intersection types," in *LICS*, 2016, pp. 237–246. 2, II-C, II-D, II-D, II-D, II-D, II-E, IV-C

[79] G. Winskel, "A presheaf semantics of value-passing processes," in *CONCUR*, 1996, pp. 98–114. IV-C