Species, Profunctors and Taylor Expansion Weighted by SMCC

A Unified Framework for Modelling Nondeterministic, Probabilistic and Quantum Programs

Takeshi Tsukada
The University of Tokyo

Kazuyuki Asada
The University of Tokyo

C.-H. Luke Ong
University of Oxford

Abstract

Motivated by a tight connection between Joyal's combinatorial species and quantitative models of linear logic, this paper introduces *weighted generalised species* (or *weighted profunctors*), where weights are morphisms of a given symmetric monoidal closed category (SMCC). For each SMCC $\mathcal{W}$, we show that the category of $\mathcal{W}$-weighted profunctors is a Lafont category, a categorical model of linear logic with exponential. As a model of programming languages, the construction of this paper gives a unified framework that induces adequate models of nondeterministic, probabilistic, algebraic and quantum programming languages by an appropriate choice of the weight SMCC.

CCS Concepts •**Theory of computation** →*Linear logic; Denotational semantics;*

Keywords quantitative model, quantum computation, generalised species, weighted species, rigid resource calculus

ACM Reference format:
Takeshi Tsukada, Kazuyuki Asada, and C.-H. Luke Ong. 2018. Species, Profunctors and Taylor Expansion Weighted by SMCC. In *Proceedings of LICS '18: 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, Oxford, United Kingdom, July 9–12, 2018 (LICS '18),* 25 pages.
DOI: 10.1145/3209108.3209157

1 Introduction

Semantics of programming languages with branching constructs such as nondeterministic, probabilistic, algebraic and quantum programming languages (e.g. [8, 11, 18, 36, 43]) is an important area of current interest. The aim of this paper is to give a unified framework for modelling these languages.

This paper is, of cause, not the first work that addresses this problem. Among others, several models (e.g. [8, 11, 28, 36]) have been constructed using the techniques of quantitative models of linear logic. For example, the probabilistic coherence space model [8, 11] is a fully abstract model for probabilistic PCF; the weighted relational model [28] gives a unified account of nondeterministic, probabilistic and algebraic programs; and Pagani et al. [36] give a model of higher-order quantum programs.

This paper proposes a general model construction of which [28] and [36] are instances in a certain sense. A notable conceptual difference is that, building on [12, 13, 42], our construction is a cross-fertilization between combinatorial species and (quantitative) models of λ-calculus.

LICS '18, Oxford, United Kingdom

DOI: 10.1145/3209108.3209157

1.1 Why combinatorics matters?

Let us first explain why combinatorics ideas would be useful at the intuitive level. We shall see that a combinatorics problem naturally arises in the operational semantics of programs.

Consider, for example, a programming language with probabilistic branching.

Given a closed term P of the unit type, the probability of convergence is usually defined as follows. First we define the set $Eval(P)$ of reduction sequences $\pi : P \longrightarrow^* \texttt{()}$, where $\texttt{()}$ is the unique value of the unit type and π is the name of this reduction sequence. Because of the branching construct, P may have many reduction sequences. Each reduction sequence $\pi \in Eval(P)$ is associated with a real number $\varpi(\pi)$ between 0 and 1, called its *probability* or *weight*. Hence $Eval(P)$ is not merely a set but a *weighted set*. Then the probability of convergence is defined as the sum $\sum_{\pi \in Eval(P)} \varpi(\pi)$. We aim to apply combinatorics techniques to enumerate the elements of $Eval(P)$ and to compute their weights.

The difference of the branching constructs (i.e. differences of nondetermisitic, probabilistic, algebraic or quantum programs) is understood as the difference of the domains of weights. For example, for a nondeterministic program, a weight is an element of the two-valued Boolean algebra; the weight function is defined by $\varpi(\pi) = \texttt{true}$ for every reduction sequence π and the sum is the disjunction; then $\sum_{\pi : M \longrightarrow^* V} \varpi(\pi) = \texttt{true}$ if and only if $\exists \pi . \pi : M \longrightarrow^* V$. This framework applies also to quantum programs as we shall see.

1.2 Two extensions of Joyal's combinatorial species

The combinatorics tool that we employ for computing $Eval(P)$ is based on Joyal's *combinatorial species* [22] (see also a textbook [6]), which is a functor $F : \mathbf{P} \longrightarrow \mathbf{Set}$ from the category $\mathbf{P}$ of finite cardinals and bijections. This notion is, indeed, closely related to Girard's *normal functor semantics* [15], pioneering work on quantitative models (see, e.g., [19] for the relationship). To the purpose of this paper, we need its weighted and higher-order extension: the weight is used to handle weights $\varpi(\pi)$ of $\pi \in Eval(P)$ and higher-order feature is used to deal with higher-order constructs of programs.

There have been extensions in each direction.

Given a set W of weights, a *W-weighted species* (see, e.g., a textbook [6]) is a species $F : \mathbf{P} \longrightarrow \mathbf{Set}$ together with a family of functions $\varpi_n : F(n) \longrightarrow W$ that respect the action of permutation. Many ideas and operations for species can be naturally extended to weighted species. Usually W is assumed to have an algebraic structure such as ring; we shall discuss below an appropriate algebraic structure for W in our setting.

Generalised species [12, 13] is a higher-order extension: Joyal's species can be seen as generalised species of type $I \longrightarrow I$ (where I is the unit type). Formally it is a profunctor $F : !\mathcal{A} \nrightarrow \mathcal{B}$ (i.e. a functor $F : \mathcal{B}^{\mathrm{op}} \times !\mathcal{A} \longrightarrow \mathbf{Set}$), where ! is a linear exponential comonad on the bicategory $\mathbf{Prof}$ of profunctors. This can be seen as a "proof-relevant version" of the relational model of linear logic.

Our previous work [42] shows that the interpretation of a program P in **Prof** is the set $Eval(P)$ (without weights).

This paper develops a common extension of the two, which we call *weighted generalised species* or *weighted profunctor*.

1.3 Key notion: weighted generalised species

The naïve combination of the above ideas leads us to consider a profunctor $F : \mathcal{A} \nrightarrow \mathcal{B}$ with a family $\varpi_{b,a} : F(b,a) \to W$ of functions parameterised by objects $a \in \mathrm{ob}(\mathcal{A}), b \in \mathrm{ob}(\mathcal{B})$, where W is a fixed set of weights. However this simple notion does not seem to suffice for modelling quantum programs.

This paper considers the situation in which the weight W varies with a and b. The weight is not a set but a category $\mathcal{W}$; an object is not a category but a functor $A : \mathcal{A} \to \mathcal{W}^{\mathrm{op}}$, and a morphism from $A : \mathcal{A} \to \mathcal{W}^{\mathrm{op}}$ to $B : \mathcal{B} \to \mathcal{W}^{\mathrm{op}}$ is what we call a *weighted profunctor* from A to B, which consists of a pair of a profunctor $F : \mathcal{A} \nrightarrow \mathcal{B}$ and a family $\varpi_{b,a} : F(b,a) \to \mathcal{W}^{\mathrm{op}}(B(b), A(a))$. To understand this construction, we note that an element $x \in F(b,a)$ of a profunctor can be seen as a "morphism" from b to a (see, e.g., [5]); then the above construction associates a "morphism" from b to a with a real morphism $\varpi_{b,a}(x) : B(b) \to A(a)$ in $\mathcal{W}^{\mathrm{op}}$ (i.e. a real morphism $A(a) \to B(b)$ in $\mathcal{W}$). Another syntactic exposition based on a *rigid* variant [42] of the *Taylor expansion* [10] will be given in Sections 3 and 4.

The relevance of this construction is justified by the following facts: (1) The resulting category has a good structure, namely, Lafont category with biproducts if the weight category $\mathcal{W}$ is an SMCC. (2) The construction has a concise categorical definition, as (the classifying 1-category of) a full sub-bicategory of the lax slice bicategory $\mathbf{Prof} /\!\!/ \mathcal{W}^{\mathrm{op}}$. (3) The construction gives us an adequate model of a programming language, in which the interpretation of a program has a syntactic counterpart, the *rigid Taylor expansion*, by which a program is interpreted as a collection of its linear approximations.

1.4 Generating series and matrices

Calculation of species and profunctors is often cumbersome. To ease the computation, in the context of weighted species, one can use the *generating series*. Let R be a ring and assume a weighted species $F : \mathbf{P} \to \mathbf{Set}$ with $\varpi_n : F(n) \to R$ such that $F(n)$ is finite for every n. Its *(exponential) generating series* is defined as $\|(F,\varpi)\| = \sum_{n=0}^{\infty} \|(F,\varpi)\|_n \, z^n$, where z is the indeterminant and the coefficient is defined by $\|(F,\varpi)\|_n := (1/n!) \sum_{x \in F(n)} \varpi_n(x)$. Many operations can be carried out in this generating series representation.

Motivated by this idea, this paper develops a concise representation for (a subclass of) weighted profunctors. It is a matrix indexed by objects of $\mathcal{A}$ and $\mathcal{B}$ whose (a,b)-entry is defined by $\|(F,\varpi)\|_{b,a} := (1/\#G) \sum_{x \in F(b,a)} \varpi_{b,a}(x)$ where G is a group describing symmetries of a and b.

This construction is applicable only if the weight category $\mathcal{W}$ has sufficient structure. For example, the sum $\sum_{x \in F(b,a)} \varpi_{b,a}(x)$ of morphisms in $\mathcal{W}(A(a), B(b))$ must be defined in order for the above definition to make sense. We characterise sufficiency of structure in terms of *enriched category theory*, namely, enrichment by Σ*-monoids* (a class of algebras with countable sum): if the SMCC structure of $\mathcal{W}$ is enriched by Σ-monoids and satisfies an additional requirement, then all computation of the Lafont category can be carried out in the matrix representation.

1.5 Contributions

This paper introduces *weighted generalised species* and *weighted profunctors* parametrised by the weight SMCC $\mathcal{W}$. The category $\mathbf{Pr} /\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ of $\mathcal{W}$-weighted profunctors is a model of linear logic (namely a Lafont category with biproducts) and an adequate model of a calculus $\lambda_{\mathcal{W}}$, into which nondeterministic, probabilistic, algebraic and quantum programs can be embedded when $\mathcal{W}$ is appropriately chosen. Assuming additional structures for $\mathcal{W}$, this paper defines a *category of matrices* $\mathbf{Mat}(\mathcal{W})$ over $\mathcal{W}$, which is also a model of linear logic and an adequate model of $\lambda_{\mathcal{W}}$. This construction generalises that of a weighted relational model [28] and of a model of quantum programs by Pagani et al. [36] (see Remark 5.12).

1.6 Related work

The relational model **MRel** [15] is perhaps the prototypical quantitative model of the lambda calculus. In an effort to generalise Girard's *quantitative domains* [15], Lamarche introduced an important extension of the relational model, namely, the category of *weighted relations* over a complete commutative semiring [29]. Characterised as the free biproduct completion of the weight semiring, the weighted relational model was further developed by Laird et al. in a series of papers [26–28]. By an appropriate choice of the weight semiring, these weighted relational models give an adequate semantics of nondeterministic and probabilistic PCF, with scalar weights from the semiring.

For modelling probabilistic PCF, a related semantics, based on *probabilistic coherence spaces* [8], was shown to be fully abstract by Ehrhard et al. [11]. In the latter paper (§5.1), the authors drew a comparison between the probabilistic coherence spaces interpretation and the sum of weights of intersection type derivations. Since linear approximations (of our present paper) can be viewed as derivations in an intersection type system, summation of the weights of all derivations can be related to the generating series (or the matrix representation) of a weighted profunctor (or the rigid Taylor expansion). In this sense, our paper confirms the observation of Ehrhard et al. in [11] from a somewhat more general perspective. A connection [11, Footnote 7, p. 313] between the probabilistic coherence spaces and the combinatorial species interpretation [19, 22, 23] is similarly clarified by our work.

In [36], Pagani et al. applied the free biproduct construction to the known model of completely positive maps to obtain an adequate semantics of an expressive quantum lambda calculus. A notable advance in the denotational semantics of higher-order quantum computation, their model can interpret not just infinitary computation (both infinite data types and recursion), but also general entanglement, a defining feature of quantum computation. In the Conclusion section of the paper [36], the authors observed that their model "demonstrates that the quantum and the classical 'universes' work well together, but also—surprisingly—that they do not mix too much, even at the higher-order types." Our work clarifies this phenomenon mathematically by organising the modelling process into two phases, namely, enumeration and summation. The reason why the model supports a certain clean separation of the two worlds (always yielding "an *infinite* list of *finite*-dimensional CPMs") can be traced to the fact that the category $\mathbf{CPM}_s$ is $\Sigma\mathbf{Mon}$-enriched, and, in particular, to the presence of the element "$1/n$" in the monoid, for every natural number n (see Section 5 for the precise formulation). In fact, in the semantics, different control flows (that we do not need to distinguish) are merged.

The relational model may be generalised in quite a different way, namely, to a 2-dimensional level categorically. As set out by Fiore [12], the conceptual basis for this class of 2-categorical models of higher-order computation lies in combinatorics and its methods. In a follow-up paper [13], Fiore et al. introduced the cartesian closed bicategory of *generalised species of structures*, which generalises both Joyal's combinatorial species [22, 23] and Girard's normal functors semantics [15], and may be viewed as a proof-relevant extension of the relational model. In recent work [42], we introduced *rigid resource calculus*, and showed that the Taylor expansion semantics (within the rigid calculus) of the nondeterministic λY-calculus coincides with the generalised species interpretation.

Building on the correspondence between linear approximations and non-idempotent intersection types, Mazza et al. [31, 37] have recently developed a general 2-operadic framework for deriving systems of intersection types that characterise normalisation properties, based on a **Rel**-valued profunctorial semantics of programs. It would be interesting to clarify how their semantics relates to the generalised species interpretation [13] (or equivalently the rigid Taylor expansion semantics [42]), and to generalise their main result [31, Theorem 4.7] to programs with such branching constructs as nondeterministic and probabilistic choice.

Melliès [32] has analysed the group-theoretic nature of the PER construction in the AJM game model [1]: his *orbital game* is a reformulation of HO-style arena games [21] with justification pointers replaced by thread indexing, modulo certain left and right group actions. A similar idea appears in our Section 5. Symmetry in a similar spirit can also be found in the model of quantum computation by Pagani et al. [36], whose construction requires invariance under certain group actions.

2 A Lambda Calculus with SMCC Data

Assume a symmetric monoidal closed category $(\mathcal{W}, \otimes, \multimap, I)$, which we call the *weight category*. Based on the typed calculus in Pagani et al. [36], this section introduces a lambda calculus $\lambda_{\mathcal{W}}$ parameterised by SMCC $\mathcal{W}$, which has the objects of $\mathcal{W}$ as base types and the morphisms as constants. The standard constructs of the lambda calculus describes "classical" control, whereas constants from $\mathcal{W}$ manipulates "non-classical" data. A goal of Sections 3, 4 and 5 is to give an adequate model of $\lambda_{\mathcal{W}}$.

The calculus $\lambda_{\mathcal{W}}$ is used as a metalanguage, which is not necessarily of practical interest but fits our model well. Its usefulness is demonstrated by embedding calculi of interest into $\lambda_{\mathcal{W}}$ with appropriate $\mathcal{W}$, adequately though not necessarily fully. A key example of $\mathcal{W}$ is the category $\mathbf{CPM}_s$, which is a model of a linear and finite quantum programming language (see [38, 39] for an account of this category as a model of quantum programs). The higher-order quantum calculus of [36] can be embedded into $\lambda_{\mathbf{CPM}_s}$.

For space reason, we omit some rules and definitions; see Appendix A. For brevity, we often treat $\mathcal{W}$ as if it were a strict SMCC.

2.1 Syntax

Figure 1 shows the syntax of the calculus. The type constructors of the calculus are those of intuitionistic linear logic with the list type $\texttt{list}\, S$ and base types a, which are objects of $\mathcal{W}$. The term constructors are the standard ones of a λ-calculus with coproduct types, constructors and a destructor of lists, nondeterministic branching $M \diamond N$, sequential execution $(M; N)$, recursion $\mathbf{Y}\, V$ and constants c^S from $\mathcal{W}$; here either $S = \mathsf{a}_1 \otimes \cdots \otimes \mathsf{a}_n \multimap \mathsf{b}_1 \otimes \cdots \otimes \mathsf{b}_m$ and $c \in \mathcal{W}(\mathsf{a}_1 \otimes \cdots \otimes \mathsf{a}_n, \mathsf{b}_1 \otimes \cdots \otimes \mathsf{b}_m)$, or $S = \mathsf{b}_1 \otimes \cdots \otimes \mathsf{b}_m$ and $c \in \mathcal{W}(I, \mathsf{b}_1 \otimes \cdots \otimes \mathsf{b}_m)$. For technical convenience, the arguments of many constructs are restricted to values. This does not lose generality; for example, the term $\texttt{inl}(M)$ can be written as $\texttt{let}\, x = M \,\texttt{in}\, \texttt{inl}(x)$ for fresh x. We use $M\,V$ as the syntactic sugar of $\texttt{let}\, x = M \,\texttt{in}\, x\,V$ for fresh x. We shall often omit type annotations.

The calculus has a type system based on the dual context linear logic . A judgement has the form $\Delta \mid \Gamma \vdash M : S$, where Δ and Γ are finite sequences of type bindings of the form $x : T$ called *type environments*. The variables in Δ and in Γ are non-linear and linear ones, respectively. The typing rules are standard, some of which are listed in Fig. 2.

2.2 Operational semantics

A *configuration* (typically C, C' etc.) is a triple of sequences $\vec{x} = x_1, \dots, x_n$ of variables and $\vec{\mathsf{a}} = \mathsf{a}_1, \dots, \mathsf{a}_n$ of atomic types, a morphism $e : I \to \mathsf{a}_1 \otimes \cdots \otimes \mathsf{a}_n$ in $\mathcal{W}$ and a term $\mid x_1 : \mathsf{a}_1, \dots, x_n : \mathsf{a}_n \vdash M : I$ (note that x_i is a linear variable). We write such a triple as $[\vec{x} = e, M]$, which intuitively means $\texttt{let}\, \vec{x} = e \,\texttt{in}\, M$.

The set of *evaluation contexts* is defined by the following grammar: $E ::= [\,] \mid E; M \mid \texttt{let}\, x = E \,\texttt{in}\, M$. The *one-step evaluation relation* on configurations is given by the rules in Fig. 3. For a sequence $\pi \in \{0, 1, 2\}^*$, we write $C \xrightarrow{\pi} C'$ if there is a sequence of the form $C = C_0 \xrightarrow{d_1} C_1 \xrightarrow{d_2} \cdots \xrightarrow{d_n} C_n = C'$ and $\pi = d_1\, d_2 \dots d_n$ where $n \geq 0$ (ϵ is the empty sequence and hence the length of π may be less than n).

A *program* P is a closed term of the unit I. We define

$$Eval(P) := \{\, \pi \mid [\epsilon = \mathrm{id}_I, P] \xrightarrow{\pi} [\epsilon = e, ()] \,\}.$$

For $\pi \in Eval(P)$, its *weight* $w(\pi)$ is (necessarily unique) $e \in \mathcal{W}(I, I)$ such that $[\epsilon = \mathrm{id}_I, P] \xrightarrow{\pi} [\epsilon = e, ()]$. Let us call a set X equipped with a function $w : X \to W$ a W*-weighted set* (or simply a *weighted set*). In this terminology $Eval(P)$ is a $\mathcal{W}(I, I)$-weighted set.

In a typical situation, we are not interested in the weighted set $Eval(P)$ itself but its summary. For example, if $\mathcal{W}(I, I)$ has sums, it may be more appropriate to consider the sum $\sum_{\pi \in Eval(P)} \varpi(\pi)$; see the examples in the next subsections.

2.3 Examples

Example 2.1 (Nondeterministic calculus)**.** Let $\mathcal{W}$ be the terminal category 1 consisting of one object I and one morphism (i.e. the identity on I), which has the trivial SMCC structure. The calculus λ_1 has nothing special except for the nondeterministic branching. A closely related variant is given by a category $\mathbf{B}$ consisting of one object I and two morphisms $0, 1 \in \mathbf{B}(I, I)$, regarded as the two-value boolean algebra, with composition given by the meet. *May-convergence of* P is defined as $\bigvee_{\pi \in Eval(P)} \varpi(\pi)$. The calculus λ_1 can be embedded into $\lambda_{\mathbf{B}}$.

Example 2.2 (Probabilistic calculus)**.** If the calculus has a probabilistic branching, each reduction sequence is associated with its *probability*, i.e. a real number p with $0 \leq p \leq 1$. This observation motivates us to consider the weight category $\mathcal{W}_{[0,1]}$ consisting of one object I and $\mathcal{W}(I, I) = [0, 1]$, where $[0, 1] = \{x \in \mathbf{R} \mid 0 < x \leq 1\}$ is the interval of real numbers, with composition defined by the multiplication. In this calculus one can express, for example, the

$$S, T ::= \mathsf{a} \mid S \multimap T \mid I \mid S \otimes T \mid !S \mid S \oplus T \mid \mathtt{list}\, S$$
$$M, N, L ::= x \mid c^S \mid \lambda x^S.M \mid V\,W \mid M \diamond N \mid \mathbf{Y}V \mid () \mid M; N \mid \mathtt{let}\, x = M\, \mathtt{in}\, N \mid !V \mid \mathtt{let}\, !x = V\, \mathtt{in}\, M \mid V \otimes W \mid \mathtt{let}\, x \otimes y = V\, \mathtt{in}\, M$$
$$\mid \mathtt{inl}^{S,T}(V) \mid \mathtt{inr}^{S,T}(V) \mid \mathtt{case}\, V\, \mathtt{of}\, (\mathtt{inl}(x) : N \mid \mathtt{inr}(y) : L) \mid \mathtt{Nil}^S \mid V{::}W \mid \mathtt{case}\, V\, \mathtt{of}\, (\mathtt{Nil} : N \mid x{::}y : L)$$
$$V, W ::= x \mid c \mid \lambda x^A.M \mid V \otimes W \mid !V \mid \mathtt{inl}^{S,T}(V) \mid \mathtt{inr}^{S,T}(V) \mid \mathtt{Nil}^S \mid V{::}W$$

Figure 1. Syntax of types, terms and values (syntactic sugar: $M\,V$ means $\mathtt{let}\, x = M\, \mathtt{in}\, x\, V$ for fresh x)

$$\frac{}{\Delta \mid\ \vdash c^S : S} \quad \frac{\Delta \mid \Gamma, x : S \vdash M : T}{\Delta \mid \Gamma \vdash \lambda x.M : S \multimap T} \quad \frac{\Delta \mid \Gamma \vdash M : T \qquad \Delta \mid \Gamma \vdash N : T}{\Delta \mid \Gamma \vdash M \diamond N : T} \quad \frac{\Delta \mid\ \vdash V : !(S \multimap T) \multimap S \multimap T}{\Delta \mid\ \vdash \mathbf{Y}V : S \multimap T} \quad \frac{\Delta \mid \Gamma_1 \vdash M : I \qquad \Delta \mid \Gamma_2 \vdash N : T}{\Delta \mid \Gamma_1, \Gamma_2 \vdash M; N : T}$$

Figure 2. Simple typing rules (excerpt)

(a) Classical control flow $[\vec{x} = e, E[(\lambda y.M)\,V]] \xrightarrow{0} [\vec{x} = e, E[M\{V/y\}]]$ $\quad [\vec{x} = e, E[M_1 \diamond M_2]] \xrightarrow{i} [\vec{x} = e, E[M_i]]$

$[\vec{x} = e, E[\mathbf{Y}\,V]] \xrightarrow{0} [\vec{x} = e, E[V\,!(\lambda x.\mathbf{Y}\,V\,x)]]$ $\quad [\vec{x} = e, E[\mathtt{case}\,\mathtt{inl}(V)\,\mathtt{of}\,(x : M \mid y : N)]] \xrightarrow{0} [\vec{x} = e, E[M\{V/x\}]]$

(b) "Non-classical" data $[\vec{x}^{\vec{\mathsf{a}}}\vec{y}^{\vec{\mathsf{b}}} = e, E[c^{\vec{\mathsf{a}} \multimap \vec{\mathsf{a}}'}(\vec{x})]] \xrightarrow{0} [\vec{z}^{\vec{\mathsf{a}}'}\vec{y}^{\vec{\mathsf{b}}} = ((c \otimes \mathrm{id}_{\vec{\mathsf{b}}}) \circ e), E[\vec{z}]]$ $\quad [x_1 \dots x_n = e, P] \xrightarrow{\epsilon} [x_{\sigma(1)} \dots x_{\sigma(n)} = \sigma \circ e, P]$

Figure 3. Operational semantics (excerpt). Here σ is a permutation $\sigma \in \mathfrak{S}_n$ of n elements, identified with the structural isomorphism $\mathsf{a}_1 \otimes \cdots \otimes \mathsf{a}_n \longrightarrow \mathsf{a}_{\sigma(1)} \otimes \cdots \otimes \mathsf{a}_{\sigma(n)}$ in $\mathcal{W}$.

probabilistic choice of M and N as $(\frac{1}{2}; M) \diamond (\frac{1}{2}; N)$, where $\frac{1}{2} : I$ is the constant corresponding to $1/2 \in \mathcal{W}(I, I)$. A configuration is (essentially) a pair of $p \in [0, 1]$ and M, and $[1, M] \xrightarrow{\pi} [p, N]$ means that the probability of this reduction sequence is p. The *probability of convergence of* P can be defined by $\sum_{\pi \in Eval(P)} w(\pi)$. If P is really "probabilistic", i.e. it has only nondeterministic branches of the form $(p; M) \diamond (1 - p; N)$, then the sum must converge. Otherwise the sum can be infinite. If we want to ensure that the above sum is always defined, we should replace $[0, 1]$ with $\mathbf{R}^{\infty}_{\geq 0} := \{x \in \mathbf{R} \mid 0 \leq x\} \cup \{\infty\}$ (with $0 \times \infty = 0$).

Example 2.3 (Algebraic calculus)**.** The commutative monoid $([0, 1], \times)$ in the previous example can be replaced with any other commutative monoid. Indeed a category $\mathcal{W}$ with one object I is an SMCC if and only if $\mathcal{W}(I, I)$ is a commutative monoid. Let R be a commutative monoid and $\mathcal{W}_R$ be a category with one object I and $\mathcal{W}_R(I, I) = R$. In $\lambda_{\mathcal{W}_R}$, one can write a sum of terms with coefficients from R, e.g. $(r; M) \diamond (r'; N)$ where $r, r' \in \mathcal{W}_R(I, I) = R$, as in the algebraic lambda calculus [44]. If R has the addition operation (i.e. R is a commutative semiring), one can define the *weight of convergence of* P by $\sum_{\pi \in Eval(P)} w(\pi)$. Here the sum may be undefined since $Eval(P)$ can be a countably infinite set. It is always defined if, for example, R is a *continuous semiring* as in [28].

Example 2.4 (Quantum calculus 1)**.** Let $\mathcal{W} = \mathbf{FdHilb}$ be the category of finite dimensional Hilbert spaces, whose object is a natural number and whose morphism $f : n \longrightarrow m$ is a complex linear function $f : \mathbf{C}^n \longrightarrow \mathbf{C}^m$. This is a compact closed category with tensor product $n \otimes m := n \times m$. The quantum lambda calculus of [36] can be embedded into $\lambda_{\mathbf{FdHilb}}$. The calculus $\lambda_{\mathbf{FdHilb}}$ has the atomic type $\mathtt{qubit} := 2$ and every unitary map U on $\mathtt{qubit}^{\otimes n}$ as constants. The creation $\mathtt{new} : I \oplus I \longrightarrow \mathtt{qubit}$ of new qubit is given by $\mathtt{new} := \lambda x.\mathtt{case}\, x\, \mathtt{of}\, (\mathtt{inl}(y) : (y; |0\rangle) \mid \mathtt{inr}(z) : (z; |1\rangle))$ where $|0\rangle$ and $|1\rangle$ be the standard basis vectors $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ of $\mathtt{qubit}$ regarded as morphisms $I \longrightarrow \mathtt{qubit}$. The measurement $\mathtt{meas} : \mathtt{qubit} \longrightarrow I \oplus I$ can be defined as the nondeterministic branching followed by projections: $\mathtt{meas} := \lambda x.((\langle 0|\, x); \mathtt{inl}(I)) \diamond ((\langle 1|\, x); \mathtt{inr}(I))$ where $\langle 0| = (1\ 0)$ and $\langle 1| = (0\ 1)$ are projections. A (typical) configuration is $[x_1, \dots, x_n = e, M]$ where e is a vector in the Hilbert space of dimension 2^n (i.e. the Hilbert space $\mathtt{qubit}^{\otimes n}$); note that e is not normalised but the length indicates the probability of the reduction, that means, the probability of $[\epsilon = 1, P] \xrightarrow{\pi} [\vec{x} = e, Q]$ is $\|e\|^2$. Hence the *probability of convergence of* P is defined as $\sum_{\pi \in Eval(P)} w(\pi) w(\pi)^*$ where $(-)^*$ is the complex conjugate.

This definition of the probability of convergence gives the same value as in [36]. Actually there exists a bijection between the reduction sequences in [36] and those of our calculus, which maps $[e, |\vec{x}\rangle, M] \xrightarrow{p} [e', |\vec{y}\rangle, M']$ in [36] (where e and e' are normalised vectors and p is the probability of this reduction sequence) to $[\vec{x} = e, M] \longrightarrow [\vec{y} = \sqrt{p}e', M']$ in $\lambda_{\mathbf{FdHilb}}$.

Example 2.5 (Quantum calculus 2)**.** Let $\mathcal{W}$ be the category $\mathbf{CPM}_s$ of completely positive maps, whose object is a natural number and whose morphism $g : n \longrightarrow m$ is a special kind of linear function from $(n \times n)$-matrices to $(m \times m)$-matrices called *completely positive maps* (see, e.g., [38] and [39]). Here we use only the following fact: given a linear function $f : n \longrightarrow m$, the mapping $A \mapsto fAf^*$ (A an $n \times n$-matrix) is completely positive (where $(-)^*$ is the adjoint operator) and thus a morphism $n \longrightarrow m$ in $\mathbf{CPM}_s$. This induces a functor $\mathbf{FdHilb} \longrightarrow \mathbf{CPM}_s$ preserving the compact closed structure , as well as a translation from $\lambda_{\mathbf{FdHilb}}$ to $\lambda_{\mathbf{CPM}_s}$. The quantum calculus of [36] can be embedded into $\lambda_{\mathbf{CPM}_s}$ via this translation. A configuration $[\vec{x} = e, P]$ of $\lambda_{\mathbf{FdHilb}}$ corresponds to $[\vec{x} = ee^*, P]$ of $\lambda_{\mathbf{CPM}_s}$. An advantage of this is that now the *probability of convergence of* P is defined as the standard sum $\sum_{\pi \in Eval(P)} w(\pi)$ in $\mathbf{CPM}_s(I, I) \cong \mathbf{R}_{\geq 0}$. This advantage is significant; see Remark 5.17.

2.4 Categorical interpretation

A $\lambda_{\mathcal{W}}$*-model* is a category equipped with the following structures: (1) a linear-non-linear category [33], (2) finite biproducts $\oplus$, (3) the initial algebra of $L_a(X) = I \oplus (a \otimes X)$ for each object a, and (4) interpretations of base types and constants, including $\mathbf{Y}$ of each type. It is straightforward to give an interpretation of $\lambda_{\mathcal{W}}$-terms in a $\lambda_{\mathcal{W}}$-model (the biproduct induces a canonical commutative-monoid enrichment, by which nondeterministic branching is interpreted). Note that the fixed-point combinator is treated as a constant and thus there is no guarantee that this interpretation is adequate. Adequacy shall be discussed for individual models.

There is an appropriate notion of *$\lambda_{\mathcal{W}}$-model morphisms*, which strongly preserves the above structure. An important property is that a $\lambda_{\mathcal{W}}$-model morphism preserves the interpretation of a program (up to the structural isomorphism).

3 Rigid Taylor Expansion

This section reviews a theory of linear approximations of $\lambda_{\mathcal{W}}$-terms, a variant of the Taylor expansion [10] that we call the *rigid Taylor expansion* [42]. The aim of this section is to give a syntactic justification (or understanding) of the definition of weighted profunctors, which is introduced in the next section. Since most results of this section are an adaptation of our previous work [42], we give only a quick overview; see [42] or Appendix B for details.

3.1 Refinement types

We first introduce *refinement intersection types* (or *refinement types* for short), which properly describe classical control flows of a given term. The syntax of refinement types is shown in Fig. 4(a). It parallels the syntax of simple types: each simple-type constructor has one or two corresponding refinement-type constructors. The intuitive "correspondence" of type constructors is formally defined by the *refinement relation*, which is a binary relation $a \lhd S$ between refinement types and simple types. Some rules are listed in Fig. 4(b).

We comment on some notable points. A refinement of the exponential type $!S$ is a list $\langle a_1, \dots, a_n \rangle$ of refinement types a_i of S. This refinement type should be read as a (non-idempotent) intersection type $a_1 \wedge \cdots \wedge a_n$; a value of this type shall be made n copies, used in accord with $a_1, \dots, a_n$ respectively. A refinement of the sum type $S \oplus T$ is either $a \oplus \bullet$ or $\bullet \oplus b$. A value of type $a \oplus \bullet$ must be of the form $\mathtt{inl}(V)$ and a describes the usage of the value V. A refinement of the list type $\mathtt{list}\, S$ must be of the form $a_1{::}a_2{::}\dots a_n{::}\mathtt{nil}$. It tells us the length of the list as well as the usage of each element. Note that a refinement type of the value V in a case analysis $\mathtt{case}\, V\, \mathtt{of}\, (\cdots)$ tells us the actual branch.

The refinement types $\langle a_1, a_2 \rangle$ and $\langle a_2, a_1 \rangle$ are different but closely related. They both say that the value of these types shall be duplicated, one copy is used as of type a_1 and the other is as of type a_2. This similarity is captured by the notion of *type isomorphisms*. We write $\varphi : a \cong a'$ to mean that refinements a and a' of S are isomorphic, of which φ is a *witness* (or a *proof*). It is defined by fairly straightforward rules, some of which are found in Fig. 4(c). Note that refinement types are isomorphic in more than one way. For example, consider $\langle \mathsf{a}, \mathsf{a} \rangle$, which is isomorphic to itself in two ways; one relates the left component to the left component, and the other relates the left component to the right component.

For each simple type S, the collection of refinement types of S and isomorphisms between them forms a *groupoid*, which means the existence of the following: (1) identity $\mathrm{id}_a : a \cong a$ for every $a \lhd S$, (2) composite $(\psi \circ \varphi) : a \cong c$ for every $\varphi : a \cong b$ and $\psi : b \cong c$ and (3) inverse $\varphi^{-1} : b \cong a$ for every $\varphi : a \cong b$. We write this groupoid as $[\![S]\!]$.

We say that an isomorphism φ is *positive* if, for each negative (i.e. contravariant) occurrence of $\langle \sigma; \psi_1, \dots, \psi_n \rangle$ in φ, we have $\sigma = \mathrm{id}$. It is *negative* if every permutation in a covariant position is the identity. The groupoid $[\![S]\!]$ has a *strict factorisation system*: positive (resp. negative) isomorphisms form a subcategory and each isomorphism φ can be uniquely decomposed as $\varphi = \varphi^+ \circ \varphi^-$ where φ^+ is a positive isomorphism and φ^- is negative. We write $[\![S]\!]^+$ (resp. $[\![S]\!]^-$) as the positive (resp. negative) subcategory, which is a groupoid.

3.2 Refinement typing rules and its term representation

The syntax of *rigid resource raw-terms* is given in Fig. 5. They are used to represent refinement type derivations. It has basically the same term constructors as $\lambda_{\mathcal{W}}$ but three crucial differences: (1) A rigid resource raw-term has only one branch of nondeterministic choice $M \diamond N$ and case analyses $\mathtt{case}\, V\, \mathtt{of}\, (\mathtt{inl}(x) : M \mid \mathtt{inr}(y) : N)$ and $\mathtt{case}\, V\, \mathtt{of}\, (\mathtt{Nil} : M \mid x{::}y : N)$; (2) A rigid resource raw-term has a list $\langle v_1, \dots, v_n \rangle$ instead of the exponential $!V$; and (3) A rigid resource raw-term has no recursion. Thanks to these changes, rigid resource raw-terms have desirable properties: *a rigid resource raw-term has a unique reduction sequence, which must terminate; and it is linear*, i.e. each variable in a resource term is used exactly once.

We define a set of rules relating resource terms and $\lambda_{\mathcal{W}}$-terms. A *refinement non-linear type environment*, ranged over by Θ, is a finite sequence of type bindings of the from $\langle x_1, \dots, x_n \rangle : \langle a_1, \dots, a_n \rangle$. We write O for refinement non-linear type environments consisting of $\langle\rangle : \langle\rangle$. A *refinement linear type environment*, ranged over by Ξ, is a finite sequence of type bindings of the from $x : a$. The *refinement relations* are defined by the following rules

$$\frac{a_i \lhd S\ (\forall i \le n) \quad \Theta \lhd \Delta}{(\langle x_1, \dots, x_n \rangle : \langle a_1, \dots, a_n \rangle, \Theta) \lhd (y : S, \Delta)} \qquad \frac{a \lhd S \quad \Xi \lhd \Gamma}{(x : a, \Xi) \lhd (y : S, \Gamma)}$$

in addition to a rule relating empty environments. Note that we only compare types but not variable names. We write $(\Theta \mid \Xi) \lhd (\Delta \mid \Gamma)$ if $\Theta \lhd \Delta$ and $\Xi \lhd \Gamma$. A *refinement type judgement* is a tuple $\Theta \lhd \Delta \mid \Xi \lhd \Gamma \vdash t : a \lhd M : S$ with $(\Theta \mid \Xi) \lhd (\Delta \mid \Gamma)$ and $a \lhd S$. We omit $\Xi \lhd \Gamma$ (resp. $\Theta \lhd \Delta \mid \Xi \lhd \Gamma$) if both Ξ and Γ (resp. the four environments) are the empty sequence. Figure 6 shows important rules. Here $\wedge$ is the component-wise concatenation, e.g.,

$$\begin{aligned}(\langle x_1, x_2 \rangle : \langle a_1, a_2 \rangle,\ \langle y_1 \rangle : \langle b_1 \rangle) \wedge (\langle\rangle : \langle\rangle,\ \langle z_1, z_2 \rangle : \langle c_1, c_2 \rangle) \\ = (\langle x_1, x_2 \rangle : \langle a_1, a_2 \rangle,\ \langle y_1, z_1, z_2 \rangle : \langle b_1, c_1, c_2 \rangle).\end{aligned}$$

By dropping some components, the rules can be seen as three different typing systems. First, by removing the left-hand-sides of $\lhd$, the rules are a variant of those of the simple type system of $\lambda_{\mathcal{W}}$. Second, dropping the resource calculus part and the simple type part results in a non-idempotent intersection type system: for example, an instance of the exponential rule is

$$\frac{x : \langle \vec{a}_1 \rangle \mid\ \vdash V : b_1 \quad \dots \quad x : \langle \vec{a}_n \rangle \mid\ \vdash V : b_n}{x : \langle \vec{a}_1, \dots, \vec{a}_n \rangle \mid\ \vdash !V : \langle b_1, \dots, b_n \rangle}.$$

Third, by ignoring the right-hand-sides of $\lhd$, the resulting system can be seen as the standard type system for the linear lambda calculus *without exponentials*; we shall discuss this point in Section 4.2.

Although we do not have the general type isomorphism rule in the type system, it is derivable in a sense. For example, assume $\Theta \lhd \Delta \mid\ \vdash \langle v_1, \dots, v_n \rangle : \langle a_1, \dots, a_n \rangle \lhd !V : S$ and consider the type isomorphism $\varphi = \langle \sigma; \vec{\mathrm{id}} \rangle : \langle a_1, \dots, a_n \rangle \cong \langle a_{\sigma(1)}, \dots, a_{\sigma(n)} \rangle$, determined by a permutation $\sigma \in \mathfrak{S}_n$. Although we do not have $\Theta \lhd \Delta \mid\ \vdash \langle v_1, \dots, v_n \rangle : \langle a_{\sigma(1)}, \dots, a_{\sigma(n)} \rangle \lhd !V : S$, by applying the permutation σ to the term as well as the refinement type, we obtain a derivable judgement $\Theta \lhd \Delta \mid\ \vdash \langle v_{\sigma(1)}, \dots, v_{\sigma(n)} \rangle : \langle a_{\sigma(1)}, \dots, a_{\sigma(n)} \rangle \lhd !V : S$. A generalisation of this idea is the *action of an isomorphism $\varphi : a \cong a'$ to a rigid resource raw-term t*; we

(a) Syntax $a, b ::= \mathsf{a} \mid a \multimap b \mid () \mid a \otimes b \mid \langle a_1, \dots, a_n \rangle \mid a \oplus \bullet \mid \bullet \oplus a \mid \mathtt{nil} \mid a{::}b$

(b) Refinement relation $\dfrac{a_1 \lhd S \quad \dots \quad a_n \lhd S}{\langle a_1, \dots, a_n \rangle \lhd !S} \qquad \dfrac{a \lhd S}{a \oplus \bullet \lhd S \oplus T} \qquad \dfrac{b \lhd T}{\bullet \oplus b \lhd S \oplus T} \qquad \dfrac{}{\mathtt{nil} \lhd \mathtt{list}\, S} \qquad \dfrac{a \lhd S \quad b \lhd \mathtt{list}\, S}{a{::}b \lhd \mathtt{list}\, S}$

(c) Type isomorphisms $\dfrac{\varphi : a' \cong a \quad \psi : b \cong b'}{\varphi \multimap \psi : a \multimap b \cong a' \multimap b'} \qquad \dfrac{\sigma \in \mathfrak{S}_n \quad \varphi_1 : a_{\sigma(1)} \cong b_1 \quad \dots \quad \varphi_n : a_{\sigma(n)} \cong b_n}{\langle \sigma; \varphi_1, \dots, \varphi_n \rangle : \langle a_1, \dots, a_n \rangle \lhd \langle b_1, \dots, b_n \rangle} \qquad \dfrac{\varphi : a \cong a' \quad \psi : b \cong b'}{\varphi{::}\psi : a{::}b \cong a'{::}b'}$

Figure 4. Syntax, refinement relation and isomorphisms of refinement intersection types (excerpt)

$$\begin{aligned} s, t, u ::= {} & [\varphi]x \mid c^S \mid \lambda x^a.t \mid v\, w \mid t \diamond \bullet \mid \bullet \diamond t \mid () \mid s; t \mid \mathtt{let}\, x = s\, \mathtt{in}\, t \mid \langle v_1, \dots, v_n \rangle \mid \mathtt{let}\, \langle x_1, \dots, x_n \rangle = v\, \mathtt{in}\, t \mid v \otimes w \mid \mathtt{let}\, x \otimes y = s\, \mathtt{in}\, t \\ & \mid \mathtt{inl}(v) \mid \mathtt{inr}(v) \mid \mathtt{let}\, \mathtt{inl}(x) = v\, \mathtt{in}\, t \mid \mathtt{let}\, \mathtt{inr}(x) = v\, \mathtt{in}\, t \mid \mathtt{nil} \mid v{::}w \mid \mathtt{let}\, \mathtt{nil} = v\, \mathtt{in}\, t \mid \mathtt{let}\, x{::}y = v\, \mathtt{in}\, t \\ v, w ::= {} & [\varphi]x \mid c \mid \lambda x^a.t \mid () \mid v \otimes w \mid \langle v_1, \dots, v_n \rangle \mid \mathtt{inl}(v) \mid \mathtt{inr}(v) \mid \mathtt{nil} \mid v{::}w. \end{aligned}$$

Figure 5. Syntax of rigid resource raw-terms

$$\frac{\varphi : a \cong a' \quad O_1 \lhd \Delta_1 \quad a \lhd S \quad O_2 \lhd \Delta_2}{(O_1, \langle x \rangle : \langle a \rangle, O_2) \lhd (\Delta_1, y : S, \Delta_2) \mid \vdash [\varphi]x : a' \lhd y : S} \qquad \frac{\varphi : a \cong a' \quad O \lhd \Delta \quad a \lhd S}{O \lhd \Delta \mid (x : a) \lhd (y : S) \vdash [\varphi]x : a' \lhd y : S} \qquad \frac{O \lhd \Delta}{O \lhd \Delta \mid \vdash c^S : S \lhd c^S : S}$$

$$\frac{\Theta \lhd \Delta \mid \Xi \lhd \Gamma \vdash t : a \lhd M : S}{\Theta \lhd \Delta \mid \Xi \lhd \Gamma \vdash t \diamond \bullet : a \lhd M \diamond N : S} \qquad \frac{\Theta_i \lhd \Delta \mid \vdash v_i : a_i \lhd V : S \quad (\forall i \le n)}{(\Theta_1 \wedge \cdots \wedge \Theta_n) \lhd \Delta \mid \vdash \langle v_1, \dots, v_n \rangle : \langle a_1, \dots, a_n \rangle \lhd !V : !S}$$

$$\frac{\Theta_0 \lhd \Delta \mid \vdash v : \langle b_1, \dots, b_n \rangle \multimap a \lhd V : !T \multimap T \qquad \Theta_i \lhd \Delta \mid \vdash w_i : b_i \lhd \lambda x.\mathrm{Y}\, V\, x : T \quad (1 \le {}^\forall i \le n)}{(\Theta_0 \wedge \cdots \wedge \Theta_n) \lhd \Delta \mid \vdash ((); (v\, \langle w_1, \dots, w_n \rangle)) : a \lhd \mathrm{Y}\, V : T}$$

$$\frac{\Theta_1 \lhd \Delta \mid \Xi_1 \lhd \Gamma_1 \vdash v : \mathtt{inl}(a) \lhd V : S \oplus T \qquad \Theta_2 \lhd \Delta \mid (\Xi_2, y : a) \lhd (\Gamma_2, z : T) \vdash t : c \lhd M : U}{(\Theta_1 \wedge \Theta_2) \lhd \Delta \mid (\Xi_1, \Xi_2) \lhd (\Gamma_1, \Gamma_2) \vdash \mathtt{let}\, \mathtt{inl}(y) = v\, \mathtt{in}\, t : c \lhd \mathtt{case}\, V\, \mathtt{of}\, (\mathtt{inl}(z) : M \mid \mathtt{inr}(z') : N) : U}$$

Figure 6. Rules relating rigid resource raw-terms and $\lambda_{\mathcal{W}}$-terms (excerpt)

write $[\varphi] \cdot t$ for the term obtained by acting φ to t. It is defined by induction on t; examples of rules are

$$\begin{aligned} [\langle \sigma; \varphi_1, \dots, \varphi_n \rangle] \cdot \langle v_1, \dots, v_n \rangle &:= \langle [\varphi_1] \cdot v_{\sigma(1)}, \dots, [\varphi_n] \cdot v_{\sigma(n)} \rangle \\ [(\varphi \multimap \psi)] \cdot (\lambda x.t) &:= \lambda x.([\psi] \cdot t)\{[\varphi]x/x\} \\ [\varphi] \cdot (v\, w) &:= ([(\mathrm{id} \multimap \varphi)] \cdot v)\, w. \end{aligned}$$

The *substitution* $t\{v/x\}$ is defined as usual except for the base case where $([\varphi]x)\{v/x\} := [\varphi] \cdot v$.

Lemma 3.1. *The type isomorphism rules are derivable, e.g.,*

$$\frac{\Theta \lhd \Delta \mid \Xi \lhd \Gamma \vdash t : a \lhd M : S \qquad \varphi : a \cong a'}{\Theta \lhd \Delta \mid \Xi \lhd \Gamma \vdash [\varphi] \cdot t : a' \lhd M : S} \quad \textit{and}$$
$$\frac{\varphi : a \cong a' \qquad \Theta \lhd \Delta \mid (\Xi, x : a') \lhd \Gamma \vdash t : b \lhd M : S}{\Theta \lhd \Delta \mid (\Xi, x : a) \lhd \Gamma \vdash t\{[\varphi]x/x\} : b \lhd M : S}.$$

Let us define an isomorphism $\varphi : (\Theta \mid \Xi) \cong (\Theta' \mid \Xi')$ as a sequence of component-wise isomorphisms. The previous lemma can be generalised to

$$\frac{\varphi : (\Theta' \mid \Xi') \cong (\Theta \mid \Xi) \qquad \Theta \lhd \Delta \mid \Xi \lhd \Gamma \vdash t : b \lhd M : S}{\Theta \lhd \Delta \mid \Xi \lhd \Gamma \vdash t\{\varphi\} : b \lhd M : S}$$

where $t\{\varphi\}$ denotes the appropriate resource raw-term.

3.3 Enumeration of reduction sequences

The operational semantics of the rigid calculus is defined analogously to that of $\lambda_{\mathcal{W}}$. A *configuration* is a triple $[\vec{x} = e, t]$ that is well-typed in an appropriate sense, and the reduction is a relation on configurations. Examples of rules are

$$\begin{aligned} &[\vec{x} = e, E[t \diamond \bullet]] \xrightarrow{1} [\vec{x} = e, E[t]] \\ &[\vec{x} = e, E[\mathtt{let}\, \langle y_1, \dots, y_n \rangle = \langle v_1, \dots, v_n \rangle\, \mathtt{in}\, t]] \\ &\qquad \xrightarrow{0} [\vec{x} = e, E[t\{v_1/y_1, \dots, v_n/y_n\}]] \end{aligned}$$

where E is an *evaluation context*, defined by the grammar: $E ::= [] \mid E; t \mid \mathtt{let}\, x = E\, \mathtt{in}\, t$. Problematic configurations such as $[\vec{x} = e, \mathtt{let}\, \mathtt{inr}(x) = \mathtt{inl}(v)\, \mathtt{in}\, t]$ are filtered out by the type system.

The next lemma follows from the fact that a rigid resource raw-term has neither nondeterministic branching nor recursion.

Lemma 3.2. *For each configuration $[\vec{x} = e, t]$, there exists a unique pair (π, e') such that $[\vec{x} = e, t] \xrightarrow{\pi} [\epsilon = e', ()]$.*

For a program $\vdash P : I$, let us consider the set $\{t \mid \vdash t() \lhd P : I\}$. For each element $t \in X$ of this set, we write $\pi(t)$ and $\varpi(t)$ to mean the unique π and e such that $[\epsilon = \mathrm{id}_I, t] \xrightarrow{\pi} [\epsilon = e, ()]$. The next theorem says that the mapping $t \mapsto \pi(t)$ is a weight-preserving surjection to $\mathit{Eval}(P)$.

Theorem 3.3. *Let P be a program. If $\vdash t : () \lhd P : I$ and $[\epsilon = \mathrm{id}_I, t] \xrightarrow{\pi} [\epsilon = e, ()]$, then $[\epsilon = \mathrm{id}_I, P] \xrightarrow{\pi} [\epsilon = e, ()]$. Conversely, if $[\epsilon = \mathrm{id}_I, P] \xrightarrow{\pi} [\epsilon = e, ()]$, then there exists t such that $\vdash t : () \lhd P : I$ and $[\epsilon = \mathrm{id}_I, t] \xrightarrow{\pi} [\epsilon = e, ()]$.*

Unfortunately this is not a bijection: different approximants may induce the same computation. For example, consider refinements

$\mathtt{let}\, \langle x_1, x_2 \rangle = \langle v_1, v_2 \rangle\, \mathtt{in}\, x_1 \otimes x_2 \qquad \mathtt{let}\, \langle x_2, x_1 \rangle = \langle v_2, v_1 \rangle\, \mathtt{in}\, x_1 \otimes x_2$

of $\mathtt{let}\, !x = !V\, \mathtt{in}\, x \otimes x$ (see [42] for further discussion).

We have proposed in our previous work [42] a way to avoid this redundancy by using the action of isomorphisms. Let $\sim$ be a congruence on rigid resource raw-terms subsuming

$$v\,([\varphi]\cdot w) \;\sim\; ([(\varphi \multimap \mathrm{id})]\cdot v)\,w$$
$$\mathtt{let}\,x = [\varphi]\cdot t\,\mathtt{in}\,u \;\sim\; \mathtt{let}\,x = t\,\mathtt{in}\,(u\{[\varphi]x/x\})$$
$$\mathtt{let}\,\langle x_1,\dots,x_n\rangle = ([\langle\sigma;\varphi_1,\dots,\varphi_n\rangle]\cdot v)\,\mathtt{in}\,t$$
$$\sim\;\mathtt{let}\,\langle x_{\sigma^{-1}(1)},\dots,x_{\sigma^{-1}(n)}\rangle = v\,\mathtt{in}\,t\{[\varphi_1]x_1/x_1,\dots,[\varphi_n]x_n/x_n\}$$

and similar rules for other let-constructs. Note that $\sim$ is defined for terms of higher-order types as well.

Theorem 3.4 ([42]). *Let P be a program and assume $\vdash t_i : () \lhd P : I$ for $i = 1, 2$. Then $t_1 \sim t_2$ if and only if $\pi(t_1) = \pi(t_2)$.*

Given a $\lambda_{\mathcal{W}}$-term, its *rigid Taylor expansion* is defined as the collection of well-typed approximations of it. We write $\widetilde{t}$ for the equivalence class of $\sim$ to which t belongs.

Definition 3.5 (Rigid Taylor expansion). Given $\Delta \mid \Gamma \vdash M : S$ and $(\Theta \mid \Xi) \lhd (\Delta \mid \Gamma)$, we define

$$[\![M]\!](b, (\Theta \mid \Xi)) := \{\, \widetilde{t} \;\mid\; \Theta \lhd \Delta \mid \Xi \lhd \Gamma \vdash t : b \lhd M : S \,\}$$

We call $[\![M]\!]$ the *rigid Taylor expansion of M*. We write $(\Theta \mid \Xi \vdash \widetilde{t} : b) \in [\![M]\!]$ to mean $\widetilde{t} \in [\![M]\!](b, (\Theta \mid \Xi))$.

Theorems 3.3 and 3.4 give a bijective correspondence between $Eval(P)$ and $[\![P]\!](\epsilon, ())$, which furthermore preserves the weights. This allows us to enumerate $Eval(P)$ by induction on the structure of P, even though $Eval(P)$ is not inductively defined.

4 Weighted Generalised Species

We have seen in the previous section that the rigid Taylor expansion of a program P is a weighted set equivalent to $Eval(P)$ up to a weight-preserving bijection, and hence in a sense adequate. This section gives a more "semantic" description of this result, based on *weighted generalised species* (or *weighted profunctors*). The result of this section extends [42], which studies a weight-free setting.

4.1 Preliminary: profunctors

We briefly recall profunctors and introduce notations. A *profunctor* F from a category $\mathcal{A}$ to a category $\mathcal{B}$ (written $F : \mathcal{A} \nrightarrow \mathcal{B}$) is a functor $F : \mathcal{B}^{\mathrm{op}} \times \mathcal{A} \longrightarrow \mathbf{Set}$. For $g \in \mathcal{B}(b', b)$, $x \in F(b, a)$ and $f \in \mathcal{A}(a, a')$, we write $x \cdot f$ for $F(b, f)(x)$ and $g \cdot x$ for $F(g, a)(x)$. Since F is a bifunctor, $(g \cdot x) \cdot f = g \cdot (x \cdot f)$, which we simply write as $g \cdot x \cdot f$.[1] The composite $G \circ F : \mathcal{A} \nrightarrow \mathcal{C}$ of $F : \mathcal{A} \nrightarrow \mathcal{B}$ and $G : \mathcal{B} \nrightarrow \mathcal{C}$ can be defined by

$$(G \circ F)(c, a) := \Big(\coprod_{b \in \mathcal{B}} G(c, b) \times F(b, a)\Big)/\!\sim$$

where $\coprod$ is the coproduct in **Set** and $\sim$ is the least equivalence relation containing $(y, f \cdot x) \sim (y \cdot f, x)$ for each $y \in G(c, b')$, $f \in \mathcal{B}(b', b)$ and $x \in F(b, a)$. We write **Prof** for the bicategory of categories, profunctors and natural transformations.

[1] In this paper, the action of profunctors is written in the diagrammatic order in the sense that $g' \cdot (g \cdot x \cdot f) \cdot f' = (g'; g) \cdot x \cdot (f; f')$, where $g'; g \triangleq g \circ g'$.

4.2 Properties of the rigid Taylor expansion

This subsection studies the properties of the rigid Taylor expansion, which shall be abstracted to the notion of weighted profunctors.

As pointed out in our previous work [42], the rigid Taylor expansion $[\![M]\!]$ of a term $\Delta \mid \Gamma \vdash M : S$ is a profunctor $[\![\Delta \mid \Gamma]\!] \nrightarrow [\![S]\!]$. Here $[\![\Delta \mid \Gamma]\!]$ and $[\![S]\!]$ are groupoids of refinements and isomorphisms. Lemma 3.1 shows that $\widetilde{\varphi : a' \cong a,\ t} \in [\![M]\!](a, (\Theta \mid \Xi))$ and $\psi : (\Theta \mid \Xi) \cong (\Theta' \mid \Xi')$ imply $[\varphi^{-1}] \cdot t\{\psi^{-1}\} \in [\![M]\!](a', (\Theta' \mid \Xi'))$.

In the situation of this paper, one can furthermore interpret the refinement types and rigid resource (raw-)terms in $\mathcal{W}$.

The interpretation of a simple type induces a functor $S : [\![S]\!] \longrightarrow \mathcal{W}^{\mathrm{op}}$, i.e. refinement types and type isomorphisms can be seen as objects and morphisms in $\mathcal{W}$, respectively. Its action on objects are defined via the following syntactic translation

$$\natural(\mathtt{a}) = \mathtt{a} \qquad \natural(a \otimes b) = \natural(a{::}b) = \natural(a) \otimes \natural(b)$$
$$\natural(()) = \natural(\mathtt{nil}) = I \qquad \natural(\mathtt{inl}(a)) = \natural(\mathtt{inr}(a)) = \natural(a)$$
$$\natural(a \multimap b) = \natural(a) \multimap \natural(b) \qquad \natural(\langle a_1, \dots, a_n\rangle) = \natural(a_1) \otimes \dots \otimes \natural(a_n)$$

of a refinement type to an IMLL formula. Its action on morphisms is defined by induction on the derivation of $\varphi : a \cong a'$, using only the structural isomorphisms in $\mathcal{W}$.

A rigid resource (raw-)term induces a term of a linear lambda calculus without exponential, by ignoring `inl` and `inr` and identifying $v{::}w$ (resp. $\langle v_1, \dots, v_n\rangle$) with $v \otimes w$ (resp. $v_1 \otimes \dots \otimes v_n$) as well as the corresponding patterns. For example, $\mathtt{let}\,x{::}y = v\,\mathtt{in}\,t$ is regarded as $\mathtt{let}\,x \otimes y = v\,\mathtt{in}\,t$. Thus we have an interpretation of rigid resource (raw-)terms in the SMCC $\mathcal{W}$; we write $\langle\!|t|\!\rangle$ for this interpretation.

Lemma 4.1. *Let $\Delta \mid \Gamma \vdash M : S$. (1) The simple type S induces a functor $S : [\![S]\!] \longrightarrow \mathcal{W}^{\mathrm{op}}$ from the groupoid of refinement types and isomorphisms. Similarly the simple type environment induces a functor $E : [\![(\Delta \mid \Gamma)]\!] \longrightarrow \mathcal{W}^{\mathrm{op}}$. (2) The rigid Taylor expansion is a profunctor $[\![M]\!] : [\![\Delta \mid \Gamma]\!] \nrightarrow [\![S]\!]$ of which each element $\widetilde{t} \in [\![M]\!](a, (\Theta \mid \Xi))$ is associated with a morphism $\langle\!|t|\!\rangle : E(\Theta \mid \Xi) \longrightarrow S(a)$ in $\mathcal{W}$. Furthermore $\langle\!|\cdot|\!\rangle$ respects the action of maps in $[\![\Delta \mid \Gamma]\!]$ and $[\![S]\!]$, i.e. $S(\varphi) \circ \langle\!|t|\!\rangle \circ E(\psi) = \langle\!|[\varphi^{-1}] \cdot t\{\psi^{-1}\}|\!\rangle$.*

The above syntactic translation of terms maps the reduction rules to valid equations of the standard linear lambda calculus, by regarding $[\vec{x} = e, t]$ as $\mathtt{let}\,\vec{x} = e\,\mathtt{in}\,t$. Thanks to the well-known soundness result of SMCCs for the linear lambda calculus, $\langle\!|\cdot|\!\rangle$ is preserved by reduction. Hence the weight of a rigid resource (row-)term coincides with the interpretation in $\mathcal{W}$.

Theorem 4.2. *If $[\epsilon = \mathrm{id}_I, t] \xrightarrow{\pi} [\epsilon = e, ()]$, then $e = \langle\!|t|\!\rangle$.*

This theorem together with Theorems 3.3 and 3.4 provides us with a compositional way for calculating the weighted set $Eval(P)$.

4.3 Weighted profunctors

We introduce the notion of *weighted profunctors* as an abstraction of the properties shown in Lemma 4.1.

Definition 4.3 (Weighted category, weighted profunctor). A *$\mathcal{W}$-weighted category* is a pair $(\mathcal{A}, A)$ of a category $\mathcal{A}$ and a functor $A : \mathcal{A} \longrightarrow \mathcal{W}^{\mathrm{op}}$. A *$\mathcal{W}$-weighted profunctor* from $(\mathcal{A}, A)$ to $(\mathcal{B}, B)$ is a pair (F, ϖ) of a profunctor $F : \mathcal{A} \nrightarrow \mathcal{B}$ (i.e. a functor $F : \mathcal{B}^{\mathrm{op}} \times \mathcal{A} \longrightarrow \mathbf{Set}$) and a family of functions $\varpi_{(b,a)} : F(b, a) \longrightarrow \mathcal{W}(A(a), B(b))$ $(a \in \mathcal{A}, b \in \mathcal{B})$ that respects the action of $\mathcal{A}$ and $\mathcal{B}$, i.e.,

$$B(g) \circ \varpi_{(b,a)}(e) \circ A(f) \;=\; \varpi_{(b',a')}(g \cdot e \cdot f)$$

for every $g : b' \to b$, $e \in F(b,a)$ and $f : a \to a'$. A *2-cell* $\alpha : (F, \varpi^F) \Rightarrow (G, \varpi^G)$ is a natural transformation $\alpha : F \Rightarrow G$ preserving weights, i.e. $\varpi^F_{(b,a)}(e) = \varpi^G_{(b,a)}(\alpha_{b,a}(e))$ for every $e \in F(b,a)$. We often omit "$\mathcal{W}$-" if it is clear from the context.

Weighted categories, weighted profunctors and 2-cells in Definition 4.3 can be organised into a bicategory, which we write as $\mathbf{Prof}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$: The composite $(G, \varpi^G) \circ (F, \varpi^F)$ of weighted profunctors $(F, \varpi^F) : (\mathcal{A}, A) \nrightarrow (\mathcal{B}, B)$ and $(G, \varpi^G) : (\mathcal{B}, B) \nrightarrow (C, C)$ consists of the composite profunctor $G \circ F$ with the weight function $\varpi_{(c,a)} : (G \circ F)(c, a) \to \mathcal{W}(A(a), C(c))$ defined by

$$\varpi_{c,a}([(y,x)]_\sim) := \varpi^G_{c,b}(y) \circ \varpi^F_{b,a}(x)$$

where $(y, x) \in G(c, b) \times F(b, a)$. This is well-defined since ϖ^G and ϖ^F respect the action of $\mathcal{B}$ morphisms.

We shall mainly use a 1-categorical version of the bicategory $\mathbf{Prof}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$, written $\mathbf{Pr}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$. This is defined as the *classifying category* $Cl(\mathbf{Prof}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}})$ [4, Section 7] of $\mathbf{Prof}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$, whose object is a 0-cell of $\mathbf{Prof}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ and whose morphism is an equivalence class of 1-cells of $\mathbf{Prof}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ modulo the existence of an iso-2-cell.

4.4 $\mathbf{Pr}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ as a $\lambda_{\mathcal{W}}$-model

We first discuss the Lafont structure of $\mathbf{Pr}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$. For space reasons, we give only the overview; see Appendix C and D for the details.

The SMCC structure of $\mathbf{Pr}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ follows from the SMCC structure of $\mathbf{Prof}$ and $\mathcal{W}$. Let $A : \mathcal{A} \to \mathcal{W}^{\mathrm{op}}$ and $B : \mathcal{B} \to \mathcal{W}^{\mathrm{op}}$ be weighted categories. The tensor product is defined by $(\mathcal{A}, A) \hat{\otimes} (\mathcal{B}, B) \triangleq (\mathcal{A} \times \mathcal{B}, A \hat{\otimes} B)$ where $A \hat{\otimes} B \triangleq (\otimes^{\mathrm{op}}) \circ (A \times B)$, i.e. $(A \hat{\otimes} B)(a, b) = A(a) \otimes B(b)$ and $(A \hat{\otimes} B)(f, g) = A(f) \otimes B(g)$. This definition uses the tensor products $\times$ and $\otimes$ of $\mathbf{Prof}$ and $\mathcal{W}$, respectively. Its action on morphisms $(F_i, \varpi_i) : (\mathcal{A}_i, A_i) \nrightarrow (\mathcal{B}_i, B_i)$ $(i = 1, 2)$ is given by $(F_1 \hat{\otimes} F_2)((b_1, b_2), (a_1, a_2)) \triangleq F_1(b_1, a_1) \times F_2(b_2, a_2)$ with the weight function $F_1(b_1, a_1) \times F_2(b_2, a_2) \ni (x_1, x_2) \mapsto \varpi_1(x_1) \otimes \varpi_2(x_2) \in \mathcal{W}(A_1(a_1) \otimes A_2(a_2), B_1(b_1) \otimes B_2(b_2))$. The closed structure is defined similarly: $(\mathcal{A}, A) \hat{\multimap} (\mathcal{B}, B) \triangleq (\mathcal{A}^{\mathrm{op}} \times \mathcal{B}, (\multimap^{\mathrm{op}}) \circ (A^{\mathrm{op}} \times B))$.

For any category $\mathcal{W}$, the category $\mathbf{Pr}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ has (small) biproducts given by the biproduct of $\mathbf{Prof}$.

We can show that $\mathbf{Pr}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ has free commutative comonoids, and thus a linear exponential comonad, following the recipe of [28, 35]. It suffices to show that $\mathbf{Pr}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ has *symmetric tensor powers* [35], i.e. the equaliser $\mathbb{P}_n(\mathcal{A}, A) \to (\mathcal{A}, A)$ of $n!$ symmetries from $(\mathcal{A}, A)^{\hat{\otimes} n}$ to itself, and show that the equaliser is preserved by the tensor product. The underlying category of $\mathbb{P}_n(\mathcal{A}, A)$ has as an object a sequence $(a_i)_{i \le n}$ of objects of $\mathcal{A}$ and as a morphism $(a_i)_i \to (a'_i)_i$ a pair of permutation σ and $(f_i : a_i \to a'_{\sigma(i)})_{i \le n}$.

Theorem 4.4. *$\mathbf{Pr}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ is a Lafont category with biproducts.*

Remark 4.5. In the proof of Theorem 4.4 (in Appendix C), we employ an equivalent but categorically simpler definition of the bicategory $\mathbf{Prof}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$, as a full sub-bicategory of the *lax-slice* bicategory of $\mathbf{Prof}$ over $\mathcal{W}^{\mathrm{op}}$. There we prove that $\mathbf{Prof}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ has the symmetric monoidal closed structure, biproducts and symmetric tensor powers in 2-dimensional category theory; hence if we can extend the construction of Lafont categories in [28, 35] to 2-dimensional category theory, we obtain a Lafont *bi*category. □

The interpretation of base type a is a functor $\star \mapsto \mathsf{a} : 1 \to \mathcal{W}^{\mathrm{op}}$. Therefore the interpretation of type $\mathsf{a}_1 \otimes \cdots \otimes \mathsf{a}_n$ is $\star \mapsto \mathsf{a}_1 \otimes \cdots \otimes \mathsf{a}_n : 1 \to \mathcal{W}^{\mathrm{op}}$. The interpretation of constant $c^{\mathsf{a}_1 \otimes \cdots \otimes \mathsf{a}_n \multimap \mathsf{b}_1 \otimes \cdots \otimes \mathsf{b}_m}$ consists of the profunctor $F(\star, \star) := \{*\}$ with the weight function $* \mapsto c \in \mathcal{W}(\mathsf{a}_1 \otimes \cdots \otimes \mathsf{a}_n, \mathsf{b}_1 \otimes \cdots \otimes \mathsf{b}_m)$. The interpretation of $\mathbf{Y}$ is defined as the rigid Taylor expansion of $x : !T \multimap T \vdash \mathbf{Y}\, x : T$, in order to establish Theorem 4.6. We expect this to coincide with the fixed-point operator of Laird's theorem [26, Thm. 4.20], though a proper comparison is left for future work.

The concrete definition of the $\lambda_{\mathcal{W}}$-model structure of $\mathbf{Pr}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ is tightly related to the rigid Taylor expansion. For example, for $\mid \Gamma_i \vdash V_i : S_i$ $(i = 1, 2)$, it is fairly easy to see that $[\![V_1]\!] \hat{\otimes} [\![V_2]\!]$ defines the same 1-cell of $\mathbf{Pr}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ as $[\![V_1 \otimes V_2]\!]$. A notable point is that the equivalence relation $\sim$ in the definition of the composition of profunctors (Section 4.1) coincides with the relation $\sim$ on rigid resource raw-terms (Section 3.3). Hence it is also easy to show that $[\![N]\!] \circ [\![M]\!] = [\![\mathtt{let}\, x = M\, \mathtt{in}\, N]\!]$ for $\mid \Gamma \vdash M : S$ and $\mid x : S \vdash N : T$.

Theorem 4.6. *The interpretation of a term M in $\mathbf{Pr}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ coincides with the rigid Taylor expansion $[\![M]\!]$.*

Corollary 4.7 (Adequacy)**.** *The interpretation of a program P in $\mathbf{Pr}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ coincides with the weighted set $Eval(P)$.*

5 Associated Matrix as Generating Series

This section introduces a concise representation for (a subclass of) weighted profunctors, inspired by the generating series of a weighted species (see e.g. [6]). Recall that the (exponential) generating series of a weighted species $(F : \mathbf{P} \to \mathbf{Set}, \{\varpi_n : F(n) \to W\}_n)$ is defined as $\|(F, \varpi)\| = \sum_{n=0}^{\infty} \|(F, \varpi)\|_n\, z^n$, where z is the indeterminant and the coefficient $\|(F, \varpi)\|_n$ is defined by

$$\|(F, \varpi)\|_n \triangleq \frac{1}{n!} \sum_{x \in F(n)} \varpi_n(x)$$

provided that this expression makes sense (e.g. $W \supseteq \mathbf{Q}$ is a ring and $F(n)$ is finite for every n).

Since a profunctor $F : \mathcal{A} \nrightarrow \mathcal{B}$ is a functor $\mathcal{B}^{\mathrm{op}} \times \mathcal{A} \to \mathbf{Set}$, the ordinary species is a special case of $\mathcal{A} = [\![!I]\!]$ and $\mathcal{B} = [\![I]\!]$ as observed in [13]. This motivates us to define

$$\|(F, \varpi)\|_{b,a} := \frac{1}{\#\mathcal{B}^-(b, b)\, \#\mathcal{A}^+(a, a)} \sum_{x \in F(b,a)} \varpi_{b,a}(x) \qquad (1)$$

($\#X$ is cardinality of the set X; $\mathcal{A}$ and $\mathcal{B}$ will be strict factorisation systems, and the superscripts $(-, +)$ refer resp. to the two classes (E, M) of morphisms). We call $\|(F, \varpi)\|$ the *associated matrix*, as it can be seen as a matrix indexed by $\mathrm{ob}(\mathcal{B})$ and $\mathrm{ob}(\mathcal{A})$, whose elements are morphisms of $\mathcal{W}$. A remarkable difference from the ordinary matrix is that the domains of elements vary with indexes.

The weight category $\mathcal{W}$ should have additional structures for Equation (1) to make sense. In particular, each hom-set $\mathcal{W}(A(a), \mathcal{B}(b))$, to which $\varpi_{b,a}(x)$ belongs, should have the summation operation $\sum$, as well as the multiplication with $1/(\#\mathcal{B}^-(b, b)\, \#\mathcal{A}^+(a, a))$. Section 5.1 defines the requirements of $\mathcal{W}$ in terms of enrichment.

Section 5.2 defines the category of matrices with elements from $\mathcal{W}$ and gives a formal definition of $\|\cdot\|$. Unfortunately $\|\cdot\|$ is not even functorial. Section 5.3 introduces a subclass of profunctors, called *P-visible* profunctors, on which $\|\cdot\|$ behaves well.

5.1 Σ-monoids and ΣMon-categories

Since $Eval(P)$ can be countably infinite, the sum in (1) can also be countably infinite. This subsection introduces an algebra with countable sum, known as Σ-*monoids* [16, 17, 20], and the notion of SMCCs whose hom-sets are Σ-monoids.

Let e and e' be expressions possibly having partial operations. We write $e \sqsubseteq e'$ to mean that, if e is defined, then e' is also defined and the values are the same; $e \simeq e'$ is a shorthand for $e \sqsubseteq e' \wedge e' \sqsubseteq e$.

Let X be a set. A *countable family in* X is a pair (I, x) of a countable set I of indexes and a function $x : I \to X$. We write x_i for $x(i)$ and $\{x_i\}_{i\in I}$ for a countable family. Countable families $\{x_i\}_{i\in I}$ and $\{y_j\}_{j\in J}$ are *equivalent* if there exists a bijection $f : I \to J$ such that $x_i = y_{f(i)}$ for every $i \in I$. Given a set X, let $\mathit{Fam}(X)$ be the set of all countable families $\{x_i\}_{i\in I}$ in X indexed by a subset of natural numbers (i.e. $I \subseteq \mathbb{N}$).

Definition 5.1 (Σ-monoids)**.** A pair $(\mathfrak{M}, \sum)$ of a nonempty set $\mathfrak{M}$ and a partial function $\sum : \mathit{Fam}(\mathfrak{M}) \rightharpoonup \mathfrak{M}$ is a Σ-*monoid* if it satisfies the following conditions: (1) for every $I, J \subseteq \mathbb{N}$ and partition $\{I_j\}_{j\in J}$ of I, we have $\sum\{x_i\}_{i\in I} \simeq \sum\{\sum_{i\in I_j} x_i\}_{j\in J}$, and (2) for a singleton $I = \{j\}$, we have $\sum\{x_i\}_{i\in I} \simeq x_j$. We say $\{x_i\}_{i\in I}$ is *summable* if $\sum\{x_i\}_{i\in I}$ is defined. A Σ-monoid is *total* (aka *complete*) if all countable families are summable. We often write $\sum_{i\in I} x_i$ for $\sum\{x_i\}_{i\in I}$. A total Σ-monoid is a commutative monoid in the usual sense, with binary sum $x_1 + x_2 := \sum_{i\in\{1,2\}} x_i$.

Example 5.2. Recall examples in Section 2.3. The two-valued Boolean algebra $\mathbf{B}(I, I)$ in Example 2.1 is a total Σ-monoid by disjunction. Both $[0, 1]$ and $\mathbf{R}^{\infty}_{\geq 0}$ in Example 2.2 are Σ-monoids by the standard sum of reals (in $\mathbf{R}^{\infty}_{\geq 0}$, $\sum_{i\in I} x_i = \infty$ if it does not converge). The latter is total though the former is not. *Continuous semirings* used in [28] and *(countably) complete semirings* used in [26] are examples of total Σ-monoids by summation. As for Examples 2.4 and 2.5, both $\mathbf{FdHilb}(n, m)$ and $\mathbf{CPM}_s(n, m)$ are non-total Σ-monoids.

Definition 5.3 (Category Σ**Mon**)**.** A *homomorphism of* Σ-*monoids* is a function $f : \mathfrak{M} \to \mathfrak{N}$ such that $f(\sum_{i\in I} x_i) \sqsubseteq \sum_{i\in I} f(x_i)$ for every $\{x_i\}_{i\in I} \in \mathit{Fam}(\mathfrak{M})$. The *category* Σ**Mon** has Σ-monoids as objects and homomorphisms of Σ-monoids as morphisms. We write Σ$\mathbf{Mon}_t$ for the full subcategory of total Σ-monoids.

We review the structure of Σ**Mon** and Σ$\mathbf{Mon}_t$ following [20].

Definition 5.4 (Bilinear map)**.** Let $\mathfrak{M}$, $\mathfrak{N}$ and $\mathfrak{L}$ be Σ-monoids. A *bilinear map* $f \in \mathrm{Bilin}(\mathfrak{M}, \mathfrak{N}; \mathfrak{L})$ is a function $\mathfrak{M} \times \mathfrak{N} \to \mathfrak{L}$ such that

$$f(\sum_{i\in I} x_i, y) \sqsubseteq \sum_{i\in I} f(x_i, y) \quad \text{and} \quad f(x, \sum_{i\in I} y_i) \sqsubseteq \sum_{i\in I} f(x, y_i).$$

The functor $\mathrm{Bilin}(\mathfrak{M}, \mathfrak{N}; -) : \Sigma\mathbf{Mon} \to \mathbf{Set}$ is representable [20, Proposition 3.5]; we write $\mathfrak{M} \otimes \mathfrak{N}$ for the representation, and identify $\Sigma\mathbf{Mon}(\mathfrak{M} \otimes \mathfrak{N}, \mathfrak{L})$ with $\mathrm{Bilin}(\mathfrak{M}, \mathfrak{N}; \mathfrak{L})$.

The category Σ**Mon** is an SMCC with ⊗ as the monoidal product. The unit is $I = \{0, 1\}$ with $1 + 1$ undefined. We have $\Sigma\mathbf{Mon}(I, \mathfrak{M}) \cong \mathfrak{M}$ as sets. The linear function space $\mathfrak{M} \multimap \mathfrak{N}$ is the set of homomorphisms with the sum defined by the point-wise sum.

Definition 5.5 (Σ**Mon**-category, Σ**Mon**-SMCC)**.** A Σ**Mon**-*category* is a locally small category $\mathcal{W}$ such that (1) each hom-set $\mathcal{W}(a, b)$ is equipped with a Σ-monoid structure, and (2) the composition is bilinear. A Σ**Mon**-category is a Σ**Mon**-*SMCC* if (1) the underlying category $\mathcal{W}$ is an SMCC, (2) the action of the tensor product on morphisms, $(f, g) \mapsto (f \otimes g)$, is bilinear, and (3) the bijections $\mathcal{W}(a \otimes b, c) \cong \mathcal{W}(a, b \multimap c)$ are homomorphisms of Σ-monoid. A Σ$\mathbf{Mon}_t$-*SMCC* is a Σ**Mon**-SMCC $\mathcal{W}$ all of whose hom-objects $\mathcal{W}(a, b)$ are total Σ-monoids.

Example 5.6. The category $\mathbf{B}(I, I)$ in Example 2.1 is a Σ$\mathbf{Mon}_t$-SMCC. The category $\mathcal{W}_{[0,1]}$ and its variant $\mathcal{W}_{\mathbf{R}^{\infty}_{\geq 0}}$ in Example 2.2 are Σ**Mon**-SMCCs; the latter is also an example of Σ$\mathbf{Mon}_t$-SMCC. In general, one-object Σ$\mathbf{Mon}_t$-SMCCs coincide with *(countably) complete semirings* in the sense of [26, Definition 2.5]. A continuous semiring used in [28] is an example of total Σ-monoid by summation. **FdHilb** and $\mathbf{CPM}_s$ are Σ**Mon**-SMCCs but not Σ$\mathbf{Mon}_t$-SMCCs.

Definition 5.7 (Reciprocal for natural numbers)**.** Let $\mathcal{W}$ be a Σ**Mon**-category and $a \in \mathrm{ob}(\mathcal{W})$. Given a natural number n, we say $r \in \mathcal{W}(a, a)$ is a *reciprocal for* n if $\sum_{i=1}^{n} r = \mathrm{id}_a$. A reciprocal for n is unique if it exists. We write $1/n$ for the reciprocal for n.

Lemma 5.8. *Let* $\mathcal{W}$ *be a* Σ**Mon**-*category. If* $\mathcal{W}(I, I)$ *has reciprocals for* n, *then so does* $\mathcal{W}(a, a)$ *for every* $a \in \mathrm{ob}(\mathcal{W})$.

5.2 Associated Matrices of Weighted Profunctors

Let $\mathcal{W}$ be a Σ$\mathbf{Mon}_t$-SMCC, fixed below. Assume that, for each $n \in \mathbb{N}$, the Σ-monoid $\mathcal{W}(I, I)$ has the reciprocal for n.

A category $\mathcal{A}$ is *countable* if the collection of morphisms is countable (then $\mathrm{ob}(\mathcal{A})$ is also countable). It is *locally-finite* if $\mathcal{A}(a, a')$ is finite for every $a, a' \in \mathrm{ob}(\mathcal{A})$. We write $\mathrm{ob}(\mathcal{A})/\mathrm{iso}$ for the collection of isomorphic classes of objects in $\mathcal{A}$.

Definition 5.9 (Matrix category)**.** The *matrix category* $\mathbf{Mat}(\mathcal{W})$ is defined by the following data. An object is a weighted category $A : \mathcal{A} \to \mathcal{W}^{\mathrm{op}}$ such that $\mathcal{A}$ is a countable, locally-finite groupoid with a strict factorisation system. A morphism $f : (\mathcal{A}, A) \to (\mathcal{B}, B)$ is a family $\{f_{a,b} : A(a) \to B(b)\}_{(a,b)\in\mathrm{ob}(\mathcal{A})\times\mathrm{ob}(\mathcal{B})}$ of morphisms in $\mathcal{W}$ that respects the action of $\mathcal{A}$- and $\mathcal{B}$-morphisms (i.e. $B(g) \circ f_{a,b} \circ A(h) = f_{a',b'}$ for every $h : a \to a'$ and $g : b' \to b$). Composition of $f = \{f_{b,a}\} : (\mathcal{A}, A) \to (\mathcal{B}, B)$ and $g = \{g_{c,b}\} : (\mathcal{B}, B) \to (C, C)$ is defined by

$$(g \circ f)_{c,a} := \sum_{[b]\in\mathrm{ob}(\mathcal{B})/\mathrm{iso}} g_{c,b} \circ f_{b,a}$$

where the sum is that of $\mathcal{W}(A(a), C(c))$. The identity $\mathrm{id} : (\mathcal{A}, A) \to (\mathcal{A}, A)$ is defined by $\mathrm{id}_{a,a'} := 1/\#\mathcal{A}(a, a') \sum_{h\in\mathcal{A}(a,a')} A(h)$.

Now we are ready to define the associated matrix formally. A profunctor $F : \mathcal{A} \nrightarrow \mathcal{B}$ is said to be *countable* if $F(a, b)$ is countable for every $a \in \mathrm{ob}(\mathcal{A})$ and $b \in \mathrm{ob}(B)$.

Definition 5.10 (Associated matrix)**.** Let $\mathcal{A}, \mathcal{B} \in \mathrm{ob}(\mathbf{Mat}(\mathcal{W}))$. Given a $\mathcal{W}$-weighted countable profunctor $F : \mathcal{A} \nrightarrow \mathcal{B}$ with weight function $\varpi_{a,b} : F(a, b) \to \mathcal{W}(A(a), B(b))$, the *associated matrix* is a morphism $\|(F, \varpi)\| : \mathcal{A} \to \mathcal{B}$ in $\mathbf{Mat}(\mathcal{W})$ given by (1).

A morphism $f = \{f_{a,b}\}_{a,b} : (\mathcal{A}, A) \to (\mathcal{B}, B)$ in $\mathbf{Mat}(\mathcal{W})$ bijectively corresponds to a weight function ϖ^f for the locally-terminal profunctor $F : \mathcal{A} \nrightarrow \mathcal{B}$ (i.e. $F(b, a) = \{*\}$ for every a and b): let us define $\varpi^f_{b,a}(*) = f_{a,b}$. Although this correspondence is not functorial, the SMCC structure of $\mathbf{Mat}(\mathcal{W})$ can be defined via the correspondence. The biproducts and symmetric tensor powers are obtained from $\mathbf{Pr}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ by applying $\|\cdot\|$.

Theorem 5.11. $\mathbf{Mat}(\mathcal{W})$ *is a Lafont category with countable biproducts.*

Remark 5.12. The *countable biproduct completion* $\mathcal{W}^{\prod}$ (cf. [26–28]) is a full subcategory of $\mathbf{Mat}(\mathcal{W})$ consisting of objects $A : \mathcal{A} \to \mathcal{W}$ with $\mathcal{A}$ discrete. (I.e. objects of $\mathcal{W}^{\prod}$ are countable lists of objects of $\mathcal{W}$.) A notable difference is that $\mathbf{Mat}(\mathcal{W})$ is a Lafont category, whereas $\mathcal{W}^{\prod}$ is not. The objects $\mathcal{A}$ of $\mathbf{Mat}(\mathcal{W})$ with nontrivial isomorphisms (i.e. those not in $\mathcal{W}^{\prod}$) are essential for $\mathbf{Mat}(\mathcal{W})$ to be a Lafont category. In a related construction in [36], a morphism is required to be invariant under the action of chosen permutations of basis vectors. This can be seen as a special case of requirements for morphisms (i.e. $B(g) \circ f_{a,b} \circ A(h) = f_{a',b'}$) in $\mathbf{Mat}(\mathcal{W})$: if $(\mathcal{A}, A)$ is an interpretation of a simple type, then $A(\varphi)$ is composed of structural isomorphisms in $\mathcal{W}$, which are permutations of basis vectors if $\mathcal{W} = \mathbf{CPM}_s$. □

5.3 P-visible Weighted Profunctors

Unfortunately, as mentioned at the beginning of this section, $\|\cdot\|$ is not functorial. This subsection introduces a subcategory of $\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$, which contains the interpretations of $\lambda_{\mathcal{W}}$-terms, and to which the restriction of $\|\cdot\|$ is a functor.

Definition 5.13 (P-visible profunctor)**.** Let S and T be simple types. A countable profunctor $F : [\![S]\!] \nrightarrow [\![T]\!]$ is *P-visible* if, for each $a \in [\![S]\!]$, $b \in [\![T]\!]$ and $x \in F(b, a)$, there exists a rigid resource term $x : a \vdash \widetilde{t} : b$ such that $\mathrm{fix}(x) \subseteq \mathrm{fix}(\widetilde{t})$ (here $\mathrm{fix}(x) = \{(\varphi, \psi) \mid \varphi \cdot x \cdot \psi = x\}$). A weighted profunctor is P-visible if so is the underlying profunctor. We write $(\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}})\restriction_{\mathrm{V}}$ for the subcategory whose objects are the interpretations of simple types and whose morphisms are the P-visible ones.

By definition, the interpretation of a $\lambda_{\mathcal{W}}$-term in $\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ lives in $(\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}})\restriction_{\mathrm{V}}$. It has the structure of a $\lambda_{\mathcal{W}}$-model induced by that of $\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$. The embedding $(\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}})\restriction_{\mathrm{V}} \to \mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ strictly preserves this structure.

Lemma 5.14. *$\|\cdot\| : (\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}})\restriction_{\mathrm{V}} \to \mathbf{Mat}(\mathcal{W})$ is a $\lambda_{\mathcal{W}}$-model morphism.*

Proof. (Sketch) The most nontrivial part is the functoriality of $\|\cdot\|$. Let $(F, \varpi^F) : (\mathcal{A}, A) \nrightarrow (\mathcal{B}, B)$ and $(G, \varpi^G) : (\mathcal{B}, B) \nrightarrow (\mathcal{C}, C)$ be P-visible profunctors. The key observation is that, thanks to P-visibility, for every $(y, x) \in G(c, b) \times F(b, a)$ and $f : b \longrightarrow b$, we have $(y, f \cdot x) = (y \cdot f, x)$ implies $f = \mathrm{id}$. Then each equivalence class of $G(c, b) \times F(b, a)$ by $\sim$ (where $\sim$ is that appears in the composition of profunctors) has exactly $\#\mathcal{B}(b, b)$ elements. Hence

$$\sum_{[(y,x)]_\sim \in (G(c,b)\times F(b,a))/\sim} \varpi^G(y) \circ \varpi^F(x) = \frac{1}{\#\mathcal{B}(b,b)} \sum_{(y,x)\in G(c,b)\times F(b,a)} \varpi^G(y) \circ \varpi^F(x).$$

A calculation using this fact and $\#\mathcal{B}(b, b) = \#\mathcal{B}^+(b, b) \times \#\mathcal{B}^-(b, b)$ shows $\|G \circ F\|_{c,a} = (\|G\| \circ \|F\|)_{c,a}$. □

Corollary 5.15. *For every program P, the interpretation of P in $\mathbf{Mat}(\mathcal{W})$ is $\sum_{\pi \in Eval(P)} \varpi(\pi)$ where the sum is that in $\mathcal{W}(I, I)$.*

So far, we have assumed that $\mathcal{W}$ is a $\Sigma\mathbf{Mon}_t$-SMCC with reciprocals n^{-1} for every natural number n. We can also deal with $\Sigma\mathbf{Mon}$-SMCCs such as $\mathbf{FdHilb}$ and $\mathbf{CPM}_s$ as follows. First $\Sigma\mathbf{Mon}_t$ is a reflexive full subcategory of $\Sigma\mathbf{Mon}$ (see [20]) and it is an exponential ideal (i.e., for every $\mathfrak{N} \in \Sigma\mathbf{Mon}_t$ and $\mathfrak{M} \in \Sigma\mathbf{Mon}$, we have $\mathfrak{M} \multimap \mathfrak{N} \in \Sigma\mathbf{Mon}_t$). A general result shows that $\Sigma\mathbf{Mon}_t$ is an SMCC and the adjunction between $\Sigma\mathbf{Mon}$ and $\Sigma\mathbf{Mon}_t$ is symmetric monoidal [20, Corollary 3.9]. Let us write $T : \Sigma\mathbf{Mon} \to \Sigma\mathbf{Mon}_t$ for the left adjoint of the inclusion $\Sigma\mathbf{Mon}_t \to \Sigma\mathbf{Mon}$. Thanks to a result in [30], a $\Sigma\mathbf{Mon}$-SMCC $\mathcal{W}$ induces a $\Sigma\mathbf{Mon}_t$-SMCC $T\mathcal{W}$ obtained by applying T to each hom-object. Let $\eta_{\mathcal{W}(I,I)}$ be the unit $\mathcal{W}(I, I) \to T(\mathcal{W}(I, I)) = (T\mathcal{W})(I, I)$, which is injective.

Theorem 5.16 (Adequacy)**.** *Assume that $\mathcal{W}$ is a $\Sigma\mathbf{Mon}$-SMCCs with reciprocals for natural numbers. For every $\lambda_{\mathcal{W}}$ program P, we have $\sum_{\pi \in Eval(P)} \varpi(\pi) \sqsubseteq \eta_{\mathcal{W}(I,I)}([\![P]\!]_{\mathbf{Mat}(T\mathcal{W})})$.*

Remark 5.17. Taking $\mathcal{W} = T(\mathbf{CPM}_s)$, this theorem shows that $\mathbf{Mat}(\mathcal{W})$ is adequate for the calculus in [36]; indeed the model $\mathbf{Mat}(\mathcal{W})$ is essentially the same model as in [36], at least on the interpretation of types, except that [36] applies a different completion to $\mathbf{CPM}_s$. Although $\mathbf{FdHilb}$ is also $\Sigma\mathbf{Mon}$-SMCC and their calculus can be embedded into $\lambda_{\mathbf{FdHilb}}$, the category $\mathbf{Mat}(T(\mathbf{FdHilb}))$ is not an adequate model. This is because the sum in $\mathbf{FdHilb}(I, I) = \mathbb{C}$ differs from what we needed; recall that the meaning of a $\lambda_{\mathbf{FdHilb}}$ program P is $\sum_{\pi \in Eval(P)} \varpi(\pi)\varpi(\pi)^*$, not $\sum_{\pi \in Eval(P)} \varpi(\pi)$. □

Acknowledgments

The authors would like to thank Marcelo Fiore for insightful discussions. This work was supported by JSPS KAKENHI 15H05706, 16K16004 and 18K11156, and EPSRC grant EP/M023974/1.

References

[1] Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full Abstraction for PCF. *Information and Computation*, 163(2):409–470, dec 2000.

[2] John C. Baez and James Dolan. Higher dimensional algebra III: n-categories and the algebra of opetopes. *Adv. Math*, 135:145–206, 1998.

[3] Marek A. Bednarczyk, Andrzej M. Borzyszkowski, and Wieslaw Pawlowski. Generalized congruences – epimorphisms in cat. *Theory and Applications of Categories*, 5:266–280, 1999.

[4] Jean Bénabou. *Introduction to bicategories*, pages 1–77. Springer Berlin Heidelberg, Berlin, Heidelberg, 1967.

[5] Jean Bénabou. Distributors at work. Course notes, TU Darmstadt, 2000.

[6] François Bergeron, Gilbert Labelle, and Pierre Leroux. *Combinatorial Species and Tree-like Structures*. Cambridge Univ. Press, 1997.

[7] Mario Jose Cáccamo, Martin Hyland, and Glynn Winskel. Lecture notes in category theory. 2005.

[8] Vincent Danos and Thomas Ehrhard. Probabilistic coherence spaces as a model of higher-order probabilistic computation. *Inf. Comput.*, 209(6):966–991, 2011.

[9] Brian Day and Ross Street. Monoidal bicategories and hopf algebroids. *Advances in Mathematics*, 129(1):99 – 157, 1997.

[10] Thomas Ehrhard and Laurent Regnier. Uniformity and the Taylor expansion of ordinary lambda-terms. *Theor. Comput. Sci.*, 403(2-3):347–372, 2008.

[11] Thomas Ehrhard, Christine Tasson, and Michele Pagani. Probabilistic coherence spaces are fully abstract for probabilistic PCF. In *POPL*, pages 309–320, 2014.

[12] Marcelo P. Fiore. Mathematical models of computational and combinatorial structures. In *FoSSaCS*, pages 25–46, 2005.

[13] Marcelo P. Fiore, Nicola Gambino, J. Martin E. Hyland, and Glynn Winskel. The cartesian closed bicategory of generalised species of structures. *J. London Maths. Soc.*, 77:203–220, 2007.

[14] Richard Garner and Michael Shulman. Enriched categories as a free cocompletion. *Advances in Mathematics*, 289:1 – 94, 2016.

[15] Jean-Yves Girard. Normal functors, power series and λ-calculus. *Ann. Pure Appl. Logic*, 37(2):129–177, 1988.

[16] Esfandiar Haghverdi. Partially additive categories and fully complete models of linear logic. In *TLCA*, pages 197–216, 2001.

[17] Esfandiar Haghverdi and Philip J. Scott. A categorical model for the geometry of interaction. *Theor. Comput. Sci.*, 350(2-3):252–274, 2006.

[18] Russell Harmer and Guy McCusker. A fully abstract game semantics for finite nondeterminism. In *LICS*, pages 422–430. IEEE Comput. Soc, 1999.

[19] Ryu Hasegawa. Two applications of analytic functors. *Theor. Comput. Sci.*, 272(1-2):113–175, 2002.

[20] Naohiko Hoshino. A representation theorem for unique decomposition categories. *Electr. Notes Theor. Comput. Sci.*, 286:213–227, 2012.

[21] J. M. E. Hyland and C.-H. Luke Ong. On full abstraction for PCF: I, II, and III. *Information and Computation*, 163(2):285–408, 2000.

[22] Andre Joyal. Une théorie combinatoire des séries formelles. *Adv. Math.*, 42:1fi??82, 1981.

[23] Andre Joyal. Foncteurs analytiques et espèces de structures. In *Combinatoire Énumérative*, volume 1234 of *Lecture Notes in Mathematics*, page 126fi??159. Springer, Berlin, 1986.

[24] G. M. Kelly and Ross Street. Review of the elements of 2-categories. In Gregory M. Kelly, editor, *Category Seminar*, pages 75–103, Berlin, Heidelberg, 1974. Springer Berlin Heidelberg.

[25] Stephen Lack. *A 2-Categories Companion*, pages 105–191. Springer New York, 2010.

[26] James Laird. Fixed points in quantitative semantics. In *LICS*, 2016.

[27] James Laird. From qualitative to quantitative semantics - by change of base. In *FoSSaCS*, pages 36–52, 2017.

[28] Jim Laird, Giulio Manzonetto, Guy McCusker, and Michele Pagani. Weighted relational models of typed lambda-calculi. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*, pages 301–310, 2013.

[29] François Lamarche. Quantitative domains and infinitary algebras. *Theor. Comput. Sci.*, 94(1):37–62, 1992.

[30] R. B. B. Lucyshyn-Wright. Relative symmetric monoidal closed categories I: Autoenrichment and change of base. *Theory and Applications of Categories*, 31:138–174, 2016.

[31] Damiano Mazza, Luc Pellissier, and Pierre Vial. Polyadic approximations, fibrations and intersection types. *PACMPL*, 2(POPL):6:1–6:28, 2018.

[32] Paul-André Melliès. Asynchronous games 1: Uniformity by group invariance. unpublished manuscript, 2003.

[33] Paul-André Melliès. *Categorical Semantics of Linear Logic*. Panoramas et Synthèses 27. 2009.

[34] Paul-André Melliès. Dialogue categories and chiralities. *Publ. Res. Inst. Math. Sci.*, (52):359–412, 2016.

[35] Paul-André Melliès, Nicolas Tabareau, and Christine Tasson. An explicit formula for the free exponential modality of linear logic. In *ICALP*, pages 247–260, 2009.

[36] Michele Pagani, Peter Selinger, and Benoît Valiron. Applying quantitative semantics to higher-order quantum computing. In *POPL*, pages 647–658, 2014.

[37] Luc Pellissier. *Réduction et Approximation Linéaires*. PhD thesis, Université Paris 13, 2017.

[38] Peter Selinger. Towards a quantum programming language. *Mathematical Structures in Computer Science*, 14(4):56, 2004.

[39] Peter Selinger and Benoît Valiron. On a fully abstract model for a quantum linear functional language. *Electronic Notes in Theoretical Computer Science*, 210(C):123–137, jul 2008.

[40] Michael Shulman. Framed bicategories and monoidal fibrations. *Theory and Applications of Categories*, 20(18):650–738, 2008.

[41] Michael Stay. Compact closed bicategories. *Theory and Applications of Categories*, 31(26):755–798, 2016.

[42] Takeshi Tsukada, Kazuyuki Asada, and C.-H. Luke Ong. Generalised species of rigid resource terms. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12, 2017.

[43] Takeshi Tsukada and C.-H. Luke Ong. Nondeterminism in game semantics via sheaves. In *LICS*, 2015.

[44] Lionel Vaux. The algebraic lambda calculus. *Mathematical Structures in Computer Science*, 19(5):1029–1059, 2009.

[45] R.J. Wood. Abstract proarrows. I. *Cah. Topologie Géom. Différ. Catégoriques*, 23:279–290, 1982.

[46] R.J. Wood. Proarrows. II. *Cah. Topologie Géom. Différ. Catégoriques*, 26:135–168, 1985.

A Supplementary Materials for Section 2

A.1 Language Definition

Figure 7 is the complete list of typing rule of the simple type system for $\lambda_{\mathcal{W}}$. One-step reduction relation is defined by the rules in Fig. 8.

Configurations $[\vec{x}_1 = e_1, M_1]$ and $[\vec{x}_2 = e_2, M_2]$ are *α-equivalent* if $M_1 = M_2\{\vec{x}_2/\vec{x}_1\}$ and $e_1 = e_2$. Given a configuration $[\vec{x} = e, M]$, a path π determines uniquely up to α-equivalence a configuration $[\vec{y} = e, N]$ such that $[\vec{x} = e, M] \xrightarrow{\pi} [\vec{y} = e, N]$ (if exists).

Lemma A.1. *Assume* $[\vec{x} = e, M] \xrightarrow{\pi} [\vec{y}_i = e_i, N_i]$ *for* $i = 1, 2$. *Then there exists* $[\vec{y}'_1 = e'_1, N'_1]$ *such that*

- $[\vec{y}_1 = e_1, N_1] \xrightarrow{\epsilon} [\vec{y}'_1 = e'_1, N'_1]$ *and*
- $[\vec{y}'_1 = e'_1, N'_1]$ *is α-equivalent to* $[\vec{y}_2 = e_2, N_2]$.

Corollary A.2. *If* $[\epsilon = \mathrm{id}_I, P] \xrightarrow{\pi} [\epsilon = e_i, ()]$ *for* $i = 1, 2$, *then* $e_1 = e_2$.

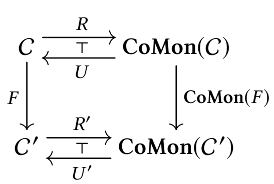
A.2 Morphisms of $\lambda_{\mathcal{W}}$-models

Definition A.3 ($\lambda_{\mathcal{W}}$-model morphism)**.** For $\lambda_{\mathcal{W}}$-models C and C', a *$\lambda_{\mathcal{W}}$-model morphism F* from C to C' is a functor $F : C \to C'$ with the following structures/properties:

- F is a *linear functor*, i.e.,
  - F is a strong monoidal functor
  - the canonical morphism $F(a \multimap b) \to F(a) \multimap' F(b)$ is isomorphic,
  - F is a comonad morphism
  - the comonad-morphism structure $\zeta : F! \Rightarrow !'F$ is a monoidal natural isomorphism.
- F preserves finite biproducts.
- F preserves the initial algebra of $L_a(X) \triangleq I \oplus (a \otimes X)$, i.e., F induces a functor L_F from the category of L_a-algebras to that of $L_{F(a)}$-algebras (by the preservation of monoidal products and coproducts), and then L_F preserves the initial object.
- F preserves the interpretation of base types up to iso, and preserves constants (inserting the canonical iso).

In this paper, we obtain not merely linear-non-linear but Lafont categories. Between Lafont categories C and C', one might define the following notion of morphism.

Definition A.4 (Lafont morphism)**.** A strong monoidal closed functor $F : C \to C'$ induces a strong monoidal functor $\mathbf{CoMon}(F) : \mathbf{CoMon}(C) \to \mathbf{CoMon}(C')$ such that $F \circ U = U' \circ \mathbf{CoMon}(F)$ (as monoidal functors) where U and U' are the forgetful functors in the following diagram:



Then by the bijective correspondence induced by the adjunctions $U \dashv R$ and $U' \dashv R'$ (see "internal adjunctions" in Section C.1), we obtain a canonical natural transformation $\varphi : \mathbf{CoMon}(F) \circ R \Rightarrow R' \circ F$ from the identity $id : U' \circ \mathbf{CoMon}(F) \Rightarrow F \circ U$. Then F is a *Lafont functor* if this φ is isomorphic.

Note that, for a strong monoidal closed functor, being a linear functor (as in Definition A.3) requires a *structure* (i.e., ζ), while being a Lafont functor is a *property*. Still, in fact, the two notions are equivalent for functors between Lafont categories:

Proposition A.5. *For a functor F between Lafont categories C and C', if F is a linear functor, then the comonad-morphism structure $\zeta : F! \Rightarrow !'F$ is necessarily the canonical one, i.e. $\zeta = U' \circ \varphi$ where φ is defined in Definition A.4. Then F is a Lafont functor.*

Conversely, any Lafont functor is a linear functor (with the canonical structure $\zeta = U' \circ \varphi$).

Proof. We first show the former statement. By $\zeta : F \circ U \circ R \Rightarrow U' \circ R' \circ F$, we can construct a natural isomorphism $\varphi' : \mathbf{CoMon}(F) \circ R \Rightarrow R' \circ F$ such that

$$U' \circ \varphi' = \zeta.$$

(To construct a comonoid homomorphism $\varphi'_A : \mathbf{CoMon}(F)(RA) \to R'FA$, the underlying morphism is given by ζ, as required. Then this is comonoid homomorphism because ζ is by definition a monoidal natural transformation and hence respects the comonoid structures.) Then, recall that the canonical natural transformation φ is defined from the identity $id : U' \circ \mathbf{CoMon}(F) \Rightarrow F \circ U$, and hence

$$(\varepsilon' \circ F) \bullet (U' \circ \varphi) = F \circ \varepsilon : U' \circ \mathbf{CoMon}F \circ R \Rightarrow F.$$

Also, by definition of comonad morphism ζ, we have

$$F \circ \varepsilon = (\varepsilon' \circ F) \bullet \zeta = (\varepsilon' \circ F) \bullet (U' \circ \varphi').$$

Since the mapping $(\varepsilon' \circ F) \bullet (U' \circ (-))$ is bijective, we have $\varphi = \varphi'$, and hence

$$\zeta = U' \circ \varphi' = U' \circ \varphi.$$

Since U' reflects isromophism, φ is isomorphic and hence F is a Lafont functor.

On the converse statement, the canonical natural transformation φ is necessarily a monoidal natural transformation (to calculate this, use the bijection $(\varepsilon' \circ F) \bullet (U' \circ (-))$), and hence we have the monoidal comonad-morphism structure $\zeta = U' \circ \varphi$. □

B Supplementary Materials for Section 3

B.1 On refinement types

Figure 9 is the list of rules for the refinement relation. Figure 10 defines isomorphisms between refinement types. Here we write $\varphi : a \overset{+}{\cong} a'$ (resp. $\varphi : a \overset{-}{\cong} a'$) to mean that φ is a positive (resp. negative) isomorphism. We write $\varphi : a \cong a'$ if the polarity of φ is not important.

For $\varphi : a \cong a'$ and $\psi : a' \cong a''$, their composite $(\psi \circ \varphi) : a \cong a''$ is defined in a natural way. For example,

$$(\psi_1 \otimes \psi_2) \circ (\varphi_1 \otimes \varphi_2) \;\triangleq\; (\psi_1 \circ \varphi_1) \otimes (\psi_2 \circ \varphi_2)$$

and

$$\langle \sigma'; (\psi_j)_j \rangle \circ \langle \sigma; (\varphi_i)_i \rangle \;\triangleq\; \langle \sigma \circ \sigma'; (\psi_j \circ \varphi_{\sigma'(j)})_j \rangle.$$

The definition of the inverse is also straightforward, e.g.,

$$(\varphi \otimes \psi)^{-1} \;\triangleq\; \varphi^{-1} \otimes \psi^{-1}$$

and

$$\langle \sigma; \varphi_1, \ldots, \varphi_n \rangle^{-1} \;\triangleq\; \langle \sigma^{-1}; \varphi_{\sigma^{-1}(1)}, \ldots, \varphi_{\sigma^{-1}(n)} \rangle.$$

The definition of the identity is obvious.

Lemma B.1. *Every $\varphi : a \cong a'$ can be uniquely factorised as $\varphi = \psi^+_1 \circ \psi^-_1$ with $\psi^-_1 : a \overset{-}{\cong} a''$ and $\psi^+_1 : a'' \overset{+}{\cong} a'$ for some a''. Similarly every $\varphi : a \cong a'$ can also be uniquely factorised as $\varphi = \psi^-_2 \circ \psi^+_1$.*

$$\frac{}{\Delta \mid x : S \vdash x : S} \qquad \frac{}{\Delta, x : S \mid\ \vdash x : S} \qquad \frac{c \in \mathcal{W}(\mathsf{a}_1 \otimes \cdots \otimes \mathsf{a}_n, \mathsf{a}'_1 \otimes \cdots \otimes \mathsf{a}'_m)}{\Delta \mid\ \vdash c : \mathsf{a}_1 \otimes \cdots \otimes \mathsf{a}_n \multimap \mathsf{a}'_1 \otimes \cdots \otimes \mathsf{a}'_m}$$

$$\frac{\Delta \mid \Gamma, x : S \vdash M : T}{\Delta \mid \Gamma \vdash \lambda x.M : S \multimap T} \qquad \frac{\Delta \mid \Gamma_1 \vdash V : S \multimap T \qquad \Delta \mid \Gamma_2 \vdash W : S}{\Delta \mid \Gamma_1, \Gamma_2 \vdash V\,W : T} \qquad \frac{\Delta \mid \Gamma \vdash M : T \qquad \Delta \mid \Gamma \vdash N : T}{\Delta \mid \Gamma \vdash M \diamond N : T} \qquad \frac{\Delta \mid\ \vdash V : !(S \multimap T) \multimap S \multimap T}{\Delta \mid\ \vdash \mathrm{Y}\,V : S \multimap T}$$

$$\frac{}{\Delta \mid\ \vdash () : I} \qquad \frac{\Delta \mid \Gamma_1 \vdash M : I \qquad \Delta \mid \Gamma_2 \vdash N : T}{\Delta \mid \Gamma_1, \Gamma_2 \vdash M; N : T} \qquad \frac{\Delta \mid \Gamma_1 \vdash M : S \qquad \Delta \mid \Gamma_2, x : S \vdash N : T}{\Delta \mid \Gamma_1, \Gamma_2 \vdash \mathtt{let}\, x = M \,\mathtt{in}\, N : T}$$

$$\frac{\Delta \mid\ \vdash V : T}{\Delta \mid\ \vdash !V : !T} \qquad \frac{\Delta \mid \Gamma_1 \vdash V : !S \qquad \Delta, x : S \mid \Gamma_2 \vdash N : T}{\Delta \mid \Gamma_1, \Gamma_2 \vdash \mathtt{let}\, !x = V \,\mathtt{in}\, N : T}$$

$$\frac{\Delta \mid \Gamma_1 \vdash V : S \qquad \Delta \mid \Gamma_2 \vdash W : T}{\Delta \mid \Gamma_1, \Gamma_2 \vdash V \otimes W : S \otimes T} \qquad \frac{\Delta \mid \Gamma_1 \vdash V : S \otimes S' \qquad \Delta \mid \Gamma_2, x : S, x' : S' \vdash M : T}{\Delta \mid \Gamma_1, \Gamma_2 \vdash \mathtt{let}\, x \otimes x' = V \,\mathtt{in}\, M : T}$$

$$\frac{\Delta \mid \Gamma \vdash V : S}{\Delta \mid \Gamma \vdash \mathtt{inl}(V) : S \oplus T} \qquad \frac{\Delta \mid \Gamma \vdash W : T}{\Delta \mid \Gamma \vdash \mathtt{inr}(W) : S \oplus T}$$

$$\frac{\Delta \mid \Gamma_1 \vdash V : S \oplus S' \qquad \Delta \mid \Gamma_2, x : S \vdash M : T \qquad \Delta \mid \Gamma_2, x' : S' \vdash N : T}{\Delta \mid \Gamma_1, \Gamma_2 \vdash \mathtt{case}\, V \,\mathtt{of}\, (\mathtt{inl}(x) : M \mid \mathtt{inr}(x') : N) : T}$$

$$\frac{}{\Delta \mid\ \vdash \mathtt{Nil} : \mathtt{list}\, T} \qquad \frac{\Delta \mid \Gamma_1 \vdash V : T \qquad \Delta \mid \Gamma_2 \vdash W : \mathtt{list}\, T}{\Delta \mid \Gamma_1, \Gamma_2 \vdash V{::}W : \mathtt{list}\, T}$$

$$\frac{\Delta \mid \Gamma_1 \vdash V : \mathtt{list}\, S \qquad \Delta \mid \Gamma_2 \vdash M : T \qquad \Delta \mid \Gamma_2, x : S, y : \mathtt{list}\, S \vdash N : T}{\Delta \mid \Gamma_1, \Gamma_2 \vdash \mathtt{case}\, V \,\mathtt{of}\, (\mathtt{Nil} : M \mid x{::}y : N) : T}$$

$$\frac{\Delta \mid x_1 : S_1, \ldots, x_n : S_n \vdash M : T \qquad \sigma \in \mathfrak{S}_n}{\Delta \mid x_{\sigma(1)} : S_{\sigma(1)}, \ldots, x_{\sigma(n)} : S_{\sigma(n)} \vdash M : T} \qquad \frac{x_1 : S_1, \ldots, x_n : S_n \mid \Gamma \vdash M : T \qquad \sigma \in \mathfrak{S}_n}{x_{\sigma(1)} : S_{\sigma(1)}, \ldots, x_{\sigma(n)} : S_{\sigma(n)} \mid \Gamma \vdash M : T}$$

Figure 7. Simple typing rules ($\mathfrak{S}_n$ is the set of permutations of n elements)

Proof. By induction on the size of a. (We need the induction hypothesis of the latter claim to prove the former when $a = a_1 \multimap a_2$.)

The only nontrivial case is that $\varphi = \langle \sigma; \varphi_1, \ldots, \varphi_n \rangle : \langle a_1, \ldots, a_n \rangle \cong \langle a'_1, \ldots, a'_n \rangle$. Let us decompose it into $\psi^+ \circ \psi^-$; the other case is similar. Then $\varphi_i : a_{\sigma(i)} \cong a'_i$. By the induction hypothesis, we have $\varphi_i = \psi_i^+ \circ \psi_i^-$ for each i. Let

$$\varphi^+ \triangleq \langle \sigma; \psi_1^+, \ldots, \psi_n^+ \rangle$$
$$\varphi^- \triangleq \langle \mathrm{id}; \psi^-_{\sigma^{-1}(1)}, \ldots, \psi^-_{\sigma^{-1}(n)} \rangle.$$

Then

$$\varphi^+ \circ \varphi^- \;=\; \langle \sigma; (\psi_i^+ \circ \psi^-_{\sigma(\sigma^{-1}(i))})_i \rangle \;=\; \varphi.$$

□

Isomorphisms between refinement type environments is defined as follows. For type bindings of non-linear type environments, we define

$$\frac{\sigma \in \mathfrak{S}_n \qquad \varphi_i : a_{\sigma(i)} \cong a' \; (\forall i \le n)}{\langle \sigma; \varphi_1, \ldots, \varphi_n \rangle : (\langle \vec{x} \rangle : \langle \vec{a} \rangle) \cong (\langle \vec{y} \rangle : \langle \vec{a'} \rangle)}.$$

Then we define

$$\frac{\varphi_i : (\langle \vec{x}_i \rangle : \langle \vec{a}_i \rangle) \cong (\langle \vec{y}_i \rangle : \langle \vec{a'}_i \rangle) \; (\forall i \le n)}{(\varphi_1, \ldots, \varphi_n) : \begin{array}{l} (\langle \vec{x}_1 \rangle : \langle \vec{a}_1 \rangle, \ldots, \langle \vec{x}_n \rangle : \langle \vec{a}_n \rangle) \\ \quad \cong (\langle \vec{y}_1 \rangle : \langle \vec{a'}_1 \rangle, \ldots, \langle \vec{y}_n \rangle : \langle \vec{a'}_n \rangle) \end{array}}$$

and

$$\frac{\varphi_i : a_i \cong a'_i \; (\forall i \le n)}{(\varphi_1, \ldots, \varphi_n) : (x_1 : a_1, \ldots, x_n : a_n) \cong (y_1 : a'_1, \ldots, y_n : a'_n)}.$$

Finally

$$\frac{\varphi : \Theta \cong \Theta' \qquad \psi : \Xi \cong \Xi'}{(\varphi, \psi) : (\Theta \mid \Xi) \cong (\Theta' \mid \Xi')}.$$

B.2 On the rigid resource calculus

Figures 11 and 12 give the complete list of rules relating rigid resource raw-terms and $\lambda_{\mathcal{W}}$-terms.

We define substitution and action of isomorphisms. We first define a special kind of substitution, $t\{[\varphi]x/y\}$: it is the same as the standard substitution but

$$([\psi]y)\{[\varphi]x\} \;\triangleq\; [\psi \circ \varphi]x.$$

The action of isomorphism is defined by the rules in Fig. 13. Then we

(a) Classical control flow

$$
\begin{aligned}
[\vec{x} = e, E[(\lambda y.M)\, V]] &\xrightarrow{0} [\vec{x} = e, E[M\{V/y\}]] \\
[\vec{x} = e, E[M_1 \diamond M_2]] &\xrightarrow{i} [\vec{x} = e, E[M_i]] \\
[\vec{x} = e, E[\mathbf{Y}\, V]] &\xrightarrow{0} [\vec{x} = e, E[V\, !(\lambda x.\mathbf{Y}\, V\, x)]] \\
[\vec{x} = e, E[(); M]] &\xrightarrow{0} [\vec{x} = e, E[M]] \\
[\vec{x} = e, E[\mathtt{let}\, y = V \,\mathtt{in}\, M]] &\xrightarrow{0} [\vec{x} = e, E[M\{V/y\}]] \\
[\vec{x} = e, E[\mathtt{let}\, !y = !V \,\mathtt{in}\, M]] &\xrightarrow{0} [\vec{x} = e, E[M\{V/y\}]] \\
[\vec{x} = e, E[\mathtt{let}\, y \otimes z = V \otimes W \,\mathtt{in}\, M]] &\xrightarrow{0} [\vec{x} = e, E[M\{V/y, W/z\}]] \\
[\vec{x} = e, E[\mathtt{case}\, \mathtt{inl}(V) \,\mathtt{of}\, (x : M \mid y : N)]] &\xrightarrow{0} [\vec{x} = e, E[M\{V/x\}]] \\
[\vec{x} = e, E[\mathtt{case}\, \mathtt{inr}(V) \,\mathtt{of}\, (x : M \mid y : N)]] &\xrightarrow{0} [\vec{x} = e, E[N\{V/x\}]] \\
[\vec{x} = e, E[\mathtt{case}\, \mathtt{Nil} \,\mathtt{of}\, (\mathtt{Nil} : M \mid y{::}z : N)]] &\xrightarrow{0} [\vec{x} = e, E[M]] \\
[\vec{x} = e, E[\mathtt{case}\, V{::}W \,\mathtt{of}\, (\mathtt{Nil} : M \mid y{::}z : N)]] &\xrightarrow{0} [\vec{x} = e, E[N\{V/y, W/z\}]]
\end{aligned}
$$

(b) "Non-classical" data

$$
\begin{aligned}
[\vec{x}^{\vec{\mathsf{a}}}\vec{y}^{\vec{\mathsf{b}}} = e, E[c^{\vec{\mathsf{a}} \multimap \vec{\mathsf{a}}'}(\vec{x})]] &\xrightarrow{0} [\vec{z}^{\vec{\mathsf{a}}'}\vec{y}^{\vec{\mathsf{b}}} = ((c \otimes \mathrm{id}_{\vec{\mathsf{b}}}) \circ e), E[\vec{z}]] \\
[x_1 \dots x_n = e, P] &\xrightarrow{\epsilon} [x_{\sigma(1)} \dots x_{\sigma(n)} = \sigma \circ e, P].
\end{aligned}
$$

Figure 8. Operational semantics. Here σ is a permutation $\sigma \in \mathfrak{S}_n$ of n elements, identified with the structural isomorphism $\mathsf{a}_1 \otimes \cdots \otimes \mathsf{a}_n \to \mathsf{a}_{\sigma(1)} \otimes \cdots \otimes \mathsf{a}_{\sigma(n)}$ in $\mathcal{W}$.

$$
\frac{}{\mathsf{a} \lhd \mathsf{a}} \qquad
\frac{a \lhd S \qquad b \lhd T}{a \multimap b \lhd S \multimap T} \qquad
\frac{}{() \lhd I} \qquad
\frac{a \lhd S \qquad b \lhd T}{a \otimes b \lhd S \otimes T} \qquad
\frac{a_1 \lhd S \qquad \dots \qquad a_n \lhd S}{\langle a_1, \dots, a_n \rangle \lhd !S}
$$

$$
\frac{a \lhd S}{a \oplus \bullet \lhd S \oplus T} \qquad
\frac{b \lhd T}{\bullet \oplus b \lhd S \oplus T} \qquad
\frac{}{\mathtt{nil} \lhd \mathtt{list}\, S} \qquad
\frac{a \lhd S \qquad b \lhd \mathtt{list}\, S}{a{::}b \lhd \mathtt{list}\, S}
$$

Figure 9. Refinement relation

$$
\frac{}{\mathrm{id}_{\mathsf{a}} : \mathsf{a} \stackrel{\pm}{\cong} \mathsf{a}} \qquad
\frac{\varphi : a' \stackrel{\mp}{\cong} a \qquad \psi : b \stackrel{\pm}{\cong} b'}{\varphi \multimap \psi : a \multimap b \stackrel{\pm}{\cong} a' \multimap b'} \qquad
\frac{}{\mathrm{id}_{()} : () \stackrel{\pm}{\cong} ()} \qquad
\frac{\varphi : a \stackrel{\pm}{\cong} a' \qquad \psi : b \stackrel{\pm}{\cong} b'}{\varphi \otimes \psi : a \otimes b \stackrel{\pm}{\cong} a' \otimes b'}
$$

$$
\frac{\varphi : a \stackrel{\pm}{\cong} a'}{\varphi \oplus \bullet : a \oplus \bullet \stackrel{\pm}{\cong} a' \oplus \bullet} \qquad
\frac{\psi : a \stackrel{\pm}{\cong} a'}{\bullet \oplus \psi : \bullet \oplus a \stackrel{\pm}{\cong} \bullet \oplus a'} \qquad
\frac{}{\mathrm{id}_{\mathtt{nil}} : \mathtt{nil} \stackrel{\pm}{\cong} \mathtt{nil}} \qquad
\frac{\varphi : a \stackrel{\pm}{\cong} a' \qquad \psi : b \stackrel{\pm}{\cong} b'}{\varphi{::}\psi : a{::}b \stackrel{\pm}{\cong} a'{::}b'}
$$

$$
\frac{\sigma \in \mathfrak{S}_n \qquad \varphi_1 : a_{\sigma(1)} \stackrel{+}{\cong} b_1 \quad \dots \quad \varphi_n : a_{\sigma(n)} \stackrel{+}{\cong} b_n}{\langle \sigma; \varphi_1, \dots, \varphi_n \rangle : \langle a_1, \dots, a_n \rangle \stackrel{+}{\cong} \langle b_1, \dots, b_n \rangle} \qquad
\frac{\varphi_1 : a_{\sigma(1)} \stackrel{-}{\cong} b_1 \quad \dots \quad \varphi_n : a_{\sigma(n)} \stackrel{-}{\cong} b_n}{\langle \mathrm{id}; \varphi_1, \dots, \varphi_n \rangle : \langle a_1, \dots, a_n \rangle \stackrel{-}{\cong} \langle b_1, \dots, b_n \rangle}
$$

Figure 10. Isomorphisms between refinement types with polarity annotation (double sign in same order)

define the general substitution $t\{u/x\}$ as the standard substitution but

$$([\varphi]x)\{u/x\} \;\triangleq\; [\varphi] \cdot u.$$

The two definitions of $t\{[\varphi]x/y\}$ coincide.

The action of $\varphi : (\Theta \mid \Xi) \cong (\Theta' \mid \Xi')$ to rigid resource raw-terms is defined as follows.

- Assume $(\Theta, \langle x_1, \dots, x_n \rangle : \langle a_1, \dots, a_n \rangle, \Theta') \lhd \Delta \mid \Xi \lhd \Gamma \vdash t : b \lhd M : S$.
  - Let $\sigma \in \mathfrak{S}_n$, which induces an isomorphism

$$
\begin{aligned}
\varphi \;:\; &(\Theta, \langle y_1, \dots, y_n \rangle : \langle a_{\sigma(1)}, \dots, a_{\sigma(n)} \rangle, \Theta' \mid \Xi) \\
&\cong (\Theta, \langle x_1, \dots, x_n \rangle : \langle a_1, \dots, a_n \rangle, \Theta' \mid \Xi).
\end{aligned}
$$

  For this case, we define $t\{\varphi\} \triangleq t\{y_1/x_{\sigma(1)}, \dots, y_n/x_{\sigma(n)}\}$.

$$\frac{\varphi : a \cong a' \qquad O_1 \lhd \Delta_1 \quad a \lhd S \quad O_2 \lhd \Delta_2}{(O_1, \langle x \rangle : \langle a \rangle, O_2) \lhd (\Delta_1, y : S, \Delta_2) \mid\ \vdash [\varphi]x : a' \lhd y : S} \qquad \frac{\varphi : a \cong a' \qquad O \lhd \Delta \quad a \lhd S}{O \lhd \Delta \mid (x : a) \lhd (y : S) \vdash [\varphi]x : a' \lhd y : S} \qquad \frac{O \lhd \Delta}{O \lhd \Delta \mid\ \vdash c^S : S \lhd c^S : S}$$

$$\frac{\Theta \lhd \Delta \mid (\Xi, x : a) \lhd (\Gamma, x : S) \vdash t : b \lhd M : T}{\Theta \lhd \Delta \mid \Xi \lhd \Gamma \vdash \lambda x.t : a \multimap b \lhd \lambda x.M : S \multimap T} \qquad \frac{\Theta_1 \lhd \Delta \mid \Xi_1 \lhd \Gamma_1 \vdash v : a \multimap b \lhd V : S \multimap T \qquad \Theta_2 \lhd \mid \Xi_2 \lhd \Gamma_2 \vdash w : a \lhd W : S}{(\Theta_1 \wedge \Theta_2) \lhd \Delta \mid (\Xi_1, \Xi_2) \lhd (\Gamma_1, \Gamma_2) \vdash v\, w : b \lhd V\, W : T}$$

$$\frac{\Theta \lhd \Delta \mid \Xi \lhd \Gamma \vdash t : a \lhd M : S}{\Theta \lhd \Delta \mid \Xi \lhd \Gamma \vdash t \diamond \bullet : a \lhd M \diamond N : S} \qquad \frac{\Theta \lhd \Delta \mid \Xi \lhd \Gamma \vdash t : a \lhd N : S}{\Theta \lhd \Delta \mid \Xi \lhd \Gamma \vdash \bullet \diamond t : a \lhd M \diamond N : S} \qquad \frac{}{O \lhd \Delta \mid\ \vdash () : () \lhd () : I}$$

$$\frac{\Theta_0 \lhd \Delta \mid\ \vdash v : \langle b_1, \ldots, b_n \rangle \multimap a \lhd V : !T \multimap T \qquad \Theta_i \lhd \Delta \mid\ \vdash w_i : b_i \lhd \lambda x.\mathrm{Y} V\, x : T \quad (1 \le {}^{\forall} i \le n)}{(\Theta_0 \wedge \cdots \wedge \Theta_n) \lhd \Delta \mid\ \vdash ((); (v\, \langle w_1, \ldots, w_n \rangle)) : a \lhd \mathrm{Y} V : T}$$

$$\frac{\Theta_1 \lhd \Delta \mid \Xi_1 \lhd \Gamma_1 \vdash s : () \lhd M : I \qquad \Theta_2 \lhd \Delta \mid \Xi_2 \lhd \Gamma_2 \vdash t : a \lhd N : S}{(\Theta_1 \wedge \Theta_2) \lhd \Delta \mid (\Xi_1, \Xi_2) \lhd (\Gamma_1, \Gamma_2) \vdash s; t : a \lhd M; N : S}$$

$$\frac{\Theta_1 \lhd \Delta \mid \Xi_1 \lhd \Gamma_1 \vdash s : a \lhd M : S \qquad \Theta_2 \lhd \Delta \mid (\Xi_2, x : a) \lhd (\Gamma_2, y : S) \vdash t : b \lhd N : T}{(\Theta_1 \wedge \Theta_2) \lhd \Delta \mid (\Xi_1, \Xi_2) \lhd (\Gamma_1, \Gamma_2) \vdash \mathtt{let}\, x = s\, \mathtt{in}\, t : b \lhd \mathtt{let}\, y = M\, \mathtt{in}\, N : T}$$

$$\frac{\Theta_i \lhd \Delta \mid\ \vdash v_i : a_i \lhd V : S \quad (\forall i \le n)}{(\Theta_1 \wedge \cdots \wedge \Theta_n) \lhd \Delta \mid\ \vdash \langle v_1, \ldots, v_n \rangle : \langle a_1, \ldots, a_n \rangle \lhd !V : !S}$$

$$\frac{\Theta_1 \lhd \Delta \mid \Xi_1 \lhd \Gamma_1 \vdash v : \langle a_1, \ldots, a_n \rangle \lhd V : !S \qquad (\Theta_2, \langle x_1, \ldots, x_n \rangle : \langle a_1, \ldots, a_n \rangle) \lhd (\Delta, y : S) \mid \Xi_2 \lhd \Gamma_2 \vdash t : b \lhd N : T}{(\Theta_1 \wedge \Theta_2) \lhd \Delta \mid (\Xi_1, \Xi_2) \lhd (\Gamma_1, \Gamma_2) \vdash \mathtt{let}\, \langle x_1, \ldots, x_n \rangle = v\, \mathtt{in}\, t : b \lhd \mathtt{let}\, !y = M\, \mathtt{in}\, N : T}$$

$$\frac{\Theta_1 \lhd \Delta \mid \Xi_1 \lhd \Gamma_1 \vdash v : a \lhd V : S \qquad \Theta_2 \lhd \Delta \mid \Xi_2 \lhd \Gamma_2 \vdash w : b \lhd W : T}{(\Theta_1 \wedge \Theta_2) \lhd \Delta \mid (\Xi_1, \Xi_2) \lhd (\Gamma_1, \Gamma_2) \vdash v \otimes w : a \otimes b \lhd V \otimes W : S \otimes T}$$

$$\frac{\Theta_1 \lhd \Delta \mid \Xi_1 \lhd \Gamma_1 \vdash v : a \otimes a' \lhd V : S \otimes S' \qquad \Theta_2 \lhd \Delta \mid (\Xi_2, x : a, y' : a') \lhd (\Gamma_1, y : S, y' : S') \vdash t : b \lhd M : T}{(\Theta_1 \wedge \Theta_2) \lhd \Delta \mid (\Xi_1, \Xi_2) \lhd (\Gamma_1, \Gamma_2) \vdash \mathtt{let}\, x \otimes x' = v\, \mathtt{in}\, t : c \lhd \mathtt{let}\, y \otimes y' = V\, \mathtt{in}\, M : T}$$

$$\frac{\Theta \lhd \Delta \mid \Xi \lhd \Gamma \vdash v : a \lhd V : S}{\Theta \lhd \Delta \mid \Xi \lhd \Gamma \vdash \mathtt{inl}(v) : a \oplus \bullet \lhd \mathtt{inl}(V) : S \oplus T} \qquad \frac{\Theta \lhd \Delta \mid \Xi \lhd \Gamma \vdash v : b \lhd V : T}{\Theta \lhd \Delta \mid \Xi \lhd \Gamma \vdash \mathtt{inr}(v) : \bullet \oplus b \lhd \mathtt{inr}(V) : S \oplus T}$$

$$\frac{\Theta_1 \lhd \Delta \mid \Xi_1 \lhd \Gamma_1 \vdash v : a \oplus \bullet \lhd V : S \oplus T \qquad \Theta_2 \lhd \Delta \mid (\Xi_2, x : a) \lhd (\Gamma_2, y : S) \vdash t : c \lhd N : U}{(\Theta_1 \wedge \Theta_2) \lhd \Delta \mid (\Xi_1, \Xi_2) \lhd (\Gamma_1, \Gamma_2) \vdash \mathtt{let}\, \mathtt{inl}(x) = v\, \mathtt{in}\, t : c \lhd \mathtt{case}\, V\, \mathtt{of}\, (\mathtt{inl}(y) : N \mid \mathtt{inr}(y') : N') : U}$$

$$\frac{\Theta_1 \lhd \Delta \mid \Xi_1 \lhd \Gamma_1 \vdash v : \bullet \oplus b \lhd V : S \oplus T \qquad \Theta_2 \lhd \Delta \mid (\Xi_2, x' : b) \lhd (\Gamma_2, y' : T) \vdash t' : c \lhd N' : U}{(\Theta_1 \wedge \Theta_2) \lhd \Delta \mid (\Xi_1, \Xi_2) \lhd (\Gamma_1, \Gamma_2) \vdash \mathtt{let}\, \mathtt{inr}(x') = v\, \mathtt{in}\, t' : c \lhd \mathtt{case}\, V\, \mathtt{of}\, (\mathtt{inl}(y) : N \mid \mathtt{inr}(y') : N') : U}$$

$$\frac{}{O \lhd \Delta \mid\ \vdash \mathtt{nil} : \mathtt{nil} \lhd \mathtt{Nil} : \mathtt{list}\, S} \qquad \frac{\Theta_1 \lhd \Delta \mid \Xi_1 \lhd \Gamma_1 \vdash v : a \lhd V : S \qquad \Theta_2 \lhd \Delta \mid \Xi_2 \lhd \Gamma_2 \vdash w : b \lhd W : \mathtt{list}\, S}{(\Theta_1 \wedge \Theta_2) \lhd \Delta \mid (\Xi_1, \Xi_2) \lhd (\Gamma_1, \Gamma_2) \vdash v{::}w : a{::}b \lhd V{::}W : \mathtt{list}\, S}$$

$$\frac{\Theta_1 \lhd \Delta \mid \Xi_1 \lhd \Gamma_1 \vdash v : \mathtt{nil} \lhd V : \mathtt{list}\, S \qquad \Theta_2 \lhd \Delta \mid \Xi_2 \lhd \Gamma_2 \vdash t : b \lhd M : T}{(\Theta_1 \wedge \Theta_2) \lhd \Delta \mid (\Xi_1, \Xi_2) \lhd (\Gamma_1, \Gamma_2) \vdash \mathtt{let}\, \mathtt{nil} = v\, \mathtt{in}\, t : b \lhd \mathtt{case}\, V\, \mathtt{of}\, (\mathtt{Nil} : M \mid y{::}y' : N) : T}$$

$$\frac{\Theta_1 \lhd \Delta \mid \Xi_1 \lhd \Gamma_1 \vdash v : a{::}a' \lhd V : \mathtt{list}\, S \qquad \Theta_2 \lhd \Delta \mid (\Xi_2, x : a, x' : a') \lhd (\Gamma_2, y : S, y' : \mathtt{list}\, S) \vdash t : b \lhd N : T}{(\Theta_1 \wedge \Theta_2) \lhd \Delta \mid (\Xi_1, \Xi_2) \lhd (\Gamma_1, \Gamma_2) \vdash \mathtt{let}\, x{::}y = v\, \mathtt{in}\, t : b \lhd \mathtt{case}\, V\, \mathtt{of}\, (\mathtt{Nil} : M \mid y{::}y' : N) : T}$$

Figure 11. Rules relating rigid resource raw-terms and $\lambda_{\mathcal{W}}$-terms

$$\frac{(\Theta_1, \langle x_1, \dots, x_n \rangle : \langle a_1, \dots, a_n \rangle, \Theta_2) \lhd \Delta \mid \Xi \lhd \Gamma \vdash t : b \lhd M : T \qquad \sigma \in \mathfrak{S}_n}{(\Theta_1, \langle x_{\sigma(1)}, \dots, x_{\sigma(n)} \rangle : \langle a_{\sigma(1)}, \dots, a_{\sigma(n)} \rangle, \Theta_2) \lhd \Delta \mid \Xi \lhd \Gamma \vdash t : b \lhd M : T}$$

$$\frac{\Theta \lhd \Delta \mid (\Xi_1, x : a, x' : a', \Xi_2) \lhd (\Gamma_1, y : S, y' : S', \Gamma_2) \vdash t : b \lhd M : T \qquad \Xi_1 \lhd \Gamma_1 \qquad \Xi_2 \lhd \Gamma_2}{\Theta \lhd \Delta \mid (\Xi_1, x' : a', x : a, \Xi_2) \lhd (\Gamma_1, y' : S', y : S, \Gamma_2) \vdash t : b \lhd M : T}$$

$$\frac{(\Theta_1, \langle \vec{x} \rangle : \langle \vec{a} \rangle, \langle \vec{x}' \rangle : \langle \vec{a}' \rangle, \Theta_2) \lhd (\Delta_1, y : S, y' : S', \Delta_2) \mid \Xi \lhd \Gamma \vdash t : b \lhd M : T \qquad \Theta_1 \lhd \Delta_1 \qquad \Theta_2 \lhd \Delta_2}{(\Theta_1, \langle \vec{x}' \rangle : \langle \vec{a}' \rangle, \langle \vec{x} \rangle : \langle \vec{a} \rangle, \Theta_2) \lhd (\Delta_1, y' : S', y : S, \Delta_2) \mid \Xi \lhd \Gamma \vdash t : b \lhd M : T}$$

Figure 12. Rules relating rigid resource raw-terms and $\lambda_{\mathcal{W}}$-terms (structural rules)

$$\begin{aligned}
[\varphi] \cdot ([\psi] x) &:= [\varphi \circ \psi] x \\
[\varphi] \cdot c &:= c \\
[(\varphi \multimap \psi)] \cdot (\lambda x.t) &:= \lambda x.([\psi] \cdot t)\{[\varphi]x/x\} \\
[\varphi] \cdot (v\, w) &:= ([(\mathrm{id} \multimap \varphi)] \cdot v)\, w \\
[\varphi] \cdot (t \diamond \bullet) &:= ([\varphi] \cdot t) \diamond \bullet \\
[\varphi] \cdot (\bullet \diamond t) &:= \bullet \diamond ([\varphi] \cdot t) \\
[\varphi] \cdot () &:= () \\
[\varphi] \cdot (s; t) &:= s; ([\varphi] \cdot t) \\
[\varphi] \cdot (\mathtt{let}\, x = s\, \mathtt{in}\, t) &:= \mathtt{let}\, x = s\, \mathtt{in}\, ([\varphi] \cdot t) \\
[\langle \sigma; \varphi_1, \dots, \varphi_n \rangle] \cdot \langle v_1, \dots, v_n \rangle &:= \langle [\varphi_1] \cdot v_{\sigma(1)}, \dots, [\varphi_n] \cdot v_{\sigma(n)} \rangle \\
[\varphi] \cdot (\mathtt{let}\, \langle x_1, \dots, x_n \rangle = v\, \mathtt{in}\, t) &:= \mathtt{let}\, \langle x_1, \dots, x_n \rangle = v\, \mathtt{in}\, ([\varphi] \cdot t)
\end{aligned}$$

$$\begin{aligned}
[(\varphi \otimes \psi)] \cdot (v \otimes w) &:= ([\varphi] \cdot v) \otimes ([\psi] \cdot w) \\
[\varphi] \cdot (\mathtt{let}\, x \otimes y = v\, \mathtt{in}\, t) &:= \mathtt{let}\, x \otimes y = v\, \mathtt{in}\, ([\varphi] \cdot t) \\
[\mathtt{inl}(\varphi)] \cdot \mathtt{inl}(v) &:= \mathtt{inl}([\varphi] \cdot v) \\
[\mathtt{inr}(\varphi)] \cdot \mathtt{inr}(v) &:= \mathtt{inr}([\varphi] \cdot v) \\
[\varphi] \cdot (\mathtt{let}\, \mathtt{inl}(x) = v\, \mathtt{in}\, t) &:= \mathtt{let}\, \mathtt{inl}(x) = v\, \mathtt{in}\, ([\varphi] \cdot t) \\
[\varphi] \cdot (\mathtt{let}\, \mathtt{inr}(x) = v\, \mathtt{in}\, t) &:= \mathtt{let}\, \mathtt{inr}(x) = v\, \mathtt{in}\, ([\varphi] \cdot t) \\
[\varphi] \cdot (\mathtt{let}\, \mathtt{nil} = v\, \mathtt{in}\, t) &:= \mathtt{let}\, \mathtt{nil} = v\, \mathtt{in}\, ([\varphi] \cdot t) \\
[\varphi] \cdot \mathtt{nil} &:= \mathtt{nil} \\
[(\varphi :: \psi)] \cdot (v :: w) &:= ([\varphi] \cdot v) :: ([\psi] \cdot w) \\
[\varphi] \cdot (\mathtt{let}\, x :: y = v\, \mathtt{in}\, t) &:= \mathtt{let}\, x :: y = v\, \mathtt{in}\, ([\varphi] \cdot t)
\end{aligned}$$

Figure 13. Action of isomorphisms to rigid resource raw-terms

- Assume $\varphi_i : a'_i \cong a_i$ for each $i \le n$. This family induces an isomorphism

$$\begin{aligned}\varphi\ &: (\Theta, \langle x'_1, \dots, x'_n \rangle : \langle a'_1, \dots, a'_n \rangle, \Theta' \mid \Xi) \\ &\cong (\Theta, \langle x_1, \dots, x_n \rangle : \langle a_1, \dots, a_n \rangle, \Theta' \mid \Xi).\end{aligned}$$

  We define $t\{\varphi\} \triangleq t\{[\varphi_1]x'_1/x_1, \dots, [\varphi_n]x'_n/x_n\}$.
- Assume $\Theta \lhd \Delta \mid (\Xi, x : a, \Xi') \lhd \Gamma \vdash t : b \lhd M : S$. Then $\varphi : a' \cong a$ induces an isomorphism

$$\varphi\ :\ (\Theta \mid \Xi, x' : a', \Xi') \cong (\Theta \mid \Xi, x : a, \Xi').$$

  We define $t\{\varphi\} \triangleq t\{[\varphi]x'/x\}$.

Every isomorphism $\varphi : (\Theta \mid \Xi) \cong (\Theta' \mid \Xi')$ can be written as a composition of above ones.

The one-step reduction relation is defined by the rules in Fig. 14, where the evaluation context is given by the grammar: $E ::= [] \mid E; t \mid \mathtt{let}\, x = E\, \mathtt{in}\, t$.

The equivalence relation $\sim$ is defined as the least congruence that contains the rules in Fig. 15.

As the refinement system can be seen as an intersection type system, it enjoys Subject Reduction (Lemma B.2) and Subject Expansion (Lemma B.3). Theorem 3.3 follows from these results.

Lemma B.2. *Let* $\mid (x_1 : \mathsf{a}_1, \dots, x_n : \mathsf{a}_n) \lhd (y_1 : \mathsf{a}_1, \dots, y_n : \mathsf{a}_n) \vdash t : () \lhd M : I$. *Suppose* $[\vec{x} = e, t] \xrightarrow{\pi} [\vec{x}' = e', t']$ *and let* a'_i *for the type of* x'_i. *Then there exists* M' *and* $\vec{y}'$ *such that* $[\vec{y} = e, M] \xrightarrow{\pi} [\vec{y}' = e', M']$ *and* $\mid (x'_1 : \mathsf{a}'_1, \dots, x'_m : \mathsf{a}'_n) \lhd (y'_1 : \mathsf{a}'_1, \dots, y'_m : \mathsf{a}'_m) \vdash t' : () \lhd M' : I$.

Proof. Similar to the standard proof of Subject Reduction. The claim is proved by induction on the length of the reduction sequence. The base case can be proved by using a kind of Substitution Lemma. □

Lemma B.3. *Let* $\mid (x_1 : \mathsf{a}_1, \dots, x_n : \mathsf{a}_n) \lhd (y_1 : \mathsf{a}_1, \dots, y_n : \mathsf{a}_n) \vdash t : () \lhd M : I$. *Suppose* $[\vec{y}' = e', M'] \xrightarrow{\pi} [\vec{y} = e, M]$ *and let* a'_i *for the type of* y'_i. *Then there exists* t' *and* $\vec{x}'$ *such that* $[\vec{y}' = e', M'] \xrightarrow{\pi} [\vec{y} = e, M]$ *and* $\mid (x'_1 : \mathsf{a}'_1, \dots, x'_m : \mathsf{a}'_n) \lhd (y'_1 : \mathsf{a}'_1, \dots, y'_m : \mathsf{a}'_m) \vdash t' : () \lhd M' : I$.

Proof. Similar to the standard proof of Subject Expansion; De-substitution Lemma is the key to the base cases. □

Careful inspection of the proof of Subject Expansion (Lemma B.3) leads to Theorem 3.4. De-subsbitution Lemma says that, if $t \lhd M\{V/x\}$, then t can be decomposed as $t = t'\{v/x\}$ so that $t' \lhd M$ and $v \lhd V$. To prove Theorem 3.4, it suffices to show that such a decomposition is unique up to (an extension of) $\sim$.

C Proof of Theorem 4.4

Here we give a complete definition of the Lafont category $\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ and its proof. In Appendix D, we give a concrete description of the structure given here. A reader who is not familiar with (2-)category theory can skip this section or see Appendix D at the same time.

To define our Lafont category, we shall use the construction in [35][28, Proposition II.3], which says that a Lafont category C can be constructed from:

- an SMCC structure $(\otimes, I, \multimap)$ of C
- countable biproducts in C

$$[\vec{x} = e, E[(\lambda x.t)\, v]] \xrightarrow{0} [\vec{x} = e, E[t\{v/x\}]]$$
$$[\vec{x} = e, E[t \diamond \bullet]] \xrightarrow{1} [\vec{x} = e, E[t]]$$
$$[\vec{x} = e, E[\bullet \diamond t]] \xrightarrow{2} [\vec{x} = e, E[t]]$$
$$[\vec{x} = e, E[(); t]] \xrightarrow{0} [\vec{x} = e, E[t]]$$
$$[\vec{x} = e, E[\mathtt{let}\, y = v\, \mathtt{in}\, t]] \xrightarrow{0} [\vec{x} = e, E[t\{v/y\}]]$$
$$[\vec{x} = e, E[\mathtt{let}\, \langle y_1, \dots, y_n\rangle = \langle v_1, \dots, v_n\rangle\, \mathtt{in}\, t]] \xrightarrow{0} [\vec{x} = e, E[t\{v_1/y_1, \dots, v_n/y_n\}]]$$
$$[\vec{x} = e, E[\mathtt{let}\, y \otimes z = v \otimes w\, \mathtt{in}\, t]] \xrightarrow{0} [\vec{x} = e, E[t\{v/y, w/z\}]]$$
$$[\vec{x} = e, E[\mathtt{let}\, \mathtt{inl}(y) = \mathtt{inl}(v)\, \mathtt{in}\, t]] \xrightarrow{0} [\vec{x} = e, E[t\{v/y\}]]$$
$$[\vec{x} = e, E[\mathtt{let}\, \mathtt{inr}(y) = \mathtt{inr}(v)\, \mathtt{in}\, t]] \xrightarrow{0} [\vec{x} = e, E[t\{v/y\}]]$$
$$[\vec{x} = e, E[\mathtt{let}\, \mathtt{nil} = \mathtt{nil}\, \mathtt{in}\, t]] \xrightarrow{0} [\vec{x} = e, E[t]]$$
$$[\vec{x} = e, E[\mathtt{let}\, y{::}z = v{::}w\, \mathtt{in}\, t]] \xrightarrow{0} [\vec{x} = e, E[t\{v/y, w/z\}]]$$
$$[\vec{x}\vec{y} = e, E[c\, \vec{x}]] \xrightarrow{0} [\vec{z}\vec{y} = (c \otimes \mathrm{id}) \circ e, E[\vec{z}]] \qquad [\vec{x} = e, t] \xrightarrow{\epsilon} [\sigma\vec{x} = \sigma \circ e, t]$$

Figure 14. Operational semantics of the rigid resource calculus

$$v\,([\varphi] \cdot w) \sim ([(\varphi \multimap \mathrm{id})] \cdot v)\, w$$
$$\mathtt{let}\, x = [\varphi] \cdot t\, \mathtt{in}\, u \sim \mathtt{let}\, x = t\, \mathtt{in}\, (u\{[\varphi]x/x\})$$
$$\mathtt{let}\, \langle x_1, \dots, x_n\rangle = ([\langle \sigma; \varphi_1, \dots, \varphi_n\rangle] \cdot v)\, \mathtt{in}\, t \sim \mathtt{let}\, \langle x_{\sigma^{-1}(1)}, \dots, x_{\sigma^{-1}(n)}\rangle = v\, \mathtt{in}\, t\{[\varphi_1]x_1/x_1, \dots, [\varphi_n]x_n/x_n\}$$
$$\mathtt{let}\, x \otimes y = ([\varphi \otimes \psi] \cdot v)\, \mathtt{in}\, t \sim \mathtt{let}\, x \otimes y = v\, \mathtt{in}\, t\{[\varphi]x/x, [\psi]y/y\}$$
$$\mathtt{let}\, \mathtt{inl}(x) = ([\mathtt{inl}(\varphi)] \cdot v)\, \mathtt{in}\, t \sim \mathtt{let}\, \mathtt{inl}(x) = v\, \mathtt{in}\, t\{[\varphi]x/x\}$$
$$\mathtt{let}\, \mathtt{inr}(x) = ([\mathtt{inr}(\varphi)] \cdot v)\, \mathtt{in}\, t \sim \mathtt{let}\, \mathtt{inr}(x) = v\, \mathtt{in}\, t\{[\varphi]x/x\}$$
$$\mathtt{let}\, x{::}y = ([\varphi{::}\psi] \cdot v)\, \mathtt{in}\, t \sim \mathtt{let}\, x{::}y = v\, \mathtt{in}\, t\{[\varphi]x/x, [\psi]y/y\}$$

Figure 15. Base cases of the relation $\sim$

- symmetric tensor powers (i.e., equalisers of "symmetry" arrows in C that are preserved by $(-) \otimes b$ for any object b in C).

The underlying category of our Lafont model is induced by a bicategory, so below we shall give the above structures for the bicategory.

C.1 Preliminaries on 2-(bi)category theory

Here we give some basic on bicategories, for which a reader may consult [4, 24, 25].

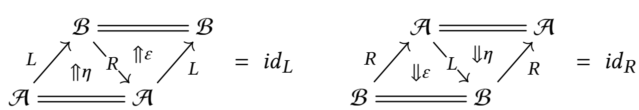
Terminology and notation In this paper, we use the notion of (2-dimensional) biproduct (i.e. "product that is also coproduct"), and so we use the terminology *2-limit* in order to refer what is historically called *bilimit* (i.e. (pseudo) "limit-for-bicategories"); but we keep to use "bi-" to refer non-universality notions such as *bi*category.

For simplicity of presentation, we omit obvious canonical iso-2-cells; for example, we treat a bicategory as if it were a 2-category, i.e. we omit the iso-2-cells of unitality and associativity.

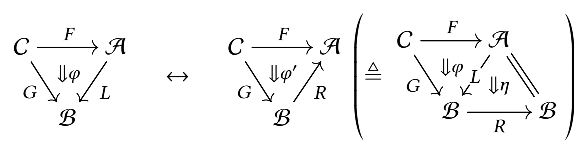
For a bicategory, we write $\circ$ for the horizontal composition of 1-cells and of 2-cells, and write $\bullet$ for the vertical composition of 2-cells; we omit $\circ$ and $\bullet$ if it is clear from the context. We write $\mathscr{B}^{1\text{-op}}$, $\mathscr{B}^{2\text{-op}}$, and $\mathscr{B}^{1,2\text{-op}}$ for the opposite bicategories of $\mathscr{B}$ on 1-cells, on 2-cells, and on both 1-cells and 2-cells, respectively.

We write $\Rightarrow$ for the (cartesian) closed structure of **Set**.

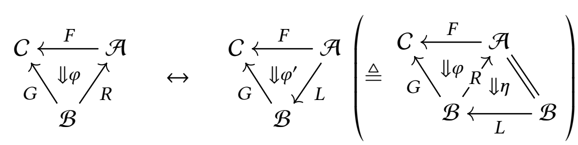
Internal Adjunction For a bicategory $\mathscr{B}$, an internal adjunction $L \dashv R : \mathcal{B} \to \mathcal{A}$ is 1-cells $L : \mathcal{A} \to \mathcal{B}$ and $R : \mathcal{B} \to \mathcal{A}$ equipped with 2-cells $\eta : Id_{\mathcal{A}} \Rightarrow RL$ and $\varepsilon : LR \Rightarrow Id_{\mathcal{B}}$ called *unit* and *counit* satisfying the following *triangular identities*:[2]



As expected, an internal adjunction induces a bijection as follows: 2-cells φ of the form on the left below bijectively corresponds to 2-cells φ' of the form below



The inverse can be defined similarly by ε, and the triangular identities ensure the bijectivity. Furthermore, it is important that there is also a "dual" of the above bijection:



Lax-slice and Pseudo-slice Bicategory Let $\mathscr{B}$ be a bicategory and $\mathcal{W}$ be a 0-cell. The *lax-slice bicategory* $\mathscr{B} /\!\!/ \mathcal{W}$ of $\mathscr{B}$ over $\mathcal{W}$ is defined as follows:

- A 0-cell is a 1-cell of the form $A : \mathcal{A} \to \mathcal{W}$,

[2] As we said, we have omitted canonical iso-2-cells; precisely we have to insert $L \cong L \circ Id_{\mathcal{A}}$ and $Id_{\mathcal{B}} \circ L \cong L$ for the LHS of the left equation, and similarly for the LHS of the right equation.

- A 1-cell from $A : \mathcal{A} \to \mathcal{W}$ to $B : \mathcal{B} \to \mathcal{W}$ is a pair of a 1-cell $F : \mathcal{A} \to \mathcal{B}$ and a 2-cell $\varphi : B \circ F \Rightarrow A$ as below[3]:

$$\begin{array}{ccc} \mathcal{A} & \xrightarrow{F} & \mathcal{B} \\ & {\scriptstyle A}\searrow \Downarrow\varphi \swarrow{\scriptstyle B} & \\ & \mathcal{W}^{\mathrm{op}} & \end{array}$$

- a 2-cell from (F, φ) to (G, ψ) is a 2-cell $\alpha : F \Rightarrow G$ such that $\varphi = (B \circ \alpha) \bullet \psi$ as below:

$$\begin{array}{ccc} \mathcal{A} & \xrightarrow{F} & \mathcal{B} \\ & {\scriptstyle A}\searrow \Downarrow\varphi \swarrow{\scriptstyle B} & \\ & \mathcal{W}^{\mathrm{op}} & \end{array} \quad = \quad \begin{array}{ccc} & \overset{F}{\frown} & \\ \mathcal{A} & \xrightarrow[G]{\Downarrow\alpha} & \mathcal{B} \\ & {\scriptstyle A}\searrow \Downarrow\psi \swarrow{\scriptstyle B} & \\ & \mathcal{W}^{\mathrm{op}} & \end{array}$$

The *pseudo-slice bicategory* $\mathscr{B}/\mathcal{W}$ of $\mathscr{B}$ over $\mathcal{W}$ is defined as the (locally-full) subbicategory of $\mathscr{B}/\!/\mathcal{W}$ where 0-cells and 2-cells are the same and 1-cells are 1-cells (F, φ) in $\mathscr{B}/\!/\mathcal{W}$ where φ is an iso-2-cell.

Classifying Category For a bicategory $\mathscr{B}$, there is a general way for obtaining a 1-category, the *classifying category* $Cl(\mathscr{B})$ [4, Section 7], which is, shortly speaking, "local-skeleton". The objects of $Cl(\mathscr{B})$ are the same as those of $\mathscr{B}$, and for objects $\mathcal{A}$ and $\mathcal{B}$, the homset $Cl(\mathscr{B})(\mathcal{A}, \mathcal{B})$ is defined as the quotient of $\mathscr{B}(\mathcal{A}, \mathcal{B})$ modulo the existence of an iso-2-cell. The identity on $\mathcal{A}$ is $[id_{\mathcal{A}}] : \mathcal{A} \to \mathcal{A}$ and the composition of $[F] : \mathcal{A} \to \mathcal{B}$ and $[G] : \mathcal{B} \to \mathcal{C}$ is $[G \circ F] : \mathcal{A} \to \mathcal{C}$.

End, coend, and (co)Yoneda lemma A reader may consult [7] for the facts in this paragraph. For functors $F, G : \mathcal{A}^{\mathrm{op}} \times \mathcal{A} \to \mathcal{B}$, we have

$$\int_a \mathcal{B}(F(a, a), G(a, a)) = \mathrm{Dinat(F, G)} \qquad (2)$$

where Dinat(F, G) is the set of all the dinatural transformations from F to G. This is used in calculation of 2-cells in **Prof**. Also, we often use the Yoneda lemma in the end form: for a functor $F : \mathcal{C} \to \mathbf{Set}$, we have

$$F(a) \cong \int_b \mathcal{C}(a, b) \Rightarrow F(b)$$

where recall that $\Rightarrow$ is the closed structure of **Set**. In calculation of the composition of profunctors, we use the *coYoneda lemma* (a.k.a. the *density formula*): for a functor $F : \mathcal{C} \to \mathbf{Set}$,

$$F(a) \cong \int^b \mathcal{C}(b, a) \times F(b).$$

Note that the above Yoneda and coYoneda lemmas contain that for a functor $F : \mathcal{C}^{\mathrm{op}} \to \mathbf{Set}$, we have

$$\int_b \mathcal{C}(b, a) \Rightarrow F(b) \;\cong\; F(a) \;\cong\; \int^b \mathcal{C}(a, b) \times F(b).$$

Basic notions on profunctors There are two ways for transforming a functor to a profunctor. For a functor $F : \mathcal{A} \to \mathcal{B}$, its *direct image* $F_* : \mathcal{A} \nrightarrow \mathcal{B}$ is defined as:

$$F_*(b, a) \triangleq \mathcal{B}(b, F(a))$$

and its *inverse image* $F^* : \mathcal{B} \nrightarrow \mathcal{A}$ is defined as:

$$F^*(a, b) \triangleq \mathcal{B}(F(a), b).$$

[3] If we reverse the direction of φ, we obtain the definition of an *oplax-slice bicategory*; note that some authors call an oplax-slice bicategory a lax-slice bicategory.

The direct image extends to a pseudofunctor $(-)_* : \mathbf{Cat} \to \mathbf{Prof}$ that maps 0-cell $\mathcal{A}$ to itself, and 2-cell $\alpha : F \Rightarrow G$ to

$$\mathcal{B}^{\mathrm{op}} \times \mathcal{A} \ni (b, a) \mapsto \mathcal{B}(b, \alpha_a) : \mathcal{B}(b, F(a)) \to \mathcal{B}(b, G(a)).$$

Similarly, the inverse image extends to a pseudofunctor $(-)^* : \mathbf{Cat} \to \mathbf{Prof}^{1,2\text{-op}}$ that maps 0-cell $\mathcal{A}$ to itself, and 2-cell $\alpha : F \Rightarrow G$ to

$$\mathcal{A}^{\mathrm{op}} \times \mathcal{B} \ni (a, b) \mapsto \mathcal{B}(\alpha_a, b) : \mathcal{B}(G(a), b) \to \mathcal{B}(F(a), b).$$

We identify a functor F with the direct image profunctor F_*, if no confusion arises. A functor $F : \mathcal{A} \to \mathcal{B}$ occurring in a diagram in **Prof** should be regarded as $F_* : \mathcal{A} \nrightarrow \mathcal{B}$.

For a functor $F : \mathcal{A} \to \mathcal{B}$, we have an internal adjunction $F_* \dashv F^*$ in **Prof**. The unit η is given by:

$$\eta : Id_{\mathcal{A}} \Rightarrow F^* \circ F_* : \mathcal{A} \nrightarrow \mathcal{A}$$

$$\eta_{a', a} : \mathcal{A}(a', a) \xrightarrow{F} \mathcal{B}(F(a'), F(a)) \cong \int^b \mathcal{B}(F(a'), b) \times \mathcal{B}(b, F(a))$$

where we used the coYoneda lemma, and the counit ε is given by:

$$\varepsilon : F_* \circ F^* \Rightarrow Id_{\mathcal{B}} : \mathcal{B} \nrightarrow \mathcal{B}$$

$$\varepsilon_{b', b} : \int^a \mathcal{B}(b', F(a)) \times \mathcal{B}(F(a), b) \to \mathcal{B}(b', b)$$

$$[a, (f, g)] \mapsto g \circ f.$$

For a profunctor $F : \mathcal{A} \nrightarrow \mathcal{B}$, i.e., a functor $F : \mathcal{B}^{\mathrm{op}} \times \mathcal{A} \to \mathbf{Set}$, we define a profunctor $F^{\mathrm{op}} : \mathcal{B}^{\mathrm{op}} \nrightarrow \mathcal{A}^{\mathrm{op}}$ as the functor

$$(\mathcal{A}^{\mathrm{op}})^{\mathrm{op}} \times \mathcal{B}^{\mathrm{op}} \xrightarrow{\cong} \mathcal{B}^{\mathrm{op}} \times \mathcal{A} \xrightarrow{F} \mathbf{Set}$$

This extends to a pseudofunctor $(-)^{\mathrm{op}} : \mathbf{Prof} \to \mathbf{Prof}^{1\text{-op}}$ that maps a 0-cell, category, $\mathcal{A}$ to $\mathcal{A}^{\mathrm{op}}$, and 2-cell $\alpha : F \Rightarrow G : \mathcal{A} \nrightarrow \mathcal{B}$ to $\alpha \circ (\cong)$ where $(\cong) : (\mathcal{A}^{\mathrm{op}})^{\mathrm{op}} \times \mathcal{B}^{\mathrm{op}} \to \mathcal{B}^{\mathrm{op}} \times \mathcal{A}$. We have the following commutativity:

$$\begin{array}{ccc} \mathbf{Cat} & \xrightarrow{(-)^{\mathrm{op}}} & \mathbf{Cat}^{2\text{-op}} \\ {\scriptstyle (-)_*}\downarrow & & \downarrow{\scriptstyle (-)^*} \\ \mathbf{Prof} & \xrightarrow[(-)^{\mathrm{op}}]{} & \mathbf{Prof}^{1\text{-op}} \end{array}$$

C.2 (Bi)category of Weighted Profunctors

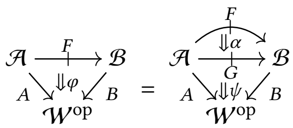
The following definition of $\mathcal{W}$-weighted profunctors (and hence 2-cells between them) are (equivalent but) different from the definition given in Definition 4.3. It will be explained just after the definition.

Definition C.1 ((Bi)category of weighted profunctors)**.** Let $\mathcal{W}$ be a category. We define a bicategory $\mathbf{Prof}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ as a "fullsub" bicategory of the lax-slice bicategory $\mathbf{Prof}/\!/\mathcal{W}^{\mathrm{op}}$ determined by 0-cells of the pseudo-slice bicategory $\mathbf{Cat}/\mathcal{W}^{\mathrm{op}}$. Specifically, $\mathbf{Prof}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ is as follows:

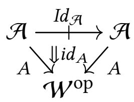
- a 0-cell, a *$\mathcal{W}$-weighted category*, is a 0-cell of $\mathbf{Cat}/\mathcal{W}^{\mathrm{op}}$, i.e. a pair $(\mathcal{A}, A)$ of a category $\mathcal{A}$ and a functor $A : \mathcal{A} \to \mathcal{W}^{\mathrm{op}}$,
- a 1-cell, a *$\mathcal{W}$-weighted profunctor*, from $(\mathcal{A}, A)$ to $(\mathcal{B}, B)$ is a pair (F, φ) of a profunctor $F : \mathcal{A} \nrightarrow \mathcal{B}$ and a natural transformation $\varphi : B \circ F \Rightarrow A$ as below:

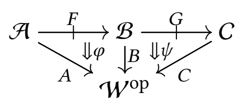
$$\begin{array}{ccc} \mathcal{A} & \overset{F}{\nrightarrow} & \mathcal{B} \\ & {\scriptstyle A}\searrow \Downarrow\varphi \swarrow{\scriptstyle B} & \\ & \mathcal{W}^{\mathrm{op}} & \end{array}$$

- a 2-cell, a $\mathcal{W}$*-weighted natural transformation* ($\mathcal{W}$*-2-cell* for short), from (F,φ) to (G,ψ) is a natural transformation α from F to G that satisfies the following equation (which means $\varphi = \psi \bullet (id_B \circ \alpha) : B \circ F \Rightarrow A$):



The horizontal identity on $(\mathcal{A}, A)$ is $(Id_{\mathcal{A}}, id_A)$ as on the left below, and the horizontal composition of $(F,\varphi) : \mathcal{A} \to \mathcal{B}$ and $(G,\psi) : \mathcal{B} \to \mathcal{C}$ is defined by the diagram on the right below:



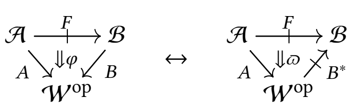


The vertical composition is naturally defined so that there is a forgetful functor from $\mathbf{Prof}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ to $\mathbf{Prof}$ that maps $(\mathcal{A}, A), (F,\varphi), \alpha$ to $\mathcal{A}, F, \alpha$, respectively.

We define $\mathbf{Pr}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}} \triangleq Cl(\mathbf{Prof}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}})$.

We sometimes write $(\mathcal{A}, A)$ and (F,φ) simply as A and F, respectively, when no confusion arises. We omit "$\mathcal{W}$-" from "$\mathcal{W}$-weighted" if $\mathcal{W}$ is clear from the context.

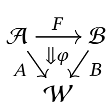
The two style of weighted profunctors (Definitions 4.3 and C.1) bijectively correspond to each other by: (i) the following bijective correspondence induced by $B_* \dashv B^*$

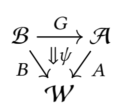


and then (ii) composing the following natural isomorphism (by the coYoneda lemma):

$$(B^* \circ A_*)(b,a) = \int^{w} \mathcal{W}(w, B(b)) \times \mathcal{W}(A(a), w) \cong \mathcal{W}(A(a), B(b)).$$

Lemma C.2. *Let $\mathscr{B}$ be a bicategory, $\mathcal{W}$ be a 0-cell in $\mathscr{B}$, and $F \dashv G : \mathscr{B}$ be an adjunction in $\mathscr{B}$ with unit η and counit ε. Given two 1-cells in $\mathscr{B}/\!/\mathcal{W}$ of the form:*





let $\psi' : A \Rightarrow B \circ F$ be the 2-cell given from ψ by $F \dashv G$, i.e. $\psi' \triangleq (\psi \circ F) \bullet (A \circ \eta)$. Then we have $(F,\varphi) \dashv (G,\psi)$ in $\mathscr{B}/\!/\mathcal{W}$ with unit η and counit ε provided that φ is iso-2-cell in **Prof** *with the inverse ψ'.*

Proof. Straightforward. □

As a corollary, we have:

Proposition C.3. *The embedding* $\mathbf{Cat}/\mathcal{W}^{\mathrm{op}} \to \mathbf{Prof}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ *maps any 1-cell (F,φ) to an internal left adjoint in* $\mathbf{Prof}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ *whose right adjoint is given by F^* and the unit and counit are those of $F_* \dashv F^*$.*

We remark that the above proposition says that the embedding $\mathbf{Cat}/\mathcal{W}^{\mathrm{op}} \to \mathbf{Prof}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ *equips* $\mathbf{Cat}/\mathcal{W}^{\mathrm{op}}$ *with proarrows* [40, 45, 46].

C.3 $\mathbf{Pr}/\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ is SMCC

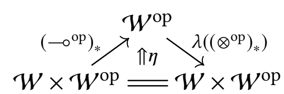
The bicategory **Prof** has the following SMCC structure[4]: For categories $\mathcal{A}_i$ $(i = 1,2)$, their monoidal product is $\mathcal{A}_1 \times \mathcal{A}_2$. For profunctors $F_i : \mathcal{A}_i \nrightarrow \mathcal{B}_i$ $(i = 1,2)$, their monoidal product $F_1 \times F_2 : (\mathcal{B}_1 \times \mathcal{B}_2)^{\mathrm{op}} \times (\mathcal{A}_1 \times \mathcal{A}_2) \to \mathbf{Set}$ is defined by: $(F_1 \times F_2)(b_1, b_2, a_1, a_2) \triangleq F_1(b_1, a_1) \times F_2(b_2, a_2)$. The monoidal product of 2-cells are defined in the obvious way. The monoidal unit is the one-object one-arrow category 1. The closed structure is $(-)^{\mathrm{op}} \times (-)$, and we have the following isomorphisms between hom-categories:

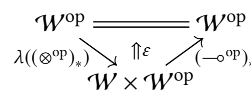
$$\lambda : \mathbf{Prof}(\mathcal{A} \times \mathcal{B}, \mathcal{C}) \xrightarrow{\cong} \mathbf{Prof}(\mathcal{A}, \mathcal{B}^{\mathrm{op}} \times \mathcal{C}) \tag{3}$$

which are pseudo-natural in $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$. Below we sometimes write $\mathcal{A} \nrightarrow\!\circ\, \mathcal{B}$ for $\mathcal{A}^{\mathrm{op}} \times \mathcal{B}$.

Proposition C.4 (Internal definition of SMCC)**.** *For a monoidal category $(\mathcal{W}, \otimes, I)$ and a functor $\multimap$: $\mathcal{W}^{\mathrm{op}} \times \mathcal{W} \to \mathcal{W}$, $(\mathcal{W}, \otimes, I, \multimap)$ is an SMCC iff $(\multimap^{\mathrm{op}})_* : (\mathcal{W}^{\mathrm{op}} \nrightarrow\!\circ\, \mathcal{W}^{\mathrm{op}}) \nrightarrow \mathcal{W}^{\mathrm{op}}$ is left adjoint in* **Prof** *to $\lambda((\otimes^{\mathrm{op}})_*) : \mathcal{W}^{\mathrm{op}} \nrightarrow (\mathcal{W}^{\mathrm{op}} \nrightarrow\!\circ\, \mathcal{W}^{\mathrm{op}})$.*

Proof. First we calculate what the structures η and ε correspond to:





By the formula (2), natural transformations η above belong to the LHS of the following:

$$\begin{aligned}
&\int_{a,a',c,c'} (\mathcal{W} \times \mathcal{W}^{\mathrm{op}})\,((c,c'),(a,a')) \Rightarrow \\
&\qquad \int^{b} \lambda((\otimes^{\mathrm{op}})_*)((c,c'),b) \times (\multimap^{\mathrm{op}})_*(b,(a,a')) \\
&= \int_{a,a',c,c'} (\mathcal{W}(c,a) \times \mathcal{W}(a',c')) \Rightarrow \int^{b} \mathcal{W}(b \otimes c, c') \times \mathcal{W}(a \multimap a', b) \\
&\cong \int_{a,a'} \mathcal{W}((a \multimap a') \otimes a, a')
\end{aligned}$$

where the isomorphism is due to the Yoneda and coYoneda lemmas. Thus, natural transformations η in the LHS bijectively correspond to dinatural transformations $(\mathrm{ev}_{a,a'} : (a \multimap a') \otimes a \to a')_{a,a'}$. Next we calculate ε:

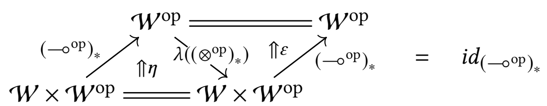
$$\begin{aligned}
&\int_{b,d} \left(\int^{c,c'} (\multimap^{\mathrm{op}})_*(d,(c,c')) \times \lambda((\otimes^{\mathrm{op}})_*)((c,c'),b) \right) \Rightarrow \mathcal{W}^{\mathrm{op}}(d,b) \\
&\cong \int_{b,d,c,c'} (\mathcal{W}(c \multimap c', d) \times \mathcal{W}(b \otimes c, c')) \Rightarrow \mathcal{W}(b,d) \\
&\cong \int_{b,c} \mathcal{W}(b, c \multimap (b \otimes c))
\end{aligned}$$

where the first isomorphism is because $(-) \Rightarrow X : \mathbf{Set} \to \mathbf{Set}^{\mathrm{op}}$ is left adjoint and hence maps coends to ends, and the second one is due to the Yoneda lemma. Thus, natural transformations ε in the LHS bijectively correspond to dinatural transformations $(\mathrm{lam}_{b,c} : b \to c \multimap (b \otimes c))_{b,c}$.

Next we show the equivalence between the triangular identities. Suppose that we are given η and ε, and hence the corresponding ev

[4] **Prof** is a compact closed bicategory [9], and hence a symmetric monoidal closed bicategory (for the definition, see [41]).

and lam. The following triangular identity for η and ε



says that the following mapping equals the identity on $\mathcal{W}(a \multimap a', d)$ for any a, a' and d. (Below the overline and underline are the parts mapped by η and ε, respectively.)

$$\begin{aligned}
&\mathcal{W}(a \multimap a', d) \\
&\qquad \ni f \\
\cong\ & \int^{c,c'} (\multimap^{\mathrm{op}})_*(d,(c,c')) \times (\mathcal{W}\times\mathcal{W}^{\mathrm{op}})\,((c,c'),(a,a')) \\
&\left(= \int^{c,c'} \mathcal{W}(c \multimap c', d) \times \overline{\mathcal{W}(c,a) \times \mathcal{W}(a',c')} \right) \\
&\mapsto \big[(a,a'),(f, id_a, id_{a'})\big] \\
\overset{\text{“}\eta\text{”}}{\to}\ & \int^{b,c,c'} (\multimap^{\mathrm{op}})_*(d,(c,c')) \times \lambda((\otimes^{\mathrm{op}})_*)((c,c'),b) \times (\multimap^{\mathrm{op}})_*(b,(a,a')) \\
&\left(= \int^{b,c,c'} \underline{\mathcal{W}(c \multimap c', d) \times \overline{\mathcal{W}(b \otimes c, c')}} \times \overline{\mathcal{W}(a \multimap a', b)} \right) \\
&\mapsto \big[(a \multimap a', a, a'),(f, \mathrm{ev}_{a,a'}, id_{a\multimap a'})\big] \\
\overset{\text{“}\varepsilon\text{”}}{\to}\ & \int^{b} \mathcal{W}^{\mathrm{op}}(d,b) \times (\multimap^{\mathrm{op}})_*(b,(a,a')) \\
&\left(= \int^{b} \underline{\mathcal{W}(b,d)} \times \mathcal{W}(a \multimap a', b) \right) \\
&\mapsto \big[a \multimap a',\ (f \circ (a \multimap \mathrm{ev}_{a,a'}) \circ \mathrm{lam}_{a\multimap a', a},\ id_{a\multimap a'})\big] \\
\cong\ & \mathcal{W}(a \multimap a', d) \\
&\mapsto f \circ (a \multimap \mathrm{ev}_{a,a'}) \circ \mathrm{lam}_{a\multimap a', a}
\end{aligned}$$

Thus, this triangular identity for η and ε is equivalent to the following triangular identity for ev and lam:

$$(a \multimap \mathrm{ev}_{a,a'}) \circ \mathrm{lam}_{a\multimap a', a} = id_{a\multimap a'}$$

(for one implication, consider d and f as $a \multimap a'$ and $id_{a\multimap a'}$, respectively).

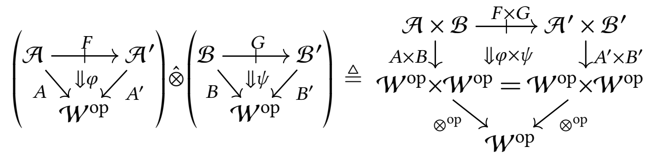
The equivalence between the other triangular identities can be shown similarly. ◻

Remark C.5. If we use the other style of definition of profunctors: i.e. $F : \mathcal{A} \nrightarrow \mathcal{B}$ iff $F : \mathcal{A}^{\mathrm{op}} \times \mathcal{B} \to \mathbf{Set}$, then the statement above becomes the following: $(\mathcal{W}, \otimes, I, \multimap)$ is an SMCC iff $\multimap_* : (\mathcal{W} \nrightarrow \mathcal{W}) \nrightarrow \mathcal{W}$ is left adjoint to $\lambda(\otimes_*) : \mathcal{W} \nrightarrow (\mathcal{W} \nrightarrow \mathcal{W})$. In this statement, we used only the symmetric monoidal closed structure of the ambient bicategory **Prof** (rather than, say, compact closed structure nor even inverse image of a functor). Thus this can be regarded as an instance of *the microcosm principle* [2][34].

For an SMC $(\mathcal{W}, \otimes, I)$, we have the following monoidal structure on $\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$: the unit is $\hat{I} \triangleq (1 \xrightarrow{I} \mathcal{W}^{\mathrm{op}})$; the monoidal product of $(\mathcal{A}, A)$ and $(\mathcal{B}, B)$ is:

$$(\mathcal{A}, A) \mathbin{\hat{\otimes}} (\mathcal{B}, B) \triangleq (\mathcal{A} \times \mathcal{B} \xrightarrow{A \times B} \mathcal{W}^{\mathrm{op}} \times \mathcal{W}^{\mathrm{op}} \xrightarrow{\otimes^{\mathrm{op}}} \mathcal{W}^{\mathrm{op}});$$

and its action on 1-cells is defined by:



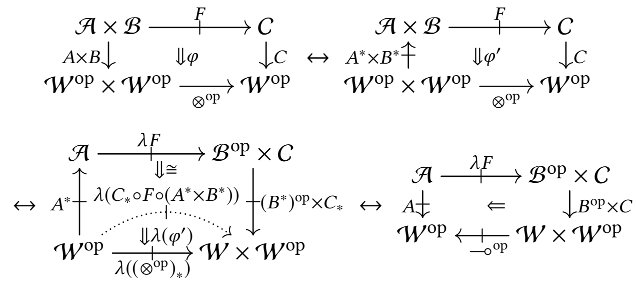
Furthermore, given a SMCC $(\mathcal{W}, \otimes, I, \multimap)$, we define:

$$(\mathcal{B}, B) \mathbin{\hat{\multimap}} (C, C) \triangleq (\mathcal{B}^{\mathrm{op}} \times C \xrightarrow{B^{\mathrm{op}} \times C} \mathcal{W} \times \mathcal{W}^{\mathrm{op}} \xrightarrow{\multimap^{\mathrm{op}}} \mathcal{W}^{\mathrm{op}}),$$

which becomes the closed structure of $\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ as follows:

Proposition C.6. *If $(\mathcal{W}, \otimes, I, \multimap)$ is a symmetric monoidal closed category, then $(\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}, \hat{\otimes}, \hat{I}, \hat{\multimap})$ is a symmetric monoidal closed bicategory.*

Proof. The closedness follows from the following bijections:



where the first correspondence is due to $A_* \dashv A^*$ and $B_* \dashv B^*$; the second one is due to (3) (whose naturality gives $\cong$ and whose action on morphisms gives $\lambda(\varphi')$); and the third one is due to $\multimap^{\mathrm{op}} \dashv \lambda((\otimes^{\mathrm{op}})_*)$ (by Lemma C.4), $A_* \dashv A^*$, and $(B^*)^{\mathrm{op}} = (B^{\mathrm{op}})_*$. It is obvious that the bijective correspondence between 2-cells from (F, φ) to (G, ψ) and those from $(\lambda F, \lambda(\varphi')'$ to $(\lambda G, \lambda(\psi')'$ is given in the same way. ◻

If $\mathscr{B}$ is a symmetric monoidal (closed) bicategory, then $Cl(\mathscr{B})$ is a symmetric monoidal (closed) category, whose structure is defined in the obvious way. Thus for symmetric monoidal (closed) category $\mathcal{W}$, we have obtained a symmetric monoidal (closed) category $\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$.

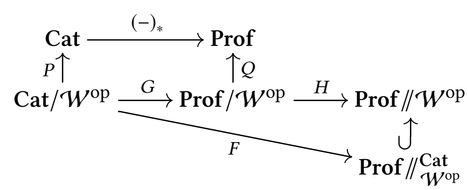
C.4 2-(co)limits of $\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$

To give the biproducts and the symmetric tensor powers in $\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$, here we consider some general results on 2-(co)limits in/around $\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$.

As in the 1-dimensional case, for any bicategory $\mathscr{B}$ and its object $\mathcal{B}$, colimits of the pseudo-slice bicategory $\mathscr{B}/\mathcal{B}$ are created by the projection $\mathscr{B}/\mathcal{B} \to \mathscr{B}$ [14, Section 14.1].

Lemma C.7. *The embedding $\mathbf{Cat}/\mathcal{W}^{\mathrm{op}} \to \mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ preserves 2-colimits.*

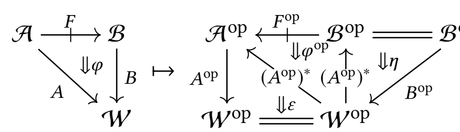
Proof. We have the obvious (pseudo) 2-functors as follows:



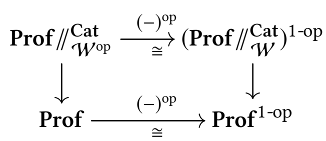
The projections P and Q create 2-colimits as mentioned above, and $(-)_*$ preserves 2-colimits; hence G preserves 2-colimits. Also, H preserves 2-colimits and hence so does HG. Thus, since $\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ is a full sub-bicategory of $\mathbf{Prof}/\!\!/\mathcal{W}^{\mathrm{op}}$, F preserves 2-colimits. □

Lemma C.8. *For any category $\mathcal{W}$, we have the following 2-isomorphism $\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}} \cong (\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}})^{1\text{-op}}$:*

- *0-cell $A : \mathcal{A} \to \mathcal{W}$ is mapped to $A^{\mathrm{op}} : \mathcal{A}^{\mathrm{op}} \to \mathcal{W}^{\mathrm{op}}$*
- *1-cell (F, φ) is mapped as follows:*



- *2-cells are mapped obviously so that we have the following commutative diagram:*



Lemma C.9. *Let $\mathscr{F} \dashv \mathscr{G} : \mathscr{A} \to \mathscr{B}$ be a (pseudo) 2-adjunction between bicategories, and let $\eta : Id_{\mathscr{B}} \Rightarrow \mathscr{G}\mathscr{F}$ and $\varepsilon : \mathscr{F}\mathscr{G} \Rightarrow Id_{\mathscr{A}}$ be its unit and counit. Also let $\varepsilon' : \mathscr{G}\mathscr{F} \Rightarrow Id_{\mathscr{B}}$ and $\eta' : Id_{\mathscr{A}} \Rightarrow \mathscr{F}\mathscr{G}$ be (internal) right adjoint to η and ε in the bicategories $\mathbf{BiCAT}(\mathscr{B}, \mathscr{B})$ and $\mathbf{BiCAT}(\mathscr{A}, \mathscr{A})$, respectively. Then, we have also $\mathscr{G} \dashv \mathscr{F}$ with unit η' and counit ε'.*

Proof. By the assumption we have the "triangular" isomodification:

$$(\varepsilon \circ \mathscr{F}) \bullet (\mathscr{F} \circ \eta) \cong id_{\mathscr{F}}. \tag{4}$$

Then we show that we have a triangular isomodification:

$$(\eta' \circ \mathscr{F}) \bullet (\mathscr{F} \circ \varepsilon') \cong id_{\mathscr{F}}.$$

Now $(-) \circ \mathscr{F} : \mathbf{BiCAT}(\mathscr{A}, \mathscr{A}) \to \mathbf{BiCAT}(\mathscr{B}, \mathscr{A})$ and $\mathscr{F} \circ (-) : \mathbf{BiCAT}(\mathscr{B}, \mathscr{B}) \to \mathbf{BiCAT}(\mathscr{B}, \mathscr{A})$ are pseudo-functors and hence they map internal adjunction to internal adjunction. Thus $\eta' \circ \mathscr{F}$ and $\mathscr{F} \circ \varepsilon'$ are right adjoint to $\varepsilon \circ \mathscr{F}$ and $\mathscr{F} \circ \eta$ in $\mathbf{BiCAT}(\mathscr{B}, \mathscr{A})$, respectively. Then the composite $(\eta' \circ \mathscr{F}) \bullet (\mathscr{F} \circ \varepsilon')$ is right adjoint to $(\varepsilon \circ \mathscr{F}) \bullet (\mathscr{F} \circ \eta)$ in $\mathbf{BiCAT}(\mathscr{B}, \mathscr{A})$. By (4), $(\eta' \circ \mathscr{F}) \bullet (\mathscr{F} \circ \varepsilon')$ is right adjoint to $id_{\mathscr{F}}$, and also trivially $id_{\mathscr{F}}$ is right adjoint to $id_{\mathscr{F}}$. Thus, since internal adjoint is unique up to iso-2-cell, there is an isomodification between $(\eta' \circ \mathscr{F}) \bullet (\mathscr{F} \circ \varepsilon')$ and $id_{\mathscr{F}}$.

The other kind of a triangular isomodification can be given in the same way. □

C.5 $\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ has Biproducts

Now we consider biproducts.

Definition C.10 (2-biproducts)**.** A bicategory $\mathscr{B}$ has *2-biproducts* if for any family $(\mathcal{A}_i)_{i \in I}$ of 0-cells, we have a 0-cell $\oplus_{i \in I}\mathcal{A}_i$ equipped with two families of 1-cells $(Pr_i : \oplus_{i \in I}\mathcal{A}_i \to \mathcal{A}_i)_{i \in I}$ and $(In_i : \mathcal{A}_i \to \oplus_{i \in I}\mathcal{A}_i)_{i \in I}$ such that

- $(\oplus_{i \in I}\mathcal{A}_i, (Pr_i)_{i \in I})$ is a 2-product of $(\mathcal{A}_i)_{i \in I}$,
- $(\oplus_{i \in I}\mathcal{A}_i, (In_i)_{i \in I})$ is a 2-coproduct of $(\mathcal{A}_i)_{i \in I}$, and
- for each $i, j \in I$, there exists an iso-2-cell:

$$\begin{cases} Pr_i \circ In_i \cong Id_{\mathcal{A}_i} : \mathcal{A}_i \to \mathcal{A}_i & (i = j) \\ Pr_j \circ In_i \cong Zero_{\mathcal{A}_i, \mathcal{A}_j} : \mathcal{A}_i \to \mathcal{A}_j & (i \neq j) \end{cases}$$

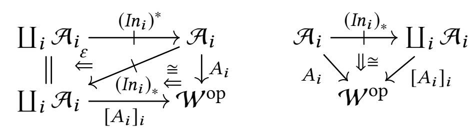
where: when $I = \emptyset$, the third condition trivially holds (without involving the notion $Zero_{\mathcal{A}_i, \mathcal{A}_j}$) and we have *zero 0-cell* $\mathcal{Z} \triangleq \oplus_{i \in \emptyset}\mathcal{A}_i$; and when $I \neq \emptyset$, we define $Zero_{\mathcal{A}_i, \mathcal{A}_j}$ as the (unique up to iso-2-cell) *zero 1-cell* $\mathcal{A}_i \to \mathcal{Z} \to \mathcal{A}_j$.

Lemma C.11. *If $\mathscr{B}$ has 2-biproducts, then $Cl(\mathscr{B})$ has biproducts.*

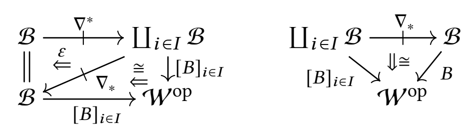
Proof. Trivial. □

Biproducts of $\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ are given by Lemma C.11 and the next lemma.

Proposition C.12. *For any category $\mathcal{W}$, the bicategory $\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ has the following 2-biproducts: for a family $(\mathcal{A}_i, A_i)_{i \in I}$ of 0-cells, the 2-biproduct $\oplus_i(\mathcal{A}_i, A_i)$ is the 0-cell $([A_i]_i)_* = [(A_i)_*]_i : \coprod_i \mathcal{A}_i \nrightarrow \mathcal{W}^{\mathrm{op}}$ equipped with the following projections and coprojections:*



Further, for 0-cell $B : \mathcal{B} \to \mathcal{W}^{\mathrm{op}}$ and set I, the diagonal and codiagonal are given by the following:



We remark that a similar proposition to the above holds for the lax-slice bicategory $\mathbf{Prof}/\!\!/\mathcal{W}^{\mathrm{op}}$, by essentially the same proof. Also we remark that, by Proposition C.3, the injections and codiagonals are internally left adjoint to the projections and diagonals, respectively.

Proof. The 2-coproduct part follows from Lemma C.7. Then since the projections and diagonals are given as right adjoint 1-cells to the injections and codiagonals respectively, the 2-product part follows from Lemma C.9 applied to $\coprod \dashv \Delta : \mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}} \to (\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}})^I$.

What remains to show is the third condition in the definition of 2-biproducts. Let $(A_i : \mathcal{A}_i \to \mathcal{W}^{\mathrm{op}})_{i \in I}$ be a family of 0-cells. For each $i \in I$, we have $\eta : Id \Rightarrow (In_i)^* \circ (In_i)_* : A_i \to A_i$, which is an iso-2-cell because the functor $In_i : \mathcal{A}_i \to \coprod_i A_i$ is fully faithful (in general, $F : \mathcal{A} \to \mathcal{B}$ is fully faithful iff the unit η is isomorphic). It is easy to check that this η is in fact a 2-cell in $\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ of the required type (ε in the definition of projection and this η cancel each other).

Let $i \neq j \in I$. Now zero 1-cell $\mathcal{Z}$ is the empty category, and the zero profunctor $Zero : \mathcal{A}_i \nrightarrow \mathcal{Z} \nrightarrow \mathcal{A}_j$ is the constant functor of the empty set. On the other hand, the profunctor $In_j{}^* \circ In_{i*} : \mathcal{A}_i \nrightarrow \coprod_i \mathcal{A}_i \nrightarrow \mathcal{A}_j$ maps $a \in \mathcal{A}_i$ and $a' \in \mathcal{A}_j$ to $(\coprod_i \mathcal{A}_i)((j, a'), (i, a))$, which is the empty set since $i \neq j$. Thus the required iso-2-cell is the identity (between the empty profunctors), and checking if this is in fact a 2-cell in $\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ is trivial, because maps from the empty set are unique. □

C.6 $\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ has Equalisers Sufficiently

Next we consider equalisers.

Let $(\mathscr{B}, \otimes, I)$ be a symmetric monoidal bicategory, A be a 0-cell, and n be a natural number. We write $A^{\otimes n}$ for the n-fold monoidal products $A \otimes \cdots \otimes A$. For $\sigma \in \mathfrak{S}_n$, we write $A^{\otimes \sigma} : A^{\otimes n} \to A^{\otimes n}$ for the structural 1-cell induced by σ and the symmetry structure of the symmetric monoidal bicategory $\mathscr{B}$. We define $A^{\otimes n}$ and $A^{\otimes\sigma}$ similarly for a symmetric monoidal (1-)category. For example, for the monoidal category $(\mathbf{Set}, \times, 1)$ and $\sigma \in \mathfrak{S}_n$, the function $A^{\times\sigma} : A^n \to A^n$ is $(a_i)_{i\le n} \mapsto (a_{\sigma(i)})_{i\le n}$. We omit $\otimes$ from $A^{\otimes\sigma}$ and $A^{\otimes n}$ and write simply A^σ and A^n, if no confusion arises.

Proposition C.13. *Let $\mathcal{W}$ be a symmetric monoidal category. Then for any 0-cell $A : \mathcal{A} \to \mathcal{W}^{\mathrm{op}}$ in $\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ and $n \in \mathbb{N}$,*

1. *we have a 2-equaliser $(G_*)^{\mathrm{op}} : C^{\mathrm{op}} \nrightarrow A^{\hat\otimes n}$ in $\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ of the parallel 1-cells $(A^{\hat\otimes\sigma} : A^{\hat\otimes n} \to A^{\hat\otimes n})_{\sigma\in\mathfrak{S}_n}$ where the functor $G : (A^{\mathrm{op}})^{\hat\otimes n} \to C$ is the 2-coequaliser in $\mathbf{Cat}/\mathcal{W}$ of $\big((A^{\mathrm{op}})^{\hat\otimes(\sigma^{-1})} : (A^{\mathrm{op}})^{\hat\otimes n} \to (A^{\mathrm{op}})^{\hat\otimes n}\big)_{\sigma\in\mathfrak{S}_n}$,*
2. *for any 0-cell $B : \mathcal{B} \to \mathcal{W}^{\mathrm{op}}$ in $\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$, $(-) \mathbin{\hat\otimes} B$ preserves the 2-equaliser in the previous item.*

Proof. By Lemmas C.7 and C.8, the composite

$$\mathscr{F} : (\mathbf{Cat}/\mathcal{W})^{1\text{-op}} \to (\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}})^{1\text{-op}} \xrightarrow{\cong} \mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$$

preserves 2-limits. Item 1 follows from this, because, for each $\sigma \in \mathfrak{S}_n$, $\mathscr{F}(A^{\mathrm{op}\,\hat\otimes\sigma^{-1}})$ is isomorphic to $A^{\hat\otimes\sigma}$.

On Item 2, first note that $\mathbf{Cat}/\mathcal{W}$ has a monoidal structure defined similarly so that $\mathbf{Cat}/\mathcal{W} \to \mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}}$ is strict 2-monoidal. Hence we have the following commutative diagram:

$$\begin{array}{ccccc}
(\mathbf{Cat}/\mathcal{W})^{1\text{-op}} & \longrightarrow & (\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}})^{1\text{-op}} & \xrightarrow{\cong} & \mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}} \\
{\scriptstyle ((-)\hat\otimes B^{\mathrm{op}})^{1\text{-op}}}\big\downarrow & & {\scriptstyle ((-)\hat\otimes B^{\mathrm{op}})^{1\text{-op}}}\big\downarrow & & \big\downarrow{\scriptstyle (-)\hat\otimes B} \\
(\mathbf{Cat}/\mathcal{W})^{1\text{-op}} & \longrightarrow & (\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}})^{1\text{-op}} & \xrightarrow{\cong} & \mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}
\end{array}$$

We only need to show that the composite

$$(\mathbf{Cat}/\mathcal{W})^{1\text{-op}} \to (\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}})^{1\text{-op}} \xrightarrow{\cong} \mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}} \xrightarrow{(-)\hat\otimes B} \mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$$

preserves 2-limits. Since the two pseudofunctors

$$(\mathbf{Cat}/\mathcal{W})^{1\text{-op}} \to (\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}})^{1\text{-op}} \xrightarrow{\cong} \mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$$

on the bottom line in above diagram preserve 2-limits, it suffices to show that the pseudofunctor $((-) \mathbin{\hat\otimes} B^{\mathrm{op}})^{1\text{-op}}$ on $(\mathbf{Cat}/\mathcal{W})^{1\text{-op}}$ preserves 2-limits, i.e. $(-) \mathbin{\hat\otimes} B^{\mathrm{op}}$ on $\mathbf{Cat}/\mathcal{W}$ preserves 2-colimits.

Now we have the following diagram:

$$\begin{array}{ccc}
\mathbf{Cat}/\mathcal{W} & \xrightarrow{(-)\hat\otimes B^{\mathrm{op}}} & \mathbf{Cat}/\mathcal{W} \\
\downarrow & & \downarrow \\
\mathbf{Cat} & \xrightarrow[(-)\times\mathcal{B}^{\mathrm{op}}]{} & \mathbf{Cat}
\end{array}$$

Here since $\mathbf{Cat}$ is 2-cartesian 2-closed, $(-) \times \mathcal{B}^{\mathrm{op}}$ is left 2-adjoint and hence preserves 2-colimits. The projection $\mathbf{Cat}/\mathcal{W} \to \mathbf{Cat}$ creates 2-colimits; hence, especially the projection on the left above preserves 2-colimits and the projection on the right above reflects 2-colimits. Therefore $(-) \mathbin{\hat\otimes} B^{\mathrm{op}}$ preserves 2-colimits. □

As an immediate corollary of the above proposition, we have:

Proposition C.14. *Let $\mathcal{W}$ be a symmetric monoidal category. Then for any object $A : \mathcal{A} \to \mathcal{W}^{\mathrm{op}}$ in $\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ and $n \in \mathbb{N}$,*

1. *we have an equaliser in $\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ of the parallel morphisms $(A^{\hat\otimes\sigma} : A^{\hat\otimes n} \to A^{\hat\otimes n})_{\sigma\in\mathfrak{S}_n}$,*
2. *for any object $B : \mathcal{B} \to \mathcal{W}^{\mathrm{op}}$ in $\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$, $(-) \mathbin{\hat\otimes} B$ preserves the equaliser in the previous item.*

C.7 $\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ is a $\lambda_{\mathcal{W}}$-model

Now we have an SMCC $\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ with countable biproducts and equalisers of $(A^{\hat\otimes\sigma})_{\sigma\in\mathfrak{S}_n}$. Hence by the construction in [35][28, Proposition II.3], we have a Lafont category $\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$. See Appendix D for a concrete description of this Lafont structure.

On the list structure, it is well known that, if C has countable coproducts and an endofunctor F on C preserves countable coproducts, then $\coprod_{n\in\mathbb{N}} F^n(I)$ is an initial algebra of $I + F(-)$. Thus, by the countable biproducts and the SMCC structure of $\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$, for any object A in $\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$, we have an initial algebra of the endofunctor $\hat I \oplus (A \mathbin{\hat\otimes} (-))$, given by $\oplus_{n\in\mathbb{N}} A^{\hat\otimes n}$.

D Concrete Description of Lafont Model $\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$

Here we give a *concrete* description of the Lafont-structure of the bicategory $\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ given in Appendix C. This concrete description is convenient for showing the equivalence with the Taylor expansion, and also should be easy to understand for readers who are not much familiar with (2-)category theory.

On the style of 1-cell of $\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$, here we use that in Definition 4.3 rather than the lax-slice style in Appendix C.

D.1 The bicategory $\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$

Let $F : \mathcal{A} \nrightarrow \mathcal{B}$ be a profunctor. For $e \in F(b, a)$ and $f : a \to a'$, we write $e \cdot f$ to mean $F(b, f)(e) \in F(b, a')$ (provided that F is clear from the context). Similarly, for $e \in F(b, a)$ and $g : b' \to b$, the expression $g \cdot e$ indicates $F(g, a)(e) \in F(b', a)$. As F is a functor from $\mathcal{B}^{\mathrm{op}} \times \mathcal{A}$, we have $F(g, a') \circ F(b, f) = F(g, f) = F(b', f) \circ F(g, a)$; hence the expression $f \cdot e \cdot g$ is unambiguous.

The concrete definition of $\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ is as follows:

- 0-cell: a weighted category $A : \mathcal{A} \to \mathcal{W}^{\mathrm{op}}$.
- 1-cell: $(F, \varpi) : (\mathcal{A}, A) \to (\mathcal{B}, B)$ is a pair of a profunctor $F : \mathcal{A} \nrightarrow \mathcal{B}$ and a *weight function* $\varpi_{(b,a)} : F(b, a) \to \mathcal{W}(A(a), B(b))$ that respects the action of $\mathcal{A}$ and $\mathcal{B}$, i.e.,
$$A(f) \circ \varpi_{(b,a)}(e) \circ B(g) \;=\; \varpi_{(b',a')}(f \cdot e \cdot g)$$
for every $g : b' \to b$, $e \in F(b, a)$ and $f : a \to a'$.
- 2-cell: $\alpha : (F, \varpi^F) \Rightarrow (G, \varpi^G)$ is a 2-cell $\alpha : F \Rightarrow G$ of $\mathbf{Prof}$ (i.e. a natural transformation $\alpha : F \Rightarrow G$ of $F, G : \mathcal{B}^{\mathrm{op}} \times \mathcal{A} \to \mathbf{Set}$) that preserves the weights, i.e.,
$$\varpi^F_{(b,a)}(e) = \varpi^G_{(b,a)}(\alpha_{b,a}(e))$$
for every $e \in F(b, a)$.

Below we omit the description of 2-cells of $\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$, since they do not (explicitly) occur in $\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$.

D.2 Symmetric Monoidal Structure

Let $(\mathcal{W}, \otimes, I)$ be an SMC.

- Let $A : \mathcal{A} \to \mathcal{W}^{\mathrm{op}}$ and $B : \mathcal{B} \to \mathcal{W}^{\mathrm{op}}$ be weighted categories. We define their tensor product as $(\mathcal{A}, A) \mathbin{\hat\otimes} (\mathcal{B}, B) \triangleq (\mathcal{A} \times \mathcal{B}, A \mathbin{\hat\otimes} B)$ where
$$A \mathbin{\hat\otimes} B \triangleq (\otimes^{\mathrm{op}}) \circ (A \times B),$$
i.e. $(A \mathbin{\hat\otimes} B)(a, b) = A(a) \otimes B(b)$ and $(A \mathbin{\hat\otimes} B)(f, g) = A(f) \otimes B(g)$.
- Given 1-cells $(F_i, \varpi_i) : (\mathcal{A}_i, A_i) \to (\mathcal{B}_i, B_i)$ $(i = 1, 2)$, we define $(F_1, \varpi_1) \mathbin{\hat\otimes} (F_2, \varpi_2) = (G, \varpi)$ as follows. The profunctor G :

$\mathcal{A}_1 \times \mathcal{A}_2 \nrightarrow \mathcal{B}_1 \times \mathcal{B}_2$ is defined by

$$(\mathcal{B}_1 \times \mathcal{B}_2)^{\mathrm{op}} \times (\mathcal{A}_1 \times \mathcal{A}_2) \cong (\mathcal{B}_1^{\mathrm{op}} \times \mathcal{A}_1) \times (\mathcal{B}_2^{\mathrm{op}} \times \mathcal{A}_2) \xrightarrow{F_1 \times F_2} \mathbf{Set} \times \mathbf{Set} \xrightarrow{\times} \mathbf{Set}.$$

More explicitly

$$G((b_1, b_2), (a_1, a_2)) \triangleq F_1(b_1, a_1) \times F_2(b_2, a_2)$$
$$G((g_1, g_2), (f_1, f_2)) \triangleq F_1(g_1, f_1) \times F_2(g_2, f_2).$$

The weight function

$$\varpi_{(b_1,b_2),(a_1,a_2)} : G((b_1, b_2), (a_1, a_2)) \to \mathcal{W}(A_1(a_1) \otimes A_2(a_2), B_1(b_1) \otimes B_2(b_2))$$

is defined by

$$\varpi_{(b_1,b_2),(a_1,a_2)}(e_1, e_2) \triangleq \left((\varpi_1)_{b_1,a_1}(e_1)\right) \otimes \left((\varpi_2)_{b_2,a_2}(e_2)\right).$$

D.3 Closed Structure

Let $(\mathcal{W}, \otimes, I, \multimap)$ be an SMCC.

Let $A : \mathcal{A} \to \mathcal{W}^{\mathrm{op}}$ and $B : \mathcal{B} \to \mathcal{W}^{\mathrm{op}}$ be weighted categories. We define their linear function space as $(\mathcal{A}, A) \mathbin{\hat{\multimap}} (\mathcal{B}, B) \triangleq (\mathcal{A}^{\mathrm{op}} \times \mathcal{B}, A \mathbin{\hat{\multimap}} B)$ where

$$A \mathbin{\hat{\multimap}} B \triangleq (\multimap^{\mathrm{op}}) \circ (A^{\mathrm{op}} \times B),$$

i.e. $(A \mathbin{\hat{\multimap}} B)(a, b) = A(a) \multimap B(b)$ and $(A \mathbin{\hat{\multimap}} B)(f, g) = A(f) \multimap B(g)$.

The equivalence between

$$(\mathcal{A}, A) \mathbin{\hat{\otimes}} (\mathcal{B}, B) \nrightarrow (C, C) \quad \text{and} \quad (\mathcal{A}, A) \nrightarrow (\mathcal{B}, B) \mathbin{\hat{\multimap}} (C, C)$$

is given as follows. Assume that $(F, \varpi) : (\mathcal{A}, A) \mathbin{\hat{\otimes}} (\mathcal{B}, B) \nrightarrow (C, C)$. Then $F : C^{\mathrm{op}} \times (\mathcal{A} \times \mathcal{B}) \to \mathbf{Set}$. Hence it can be identified with $F' : (\mathcal{B}^{\mathrm{op}} \times C)^{\mathrm{op}} \times \mathcal{A} \to \mathbf{Set}$. Given $e \in F'((b, c), a) = F(c, (a, b))$, we define

$$\varpi'(e) \triangleq \lambda(\varpi(e)).$$

The pseudo-inverse is obvious.

D.4 Biproducts

We only describe the binary case for simplicity.

- Let $A : \mathcal{A} \to \mathcal{W}^{\mathrm{op}}$ and $B : \mathcal{B} \to \mathcal{W}^{\mathrm{op}}$ be weighted categories. We define their biproduct as $(\mathcal{A}, A) \oplus (\mathcal{B}, B) \triangleq (\mathcal{A} + \mathcal{B}, [A, B])$ where $[A, B] : \mathcal{A} + \mathcal{B} \to \mathcal{W}^{\mathrm{op}}$ is the canonical functor given by the coproduct structure of **Cat**.
- Given 1-cells $(F_i, \varpi_i) : (A_i, \mathcal{A}_i) \to (B_i, \mathcal{B}_i)$ $(i = 1, 2)$, we define $(F_1, \varpi_1) \oplus (F_2, \varpi_2) = (G, \varpi)$ as follows. The profunctor $G : (\mathcal{A}_1 + \mathcal{A}_2) \nrightarrow (\mathcal{B}_1 + \mathcal{B}_2)$ is defined by

$$G(b, a) \triangleq \begin{cases} F_1(b, a) & (\text{if } a \in \mathcal{A}_1 \text{ and } b \in \mathcal{B}_1) \\ F_2(b, a) & (\text{if } a \in \mathcal{A}_2 \text{ and } b \in \mathcal{B}_2) \\ \emptyset & (\text{otherwise}) \end{cases}$$

$$G(g, f) \triangleq \begin{cases} F_1(g, f) & (\text{if } f \text{ in } \mathcal{A}_1 \text{ and } g \text{ in } \mathcal{B}_1) \\ F_2(g, f) & (\text{if } f \text{ in } \mathcal{A}_2 \text{ and } g \text{ in } \mathcal{B}_2) \\ \mathrm{id}_\emptyset & (\text{otherwise}). \end{cases}$$

The weight function $\varpi_{b,a} : G(b, a) \to \mathcal{W}([A_1, A_2](a), [B_1, B_2](b))$ is defined by

$$\varpi_{b,a}(e) = \begin{cases} (\varpi_1)_{b,a}(e) & (\text{if } a \in \mathcal{A}_1 \text{ and } b \in \mathcal{B}_1) \\ (\varpi_2)_{b,a}(e) & (\text{if } a \in \mathcal{A}_2 \text{ and } b \in \mathcal{B}_2). \end{cases}$$

D.5 Symmetric Tensor Powers

Given a weighted category $A : \mathcal{A} \to \mathcal{W}^{\mathrm{op}}$ and $n \in \mathbb{N}$, the equaliser $\mathbb{P}_n(\mathcal{A}, A)$ of $((\mathcal{A}, A)^{\hat{\otimes}\sigma})_{\sigma \in \mathfrak{S}_n}$ is called *symmetric tensor powers* [35], which is used for defining the linear exponential comonad later.

***The coequaliser in* Cat** The construction of the 2-equaliser in Appendix C constructs the 2-equaliser in $\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ by a certain 2-coequaliser that is created finally in **Cat** (by the 2-colimit-creation of the projection $\mathbf{Cat}/\mathcal{W} \to \mathbf{Cat}$). Any 2-coequaliser in **Cat** can be calculated by a generalised congruence [3], but in the current case we have the following simple 2-coequaliser.

Let $\mathcal{A}$ be a category and $n \in \mathbb{N}$. Recall that the functor $\mathcal{A}^\sigma : \mathcal{A}^n \to \mathcal{A}^n$ induced by a permutation $\sigma \in \mathfrak{S}_n$ maps $(a_i)_{i \le n}$ to $(a_{\sigma(i)})_{i \le n}$.

We define a category $\mathbb{P}^*_n(\mathcal{A})$, the vertex of the 2-coequaliser of $(\mathcal{A}^\sigma)_{\sigma \in \mathfrak{S}_n}$, as follows:

- The objects of $\mathbb{P}^*_n(\mathcal{A})$ are lists $(a_1, \dots, a_n)$ of objects in $\mathcal{A}$ of length n.
- A morphism $(a_1, \dots, a_n) \to (a'_1, \dots, a'_n)$ in $\mathbb{P}^*_n(\mathcal{A})$ is a pair of a permutation $\sigma \in \mathfrak{S}_n$ and a family $(f_i : a_{\sigma(i)} \to a'_i)_{i=1}^n$ of arrows in $\mathcal{A}$.

Now we have the obvious embedding functor $E_{\mathcal{A}} : \mathcal{A}^n \to \mathbb{P}^*_n(\mathcal{A})$, which does not change objects, and maps morphism $(f_i)_i$ to $(id, (f_i)_i)$. Then it can be easily checked that this functor $E_{\mathcal{A}}$ is an 2-coequaliser of $(\mathcal{A}^\sigma)_{\sigma \in \mathfrak{S}_n}$. (For example, this coequalises $\mathcal{A}^\sigma$ and $Id_{\mathcal{A}^n}$ as follows: $\mathcal{A}^\sigma((a_i)_i) = (a_{\sigma(i)})_i$ is isomorphic to $Id_{\mathcal{A}^n}((a_i)_i) = (a_i)_i$ by $(\sigma^{-1}, (id_{a_i})_i) : (a_{\sigma(i)})_i \to (a_i)_i$ and its inverse $(\sigma, (id_{a_{\sigma(i)}})_i) : (a_i)_i \to (a_{\sigma(i)})_i$.)

Then by the construction in Appendix C, we have an equaliser $((E_{\mathcal{A}^{\mathrm{op}}})_*)^{\mathrm{op}} = (E^{\mathrm{op}}_{\mathcal{A}^{\mathrm{op}}})^* : (\mathbb{P}^*_n(\mathcal{A}^{\mathrm{op}}))^{\mathrm{op}} \nrightarrow ((\mathcal{A}^{\mathrm{op}})^n)^{\mathrm{op}} = \mathcal{A}^n$ of parallel arrows $((\mathcal{A}, A)^{\hat{\otimes}\sigma})_\sigma$ in $\mathbf{Prof}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$. Below we give a concrete description of this 2-equaliser.

The equaliser $\mathbb{P}_n(\mathcal{A}, A)$ ***of parallel morphisms*** $((\mathcal{A}, A)^{\hat{\otimes}\sigma})_\sigma$ We define the weighted category as $\mathbb{P}_n(\mathcal{A}, A) \triangleq (\mathbb{P}_n(\mathcal{A}), \mathbb{P}_n(A))$ where $\mathbb{P}_n(\mathcal{A}) \triangleq (\mathbb{P}^*_n(\mathcal{A}^{\mathrm{op}}))^{\mathrm{op}}$, which is specifically as follows:

- The objects of $\mathbb{P}_n(\mathcal{A})$ are lists $(a_1, \dots, a_n)$ of objects in $\mathcal{A}$ of length n.
- A morphism $(a_1, \dots, a_n) \to (a'_1, \dots, a'_n)$ in $\mathbb{P}_n(\mathcal{A})$ is a pair of a permutation $\sigma \in \mathfrak{S}_n$ and a family $(f_i : a_i \to a'_{\sigma(i)})_{i=1}^n$ of arrows in $\mathcal{A}$. In other words,

$$\mathbb{P}_n(\mathcal{A})((a_i)_i, (a'_i)_i) \triangleq \coprod_{\sigma \in \mathfrak{S}_n} \mathcal{A}^n((a_i)_i, (a'_{\sigma(i)})_i) = \coprod_{\sigma \in \mathfrak{S}_n} \prod_{i=1}^n \mathcal{A}(a_i, a'_{\sigma(i)}).$$

Also, the functor $\mathbb{P}_n(A)$ is defined as follows:

- The functor $\mathbb{P}_n(A) : \mathbb{P}_n(\mathcal{A}) \to \mathcal{W}^{\mathrm{op}}$ maps an object $(a_1, \dots, a_n)$ to $\otimes_{i=1}^n A(a_i)$
- The functor $\mathbb{P}_n(A) : \mathbb{P}_n(\mathcal{A}) \to \mathcal{W}^{\mathrm{op}}$ maps an arrow $(\sigma, (f_i)_i) : (a_1, \dots, a_n) \to (a'_1, \dots, a'_n)$ to

$$\otimes_i A(a'_i) \xrightarrow{\cong} \otimes_i A(a'_{\sigma(i)}) \xrightarrow{\otimes_i A(f_i)} \otimes_i A(a_i).$$

The equaliser 1-cell $(\mathsf{eq}^A_n, \varpi) : (\mathbb{P}_n(\mathcal{A}), \mathbb{P}_n(A)) \nrightarrow (\mathcal{A}, A)^{\hat{\otimes} n}$ is given by:

$$\mathsf{eq}^A_n((a'_1, \dots, a'_n), (a_1, \dots, a_n)) = \coprod_{\sigma \in \mathfrak{S}_n} \prod_{i=1}^{n} \mathcal{A}(a'_i, a_{\sigma(i)})$$

and

$$\varpi_{(a'_i)_i, (a_i)_i}(\sigma, (f_i)_{i=1}^n) \triangleq \otimes_i A(a_i) \xrightarrow{\cong} \otimes_i A(a_{\sigma(i)}) \xrightarrow{\otimes_{i=1}^n A(f_i)} \otimes_i A(a'_i).$$

The functor by the equaliser The construction $\mathbb{P}_n(-)$ extends to a functor on $\mathbf{Pr}/\!\!/^{\mathbf{Cat}}_{\mathcal{W}^{\mathrm{op}}}$ as follows: Given a weighted profunctor $(F, \varpi) : (\mathcal{A}, A) \nrightarrow (\mathcal{B}, B)$, we define

$$(\mathbb{P}_n(F), \mathbb{P}_n(\varpi)) : (\mathbb{P}_n(\mathcal{A}), \mathbb{P}_n(A)) \nrightarrow (\mathbb{P}_n(\mathcal{B}), \mathbb{P}_n(B))$$

by

$$\mathbb{P}_n(F)((b_1, \dots, b_n), (a_1, \dots, a_n)) \triangleq \coprod_{\sigma \in \mathfrak{S}_n} \prod_{i=1}^{n} F\left(b_i, a_{\sigma(i)}\right)$$

and

$$\begin{aligned}
&\mathbb{P}_n(\varpi)_{(b_i)_i, (a_i)_i} : \\
&\mathbb{P}_n(F)\big((b_i)_i, (a_i)_i\big) \to \mathcal{W}\big(\mathbb{P}_n(A)((a_i)_i), \mathbb{P}_n(B)((b_i)_i)\big) \\
&\left(= \coprod_{\sigma \in \mathfrak{S}_n} \prod_{i=1}^{n} F\left(b_i, a_{\sigma(i)}\right) \to \mathcal{W}\big(\otimes_i A(a_i), \otimes_i B(b_i)\big)\right) \\
&\big(\sigma, (e_i)_{i=1}^n\big) \mapsto \left(\otimes_i A(a_i) \xrightarrow{\cong} \otimes_i A(a_{\sigma(i)}) \xrightarrow{\otimes_i(\varpi_{b_i, a_{\sigma(i)}}(e_i))} \otimes_i B(b_i)\right)
\end{aligned}$$

where note that $\varpi_{b_i, a_{\sigma(i)}} : F\left(b_i, a_{\sigma(i)}\right) \to \mathcal{W}\left(A(a_{\sigma(i)}), B(b_i)\right)$.

D.6 Linear Exponential Comonad

The underlying functor of the comonad is defined as:

$$\begin{aligned}
\mathbb{P}(\mathcal{A}, A) &\triangleq \oplus_{n \in \mathbb{N}} \mathbb{P}_n(\mathcal{A}, A) = \left(\coprod_n \mathbb{P}_n(\mathcal{A}), [\mathbb{P}_n(A)]_n\right) \\
\mathbb{P}(F, \varpi) &\triangleq \oplus_{n \in \mathbb{N}} \mathbb{P}_n(F, \varpi).
\end{aligned}$$

The comonad structure The counit $\varepsilon = (F, \varpi) : \oplus_{n \in \mathbb{N}} \mathbb{P}_n(\mathcal{A}, A) \to (\mathcal{A}, A)$ of the comonad is given by:

$$F(a', (n, (a_i)_{i \le n})) = \begin{cases} \mathbb{P}_n(\mathcal{A})((a'), (a_1)) & (n = 1) \\ \emptyset & (\text{otherwise}) \end{cases}$$

and: $\varpi_{a', (n, (a_i)_{i \le n})}$ is the empty-function when $n \neq 1$, and when $n = 1$,

$$\begin{aligned}
\varpi_{a', (1, (a_1))} : \mathbb{P}_n(\mathcal{A})((a'), (a_1)) &\to \mathcal{W}(\mathbb{P}_1(A)(a_1), A(a')) \\
\big(= \mathcal{A}(a', a_1) &\to \mathcal{W}(A(a_1), A(a'))\big) \\
f &\mapsto A(f).
\end{aligned}$$

The comultiplication

$$\nu = (F, \varpi) : \oplus_n \mathbb{P}_n(\mathcal{A}, A) \to \oplus_n \mathbb{P}_n(\oplus_m \mathbb{P}_m(\mathcal{A}, A))$$

of the comonad where

$$\oplus_n \mathbb{P}_n(\oplus_m \mathbb{P}_m(\mathcal{A}, A)) = \left(\coprod_n \mathbb{P}_n\Big(\coprod_m \mathbb{P}_m(\mathcal{A})\Big), [\mathbb{P}_n([\mathbb{P}_m(A)]_m)]_n\right)$$

is given as follows:

$$\begin{aligned}
&F\left(\left(n', \left((m'_i, (a'_{i,j})_{j \le m'_i})\right)\right)_{i \le n'}, (n, (a_i)_{i \le n})\right) \\
&= \begin{cases} \mathbb{P}_n(\mathcal{A})\left((a'_{i,j})_{i \le n', j \le m'_i}, (a_i)_{i \le n}\right) & (n = \sum_{i \le n'} m'_i) \\ \emptyset & (\text{otherwise}). \end{cases}
\end{aligned}$$

When $n \neq \sum_{i \le n'} m'_i$, the weight function

$$\varpi_{\left(n', \left((m'_i, (a'_{i,j})_{j \le m'_i})\right)\right)_{i \le n'}, (n, (a_i)_{i \le n})}$$

from the empty set is unique, and when $n = \sum_{i \le n'} m'_i$, we have:

$$\begin{aligned}
&\varpi_{\left(n', \left((m'_i, (a'_{i,j})_{j \le m'_i})\right)\right)_{i \le n'}, (n, (a_i)_{i \le n})} : \\
&\mathbb{P}_n(\mathcal{A})\left((a'_{i,j})_{i \le n', j \le m'_i}, (a_i)_{i \le n}\right) \to \\
&\quad \mathcal{W}\Big([\mathbb{P}_n(A)]_n\big((n, (a_i)_{i \le n})\big), \\
&\qquad [\mathbb{P}_{n'}([\mathbb{P}_{m'}(A)]'_m)]'_n\left(\left(n', \left((m'_i, (a'_{i,j})_{j \le m'_i})\right)\right)_{i \le n'}\right)\Big) \\
&\left(= \coprod_{\sigma \in \mathfrak{S}_n} \mathcal{A}^n\left((a'_{i,j})_{i \le n', j \le m'_i}, (a_{\sigma(i)})_{i \le n}\right) \to \mathcal{W}\left(\otimes_{i \le n} a_i,\ \otimes_{i \le n'} \otimes_{j \le m'_i} a'_{i,j}\right)\right) \\
&(\sigma, (f_i)_{i \le n}) \mapsto \\
&\quad \left(\otimes_{i \le n} a_i \xrightarrow{\cong} \otimes_{i \le n} a_{\sigma(i)} \xrightarrow{\otimes_{i \le n} A(f_i)} \otimes_{i \le n'} \otimes_{j \le m'_i} a'_{i,j}\right).
\end{aligned}$$

The comonoid structure The cofree comonoid structure of $\mathbb{P}(\mathcal{A}, A)$ is given as follows: the counit $(F, \varpi) : (\coprod_n \mathbb{P}_n(\mathcal{A}), [\mathbb{P}_n(A)]_n) \to (1, I)$ of the comonoid is given by:

$$F\big(*, (n, (a_i)_{i \le n})\big) \triangleq \begin{cases} \{*\} & (n = 0) \\ \emptyset & (\text{otherwise}) \end{cases}$$

and

$$\varpi_{*, (n, (a_i)_{i \le n})} \triangleq \begin{cases} * \mapsto id_I \in \mathcal{W}(I, I) & (n = 0) \\ \text{the empty-function} & (\text{otherwise}) \end{cases}$$

The comultiplication

$$\begin{aligned}
(F, \varpi) : &\left(\coprod_n \mathbb{P}_n(\mathcal{A}),\ [\mathbb{P}_n(A)]_n\right) \to \\
&\left(\coprod_n \mathbb{P}_n(\mathcal{A}) \times \coprod_n \mathbb{P}_n(\mathcal{A}),\ \otimes^{\mathrm{op}} \circ \big([\mathbb{P}_n(A)]_n \times [\mathbb{P}_n(A)]_n\big)\right)
\end{aligned}$$

of the comonoid is given as follows:

$$\begin{aligned}
&F\left(\left((n', (a'_i)_{i \le n'}),\ (n'', (a''_i)_{i \le n''})\right),\ (n, (a_i)_{i \le n})\right) \\
&\triangleq \begin{cases} \mathbb{P}_n(\mathcal{A})\left((a'_1, \dots a'_{n'}, a''_1, \dots a''_{n''}), (a_i)_{i \le n}\right) & (n = n' + n'') \\ \emptyset & (\text{otherwise}) \end{cases}
\end{aligned}$$

When $n \neq n' + n''$, the weight function

$$\varpi_{\left((n', (a'_i)_{i \le n'}),\ (n'', (a''_i)_{i \le n''})\right),\ (n, (a_i)_{i \le n})}$$

from the empty set is unique, and when $n = n' + n''$, we have:

$$
\begin{aligned}
&\varpi_{\left((n',(a'_i)_{i\leq n'}),\ (n'',(a''_i)_{i\leq n''})\right),\ (n,(a_i)_{i\leq n})} : \\
&\mathbb{P}_n(\mathcal{A})\left((a'_1, \dots a'_{n'}, a''_1, \dots a''_{n''}), (a_i)_{i\leq n}\right) \to \\
&\qquad \mathcal{W}\Big([\mathbb{P}_n(A)]_n\left((n,(a_i)_{i\leq n})\right), \\
&\qquad\qquad [\mathbb{P}_{n'}([\mathbb{P}_{m'}(A)]'_m)]'_n\left((n',(a'_i)_{i\leq n'}),\ (n'',(a''_i)_{i\leq n''})\right)\Big) \\
&\left(= \coprod_{\sigma\in\mathfrak{S}_n} \mathcal{A}^n\left((a'_1, \dots a'_{n'}, a''_1, \dots a''_{n''}), (a_{\sigma(i)})_{i\leq n}\right) \to \right. \\
&\qquad\qquad \left. \mathcal{W}\left(\otimes_{i\leq n} a_i,\ (\otimes_{i\leq n'} a'_i) \otimes (\otimes_{i\leq n''} a''_i)\right) \right) \\
&\left(\sigma, (f_i)_{i\leq n}\right) \mapsto \\
&\qquad \left(\otimes_{i\leq n} a_i \xrightarrow{\cong} \otimes_{i\leq n} a_{\sigma(i)} \xrightarrow{\otimes_{i\leq n} A(f_i)} (\otimes_{i\leq n'} a'_i) \otimes (\otimes_{i\leq n''} a''_i)\right).
\end{aligned}
$$