Categorifying Non-Idempotent Intersection Types

Giulio Guerrieri
University of Bath, Department of Computer Science, Bath, UK

Federico Olimpieri
Institut de Mathématiques de Marseille (I2M), Aix-Marseille Université, Marseille, France

Abstract

Non-idempotent intersection types can be seen as a syntactic presentation of a well-known denotational semantics for the lambda-calculus, the category of sets and relations. Building on previous work, we present a categorification of this line of thought in the framework of the bang calculus, an untyped version of Levy's call-by-push-value. We define a bicategorical model for the bang calculus, whose syntactic counterpart is a suitable category of types. In the framework of distributors, we introduce intersection type distributors, a bicategorical proof relevant refinement of relational semantics. Finally, we prove that intersection type distributors characterize normalization at depth 0.

2012 ACM Subject Classification Theory of computation → Lambda calculus; Theory of computation → Linear logic; Theory of computation → Categorical semantics

Keywords and phrases Linear logic, bang calculus, non-idempotent intersection types, distributors, relational semantics, combinatorial species, symmetric sequences, bicategory, categorification

Digital Object Identifier 10.4230/LIPIcs.CSL.2021.25

Funding This work is partially supported by EPSRC Project EP/R029121/1 *Typed lambda-calculi with sharing and unsharing.*

Acknowledgements The authors thank Lionel Vaux Auclair for insightful discussions and comments.

1 Introduction

Since Girard's introduction of *linear logic* [32], the notion of linearity has played a central role in the Logic-in-Computer-Science community. A program is linear when it uses its inputs only *once* during computation (inputs cannot be copied or deleted); while a non-linear program may call its inputs at will. Via the exponential modalities ! and ?, linear logic gives a logical status to the operations of erasing and copying data.

Another way to study linearity is provided by some type systems. *Intersection types* were introduced by Coppo and Dezani [14, 15] as an extension of simple types by means of the (associative, commutative and idempotent) intersection connective $a \cap b$: a term of type $a \cap b$ can be seen as a program of both type a and type b. This kind of type systems have proven to be very useful to characterize various notion of normalization in the λ-calculus [37]. If we impose *non-idempotency* to the intersection [31, 16] (i.e. $a \cap a \neq a$), we get a "resource-sensitive" intersection type system, in the sense that the arrow type encodes the *exact* number of times that a term needs its input during computation: intuitively, a term typed $a \cap a \cap b$ can be used twice as a program of type a and once as a program of type b. Non-idempotent intersection types allow *combinatorial* characterization of normalization properties and of the execution time of programs [9, 16, 4] and proof-nets [19, 20]. Also, De Carvalho's non-idempotent intersection type system $\mathcal{R}$ is a syntactic presentation of the categorical semantics of λ-calculus given in the category of sets and relations [16, 17]. There is a strong connection between linear logic and non-idempotent intersection types [18].

Inspired by [34, 41, 48, 43], we propose here a *categorification* of this kind of semantics. Roughly, categorification consists in replacing set-theoretic notions with category-theoretic ones. In general, this process gives both more fine-grained structures and general points of view. Melliès and Zeilberger [42] followed this approach to present a categorical definition of what a type system is: a type system is a *functor* between a category of type derivations and a category of terms. Since we are interested in categorical semantics with an intersection type presentation, the first natural thing to do is replacing the category of sets and relations with the bicategory of *distributors* [6, 10]. Distributor-induced semantics of programming languages were already presented in [12, 27]. In particular, Fiore, Gambino, Hyland and Winskel introduced the bicategory of *generalized species of structure* [27], a very rich framework that generalizes both relational semantics and Joyal's *combinatorial species* [35, 27, 30, 48]. As shown in [12, 29], distributors can also lead to a generalization of Scott's semantics.

Mazza, Pellissier and Vial [41], inspired by [42] and Hyland's project of categorification of the theory of the λ-calculus [34], presented a general approach to intersection types rooted in the notion of multicategory. In their framework, the λ-calculus is seen as a 2-operad, where 2-cells consist of reduction paths. Intersection type systems are seen as a special kind of *fibrations*. Via a Grothendieck construction, with these fibrations they associate an *approximation presheaf* that interprets terms as *discrete distributors*. Thanks to this categorical approach, they are able to prove a parametric normalization theorem for a class of intersection type systems in a modular and elegant way. Their method relies on a Curry-Howard style correspondence between intersection type derivations and a kind of λ-terms *approximants*, the *polyadic terms*. However, their approach does not provide a denotational model and it does not support subtyping for intersection types. This latter feature is strictly linked to the fact that approximation presheaves action on types is restricted to discrete categories [41]. It is then natural to ask what happens when we take the standpoint of denotational semantics and we take into account categories with non-trivial morphisms.

Recently, Tsukada, Asada and Ong [48, 49] presented the *rigid Taylor expansion*[1] semantics for an η-expanded fragment of non-deterministic simply-typed λ-calculus with fixed point combinator, then extended to probabilistic and quantum computation: the linear approximants are still polyadic terms. They proved that this semantics is naturally isomorphic to the generalized species semantics. This time, the standpoint is the one of denotational semantics and distributors ranges over groupoids, but subtyping is not taken into account. The groupoid structure of the model gives the possibility to define an *action* of type isomorphisms on polyadic terms. A quotient induced by this action guarantees the preservation, up to isomorphism, of the semantics under reduction. Concretely one has that $[\![M]\!] \cong [\![N]\!]$ whenever $M \to N$ and the natural isomorphism is given by *reduction* of polyadic terms.

Inspired by these lines of thought, Olimpieri [43] introduced *intersection type distributors*, a categorified version of intersection type disciplines, where subtyping and denotational semantics are both taken into account. Intersection type distributors are a *syntactic presentation* of bicategorical denotational semantics for the λ-calculus given by Kleisli bicategories of distributors for suitable pseudomonads. Each pseudomonad taken into account gives rise to a notion of intersection type, with specific resource behavior[2]. The semantics obtained by this method is *proof relevant*: given a term M, a type context Δ and a type a, we set $\mathsf{T}_U(M)(\Delta, a) = \left\{ \begin{array}{c} \tilde{\pi} \\ \vdots \\ \Delta \vdash M : a \end{array} \mid \pi \text{ is a type derivation for } M \right\}$ where $\mathsf{T}_U(M)$ is the intersection type distributor that interprets M in an appropriate category U of types, and

[1] The rigid Taylor expansion is a deterministic variant of Ehrhard and Regnier's Taylor expansion [25, 26].

[2] It is worth noting that this new semantic setting is not a special case of [41], as standard polyadic terms fail dramatically subject reduction for intersection type distributors. The failure of subject reduction happens because standard polyadic terms [48, 41] cannot encode all the *qualitative* information produced by the *subtyping* feature of intersection type distributors. A counterexample is in Appendix A.

$\tilde{\pi}$ is an equivalence class of derivations. The equivalence relation on derivations is induced by the composition of distributors, which generalizes the quotient of [48]. We have that, if $M \to N$, then $\mathsf{T}_U(M) \cong \mathsf{T}_U(N)$. Categorification then allows us to pass from a semantics of *types* to a semantics of *derivations*. Note that, in our setting, the semantics of a term M associates with every type context Δ and type a the set of derivations for M with conclusion $\Delta \vdash M : a$; while more coarse-grained models such as relational semantics can only say if *there is* a type derivation for M with conclusion $\Delta \vdash M : a$.

In the present paper, we introduce *non-idempotent* intersection type distributors in an untyped call-by-push-value setting [33, 24, 39, 47], the *bang calculus*. Levy's call-by-push-value paradigm subsumes call-by-name (CbN) and call-by-value (CbV), from both the operational and denotational semantics standpoints [39, 33]. In this respect, our work is more general than [43] (which considers only the CbN λ-calculus). Moreover, inspired by linear logic, the bang calculus internalizes in the syntax the !-operator, which semantically corresponds to the monadic operator to handle resources. In this way, it is more natural to link syntax and semantics and to disentangle our investigation from the evaluation mechanism. Here we focus on a particular monadic construction (the symmetric strict monoidal completion, see Section 2) and we do not extend the more general and abstract method of [43] to the bang calculus because in this way we can avoid introducing too much categorical background.

Our categorical approach allows the introduction of a suitable *category of types*, where morphisms between types are a generalization of *subtyping*. Given a type morphism $a' \to a$, the intuition is that the type a' somehow *refines* the type a. We prove that non-idempotent intersection type distributors characterize normalization at depth 0 in the bang calculus. Normalization at depth 0 in the bang calculus is a notion that encompasses both CbN solvability [2, 37] and CbV potential valuability [45, 11]. The argument to prove this result is combinatorial and standard (similar results for the bang calculus are proved in [24, 8] using relational semantics), but thanks to the categorified setting we gain a much more fine-grained understanding of the dynamics of type derivations under reduction. Indeed, in our setting, subject reduction and expansion (Theorem 12) clearly open the possibility to define an explicit *deterministic* reduction relation on (equivalent classes) of type derivations, but the investigation of this line of thought is left to future work. We just notice that the substitution operation on type derivations is strictly linked to morphism composition, respecting the basic intuition of categorical semantics: substitution corresponds to composition.

Outline. Some preliminaries are in Section 2. Section 3 shows how the category of distributors Dist can be seen as a generalization of the categories Rel of sets and relations and Polr of preorders. In Section 4 we define a proof-relevant denotational model of the bang calculus in Dist as a generalization of non-idempotent intersection type systems and we prove a semantic characterization of depth 0 normalization in the bang calculus. Section 5 concludes. In Appendix A we recall some basic notions for bicategories and coends, we prove Lemma 11 and we show the failure of subject reduction with subtyping for polyadic terms.

2 Preliminaries

The bang calculus. The syntax and operational semantics of the *bang calculus* [33] are defined in Figure 1.[3] Terms are built up from a countably infinite set of *variables* (denoted by $x, y, z, \dots$). Terms of the form $S^!$ (resp. $\lambda x.S$; ST) are called *boxes* (resp. *abstractions*;

[3] Syntax and reduction rule of the bang calculus are presented as in [33], which are slightly different from [24]. But unlike [33] (and akin to [46]), here we do not use der as a primitive, since der and its associated rule $\mathsf{der}(S^!) \mapsto_{\mathrm{d}} S$ can be simulated in our setting by defining $\mathsf{der} = \lambda x.x$, because $(\lambda x.x)S^! \mapsto_{\mathrm{b}} S$.

Terms:	$S, T, U ::= x \mid \lambda x.S \mid ST \mid S^!$		(set: $!\Lambda$)
Contexts:	$\mathtt{C} ::= [\cdot] \mid \lambda x.\mathtt{C} \mid \mathtt{C}S \mid S\mathtt{C} \mid \mathtt{C}^!$		(set: $!\Lambda_{\mathtt{C}}$)
Ground Contexts:	$\mathtt{G} ::= [\cdot] \mid \lambda x.\mathtt{G} \mid \mathtt{G}S \mid S\mathtt{G}$		(set: $!\Lambda_{\mathtt{G}}$)
Root-step:	$(\lambda x.S)T^! \mapsto_{\mathrm{b}} S\{T/x\}$		
$\to_{\mathrm{b}}$ *-reduction*:	$S \to_{\mathrm{b}} T \;\Leftrightarrow\; \exists\, \mathtt{C} \in !\Lambda_{\mathtt{C}}, \exists\, S', T' \in !\Lambda : S = \mathtt{C}[S'],\, T = \mathtt{C}[T'],\, S' \mapsto_{\ell} T'$		
$\to_{\mathrm{b_g}}$ *-reduction*:	$S \to_{\mathrm{b_g}} T \;\Leftrightarrow\; \exists\, \mathtt{G} \in !\Lambda_{\mathtt{G}}, \exists\, S', T' \in !\Lambda : S = \mathtt{G}[S'],\, T = \mathtt{G}[T'],\, S' \mapsto_{\ell} T'$		

Figure 1 The bang calculus: its syntax and reduction rules.

(linear) applications). The set of boxes is denoted by $!\Lambda_!$. The set of free variables of a term S, denoted by $\mathsf{fv}(S)$, is defined as expected, λ being the only binding construct. All terms are considered up to α-conversion. Given $S, T \in !\Lambda$ and a variable x, $S\{T/x\}$ denotes the term obtained by the *capture-avoiding substitution* of T for each free occurrence of x in S.

Contexts $\mathtt{C}$ and (with exactly one hole $[\cdot]$) are defined in Figure 1. We write $\mathtt{C}[S]$ for the term obtained by capture-allowing substitution of the term S for the hole $[\cdot]$ in the context $\mathtt{C}$. *Ground contexts* $\mathtt{G}$ are the restriction to contexts where the hole is not inside any !.

The *bang calculus* is the set $!\Lambda$ endowed with reduction $\to_{\mathrm{b}}$ (Figure 1), which is confluent [33]. Intuitively in the root-step $\mapsto_{\mathrm{b}}$ the box-construct ! marks the only terms that can be erased and duplicated: a *β-like redex* $(\lambda x.S)T$ can be fired only when its argument is a box, *i.e.* $T = U^!$: if it is so, the content U of the box T replaces any free occurrence of x in S.

Reduction $\to_{\mathrm{b_g}} \subseteq \to_{\mathrm{b}}$ is said *at depth 0* and defined as the closure of $\mapsto_{\mathrm{b}}$ under ground contexts (see Figure 1): it does not reduce inside boxes. It has the diamond-property [33].

▶ **Example 1.** Let $\Delta = \lambda x.xx^!$. Then $\Delta\Delta^! \to_{\mathrm{b_g}} \Delta\Delta^! \to_{\mathrm{b_g}} \dots$ (and so $\Delta\Delta^! \to_{\mathrm{b}} \Delta\Delta^! \to_{\mathrm{b}} \dots$).

▶ **Definition 2** (Clash). *A* clash *is a term of the form $S^!T$ or $T(\lambda x.S)$.*

Let $S \in !\Lambda$: S is clash-free *if and only if it contains no clash; S is* clash-free at depth 0 *if and only if each clash occurring in S is under the scope of a* !.

For instance, $(\lambda z.x)(x^!y)^!$ is clash-free at depth 0 but not clash-free. Roughly, a clash is a "meaningless" term that cannot inherently be typed (see [24, 8]): boxes cannot be applied, abstractions cannot be the argument of an application.

The bang calculus can be extended (see [24]) with the reduction $\to_\sigma = \to_{\sigma_1} \cup \to_{\sigma_2} \cup \to_{\sigma_3}$ where $\to_{\sigma_1}$, $\to_{\sigma_2}$ and $\to_{\sigma_3}$ are the contextual closure of the following rules, respectively:

$$(\lambda x.S)TU \mapsto_{\sigma_1} (\lambda x.SU)T \qquad (\lambda y.\lambda x.S)T \mapsto_{\sigma_2} \lambda x.(\lambda y.S)T \qquad U((\lambda x.S)T) \mapsto_{\sigma_3} (\lambda x.US)T$$

with $x \notin \mathsf{fv}(U)$ in $\mapsto_{\sigma_1}$ and $\mapsto_{\sigma_3}$, while $x \notin \mathsf{fv}(T) \cup \{y\}$ in $\mapsto_{\sigma_2}$. We set $\to_{\mathrm{b}\sigma} = \to_{\mathrm{b}} \cup \to_\sigma$ and $\to_{\mathrm{b}\sigma_{\mathrm{g}}} = \to_{\mathrm{b_g}} \cup \to_{\sigma_{\mathrm{g}}}$, where $\to_{\sigma_{\mathrm{g}}} = \to_{\sigma_{1_{\mathrm{g}}}} \cup \to_{\sigma_{2_{\mathrm{g}}}} \cup \to_{\sigma_{3_{\mathrm{g}}}}$ and $\to_{\sigma_{i_{\mathrm{g}}}}$ is the closure under ground contexts of $\mapsto_{\sigma_i}$, for $i \in \{1, 2, 3\}$. Reductions $\to_\sigma$ and $\to_{\sigma_{\mathrm{g}}}$ are strongly normalizing [24] and can "unveil" hidden b-redexes and hidden clashes. For instance,

$$((\lambda x.\Delta)x)\Delta^! \to_{\sigma_{1_{\mathrm{g}}}} (\lambda x.\Delta\Delta^!)x \qquad\qquad x((\lambda y.\lambda x.z)y) \to_{\sigma_{2_{\mathrm{g}}}} x(\lambda x.(\lambda y.z)y)$$

where $((\lambda x.\Delta)x)\Delta^!$ is b-normal but $(\lambda x.\Delta\Delta^!)x$ is not ($\to_{\mathrm{b_g}}$ can fire the b-redex $\Delta\Delta^!$), and $x((\lambda y.\lambda x.z)y)$ is clash-free but $x(\lambda x.(\lambda y.z)y)$ is not (not even at depth 0).

Integers and Permutations. For $n \in \mathbb{N}$, we set $[n] = \{1, \dots, n\}$, so $[0] = \emptyset$. The set of permutations over $[n]$ is denoted by S_n. We define the category $\mathbb{P}$ of integers and permutations:

- the objects of $\mathbb{P}$ are $\mathrm{ob}(\mathbb{P}) = \{[n] \mid n \in \mathbb{N}\}$; the identity on $[n]$ is denoted by 1_n;
- the homset from $[n]$ to $[m]$ is $\mathbb{P}[[n],[m]] = \begin{cases} S_n & \text{if } n = m \\ \emptyset & \text{otherwise;} \end{cases}$
- the category $\mathbb{P}$ is symmetric strict monoidal, with tensor product given by addition: $[n] \oplus [m] = [n+m]$. Given $\sigma \in S_{k_1}$ and $\tau \in S_{k_2}$, we define $\sigma \oplus \tau \in S_{k_1+k_2}$ as

$$(\sigma \oplus \tau)(i) = \begin{cases} \sigma(i) & \text{if } 1 \leq i \leq k_1 \\ \tau(i - k_1) + k_1 & \text{otherwise.} \end{cases}$$

Given $k_1, \dots, k_n \in \mathbb{N}$ and $\sigma \in S_n$, we define $\bar{\sigma} \colon [\sum_{i \in [n]} k_i] \to [\sum_{i \in [n]} k_{\sigma(i)}]$ as $\bar{\sigma}(\sum_{r=1}^{l-1} k_r + p) = \sum_{r=1}^{l-1} k_{\sigma(r)} + p$, where $l \in [n]$ and $1 \leq p \leq k_{\sigma(l)}$.

Symmetric strict monoidal completion. For a list $\vec{a} = \langle a_1, \dots, a_k \rangle$, we set $\mathsf{len}(\vec{a}) = k$. Lists are denoted by $\vec{a}, \vec{b}, \vec{c} \dots$, *concatenation* of two lists $\vec{a}$ and $\vec{b}$ is denoted by $\vec{a} \oplus \vec{b}$.

Let A be a small category. For each object $a \in \mathrm{ob}(A)$, the identity morphism on a is denoted by 1_a. The *symmetric strict monoidal completion* $!A$ of A is the category where:

- $\mathrm{ob}(!A) = \{\langle a_1, \dots, a_n \rangle \mid a_i \in A \text{ and } n \in \mathbb{N}\}$;
- $!A[\langle a_1, \dots, a_n \rangle, \langle a'_1, \dots, a'_{n'} \rangle] = \begin{cases} \{\langle \sigma, f_1, \dots, f_n \rangle \mid f_i : a_i \to a'_{\sigma(i)}\ ,\ \sigma \in S_n\} & \text{if } n = n'; \\ \emptyset & \text{otherwise;} \end{cases}$
- for $\vec{a} = \langle a_1, \dots, a_n \rangle \in \mathrm{ob}(!A)$, the identity on $\vec{a}$ is $1_{\vec{a}} = \langle 1_n, 1_{a_1}, \dots, 1_{a_n} \rangle$;
- for $f = \langle \sigma, f_1, \dots, f_n \rangle \colon \vec{a} \to \vec{b}$ and $g = \langle \tau, g_1, \dots, g_n \rangle \colon \vec{b} \to \vec{c}$, the composition is $g \circ f = \langle \tau\sigma, g_{\sigma(1)} \circ f_1, \dots, g_{\sigma(n)} \circ f_n \rangle$;
- the monoidal structure is given by list concatenation. The tensor product is symmetric, with symmetries given by the morphisms of the shape (where $\sigma \colon [n] \to [n]$ is a permutation)

$$\langle \sigma, \vec{1} \rangle \colon \langle a_1, \dots, a_n \rangle \to \langle a_{\sigma(1)}, \dots, a_{\sigma(n)} \rangle$$

Given a permutation $\sigma \colon [n] \to [n]$ and $\vec{a}_1, \dots, \vec{a}_n \in \mathrm{ob}(!A)$ with $\mathsf{len}(\vec{a}_i) = k_i$ we define $\sigma^\star \colon \bigoplus_{i=1}^n \vec{a}_i \to \bigoplus_{i=1}^n \vec{a}_{\sigma(i)}$ as $\langle \bar{\sigma}, 1_{a_1}, \dots, 1_{a_k} \rangle$, where $k = \sum_{i \in [n]} k_i$.

We use the following shortenings: $!A^n = (!A)^n$ and $!A^{\mathrm{op}} = (!A)^{\mathrm{op}}$.

Bicategory. We assume the reader to be familiar with bicategories [3, 6] and two-dimensional monads [5]. Some basic notions are briefly recalled in Appendix A. For a diagram $F \colon C \to D$, its colimit is denoted by $\varinjlim_{c \in C} F(c)$. Given a bicategory $\mathcal{C}$, $\mathcal{C}^{\mathrm{op}}$ is the bicategory obtained by reversing the 1-cells of $\mathcal{C}$, but not the 2-cells.

3 Rel, Polr, Dist

We sketch the structure of some categories providing denotational models of linear logic. We use linear logic notations for cartesian products, comonads modelling exponentials, etc.

Rel. A simple model of linear logic is the category Rel of sets and relations. It is a prototype of *quantitative* semantics: the interpretation of a program gives information about its resource consumption during computation. Intuitively, linear logic formulas are interpreted by sets, linear logic proofs by relations, and an element in a set represents a non-idempotent intersection type. For the bang calculus, this model has been studied in [24, 33].

Objects of Rel are sets, and morphisms of Rel are binary relations. Identities are diagonal relations. Composition of morphisms in Rel is the usual composition of relations

$$g \circ f = \{\langle x, z\rangle \mid \exists\, y \in Y : \langle x, y\rangle \in f,\ \langle y, z\rangle \in g\} \text{ for } f \subseteq X \times Y \text{ and } g \subseteq Y \times Z.$$

For $X_1, X_2 \in \mathrm{ob}(\mathrm{Rel})$, the cartesian product $X_1 \,\&\, X_2$ in Rel is the disjoint union of sets $X_1 \sqcup X_2 = (\{1\} \times X_1) \cup (\{2\} \times X_2)$, where projections $\pi_i \colon X_1 \,\&\, X_2 \to X_i$ (for $i \in \{1, 2\}$) are injections $\{\langle\langle i, x\rangle, x\rangle \mid x \in X_i\}$, and the terminal (and initial) object $\top$ is the empty set $\emptyset$.

Rel is a *symmetric monoidal* category, where the tensor $X \otimes Y$ is the cartesian product of sets $X \times Y$ and its unit $\mathbf{1}$ is an arbitrary singleton set. It is *closed*, with $X \multimap Y = X \times Y$ and evaluation $\mathrm{ev}_{X,Y} \colon (X \multimap Y) \times X \to Y$ defined by $\{\langle\langle\langle x, y\rangle, x\rangle, y\rangle \mid x \in X, y \in Y\}$.

Rel comes with an *exponential comonad* $\langle !, \mathrm{der}, \mathrm{dig}\rangle$. The functor $!$ is given by $!X = \mathcal{M}_{\mathrm{f}}(X)$ (finite multisets over X) and, for a morphism $f \in \mathrm{Rel}[X, Y]$, $!f = \{\langle[x_1, \ldots, x_n], [y_1, \ldots, y_n]\rangle \mid n \in \mathbb{N},\ \langle x_1, y_1\rangle, \ldots, \langle x_n, y_n\rangle \in f\}$. Dereliction $\mathrm{der}_X \in \mathrm{Rel}[!X, X]$ is $\{\langle[x], x\rangle \mid x \in X\}$, and digging $\mathrm{dig}_X \in \mathrm{Rel}[!X, !!X]$ is $\{\langle m_1 + \cdots + m_k, [m_1, \ldots, m_k]\rangle \mid m_1, \ldots, m_k \in !X\}$ (for two finite multisets $\bar{a} = [a_1, \ldots, a_k]$ and $\bar{b} = [b_1, \ldots, b_n]$, we set $\bar{a} + \bar{b} = [a_1, \ldots, a_k, b_1, \ldots, b_n]$).

Polr. To work within a more informative setting, providing not only *quantitative*, but also *qualitative* information, consider the category Polr of preordered sets and monotonic relations [21, 23]. Intuitively, given two types a and b, if $a \leq b$ then a is an approximant of b. All the constructions in Polr are a refinement and generalization of the ones for Rel.

In Polr, objects are preordered sets; a morphism f from $\mathcal{X} = \langle|\mathcal{X}|, \leq_{\mathcal{X}}\rangle$ to $\mathcal{Y} = \langle|\mathcal{Y}|, \leq_{\mathcal{Y}}\rangle$ is a *monotonic* relation[4] from $|\mathcal{X}|$ to $|\mathcal{Y}|$, *i.e.*, if $\langle x, y\rangle \in f$ with $x' \leq_{\mathcal{X}} x$ and $y \leq_{\mathcal{Y}} y'$ then $\langle x', y'\rangle \in f$. The identity at $\mathcal{X}$ is $\{\langle x, x'\rangle \mid x \leq_{\mathcal{X}} x'\}$. Composition preserves monotonicity.

In Polr the cartesian product $\mathcal{X}_1 \,\&\, \mathcal{X}_2$ is the disjoint union of sets $|\mathcal{X}_1| \sqcup |\mathcal{X}_2|$ with the preorder $\leq_{\mathcal{X}_1} \sqcup \leq_{\mathcal{X}_2}$ defined as $\langle i, x\rangle \leq_{\mathcal{X}_1 \& \mathcal{X}_2} \langle j, y\rangle$ if $i = j$ and $x \leq_{\mathcal{X}_i} y$. The terminal object $\top$ is $\emptyset$ with the empty order. Projections $\pi_i \colon \mathcal{X}_1 \,\&\, \mathcal{X}_2 \to \mathcal{X}_i$ are $\pi_i = \{\langle\langle i, x\rangle, x'\rangle \mid x \leq_{\mathcal{X}_i} x'\}$.

Polr has a symmetric monoidal structure. The tensor $\mathcal{X}_1 \otimes \mathcal{X}_2$ is the cartesian product of sets with the product order. The endofunctor $\mathcal{X} \otimes \_$ admits a right adjoint $\_ \multimap \mathcal{Y}$ defined as follows: $|\mathcal{X} \multimap \mathcal{Y}| = |\mathcal{X}| \times |\mathcal{Y}|$ and $\langle x, y\rangle \leq_{\mathcal{X} \multimap \mathcal{Y}} \langle x', y'\rangle$ if $x' \leq_{\mathcal{X}} x$ and $y \leq_{\mathcal{Y}} y'$. The evaluation morphism $\mathrm{ev}_{\mathcal{X}_1, \mathcal{X}_2} \colon (\mathcal{X}_1 \multimap \mathcal{X}_2) \,\&\, \mathcal{X}_1 \to \mathcal{X}_2$ is $\{\langle\langle\langle x, y\rangle, x'\rangle, y'\rangle \mid x' \leq x,\ y \leq y'\}$.

Polr has exponential comonad $\langle !, \mathrm{der}, \mathrm{dig}\rangle$.[5] The endofunctor $! \colon \mathrm{Polr} \to \mathrm{Polr}$ is given by $!\mathcal{X} = \langle \mathcal{M}_{\mathrm{f}}(|\mathcal{X}|), \leq_{\mathcal{X}}\rangle$ with $[x_1, \ldots, x_n] \leq_{!\mathcal{X}} [x'_1, \ldots, x'_{n'}]$ if $n = n'$ and there is $\sigma \in S_n$ such that $x_i \leq x'_{\sigma(i)}$ for all $1 \leq i \leq n$; for $f \in \mathrm{Polr}[\mathcal{X}, \mathcal{Y}]$, we set $!f = \{\langle[x_1, \ldots, x_n], [y_1, \ldots, y_k]\rangle \mid \langle x_i, y_i\rangle \in f,\ k \in \mathbb{N}\}$. Dereliction $\mathrm{der}_{\mathcal{X}} \colon !\mathcal{X} \to \mathcal{X}$ is $\{\langle[x], x'\rangle \mid x \leq_{\mathcal{X}} x'\}$, and digging $\mathrm{dig}_{\mathcal{X}} \colon !\mathcal{X} \to !!\mathcal{X}$ is $\{\langle m, [m_1, \ldots, m_k]\rangle \mid m \leq_{!\mathcal{X}} m_1 + \cdots + m_k\}$.

Rel is the full subcategory of Polr where objects are sets equipped with the discrete order.

Polr as a model of the bang calculus. A categorical model of the bang calculus [23, 24] consists of a $\star$-autonomous category $(A, \otimes, I, \multimap, (-)^{\perp})$, cartesian with product $\&$ and terminal object $\top$ (and, by $\star$-autonomy, cocartesian with coproduct $\oplus$ and initial object 0), endowed with a comonad $(!, \mathrm{der}, \mathrm{dig})$ with suitable Seely isomorphisms [23, 33]. Also, we

[4] In [21, 23], monotonicity is slightly different, so that the type system generated by the model is covariant on the left of $\vdash$ and contravariant on the right of $\vdash$. With our definition, the type system generated by the model is contravariant on the left of $\vdash$ and covariant on the right of $\vdash$, in accordance with [1].

[5] Akin to [21] and unlike [23], our exponential comonad is based on finite multiset construction. But our preorder on $!\mathcal{X}$ is different from [21]: there $[a] \leq_{!\mathcal{X}} [a, a]$ (idempotency is a sort of approximation), here $[a]$ and $[a, a]$ are incomparable, so that approximation is completely independent from idempotence.

Types:

$$a := x \in \mathcal{X} \mid [a_1, \ldots, a_k] \multimap a \mid [a_1, \ldots, a_k]$$

Preorder $\leq_U$ in U:

$$\frac{x \leq_{\mathcal{X}} x'}{x \leq_U x'} \qquad \frac{m' \leq_U m \qquad a \leq_U a'}{(m \multimap a) \leq_U (m' \multimap a')}$$

$$\frac{\sigma \in S_k \quad a_1 \leq_U a'_{\sigma(1)} \quad \overset{k \in \mathbb{N}}{\cdots} \quad a_k \leq_U a'_{\sigma(k)}}{[a_1, \ldots, a_k] \leq_U [a'_1, \ldots, a'_k]}$$

Derivation rules:

$$\frac{a' \leq_U a}{x_1 : [\,], \ldots, x_i : [a'], \ldots x_n : [\,] \vdash x_i : a}$$

$$\frac{\Gamma \vdash S : m \multimap a \qquad \Gamma' \vdash T : m \qquad \Delta \leq_{U^n} \Gamma \otimes \Gamma'}{\Delta \vdash ST : a}$$

$$\frac{\Gamma_1 \vdash S : a_1 \quad \overset{k \in \mathbb{N}}{\cdots} \quad \Gamma_k \vdash S : a_k \quad \Delta \leq_{U^n} \bigotimes_{i=1}^k \Gamma_i}{\bigotimes_{i=1}^k \Gamma_i \vdash S^! : [a_1, \ldots, a_k]}$$

$$\frac{\Delta, x : m \vdash S : a}{\Delta \vdash \lambda x.S : m \multimap a}$$

Figure 2 Non-idempotent intersection type system $\mathcal{R}_{\leq}$ associated with the preorder U in Polr.

require that $0 \cong \top$. An *extensional model* of the bang calculus is then an object $U \in \mathrm{ob}(A)$ such that $U \cong {!U} \,\&\, ({!U} \multimap U)$. To have a *non-extensional model* for the bang calculus a retraction ${!U} \,\&\, ({!U} \multimap U) \lhd U$ is enough.

We build a retraction in the category Polr. We define a family of preoders as follows:

$$U_0 = \mathcal{X} \text{ (any preorder)} \qquad U_{n+1} = {!U_n} \sqcup (({!U_n} \multimap U_n) \sqcup \mathcal{X}) \tag{1}$$

We define a family of canonical inclusions $(\iota_n : U_n \hookrightarrow U_{n+1})_{n \in \mathbb{N}}$ as $\iota_0 = \iota_{\mathcal{X}}$ (the inclusion $\mathcal{X} \hookrightarrow {!\mathcal{X}} \sqcup (({!\mathcal{X}} \multimap \mathcal{X}) \sqcup \mathcal{X})$) and $\iota_{n+1} = {!\iota_n} \sqcup (({!\iota_n} \multimap \iota_n) \sqcup 1_{\mathcal{X}})$, so the preorder U_n is just the restriction to the elements of U_n of the preorder U_{n+1}. We set $U = \varinjlim_{n \in \mathbb{N}} U_n$, that is a directed colimit of the directed diagram $\langle \iota_i \rangle_{i \in \mathbb{N}}$. It is easy to check that there exists a canonical inclusion $\iota : {!U} \sqcup ({!U} \multimap U) \hookrightarrow U$ and that we have a retraction ${!U} \,\&\, ({!U} \multimap U) \lhd U$.

We can define the interpretation of the terms of the bang calculus in Polr. Let $S \in {!\Lambda}$ and $\mathsf{fv}(S) \subseteq \vec{x} = \langle x_1, \ldots, x_n \rangle$ with the x_i's pairwise distinct. The *semantics* (or *denotation*) of S is a monotonic relation $[\![S]\!]_{\vec{x}} : {!U^{\otimes n}} \to U$ defined by induction as follows:

- $[\![x_i]\!]_{\vec{x}} = \{\langle\langle [\,], \ldots, [a'], \ldots, [\,] \rangle, a\rangle \mid a' \leq a\}$ ($[a']$ is in the i^{th} position in $\langle [\,], \ldots, [a'], \ldots, [\,] \rangle$);
- $[\![\lambda y.T]\!]_{\vec{x}} = \{\langle \Delta, \iota(\langle m, a \rangle) \rangle \mid \langle \Delta \oplus \langle m \rangle, a \rangle \in [\![T]\!]_{\vec{x} \oplus \langle y \rangle}\}$, where $y \notin \vec{x}$;
- $[\![ST]\!]_{\vec{x}} = \bigcup_{m \in !U} \bigcup_{\Gamma, \Gamma' \in U^n} \{\langle \Delta, a \rangle \mid \langle \Gamma, \iota(\langle m, a \rangle) \rangle \in [\![S]\!]_{\vec{x}}\,,\ \langle \Gamma', \iota(m) \rangle \in [\![T]\!]_{\vec{x}} \text{ and } \Delta \leq_{U^n} \Gamma \otimes \Gamma'\}$;
- $[\![T^!]\!]_{\vec{x}} = \bigcup_{k \in \mathbb{N}} \bigcup_{\Gamma_1, \ldots, \Gamma_k \in U^n} \{\langle \Delta, [a_1, \ldots, a_k] \rangle \mid \langle \Gamma_i, a_i \rangle \in [\![T]\!]_{\vec{x}} \text{ and } \Delta \leq_{U^n} \bigotimes_{i=1}^k \Gamma_i\}$ where if $\Gamma = \langle m_1, \ldots, m_n \rangle$ and $\Gamma' = \langle m'_1, \ldots, m'_n \rangle$ then $\Gamma \otimes \Gamma' = \langle m_1 + m'_1, \ldots, m_n + m'_n \rangle$.

Ehrhard [23] showed this is a denotational semantics. By setting $m \multimap a = \langle m, a \rangle \in {!U} \times U$, we can give a type-theoretic description of the preorder U as in Figure 2. Such a type system $\mathcal{R}_{\leq}$ is similar to de Carvalho's non-idempotent intersection type system $\mathcal{R}$ [16, 17]. The main difference is that in $\mathcal{R}_{\leq}$ types are elements of a *preorder* U (an object of Polr), while in $\mathcal{R}$ types are elements of a *set* U (an object of Rel). The additional information provided by the preorder accounts for *approximation*: if $a \leq_U b$ then the type a approximates the type b. This is evident in the rule for the variable in Figure 2: a' can be seen as a *subtype* of a.

By easy inspection of the definition, $\langle \Delta, a \rangle \in [\![S]\!]_{\vec{x}}$ if and only if $\Delta \vdash S : a$. In other words, the semantics of a term S is the set of conclusions of the type derivations for S. The semantics is then a *semantics of types* in the non-idempotent intersection type system $\mathcal{R}_{\leq}$.

We now try to shift our standpoint. In system $\mathcal{R}_{\leq}$, let us try to define a *semantics of proofs*. Given a term S, a context Δ and type a, we set $[\![S]\!]_{\vec{x}}(\Delta, a) = \left\{ \begin{array}{c} \pi \\ \vdots \\ \Delta \vdash S : a \end{array} \,\middle|\, \pi \in \mathcal{R}_{\leq} \right\}$.

It is easy to see that this proof-relevant structure is not a denotational semantics (not even up to isomorphism). Indeed, reduction over type derivations in system $\mathcal{R}_{\leq}$ is non-deterministic, since it deals with multisets. Take $(\lambda z.(yz^!)z^!)S^! \to_{\mathrm{b_g}} (yS^!)S^!$ and the following type derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{y\colon [[a]\multimap [a]\multimap c]\vdash y\colon [a]\multimap [a]\multimap c \quad \dfrac{\overline{z\colon [a]\vdash z\colon a}\,z_1}{z\colon [a]\vdash z^!\colon [a]}}{y\colon [[a]\multimap [a]\multimap c], z\colon [a]\vdash yz^!\colon [a]\multimap c} \quad \dfrac{\overline{z\colon [a]\vdash z\colon a}\,z_2}{z\colon [a]\vdash z^!\colon [a]}}{y\colon [[a]\multimap [a]\multimap c], z\colon [a,a]\vdash (yz^!)z^!\colon c}}{y\colon [[a]\multimap [a]\multimap c]\vdash \lambda z.(yz^!)z^!\colon [a,a]\multimap c} \quad \dfrac{\begin{matrix}\pi_1\\ \vdots\\ \Gamma_1\vdash S\colon a\end{matrix} \quad \begin{matrix}\pi_2\\ \vdots\\ \Gamma_2\vdash S\colon a\end{matrix}}{\Gamma_1\otimes\Gamma_2\vdash S^!\colon [a,a]}}{\Gamma_1\otimes\Gamma_2, y\colon [[a]\multimap [a]\multimap c]\vdash (\lambda z.(yz^!)z^!)S^!\colon c}$$

Suppose y is not free in S and $\pi_1 \neq \pi_2$ (*e.g.* take $S = wx^!$). Then if we consider the reduct $(yS^!)S^!$ we have two possible choices for the typing, $\pi\{\pi_1/z_1, \pi_2/z_2\}$ or $\pi\{\pi_2/z_1, \pi_1/z_2\}$. This non-determinism stems from the multiset structure, but we shall see that simply passing to a list-oriented framework does not solve the problem. A natural way to make this kind of structure a denotational semantics is the lifting to Set enriched distributors.

From Rel and Polr to Dist. We recall a basic but pivotal fact: a relation $f \subseteq X \times Y$ can be identified with its characteristic function $\chi_f\colon X \times Y \to \mathbf{2}$ where $\mathbf{2} = \{0,1\}$ is the two-element boolean algebra with sum (join) and product (meet). Composition is then defined as

$$\chi_{g\circ f}(x,z) = \sum_{y\in Y} \chi_g(y,z)\cdot\chi_f(x,y) \qquad \text{where } \chi_f\colon X\times Y\to\mathbf{2} \text{ and } \chi_g\colon Y\times Z\to\mathbf{2}\,. \quad (2)$$

All the constructions in Rel and Polr can be reformulated in this characteristic function perspective. For instance, in Rel, the identity at X becomes the characteristic function of X.

In Polr, a monotonic relation f from $\mathcal{X} = \langle|\mathcal{X}|, \leq_{\mathcal{X}}\rangle$ to $\mathcal{Y} = \langle|\mathcal{Y}|, \leq_{\mathcal{Y}}\rangle$ can be seen as a monotonic characteristic function $\chi_f\colon \mathcal{X}^{\mathrm{op}} \times \mathcal{Y} \to \mathbf{2}$, where $\mathcal{X}^{\mathrm{op}} = \langle|\mathcal{X}|, \geq_{\mathcal{X}}\rangle$ and $\mathbf{2}$ is endowed with the boolean order. Any preorder $\mathcal{X} = \langle|\mathcal{X}|, \leq_{\mathcal{X}}\rangle$ forms a category where $\mathrm{ob}(\mathcal{X}) = |\mathcal{X}|$ and $\mathcal{X}[x,x']$ is a singleton (if $x \leq_{\mathcal{X}} x'$) or the empty set (otherwise), so $\mathcal{X}^{\mathrm{op}}$ is the opposite category of $\mathcal{X}$. Thus, $\chi_f\colon \mathcal{X}^{\mathrm{op}} \times \mathcal{Y} \to \mathbf{2}$ is a bifunctor, contravariant in $\mathcal{X}$ and covariant in $\mathcal{Y}$. The semantics of a term S is then a Polr morphism $[\![S]\!]_{\vec{x}} : (!U^{\otimes n})^{\mathrm{op}} \times U \to \mathbf{2}$.

It is then natural to generalize the characteristic function viewpoint to generic categories, which gives rise to the notion of *distributor* (also known as *profunctors*).

Dist. For two small categories A, B, a *distributor* $F\colon A \nrightarrow B$ is a functor $F : A^{\mathrm{op}} \times B \to \mathrm{Set}$. Composition of distributors relies on the notion of *coend*, a kind of colimit (a coequalizer).

▶ **Definition 3** (Coend, [40])**.** *Let $F\colon C^{\mathrm{op}} \times C \to D$ be a functor. A* cowedge *for F is an object $T \in D$ together with a family of morphisms $w_c\colon F(c,c) \to T$ such that diagram* (3) *below commutes, for $f\colon c \to c'$. A* coend *for F, denoted by $\int^{c\in C} F(c,c)$, is a universal cowedge.*

$$\begin{tikzcd} F(c',c) \arrow[r, "F(f,1)"] \arrow[d, "F(1,f)"'] & F(c,c) \arrow[d, "w_c"] \\ F(c',c') \arrow[r, "w_{c'}"] & T \end{tikzcd} \quad (3)$$

We now define the bicategory Dist of *distributors*. For a proper presentation of the structure of this bicategory we refer to [10, 12, 27, 30].

- *0-cells* are small categories $A, B, C \ldots$; *1-cells* $F\colon A \nrightarrow B$ are distributors, *i.e.* functors $F\colon A^{\mathrm{op}} \times B \to \mathrm{Set}$; *2-cells* $\alpha\colon F \Rightarrow G$ are natural transformations.
- Given any 0-cells A and B, 1-cells and 2-cells are organized as a category $\mathrm{Dist}(A, B)$. Composition $\alpha \star \beta$ in $\mathrm{Dist}(A, B)$ is called *vertical composition.* We define the *zero distributor* $\emptyset_{A,B} \in \mathrm{ob}(\mathrm{Dist}(A, B))$ as $\emptyset_{A,B}(a, b) = \emptyset$ for all $a \in \mathrm{ob}(A)$ and $b \in \mathrm{ob}(B)$.
- For $A \in \mathrm{Dist}$, the identity $1_A\colon A \nrightarrow A$ is Yoneda's embedding $1_A(a', a) = A[a', a]$.
- For 1-cells $F\colon A \nrightarrow B$ and $G\colon B \nrightarrow C$, their composition is given by

$$(G \circ F)(a, c) = \int^{b \in B} G(b, c) \times F(a, b)$$

Note the analogy with (2). Composition is only associative up to canonical isomorphisms. For this reason Dist is a bicategory [6].
- The cartesian product $A \,\&\, B$ is the disjoint union $A \sqcup B$ of categories. The terminal object $\top$ is given by the empty category. The bicategory Dist admits also coproducts, with $A \oplus B = A \sqcup B$ (the canonical inclusions are denoted by ι_A and ι_B) and $0 = \top$.
- There is a symmetric monoidal structure on Dist given by the cartesian product of categories: $A \otimes B = A \times B$, with any one-object category as a unit. The bicategory of distributors is monoidal closed, with linear exponential object $A \multimap B = A^{\mathrm{op}} \times B$.

The symmetric strict monoidal completion of a small category A (Section 2) lifts to an endofunctor in Cat, by setting $!F(\langle a_1, \ldots, a_n\rangle) = \langle F(a_1), \ldots, F(a_n)\rangle$ for any functor $F\colon A \to B$. The endofunctor ! can be extended to Dist, determining a pseudocomonad $(!, \mathrm{dig}_A, \mathrm{der}_A)$ on Dist [27, 30]. The two components of the pseudocomonad are defined as follows: $\mathrm{dig}_A(\vec{a}, \langle \vec{a_1}, \ldots, \vec{a_n}\rangle) = !A[\vec{a}, \bigoplus_{i=1}^{n} \vec{a_i}]$ and $\mathrm{der}_A(\vec{a}, a) = !A[\vec{a}, \langle a\rangle]$. The Kleisli bicategory $Kl(!)(\mathrm{Dist})$ is the bicategory of *categorical symmetric sequences* [30], biequivalent to the bicategory of *generalized species of structure* [27, 28]. There are Seely equivalences $!(A \,\&\, B) \simeq\, !A \times !B$ and $!\top \simeq 1$, pseudonatural in both A and B [27].

4 A Type-Theoretic Non-Extensional Model for the Bang Calculus

Distributors-Induced Model for the Bang Calculus. The bicategory of distributors fulfills a bicategorical generalization of the categorical model of the bang calculus shown in Section 3.[6] However, we leave the proper development of a general notion of bicategorical model for the bang calculus to future work, since the notion of symmetric monoidal bicategory is highly non-trivial. For our purpose, it is enough to present a denotational model inside a particular bicategory, *i.e.*, the bicategory of distributors. A denotational model in this setting will be an interpretation of bang terms as suitable 1-cells, such that $[\![S]\!]_{\vec{x}} \cong [\![T]\!]_{\vec{x}}$ if $S \to_\ell T$. In particular, we want $[\![S]\!]_{\vec{x}}\colon (!U^{\otimes n})^{\mathrm{op}} \times U \to \mathrm{Set}$ (for $\mathsf{len}(\vec{x}) = n$), with $!U \,\&\, (!U \multimap U) \lhd U$. The intuition is that, in Dist, 0-cells represent types (and in our untyped setting, they satisfy a retraction), 1-cells represent type derivations and 2-cells represent reduction on derivations.

We build the retraction in Dist, in analogy with the construction (1) in Polr. Indeed, they are both special cases of the free-algebra construction for an (unpointed) endofunctor [36]. We recall that, in Dist, $A \,\&\, B = A \sqcup B$, $A \otimes B = A \times B$ (so $A^{\otimes n} = A^n$) and $A \multimap B = A^{\mathrm{op}} \times B$.

[6] The only delicate point is the $\star$-autonomy of the bicategory, since it does not exist in the literature a notion of $\star$-autonomous bicategory. However it is possible to equip distributors with a dualizing pseudo-endofunctor, as shown for example in [12, 27].

Types:

$$a := x \in A \mid \langle a_1, \dots, a_k \rangle \Rightarrow a \mid \langle a_1, \dots, a_k \rangle$$

Morphisms in U:

$$\frac{f \in A[x, x']}{f \in U[x, x']} \qquad \frac{\langle \sigma, \vec{f} \rangle \colon \vec{a}' \to \vec{a} \qquad f \colon a \to a'}{\langle \sigma, \vec{f} \rangle \Rightarrow f \colon (\vec{a} \Rightarrow a) \to (\vec{a}' \Rightarrow a')}$$

$$\frac{\sigma \in S_n \quad f_1 \colon a_1 \to a'_{\sigma(1)} \quad \cdots \quad f_n \colon a_n \to a'_{\sigma(n)}}{\langle \sigma, f_1, \dots, f_n \rangle \colon \langle a_1, \dots, a_n \rangle \to \langle a'_1, \dots, a'_n \rangle}$$

Derivation rules:

$$\frac{f \colon a' \to a}{x_1 : \langle\rangle, \dots, x_i : \langle a' \rangle, \dots x_n : \langle\rangle \vdash x_i : a}$$

$$\frac{\Gamma \vdash S : \vec{a} \Rightarrow a \qquad \Gamma' \vdash T : \vec{a} \qquad \eta \colon \Delta \to \Gamma \otimes \Gamma'}{\Delta \vdash ST : a}$$

$$\frac{\Gamma_1 \vdash S : a_1 \quad \overset{k \in \mathbb{N}}{\cdots} \quad \Gamma_k \vdash S : a_k \qquad \eta \colon \Delta \to \bigotimes_{i=1}^{k} \Gamma_i}{\Delta \vdash S^! : \langle a_1, \dots, a_k \rangle}$$

$$\frac{\Delta, x : \vec{a} \vdash S : a}{\Delta \vdash \lambda x.S : \vec{a} \Rightarrow a}$$

Figure 3 Non-idempotent intersection type system $\mathcal{R}_{\to}$ associated with the 0-cell U in Dist.

▶ **Definition 4.** *Let A be a small category. We define a family of small categories $(U_n)_{n \in \mathbb{N}}$ by:*

$$U_0 = A \qquad U_{n+1} = !U_n \sqcup ((!U_n^{\mathrm{op}} \times U_n) \sqcup A)$$

We define a family of inclusions $(\iota_n : U_n \hookrightarrow U_{n+1})_{n \in \mathbb{N}}$ in the canonical way:

$$\iota_0 = \iota_A \qquad \iota_{n+1} = !(\iota_n) \sqcup ((!(\iota_n)^{\mathrm{op}} \times \iota_n) \sqcup 1_A)$$

Then we set $U_A = \varinjlim_{n \in \mathbb{N}} U_n$. From now on, the 0-cell U_A will be simply denoted by U, keeping the parameter A implicit. We denote by $\xi_n \colon !U_n \sqcup (!U_n^{\mathrm{op}} \times U_n) \hookrightarrow U_n$ the canonical inclusions.

▶ **Lemma 5** (Inclusion)**.** *There exists a canonical inclusion $\iota : !U \sqcup (!U^{\mathrm{op}} \times U) \hookrightarrow U$.*

Proof. Since U is a filtered colimit, we have $!U \sqcup (!U^{\mathrm{op}} \times U) \cong \varinjlim_{n \in \mathbb{N}} !U_n \sqcup (\varinjlim_{n \in \mathbb{N}} !U_n^{\mathrm{op}} \times \varinjlim_{n \in \mathbb{N}} U_n)$, and so we can explicitly define the inclusion functor as $\iota(a) = y_{j+1}(\xi_j(a))$ where $j = \min\{n \in \mathbb{N} \mid a \in U_n \sqcup (!U_n^{\mathrm{op}} \times U_n)\}$ and $y_{j+1} \colon U_{j+1} \to U$ is the canonical injection of U_{j+1}. ◀

▶ **Theorem 6** (Retraction)**.** *We have that $!U \,\&\, (!U \multimap U) \lhd U$ in* Dist.

So, the 0-cell U is a (non-extensional) denotational model of the bang calculus. By seeing the objects of A (resp. U) as the *atomic types* (resp. *types*) and setting $\vec{a} \Rightarrow a = \langle \vec{a}, a \rangle \in !U \times U$, we give in Figure 3 a type-theoretic description of the 0-cell U. This *non-idempotent intersection type system*, called $\mathcal{R}_{\to}$, is the generalization in Dist of the system $\mathcal{R}_{\leq}$ in Figure 2 associated with Polr. A morphism $f \colon a \to b$ in Figure 3 can be seen as a witness in Dist of the subtyping relation between a and b, generalizing $a \leq_U b$ of Polr.

Semantics of Bang Terms. We now present the *semantics* (or denotation) of bang terms as distributors in the bicategory Dist. We recall that $\iota \colon !U \,\&\, (!U^{\mathrm{op}} \times U) \hookrightarrow U$. Let $\Gamma = \langle \vec{b}_1, \dots, \vec{b}_n \rangle$, $\Delta = \langle \vec{b'_1}, \dots, \vec{b'_n} \rangle \in !U^n$. A morphism $\eta \colon \Gamma \to \Delta$ is a list of morphisms $\eta = \langle \langle \sigma_1, \vec{f_1} \rangle, \dots, \langle \sigma_n, \vec{f_n} \rangle \rangle \colon \Gamma \to \Delta$ where $\langle \sigma_i, \vec{f_i} \rangle \colon \vec{b}_i \to \vec{b}'_i$. We set $\Gamma \otimes \Delta = \langle \vec{b}_1 \oplus \vec{b'}_1, \dots, \vec{b}_n \oplus \vec{b'}_n \rangle$. This tensor product inherits the relevant structure from $\oplus$. In particular, the symmetries $\vec{\sigma} \colon \bigotimes_{i=1}^{k} \Gamma_i \to \bigotimes_{i=1}^{k} \Gamma_{\sigma(i)}$ are built from the $\sigma^{\star}$ construction presented in Section 2.

▶ **Definition 7** (Semantics)**.** *Let $S \in !\Lambda$ and $\mathsf{fv}(S) \subseteq \vec{x} = \langle x_1, \dots, x_n \rangle$, with the x_i's pairwise distinct. The* semantics *$[\![S]\!]_{\vec{x}} \colon !U^{\otimes n} \nrightarrow U$ of S with respect to $\vec{x}$ is defined by induction on S:*

- $[\![x_i]\!]_{\vec{x}}(\Delta, a) = !U^n[\Delta, \langle \langle\rangle, \dots, \langle a \rangle, \dots \langle\rangle \rangle]$ *($\langle a \rangle$ is in the i^{th} position in $\langle \langle\rangle, \dots, \langle a \rangle, \dots, \langle\rangle \rangle$);*

$$[g\colon a \to b]\left(\dfrac{f\colon a' \to a}{x_1 : \langle\rangle, \dots, x_i : \langle a' \rangle, \dots, x_n : \langle\rangle \vdash x_i : a}\right) = \dfrac{g \circ f\colon a' \to b}{x_1 : \langle\rangle, \dots, x_i : \langle a' \rangle, \dots, x_n : \langle\rangle \vdash x_i : b}$$

$$[\langle \sigma, \vec{g} \rangle \Rightarrow g\colon (\vec{a} \Rightarrow a) \to (\vec{b} \Rightarrow b)]\left(\dfrac{\begin{matrix}\vdots\, \pi \\ \Delta, x : \vec{a} \vdash S : a\end{matrix}}{\Delta \vdash \lambda x.S : \vec{a} \Rightarrow a}\right) = \dfrac{\begin{matrix}\vdots\, [g]\pi\{\langle 1, \langle \sigma, \vec{g} \rangle \rangle\} \\ \Delta, x : \vec{b} \vdash S : b\end{matrix}}{\Delta \vdash \lambda x.S : \vec{b} \Rightarrow b}$$

$$[g\colon a \to b]\left(\dfrac{\begin{matrix}\vdots\, \pi_1 \\ \Gamma_1 \vdash S\colon \vec{a} \Rightarrow a\end{matrix} \quad \begin{matrix}\vdots\, \pi_2 \\ \Gamma_2 \vdash T : \vec{a}\end{matrix} \quad \eta\colon \Delta \to \Gamma_1 \otimes \Gamma_2}{\Delta \vdash ST\colon a}\right) = \dfrac{\begin{matrix}\vdots\, [1 \Rightarrow g]\pi_1 \\ \Gamma_1 \vdash S : \vec{a} \Rightarrow b\end{matrix} \quad \begin{matrix}\vdots\, \pi_2 \\ \Gamma_2 \vdash T : \vec{a}\end{matrix} \quad \eta\colon \Delta \to \Gamma_1 \otimes \Gamma_2}{\Delta \vdash ST : b}$$

$$[(\sigma, \vec{g})\colon \vec{a} \to \vec{b}]\left(\dfrac{\left(\begin{matrix}\vdots\, \pi_i \\ \Gamma_i \vdash S : a_i\end{matrix}\right)_{i=1}^{k} \quad \eta : \Delta \to \bigotimes_{i=1}^{k} \Gamma_i}{\Delta \vdash S^! : \vec{a} = \langle a_1, \dots, a_k \rangle}\right) = \dfrac{\left(\begin{matrix}\vdots\, \pi'_i \\ \Gamma_{\sigma^{-1}(i)} \vdash S : a_{\sigma^{-1}(i)}\end{matrix}\right)_{i=1}^{k} \quad \vec{\sigma}^{-1} \circ \eta : \Delta \to \bigotimes_{i=1}^{k} \Gamma_{\sigma^{-1}(i)}}{\Delta \vdash S^! : \vec{b} = \langle b_1, \dots, b_k \rangle}$$

Figure 4 Left action on derivations. In the last identity, on the right, $\pi'_i = [g_{\sigma^{-1}(i)}]\pi_{\sigma^{-1}(i)}$.

- $[\![\lambda y.S]\!]_{\vec{x}}(\Delta, a) = \begin{cases} [\![S]\!]_{\vec{x} \oplus \langle y \rangle}(\Delta \oplus \langle \vec{a} \rangle, a') & \textit{if } a = \iota(\langle \vec{a}, a' \rangle) \\ \emptyset & \textit{otherwise.} \end{cases}$, *where* $y \notin \vec{x}$;
- $[\![ST]\!]_{\vec{x}}(\Delta, a) = \displaystyle\int^{\vec{a} \in !U} \int^{\Gamma_1, \Gamma_2 \in !U^n} [\![S]\!]_{\vec{x}}(\Gamma_1, \iota(\langle \vec{a}, a \rangle)) \times [\![T]\!]_{\vec{x}}(\Gamma_2, \iota(\vec{a})) \times (!U^n)(\Delta, \Gamma_1 \otimes \Gamma_2)$;
- $[\![S^!]\!]_{\vec{x}}(\Delta, a) = \begin{cases} \displaystyle\int^{\Gamma_1, \dots, \Gamma_k \in !U^n} \prod_{i=1}^{k} [\![S]\!]_{\vec{x}}(\Gamma_i, a_i) \times (!U^n)(\Delta, \bigotimes_{i=1}^{k} \Gamma_i) & \textit{if } a = \iota(\langle a_1, \dots, a_k \rangle) \\ \emptyset & \textit{otherwise.} \end{cases}$

Given $\langle \Delta, a \rangle \in !U^n \times U$ we call *points* the elements of $[\![S]\!]_{\vec{x}}(\Delta, a)$. From now on, when we write $[\![S]\!]_{\vec{x}}$ we always assume that $\mathsf{fv}(S) \subseteq \vec{x} = \langle x_1, \dots, x_n \rangle$ and the x_i's are pairwise distinct.

The semantics of a term S is a functor $[\![S]\!]_{\vec{x}}\colon (!U^n)^{\mathrm{op}} \times U \to \mathrm{Set}$. As such, it must be defined on the objects of the category $(!U^n)^{\mathrm{op}} \times U$ (as done in Definition 7) *and* on the morphisms of the category $(!U^n)^{\mathrm{op}} \times U$. The action on morphisms (omitted in Definition 7) is given by induction on S and, in the application and bang cases, also by the universal property of the coend construction. The variable case is just the hom-functor. An explicit definition of the application and bang cases can be given by considering coends as coequalizers [44].

Non-idempotent Intersection Type Distributors. We aim to define the *non-idempotent intersection type distributor* $\mathsf{T}_U(S)_{\vec{x}}$ for any term S. Let π be a type derivation in system $\mathcal{R}_{\to}$, as defined in Figure 3. The *left* and *right actions* of morphisms on π are defined in Figures 4 and 5, respectively (by induction on π). Given $f : a \to a'$ and $\theta\colon \Delta' \to \Delta$, the left and right actions may change the conclusion of a type derivation:

$$\text{left:}\quad [f]\left(\begin{matrix}\pi \\ \vdots \\ \Delta \vdash S : a\end{matrix}\right) \rightsquigarrow \begin{matrix}[f]\pi \\ \vdots \\ \Delta \vdash S : a'\end{matrix} \qquad\qquad \text{right:}\quad \left(\begin{matrix}\pi \\ \vdots \\ \Delta \vdash S : a\end{matrix}\right)\{\theta\} \rightsquigarrow \begin{matrix}\pi\{\theta\} \\ \vdots \\ \Delta' \vdash S : a\end{matrix}$$

Notice the contravariance of the right action, and that $[f](\pi\{\theta\}) = ([f]\pi)\{\theta\}$.

We define $\sim$ as the smallest congruence on type derivations generated by the rules in Figure 6. We denote by $\tilde{\pi}$ the equivalence class of π modulo $\sim$. Note that $[f]\tilde{\pi}\{\theta\} = \widetilde{[f]\pi\{\theta\}}$.

▶ **Example 8.** We give a couple of examples of the equivalence $\sim$ between type derivations in system $\mathcal{R}_{\to}$. The intuition is that $\sim$ equalizes type derivations for the same term and with the same conclusion, where the "same" permutations are performed at different moments.

$$\frac{f : a' \to a}{x_1 : \langle\rangle, \ldots, x_i : \langle a'\rangle, \ldots, x_n : \langle\rangle \vdash x_i : a}\{\langle g : b \to a'\rangle\} \quad = \quad \frac{f \circ g : b \to a}{x_1 : \langle\rangle, \ldots, x_i : \langle b\rangle, \ldots, x_n : \langle\rangle \vdash x_i : a}$$

$$\left(\frac{\begin{array}{c}\pi\\ \vdots\\ \Delta, x : \vec{a} \vdash S : a\end{array}}{\Delta \vdash \lambda x.S : \vec{a} \Rightarrow a}\right)\{\theta\} \quad = \quad \frac{\begin{array}{c}\pi\{\theta \oplus \langle 1\rangle\}\\ \vdots\\ \Delta', x : \vec{a} \vdash S : a\end{array}}{\Delta' \vdash \lambda x.S : \vec{a} \Rightarrow a}$$

$$\left(\frac{\begin{array}{c}\pi_1\\ \vdots\\ \Gamma_1 \vdash S\colon \vec{a} \Rightarrow a\end{array} \quad \begin{array}{c}\pi_2\\ \vdots\\ \Gamma_2 \vdash T : \vec{a}\end{array} \quad \eta : \Delta \to \Gamma_1 \otimes \Gamma_2}{\Delta \vdash ST\colon a}\right)\{\theta\} = \frac{\begin{array}{c}\pi_1\\ \vdots\\ \Gamma_1 \vdash S\colon \vec{a} \Rightarrow a\end{array} \quad \begin{array}{c}\pi_2\\ \vdots\\ \Gamma_2 \vdash T : \vec{a}\end{array} \quad \eta \circ \theta\colon \Delta' \to \Gamma_1 \otimes \Gamma_2}{\Delta' \vdash ST : a}$$

$$\left(\frac{\left(\begin{array}{c}\pi_i\\ \vdots\\ \Gamma_i \vdash S : a_i\end{array}\right)_{i=1}^{k} \quad \eta : \Delta \to \bigotimes_{i=1}^{k} \Gamma_i}{\Delta \vdash S^! : \langle a_1, \ldots, a_k\rangle}\right)\{\theta\} \quad = \quad \frac{\left(\begin{array}{c}\pi_i\\ \vdots\\ \Gamma_i \vdash S : a_i\end{array}\right)_{i=1}^{k} \quad \eta \circ \theta\colon \Delta' \to \bigotimes_{i=1}^{k} \Gamma_i}{\Delta' \vdash S^! : \langle a_1, \ldots, a_k\rangle}$$

Figure 5 Right action on derivations, where $\theta\colon \Delta' \to \Delta$.

Let $f\colon a' \to a$ be a morphism between types a' and a. One can think of them as, *e.g.* $a = \langle *, \langle * \rangle \Rightarrow * \rangle$ and $a' = \langle \langle * \rangle \Rightarrow *, * \rangle$ with $f = \sigma \Rightarrow 1$ being the obvious permutation.

1. Let us type the term $xx^!$ with the following type derivation π (where $A = \langle \langle a \rangle \Rightarrow a, a\rangle$, $A' = \langle a', \langle a\rangle \Rightarrow a\rangle$ and $(1\,2) \in S_2$ is the swap permutation on $\{1, 2\}$):

$$\frac{\dfrac{1_{\langle a\rangle \Rightarrow a}\colon (\langle a\rangle \Rightarrow a) \to (\langle a\rangle \Rightarrow a)}{x : \langle\langle a\rangle \Rightarrow a\rangle \vdash x : \langle a\rangle \Rightarrow a} \quad \dfrac{\dfrac{1_a\colon a \to a}{x : \langle a\rangle \vdash x : a} \quad 1_{\langle a\rangle}\colon \langle a\rangle \to \langle a\rangle}{x : \langle a\rangle \vdash x^! : \langle a\rangle} \quad \langle (1\,2), f, 1_{\langle a\rangle \Rightarrow a}\rangle\colon A' \to A}{x : \langle a', \langle a\rangle \Rightarrow a\rangle \vdash xx^! : a}$$

   Now consider the following type derivation π' (with $A'' = \langle \langle a\rangle \Rightarrow a, a'\rangle$)

$$\frac{\dfrac{1_{\langle a\rangle \Rightarrow a}\colon (\langle a\rangle \Rightarrow a) \to (\langle a\rangle \Rightarrow a)}{x : \langle\langle a\rangle \Rightarrow a\rangle \vdash x : \langle a\rangle \Rightarrow a} \quad \dfrac{\dfrac{f\colon a' \to a}{x : \langle a'\rangle \vdash x : a} \quad 1_{\langle a'\rangle}\colon \langle a'\rangle \to \langle a'\rangle}{x : \langle a'\rangle \vdash x^! : \langle a\rangle} \quad \langle (1\,2), 1_{a'}, 1_{\langle a\rangle \Rightarrow a}\rangle\colon A' \to A''}{x : \langle a', \langle a\rangle \Rightarrow a\rangle \vdash xx^! : a}$$

   Compared to π, π' brings forward the morphism f. By the second rule in Figure 6, $\pi \sim \pi'$.

2. Let us type the term $(\lambda x.x)z^!$ (we omit the index on the identity morphisms 1):

$$\pi = \frac{\dfrac{\dfrac{f\colon a' \to a}{x : \langle a'\rangle \vdash x : a}}{\vdash \lambda x.x : \langle a'\rangle \Rightarrow a} \quad \dfrac{\dfrac{1\colon a' \to a'}{z : \langle a'\rangle \vdash z : a'} \quad 1\colon \langle a'\rangle \to \langle a'\rangle}{z : \langle a'\rangle \vdash z^! : \langle a'\rangle} \quad 1\colon \langle a'\rangle \to \langle a'\rangle}{z : \langle a'\rangle \vdash (\lambda x.x)z^! : a}$$

   Now consider the following derivation (note the different position of f with respect to π)

$$\pi' = \frac{\dfrac{\dfrac{1\colon a \to a}{x : \langle a\rangle \vdash x : a}}{\vdash \lambda x.x : \langle a\rangle \Rightarrow a} \quad \dfrac{\dfrac{f\colon a' \to a}{z : \langle a'\rangle \vdash z : a} \quad 1\colon \langle a'\rangle \to \langle a'\rangle}{z : \langle a'\rangle \vdash z^! : \langle a\rangle} \quad 1\colon \langle a'\rangle \to \langle a'\rangle}{z : \langle a'\rangle \vdash (\lambda x.x)z^! : a}$$

   According to the first rule in Figure 6, $\pi \sim \pi'$.

$$
\dfrac{\begin{matrix}\pi_1\\ \vdots \\ \Gamma_1 \vdash S : \vec{b} \Rightarrow a\end{matrix} \quad \begin{matrix}[\langle \sigma, \vec{f}\rangle]\pi_2 \\ \vdots \\ \Gamma_2 \vdash T : \vec{b}\end{matrix} \quad \eta : \Delta \to \Gamma_1 \otimes \Gamma_2}{\Delta \vdash ST : a}
\sim
\dfrac{\begin{matrix}[\langle \sigma, \vec{f}\rangle \Rightarrow 1]\pi_1\\ \vdots \\ \Gamma_1 \vdash S : \vec{a} \Rightarrow a\end{matrix} \quad \begin{matrix}\pi_2 \\ \vdots \\ \Gamma_2 \vdash T : \vec{a}\end{matrix} \quad \eta : \Delta \to \Gamma_1 \otimes \Gamma_2}{\Delta \vdash ST : a}
$$

$$
\dfrac{\begin{matrix}\pi_1\{\theta_1\}\\ \vdots \\ \Gamma_1 \vdash S : \vec{a} \Rightarrow a\end{matrix} \quad \begin{matrix}\pi_2\{\theta_2\} \\ \vdots \\ \Gamma_2 \vdash T : \vec{a}\end{matrix} \quad \eta : \Delta \to \Gamma_1 \otimes \Gamma_2}{\Delta \vdash ST : a}
\sim
\dfrac{\begin{matrix}\pi_1\\ \vdots \\ \Gamma_1' \vdash S : \vec{a} \Rightarrow a\end{matrix} \quad \begin{matrix}\pi_2 \\ \vdots \\ \Gamma_2' \vdash T : \vec{a}\end{matrix} \quad \theta \circ \eta : \Delta \to \Gamma_1' \otimes \Gamma_2'}{\Delta \vdash ST : a}
$$

$$
\dfrac{\left(\begin{matrix}\pi_i\{\theta_i\}\\ \vdots \\ \Gamma_i \vdash S : a_i\end{matrix}\right)_{i=1}^{k} \quad \eta : \Delta \to \bigotimes_{i=1}^{k} \Gamma_i}{\Delta \vdash S^! : \langle a_1, \dots, a_k \rangle}
\sim
\dfrac{\left(\begin{matrix}\pi_i\\ \vdots \\ \Gamma_i' \vdash S : a_i\end{matrix}\right)_{i=1}^{k} \quad \bigotimes_{i=1}^{k} \theta_i \circ \eta : \Delta \to \bigotimes_{i=1}^{k} \Gamma_i'}{\Delta \vdash S^! : \langle a_1, \dots, a_k \rangle}
$$

Figure 6 Congruence on type derivations, where $\langle \sigma, \vec{f}\rangle \colon \vec{a} \to \vec{b}$ and $\theta = \theta_1 \otimes \theta_2$ with $\theta_i \colon \Gamma_i \to \Gamma_i'$.

Let S be a term and $\mathsf{fv}(S) \subseteq \vec{x} = \{x_1, \dots, x_n\}$ with the x_i's pairwise distinct. With any $\langle \Delta, a\rangle \in \mathrm{ob}((!U^n)^{\mathrm{op}} \times U)$, the distributor $\mathsf{T}_U(S)_{\vec{x}} \colon !U^n \nrightarrow U$ associates the set of (equivalence classes of) type derivations for S with conclusion $\Delta \vdash S : a$. Formally, $\mathsf{T}_U(S)_{\vec{x}}$ is defined by:

1. for $\langle \Delta, a\rangle \in \mathrm{ob}((!U^n)^{\mathrm{op}} \times U)$, $\mathsf{T}_U(S)_{\vec{x}}(\Delta, a) = \left\{ \begin{matrix}\tilde{\pi} \\ \vdots \\ \Delta \vdash S : a\end{matrix} \;\middle|\; \pi \text{ is a type derivation for } S \right\}$;
2. for $f \colon a \to a'$ and $\eta \colon \Delta' \to \Delta$, $\mathsf{T}_U(S)_{\vec{x}}(\eta, f) \colon \mathsf{T}_U(S)_{\vec{x}}(\Delta, a) \to \mathsf{T}_U(S)_{\vec{x}}(\Delta', a')$ such that $\mathsf{T}_U(S)_{\vec{x}}(\eta, f)(\tilde{\pi}) = \widetilde{[f]\pi\{\eta\}} \in \mathsf{T}_U(S)_{\vec{x}}(\Delta', a')$ for any $\tilde{\pi} \in \mathsf{T}_U(S)_{\vec{x}}(\Delta, a)$.

▶ **Lemma 9** (Functoriality). *For any $S \in !\Lambda$, $\mathsf{T}_U(S)_{\vec{x}}$ is a functor from $(!U^n)^{\mathrm{op}} \times U$ to* Set.

The following theorem states that the distributor semantics induced by our category of types U can be seen in a completely type-theoretic way. The semantics $[\![S]\!]_{\vec{x}}(\Delta, a)$ of a term S is equal to the set of (equivalence classes of) type derivations whose conclusion is the sequent $\Delta \vdash S : a$. For this reason we have a bicategorical *proof relevant* semantics. This is a major improvement over relational semantics, where the elements of the denotation of a term are only witnesses of typability. Said differently, the relational semantics of S is just the set of conclusions of the type derivations for S, while the distributor semantics of S provides, for any conclusion, the set of type derivations for S with such a conclusion.

▶ **Theorem 10** (Proof-relevance). *Let $S \in !\Lambda$. There is an isomorphism of functors*

$$\psi \colon [\![S]\!]_{\vec{x}} \cong \mathsf{T}_U(S)_{\vec{x}} \quad \textit{which is natural in } \langle \Delta, a\rangle \in \mathrm{ob}((!U^n)^{\mathrm{op}} \times U).$$

Proof. By induction on the structure of S. The core of the proof is the remark that we can write the equivalence relation induced by the coend in the application and box cases with the rules in Figure 6. ◀

For any type derivation π in system $\mathcal{R}_{\to}$ we define its *size* $\mathrm{s}(\pi)$ in Figure 7 (by induction on π). It counts the number of rules for application in π. Note that if $\pi \sim \pi'$ then $\mathrm{s}(\pi) = \mathrm{s}(\pi')$. We also have that size is invariant under morphisms action: $\mathrm{s}([f]\pi) = \mathrm{s}(\pi\{\eta\}) = \mathrm{s}(\pi)$.

Let $\psi \colon [\![S]\!]_{\vec{x}} \cong \mathsf{T}_U(S)_{\vec{x}}$ be the natural isomorphism of Theorem 10. For $\alpha \in [\![S]\!]_{\vec{x}}(\Delta, a)$ we set $\mathrm{s}(\alpha) = \mathrm{s}(\psi_{\Delta,a}(\alpha))$, *i.e.* the size of a point α is the size of its derivation $\psi_{\Delta,a}(\alpha)$.

Substitution and Reduction. We prove both subject reduction and expansion for non-idempotent intersection type distributors. We enrich this result with a quantitative flavor,

$$\mathrm{s}\left(\dfrac{f : a' \to a}{x_1 : \langle\rangle, \dots, x_i : \langle a' \rangle, \dots, x_n : \langle\rangle \vdash x_i : a}\right) = 0 \qquad \mathrm{s}\left(\dfrac{\left(\begin{array}{c}\pi_i \\ \vdots \\ \Gamma_i \vdash S : a_i\end{array}\right)_{i=1}^{k} \quad \theta : \Delta \to \bigotimes_{i=1}^{k} \Gamma_i}{\Delta \vdash\, !S : \langle a_1, \dots, a_k \rangle}\right) = \sum_{i \in [k]} \mathrm{s}(\pi_i)$$

$$\mathrm{s}\left(\dfrac{\begin{array}{c}\pi' \\ \vdots \\ \Delta, x : \vec{a} \vdash S : a\end{array}}{\Delta \vdash \lambda x.S : \vec{a} \Rightarrow a}\right) = \mathrm{s}(\pi') \qquad \mathrm{s}\left(\dfrac{\begin{array}{c}\pi_1 \\ \vdots \\ \Gamma_1 \vdash S : \vec{a} \Rightarrow a\end{array} \quad \begin{array}{c}\pi_2 \\ \vdots \\ \Gamma_2 \vdash T : \vec{a}\end{array} \quad \theta : \Delta \to \Gamma_1 \otimes \Gamma_2}{\Delta \vdash ST : a}\right) = \mathrm{s}(\pi_1) + \mathrm{s}(\pi_2) + 1$$

Figure 7 Size of type derivations in system $\mathcal{R}_{\to}$.

accounting for how the size of points is affected by a reduction step. In this way, we can give a combinatorial proof for the characterization of terms that are normalizable at depth 0.

The key ingredient is the substitution lemma below. We set:

$$Sub_{S,x,T}(\Delta, a) = \int^{\vec{a} \in !U} \int^{\Gamma_0, \Gamma_1 \in !U^n} [\![S]\!]_{\vec{x} \oplus \langle x \rangle}(\Gamma_0 \oplus \langle \vec{a} \rangle, a) \times [\![T^!]\!]_{\vec{x}}(\Gamma_1, \vec{a}) \times !U^n(\Delta, \Gamma_0 \otimes \Gamma_1).$$

▶ **Lemma 11** (Substitution)**.** *Let S and T be terms. There is an isomorphism of functors*

$$\varphi \colon Sub_{S,x,T} \cong [\![S\{T/x\}]\!]_{\vec{x}}$$

natural in $\langle \Delta, a \rangle \in \mathrm{ob}((!U^n)^{\mathrm{op}} \times U)$ *and such that* $s\left(\varphi_{\Delta,a}(\widetilde{\langle \alpha_1, \alpha_2, \eta \rangle})\right) = s(\alpha_1) + s(\alpha_2)$.

Proof. By induction on the structure of S, *via* lengthy coend manipulations. The proof of the application and list cases strongly relies on the fact that the tensor product of $!U$ is symmetric. The proof of the preservation of sizes relies on the fact that size is invariant under morphism actions and equivalence. Details are in Appendix A. ◀

▶ **Theorem 12** (Subject reduction and expansion)**.** *Let S, T be two terms.*

1. *If $S \to_{\mathrm{b}} T$ then there is a natural isomorphism $[\![S]\!]_{\vec{x}}(\Delta, a) \cong [\![T]\!]_{\vec{x}}(\Delta, a)$.*
2. *If $S \to_{\mathrm{b_g}} T$ then $[\![S]\!]_{\vec{x}}(\Delta, a) \cong [\![T]\!]_{\vec{x}}(\Delta, a)$ via a natural isomorphism $\varphi_{\Delta,a}$ such that $s(\varphi_{\Delta,a}(\alpha)) = s(\alpha) - 1$ for any $\alpha \in [\![S]\!]_{\vec{x}}(\Delta, a)$.*
3. *If $S \to_{\sigma} T$ then $[\![S]\!]_{\vec{x}}(\Delta, a) \cong [\![T]\!]_{\vec{x}}(\Delta, a)$ via a natural isomorphism $\varphi_{\Delta,a}$ such that $s(\varphi_{\Delta,a}(\alpha)) = s(\alpha)$ for any $\alpha \in [\![S]\!]_{\vec{x}}(\Delta, a)$.*

Proof. We prove the base case of Item 2, which follows from the substitution lemma (Lemma 11). Let $S = (\lambda x.S_1)S_2^! \mapsto_{\mathrm{b}} S_1\{S_2/x\} = T$. By definition, we have

$$[\![S]\!]_{\vec{x}}(\Delta, a) = \int^{\vec{a} \in !U} \int^{\Gamma_1, \Gamma_2 \in !U^n} [\![\lambda x.S_1]\!]_{\vec{x}}(\Gamma_1, \iota(\langle \vec{a}, a \rangle)) \times [\![S_2^!]\!]_{\vec{x}}(\Gamma_2, \vec{a}) \times !U^n(\Delta, \Gamma_1 \otimes \Gamma_2).$$

By definition of an abstraction's denotation we have $[\![\lambda x.S_1]\!]_{\vec{x}}(\Gamma_1, \iota(\langle \vec{a}, a \rangle)) = [\![S_1]\!]_{\vec{x} \oplus \langle x \rangle}(\Gamma_1 \oplus \langle \vec{a} \rangle, a)$. Then, $[\![S]\!]_{\vec{x}}(\Delta, a) = Sub_{S_1,x,S_2}(\Delta, a)$. By Lemma 11, $\varphi_{\Delta,a} : [\![(\lambda x.S_1)S_2]\!]_{\vec{x}}(\Delta, a) \cong [\![S_1\{S_2/x\}]\!]_{\vec{x}}(\Delta, a)$. Again by Lemma 11, $\mathrm{s}(\varphi_{\Delta,a}(\beta)) = \mathrm{s}(\alpha_1) + \mathrm{s}(\alpha_2)$ for $\beta = \widetilde{\langle \alpha_1, \alpha_2, \eta \rangle} \in [\![S]\!]_{\vec{x}}(\Delta, a)$. By definition, we have that $\mathrm{s}(\beta) = \mathrm{s}(\alpha_1) + \mathrm{s}(\alpha_2) + 1$. So, we can conclude.

For Item 3, a step $\to_{\sigma}$ just requires to rearrange the rule order in a type derivation. ◀

Roughly, Theorem 12.2 states that if $S \to_{\mathrm{b_g}} T$ then for every type derivation for S there is a type derivation for T, with the same conclusion, whose size decreases by 1. In Theorem 12.1 such a quantitative account does not hold. Indeed, consider $((\lambda x.x)y^!)^! \to_{\mathrm{b}} y^!$: each of the two terms can be typed with a derivation of size 0 (take the rule for boxes with 0 premises).

▶ **Example 13.** We provide a simple example of reduction of type derivations to ease the understanding of the congruence's role in establishing the natural isomorphisms. Consider $S = (\lambda x.x)y^!$. We type it with the following type derivations:

$$\pi_1 = \cfrac{\cfrac{\cfrac{h\circ f\colon a\to b}{x\colon\langle a\rangle\vdash x:b}}{\vdash\lambda x.x:\langle a\rangle\Rightarrow b} \qquad \cfrac{\cfrac{g\colon c\to a}{y:\langle c\rangle\vdash y:a \quad 1}}{y:\langle c\rangle\vdash y^!:\langle a\rangle \quad 1}}{y:\langle c\rangle\vdash(\lambda x.x)y^!:b} \qquad \pi_2 = \cfrac{\cfrac{\cfrac{h\circ f'\colon a'\to b}{x:\langle a\rangle\vdash x:b}}{\vdash\lambda x.x:\langle d\rangle\Rightarrow b} \qquad \cfrac{\cfrac{g'\colon c\to a'}{y:\langle c\rangle\vdash y:a' \quad 1}}{y:\langle c\rangle\vdash y^!:\langle a'\rangle \quad 1}}{y:\langle c\rangle\vdash(\lambda x.x)y^!:b}$$

Suppose that $f\circ g = f'\circ g'$ and $h\colon b\to b$, $f\colon a\to b$, $f'\colon a'\to b$. We have that $\pi_1\sim\pi_2$. Indeed, by the first rule of Figure 6:

$$\pi_1 \sim \cfrac{\cfrac{\cfrac{h\colon b\to b}{x:\langle b\rangle\vdash x:b}}{\vdash\lambda x.x:\langle b\rangle\Rightarrow b} \qquad \cfrac{\cfrac{f\circ g\colon c\to b}{y:\langle c\rangle\vdash y:b \quad 1}}{y:\langle c\rangle\vdash y^!:\langle b\rangle \quad 1}}{y:\langle c\rangle\vdash(\lambda x.x)y^!:b} \qquad \pi_2 \sim \cfrac{\cfrac{\cfrac{h\colon b\to b}{x:\langle b\rangle\vdash x:b}}{\vdash\lambda x.x:\langle b\rangle\Rightarrow b} \qquad \cfrac{\cfrac{f'\circ g'\colon c\to b}{y:\langle c\rangle\vdash y:b \quad 1}}{y:\langle c\rangle\vdash y^!:\langle b\rangle \quad 1}}{y:\langle c\rangle\vdash(\lambda x.x)y^!:b}$$

and, by the hypothesis $f\circ g = f'\circ g'$, we conclude that $\pi_1\sim\pi_2$ by transitivity. In particular, this means that the quotient identify all couple of morphisms leading to the same composition.

Now, we have that $S\to_{\mathrm{b_g}} y$. Consider the following type derivation of y:

$$\pi_3 = \cfrac{h\circ(f\circ g):c\to b}{y:\langle c\rangle\vdash y:b} \qquad \text{(note that } \mathsf{s}(\pi_1)=\mathsf{s}(\pi_2)=1 \text{ and } \mathsf{s}(\pi_3)=0\text{).}$$

By an easy inspection of the definitions we have that for $\varphi_{\langle c\rangle,b} : [\![S]\!]_{\langle y\rangle}(\langle c\rangle,b)\cong[\![y]\!]_{\langle y\rangle}(\langle c\rangle,b)$, $\varphi_{\langle c\rangle,b}(\tilde{\pi_1}) = \pi_3$, where we keep implicit the isomorphism given by Theorem 10. There is then a nice correspondence between *substitution* on the term side and *composition* on the morphism side, that validates the basic intuition of categorical semantics[7].

We prove that non-idempotent intersection type distributors characterize normalization at depth 0, when normal forms are clash-free at depth 0. First, we characterize syntactically the normal forms for $\to_{\mathrm{b}\sigma_{\mathrm{g}}}$ that are clash-free at depth 0. Consider the subsets $!\Lambda_{\mathsf{d}}$, $!\Lambda_{\mathsf{n}}$, $!\Lambda_{\ell}$ (whose elements are denoted by D, N, L, respectively) of $!\Lambda$:

$$(!\Lambda_{\mathsf{d}})\quad D ::= x \mid DS^! \mid DD' \qquad (!\Lambda_{\mathsf{n}})\quad N ::= S^! \mid D \mid (\lambda x.N)D \qquad (!\Lambda_{\ell})\quad L ::= N \mid \lambda x.L$$

All terms in $!\Lambda_{\mathsf{d}}$ are not closed (they have a free "head variable") and are neither a box nor a β-like redex nor an abstraction. Clearly, $!\Lambda_{\mathsf{d}}\subsetneq{}!\Lambda_{\mathsf{n}}$ and $!\Lambda_{!}\subsetneq{}!\Lambda_{\mathsf{n}}\subsetneq{}!\Lambda_{\ell}$ with $!\Lambda_{\mathsf{d}}\cap{}!\Lambda_{!}=\emptyset$.

▶ **Proposition 14** (Syntactic characterization of clash-free at depth 0 normal forms for $\to_{\mathrm{b}\sigma_{\mathrm{g}}}$)**.**

1. *A term S is normal for $\to_{\mathrm{b}\sigma_{\mathrm{g}}}$, clash-free at depth 0 and is neither a box nor a β-like redex (i.e. nor of the form $(\lambda x.S)T$) nor an abstraction iff $S\in{}!\Lambda_{\mathsf{d}}$.*
2. *A term S is normal for $\to_{\mathrm{b}\sigma_{\mathrm{g}}}$, clash-free at depth 0 and is not an abstraction iff $S\in{}!\Lambda_{\mathsf{n}}$.*
3. *A term S is normal for $\to_{\mathrm{b}\sigma_{\mathrm{g}}}$ and clash-free at depth 0 iff $S\in{}!\Lambda_{\ell}$.*

▶ **Lemma 15** (Semantics vs. clash-free at depth 0)**.** *Let S be a term.*

1. *If $[\![S]\!]_{\vec{x}}\neq\emptyset_{!U^n,U}$ then S is clash-free at depth 0.*
2. *If S is normal for $\to_{\mathrm{b}\sigma_{\mathrm{g}}}$ and clash-free at depth 0, then $[\![S]\!]_{\vec{x}}\neq\emptyset_{!U^n,U}$.*

Proof. **1.** By induction on $S\in{}!\Lambda$.

[7] The natural isomorphism $\varphi_{\langle c\rangle,b} : [\![S]\!]_{\langle y\rangle}(\langle c\rangle,b)\cong[\![y]\!]_{\langle y\rangle}(\langle c\rangle,b)$ is a particular instance of Yoneda's lemma for coends (see Lemma 20 in Appendix A), also known as the density formula for coends [40].

2. According to Proposition 14, we can proceed by induction on $S \in !\Lambda_\ell$. ◀

▶ **Theorem 16** (Normalization at depth 0). *Let S be a term. The following are equivalent:*
1. *S is typable in system $\mathcal{R}_{\to}$;*
2. *$[\![S]\!]_{\vec{x}} \neq \emptyset_{!U^n,U}$;*
3. *S is strongly $\to_{\mathrm{b}\sigma_{\mathrm{g}}}$-normalizable with a normal form for $\to_{\mathrm{b}\sigma_{\mathrm{g}}}$ that is clash-free at depth 0;*
4. *S is weakly $\to_{\mathrm{b}\sigma_{\mathrm{g}}}$-normalizable with a normal form for $\to_{\mathrm{b}\sigma_{\mathrm{g}}}$ that is clash-free at depth 0;*
5. *$S \to_{\mathrm{b}\sigma}^{*} T$ for some term T that is normal for $\to_{\mathrm{b}\sigma_{\mathrm{g}}}$ and clash free at depth 0.*

Proof. The equivalence (1) $\Leftrightarrow$ (2) is given by Theorem 10. The implication (5) $\Rightarrow$ (2) follows from Lemma 15.2 and Theorem 12. The implication (4) $\Rightarrow$ (5) holds because $\to_{\mathrm{b}\sigma_{\mathrm{g}}} \subseteq \to_{\mathrm{b}\sigma}$. The implication (3) $\Rightarrow$ (4) is trivial.

For the implication (2) $\Rightarrow$ (3), as $[\![S]\!]_{\vec{x}} \neq \emptyset_{!U^n,U}$, there is a point $\alpha \in [\![S]\!]_{\vec{x}}(\Delta, a)$ for some $\langle \Delta, a\rangle \in \mathrm{ob}(!U^n \times U)$. Let k_S be the sum of the lengths of all $\to_{\sigma_{\mathrm{g}}}$-reduction sequences from S to a normal form for $\to_{\sigma_{\mathrm{g}}}$ (such a k_S exists because $\to_{\sigma_{\mathrm{g}}}$ is strongly normalizing [24]). We prove (3) by induction on $(\mathrm{s}\,(\alpha)\,, k_S)$ ordered lexicographically. If S is normal for $\to_{\mathrm{b}\sigma_{\mathrm{g}}}$, we are done by Lemma 15.1, as $\alpha \in [\![S]\!]_{\vec{x}}(\Delta, a)$ implies $[\![S]\!]_{\vec{x}} \neq \emptyset_{!U^n,U}$. Suppose $S \to_{\mathrm{b}\sigma_{\mathrm{g}}} S'$.
1. If $S \to_{\sigma_{\mathrm{g}}} S'$, let $\varphi\colon [\![S]\!]_{\vec{x}} \cong [\![S']\!]_{\vec{x}}$ be the natural isomorphism of Theorem 12.3. Thus, $\varphi_{\Delta,a}(\alpha) \in [\![S']\!]_{\vec{x}}(\Delta, a)$ and $\mathrm{s}\,(\alpha) = \mathrm{s}\,(\varphi_{\Delta,a}(\alpha))$ but $k_{S'} = k_S - 1$.
2. If $S \to_{\mathrm{b}_{\mathrm{g}}} S'$, let $\varphi\colon [\![S]\!]_{\vec{x}} \cong [\![S']\!]_{\vec{x}}$ be the natural isomorphism of Theorem 12.2. Thus, $\varphi_{\Delta,a}(\alpha) \in [\![S']\!]_{\vec{x}}(\Delta, a)$ and $\mathrm{s}\,(\varphi_{\Delta,a}(\alpha)) = \mathrm{s}\,(\alpha) - 1$.

In both cases, by *i.h.*, (3) holds for S'. Therefore, (3) holds for S. ◀

5 Conclusions

In this paper, we recalled some well-known and linear-logic based categorical semantics with an intersection type presentation. We showed that they can be generalized in the bicategory of distributors. We defined non-idempotent intersection type distributors in the bang calculus and provided a syntactic presentation of them as a non-idempotent intersection type system generalizing De Carvalho's system $\mathcal{R}$ [16, 17]. We proved that non-idempotent intersection type distributors determine a proof-relevant denotational semantics, and characterize normalization at depth 0 in the bang calculus via a combinatorial proof.

Perspectives. Reconciling the different methods used here and in [43, 41, 48] to categorify – non-idempotent or possibly idempotent – intersection types is the first and natural open question. The (non-trivial) answer should rely on a *subtyping-aware polyadic calculus* to be defined. This would allow [41, 48] to have a denotational semantics that supports subtyping.

Another line of research is the study of the extensional collapse [22] in the bicategorical setting of distributors, which should shed new light on the link between non-idempotent and idempotent intersection types. Relating the methods of [29, 43] should be a first step.

A relevant question immediately arises also for what concerns typed call-by-push-value [23, 39]. The extension of our work to that framework is tricky, since the semantics of types adds technical machinery. Moreover, we believe that difficulties similar to the ones found in [13] in order to define the Taylor expansion could arise also in our perspective.

Other interesting perspectives are the investigation of the relationship between our categorified rigid framework and rigid intersection types [50], and an extension of our approach to probabilistic computation. This extension is far from trivial, but the results of Tsukada, Asada and Ong [49] are encouraging, and the study of probabilistic Taylor expansion [38] and probabilistic intersection types [7] might be a starting point.

References

1 Fabio Alessi, Franco Barbanera, and Mariangiola Dezani-Ciancaglini. Intersection types and lambda models. *Theoretical Computer Science*, 355(2):108–126, 2006. Logic, Language, Information and Computation. doi:10.1016/j.tcs.2006.01.004.

2 Hendrik Pieter Barendregt. *The lambda calculus - its syntax and semantics*, volume 103 of *Studies in logic and the foundations of mathematics*. North-Holland, 1984.

3 Jean Bénabou. Introduction to bicategories. In *Reports of the Midwest Category Seminar*, pages 1–77, Berlin, Heidelberg, 1967. Springer Berlin Heidelberg.

4 Alexis Bernadet and Stéphane Jean Lengrand. Non-idempotent intersection types and strong normalisation. *Logical Methods in Computer Science*, Volume 9, Issue 4, 2013. doi:10.2168/LMCS-9(4:3)2013.

5 Robert Blackwell, Gregory Maxwell Kelly, and John Power. Two-dimensional monad theory. *Journal of Pure and Applied Algebra*, 59(1):1–41, 1989. doi:10.1016/0022-4049(89)90160-6.

6 Francis Borceux. *Handbook of Categorical Algebra*, volume 1 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1994. doi:10.1017/CBO9780511525858.

7 Flavien Breuvart and Ugo Dal Lago. On Intersection Types and Probabilistic Lambda Calculi. In *Proceedings of the 20th International Symposium on Principles and Practice of Declarative Programming, PPDP 2018, Frankfurt am Main, Germany, September 03-05, 2018*, pages 8:1–8:13, 2018. doi:10.1145/3236950.3236968.

8 Antonio Bucciarelli, Delia Kesner, Alejandro Ríos, and Andrés Viso. The bang calculus revisited. In *Functional and Logic Programming - 15th International Symposium, FLOPS 2020*, volume 12073 of *Lecture Notes in Computer Science*, pages 13–32. Springer, 2020. doi:10.1007/978-3-030-59025-3_2.

9 Antonio Bucciarelli, Delia Kesner, and Daniel Ventura. Non-idempotent intersection types for the lambda-calculus. *Logic Journal of the IGPL*, 25(4):431–464, 2017. doi:10.1093/jigpal/jzx018.

10 Jean Bénabou. Distributors at work. Lecture notes of a course given at TU Darmstadt, 2000. URL: http://www2.mathematik.tu-darmstadt.de/~streicher/FIBR/DiWo.pdf.

11 Alberto Carraro and Giulio Guerrieri. A Semantical and Operational Account of Call-by-Value Solvability. In *Foundations of Software Science and Computation Structures, FOSSACS 2014*, volume 8412 of *Lecture Notes in Computer Science*, pages 103–118, Berlin, Heidelberg, 2014. Springer. doi:10.1007/978-3-642-54830-7_7.

12 Gian Luca Cattani and Glynn Winskel. Profunctors, open maps and bisimulation. *Mathematical Structures in Computer Science*, 15(3):553–614, 2005. doi:10.1017/S0960129505004718.

13 Jules Chouquet and Christine Tasson. Taylor expansion for Call-by-Push-Value. In *28th EACSL Annual Conference on Computer Science Logic, CSL 2020*, volume 152 of *LIPIcs*, pages 16:1–16:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CSL.2020.16.

14 Mario Coppo and Mariangiola Dezani-Ciancaglini. A new type-assignment for lambda terms. *Arch. Math. Log.*, 19(1):139–156, 1978. doi:10.1007/BF02011875.

15 Mario Coppo and Mariangiola Dezani-Ciancaglini. An extension of the basic functionality theory for the λ-calculus. *Notre Dame Journal of Formal Logic*, 21(4):685–693, 1980. doi:10.1305/ndjfl/1093883253.

16 Daniel de Carvalho. *Semantique de la logique lineaire et temps de calcul*. PhD thesis, Aix-Marseille Université, 2007.

17 Daniel de Carvalho. Execution time of λ-terms via denotational semantics and intersection types. *Math. Struct. Comput. Sci.*, 28(7):1169–1203, 2018. doi:10.1017/S0960129516000396.

18 Daniel de Carvalho. Taylor expansion in linear logic is invertible. *Log. Methods Comput. Sci.*, 14(4), 2018. doi:10.23638/LMCS-14(4:21)2018.

19 Daniel de Carvalho, Michele Pagani, and Lorenzo Tortora de Falco. A semantic measure of the execution time in linear logic. *Theoretical Computer Science*, 412(20):1884–1902, 2011. doi:10.1016/j.tcs.2010.12.017.

20 Daniel de Carvalho and Lorenzo Tortora de Falco. A semantic account of strong normalization in linear logic. *Inf. Comput.*, 248:104–129, 2016. doi:10.1016/j.ic.2015.12.010.

21 Thomas Ehrhard. Collapsing non-idempotent intersection types. In *Computer Science Logic (CSL'12) - 26th International Workshop/21st Annual Conference of the EACSL, CSL 2012*, volume 16 of *LIPIcs*, pages 259–273. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012. doi:10.4230/LIPIcs.CSL.2012.259.

22 Thomas Ehrhard. The Scott model of linear logic is the extensional collapse of its relational model. *Theoretical Computer Science*, 424:20–45, 2012. doi:10.1016/j.tcs.2011.11.027.

23 Thomas Ehrhard. Call-by-push-value from a linear logic point of view. In *Programming Languages and Systems - 25th European Symposium on Programming, ESOP 2016*, volume 9632 of *Lecture Notes in Computer Science*, pages 202–228. Springer, 2016. doi:10.1007/978-3-662-49498-1_9.

24 Thomas Ehrhard and Giulio Guerrieri. The bang calculus: An untyped lambda-calculus generalizing call-by-name and call-by-value. In *Proceedings of the 18th International Symposium on Principles and Practice of Declarative Programming, PPDP 2016*, pages 174–187. Association for Computing Machinery, 2016. doi:10.1145/2967973.2968608.

25 Thomas Ehrhard and Laurent Regnier. Böhm trees, Krivine's machine and the Taylor expansion of lambda-terms. In *Logical Approaches to Computational Barriers, Second Conference on Computability in Europe, CiE 2006*, volume 3988 of *Lecture Notes in Computer Science*, pages 186–197. Springer, 2006. doi:10.1007/11780342_20.

26 Thomas Ehrhard and Laurent Regnier. Uniformity and the Taylor expansion of ordinary λ-terms. *Theoretical Computer Science*, 403(2-3), 2008. doi:10.1016/j.tcs.2008.06.001.

27 Marcelo Fiore, Nicola Gambino, Martin Hyland, and Glynn Winskel. The cartesian closed bicategory of generalised species of structures. *J. of the London Mathematical Society*, 77(1):203–220, 2008. doi:10.1112/jlms/jdm096.

28 Marcelo Fiore, Nicola Gambino, Martin Hyland, and Glynn Winskel. Relative pseudomonads, Kleisli bicategories, and substitution monoidal structures. *Selecta Mathematica*, 24(3):2791–2830, November 2017. doi:10.1007/s00029-017-0361-3.

29 Zeinab Galal. A Profunctorial Scott Semantics. In *5th International Conference on Formal Structures for Computation and Deduction (FSCD 2020)*, volume 167 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:18, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.FSCD.2020.16.

30 Nicola Gambino and André Joyal. On operads, bimodules and analytic functors. *Memoirs of the American Mathematical Society*, 249(1184):0–0, September 2017. doi:10.1090/memo/1184.

31 Philippa Gardner. Discovering needed reductions using type theory. In *Theoretical Aspects of Computer Software, International Conference TACS '94*, volume 789 of *Lecture Notes in Computer Science*, pages 555–574. Springer, 1994. doi:10.1007/3-540-57887-0_115.

32 Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987. doi:10.1016/0304-3975(87)90045-4.

33 Giulio Guerrieri and Giulio Manzonetto. The bang calculus and the two Girard's translations. In *Proceedings Joint International Workshop on Linearity & Trends in Linear Logic and Applications, Linearity-TLLA@FLoC 2018*, volume 292 of *EPTCS*, pages 15–30, 2018. doi:10.4204/EPTCS.292.2.

34 Martin Hyland. Classical lambda calculus in modern dress. *Mathematical Structures in Computer Science*, 27(5):762–781, 2017. doi:10.1017/S0960129515000377.

35 André Joyal. Foncteurs analytiques et espèces de structures. In *Combinatoire énumérative*, pages 126–159, Berlin, Heidelberg, 1986. Springer Berlin Heidelberg.

36 Gregory Maxwell Kelly. A unified treatment of transfinite constructions for free algebras, free monoids, colimits, associated sheaves, and so on. *Bulletin of the Australian Mathematical Society*, 22(1):1–83, 1980. doi:10.1017/S0004972700006353.

37 Jean-Louis Krivine. Lambda-calculus, types and models. In *Ellis Horwood series in computers and their applications*, 1993.

38 Ugo Dal Lago and Thomas Leventis. On the Taylor expansion of probabilistic lambda-terms. In Herman Geuvers, editor, *4th International Conference on Formal Structures for Computation and Deduction, FSCD 2019, June 24-30, 2019, Dortmund, Germany*, volume 131 of *LIPIcs*, pages 13:1–13:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.FSCD.2019.13.

39 Paul Blain Levy. Call-by-Push-Value: A Subsuming Paradigm. In *Typed Lambda Calculi and Applications, 4th International Conference, TLCA'99*, volume 1581 of *Lecture Notes in Computer Science*, page 228–242. Springer, 1999. doi:10.1007/3-540-48959-2_17.

40 Fosco Loregian. This is the (co)end, my only (co)friend, 2015. arXiv:1501.02503.

41 Damiano Mazza, Luc Pellissier, and Pierre Vial. Polyadic approximations, fibrations and intersection types. *Proc. ACM Program. Lang.*, 2(POPL):6:1–6:28, 2018. doi:10.1145/3158094.

42 Paul-André Melliès and Noam Zeilberger. Functors are type refinement systems. *SIGPLAN Not.*, 50(1):3–16, January 2015. doi:10.1145/2775051.2676970.

43 Federico Olimpieri. Intersection Type Distributors, 2020. arXiv:2002.01287.

44 Federico Olimpieri. *Intersection Types and Resource Calculi in the Denotational Semantics of Lambda-Calculus*. PhD thesis, Aix-Marseille Université, 2020.

45 Luca Paolini and Simona Ronchi Della Rocca. Call-by-value solvability. *RAIRO Theor. Informatics Appl.*, 33(6):507–534, 1999. doi:10.1051/ita:1999130.

46 José Espírito Santo, Luís Pinto, and Tarmo Uustalu. Modal embeddings and calling paradigms. In *4th International Conference on Formal Structures for Computation and Deduction, FSCD 2019*, volume 131 of *LIPIcs*, pages 18:1–18:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.FSCD.2019.18.

47 Alex K. Simpson. Reduction in a linear lambda-calculus with applications to operational semantics. In *Term Rewriting and Applications, 16th International Conference, RTA 2005*, volume 3467 of *Lecture Notes in Computer Science*, pages 219–234. Springer, 2005. doi:10.1007/978-3-540-32033-3_17.

48 Takeshi Tsukada, Kazuyuki Asada, and C.-H. Luke Ong. Generalised Species of Rigid Resource Terms. In *Proceedings of the 32rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS 2017, pages 1–12. IEEE Computer Society, 2017. doi:10.1109/LICS.2017.8005093.

49 Takeshi Tsukada, Kazuyuki Asada, and C.-H. Luke Ong. Species, profunctors and Taylor expansion weighted by SMCC: A unified framework for modelling nondeterministic, probabilistic and quantum programs. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, pages 889–898. IEEE Computer Society, 2018. doi:10.1145/3209108.3209157.

50 Pierre Vial. Infinitary intersection types as sequences: A new answer to Klop's problem. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017*, pages 1–12. IEEE Computer Society, 2017. doi:10.1109/LICS.2017.8005103.

A Appendix

Bicategories in a Nutshell [3, 6]. Intuitively, a bicategory is a category with "morphisms between morphisms", that is, where each hom-set itself carries the structure of a category, but the composition of morphisms is only associative up to an isomorphism, and similarly for the identities laws. Formally, a *bicategory* $\mathcal{C}$ consists of:

- a set $\mathrm{ob}(\mathcal{C})$ of *objects*, also called 0-*cells* and denoted by $A, B, C, \ldots$;
- for all $A, B \in \mathrm{ob}(\mathcal{C})$, a category $\mathcal{C}(A, B)$; objects in $\mathcal{C}(A, B)$ are called 1-*cells* or *morphisms* from A to B; while arrows in $\mathcal{C}(A, B)$ (between 1-cells from A to B) are called 2-*cells* or 2-*morphisms*; composition of 2-cells is generally called *vertical composition*;
- for every $A, B, C \in \mathrm{ob}(\mathcal{C})$, a bifunctor

  $$\circ_{A,B,C} \colon \mathcal{C}(B, C) \times \mathcal{C}(A, B) \to \mathcal{C}(A, C)$$

  called *horizontal composition* (often the indices A, B, C in $\circ_{A,B,C}$ are omitted); hence, for all 1-cells $F \colon A \to B$, $F' \colon A \to B$ and $G \colon B \to C$, $G' \colon B \to C$, and for all 2-cells $\alpha \colon F \Rightarrow F'$ and $\beta \colon G \Rightarrow G'$, we have

  $$\text{a 1-cell } G \circ_{A,B,C} F \colon A \to C \quad \text{a 2-cell } \beta \circ_{A,B,C} \alpha \colon (G \circ_{A,B,C} F) \Rightarrow (G' \circ_{A,B,C} F');$$

- for every $A \in \mathrm{ob}(\mathcal{C})$ a functor $1_A \colon 1 \to \mathcal{C}(A, A)$; with an abuse of notation we identify $1_A(\star)$ with 1_A and we call it the identity of A;
- for all 1-cells $F \colon A \to B$, $G \colon B \to C$ and $H \colon C \to D$, a family of invertible 2-cells

  $$\alpha_{H,G,F} \colon H \circ (G \circ F) \cong (H \circ G) \circ F$$

  expressing the associativity laws;
- for every 1-cell $F \colon A \to B$, two families of invertible 2-cells

  $$\lambda_F \colon 1_B \circ F \cong F \qquad \rho_F \colon F \cong F \circ 1_A$$

  expressing the identity laws.

This data is subject to additional coherence axioms. A 2-*category* is a bicategory where the associativity and identities are strict equalities, not only isomorphisms.

▶ **Definition 17** (Retraction). *Let D, E be 0-cells in a bicategory $\mathcal{C}$. A* retraction *of D to E is a couple of 1-cells $i \colon E \to D$, $j \colon D \to E$ together with an invertible 2-cell β such that the diagram below commute. We write $E \lhd D$ is there is a retraction of D to E.*

$$\begin{tikzcd} E \arrow[r, "i"] \arrow[d, "1_E"'] & D \arrow[dl, bend left, "j"] \arrow[dl, phantom, "\Swarrow\beta"] \\ E & \end{tikzcd}$$

Coends. Given a functor $F : C^{op} \times C \to \mathrm{Set}$ we recall that the coend is the coequalizer of the following diagram

$$\sum_{c,c' \in C} C(c', c) \times F(c, c') \rightrightarrows \sum_{c \in C} F(c, c) \to \int^{c \in C} F(c, c)$$

where the parallel arrows are given by left and right actions of F on morphisms $f \in C(c', c)$. Since we work with coends in the category of set, we have that this coequalizer is actually given by the quotient $\sum_{c \in C} F(c, c)/\sim$ where the equivalence relation is generated by the rule $x \sim y$ iff $F(f, c')(x) = y, F(c, f)(y) = x$, for $f : c' \to c$.

We list the three fundamental lemmas of coend calculus [40].

▶ **Lemma 18.** *Every cocontinuous functor preserves coends.*

▶ **Lemma 19** (Fubini [40])**.** *Let $F : C^{op} \times C \times D^{op} \times D \to Set$ be a functor. We have*

$$\int^{\langle c,d\rangle \in C\times D} F(c,c,d,d) \cong \int^{c\in C}\int^{d} F(c,c,d,d) \cong \int^{d\in D}\int^{c\in C} F(c,c,d,d).$$

▶ **Lemma 20** (Yoneda Ninja [40])**.** *Let $K, H : C \to Set$ be, respectively, a contravariant and a covariant functor. We have the following natural isomorphisms*

$$K(-) \cong \int^{c\in C} K(c) \times C(-,c) \qquad H(-) \cong \int^{c\in C} H(c) \times C(c,-).$$

Denotation under Reduction. In what follows we do not explicitly state, for readability reasons, when we apply Lemmas 18 and 19. For $\vec{\Gamma} = \langle \Gamma_1, \ldots, \Gamma_n\rangle$ we set $\bigotimes \vec{\Gamma} = \bigotimes_{i=1}^{n} \Gamma_i$.

▶ **Lemma 11** (Substitution)**.** *Let S and T be terms. There is an isomorphism of functors*

$$\varphi\colon Sub_{S,x,T} \cong [\![S\{T/x\}]\!]_{\vec{x}}$$

natural in $\langle \Delta, a\rangle \in \mathrm{ob}((!U^n)^{\mathrm{op}} \times U)$ and such that $s\left(\varphi_{\Delta,a}(\widetilde{\langle \alpha_1, \alpha_2, \eta\rangle})\right) = s(\alpha_1) + s(\alpha_2)$.

Proof. By induction on the structure of $S \in !\Lambda$, via lengthy coend manipulations.

If $S = x$ then

$$Sub_{S,x,T}(\Delta, a) = \int^{\Gamma_0,\Gamma_1\in !U^n}\int^{\vec{a}\in !U} [\![x]\!]_{\vec{x}}(\Gamma_0 \oplus \langle \vec{a}\rangle, a) \times [\![T^!]\!]_{\vec{x}}(\Gamma_1, \vec{a}) \times !U(\Delta, \Gamma_0 \otimes \Gamma_1).$$

By definition we have

$$\cong \int^{\Gamma_0,\Gamma_1\in !U^n}\int^{\vec{a}\in !U} !U^n(\Gamma_0 \oplus \langle \vec{a}\rangle, \langle\langle\rangle, \ldots, \langle\rangle, \langle a\rangle\rangle) \times [\![T^!]\!]_{\vec{x}}(\Gamma_1, \vec{a}) \times !U(\Delta, \Gamma_0 \otimes \Gamma_1).$$

Then, by the structure of the product category

$$\cong \int^{\Gamma_0,\Gamma_1\in !U^n}\int^{\vec{a}\in !U} !U^n(\Gamma_0, \langle\langle\rangle, \ldots, \langle\rangle\rangle) \times !U(\vec{a}, \langle a\rangle) \times [\![T^!]\!]_{\vec{x}}(\Gamma_1, \vec{a}) \times !U(\Delta, \Gamma_0 \otimes \Gamma_1).$$

Then, by Yoneda (Lemma 20) we have

$$\cong \int^{\Gamma_1\in !U^n}\int^{\vec{a}\in !U} !U(\vec{a}, \langle a\rangle) \times [\![T^!]\!]_{\vec{x}}(\Gamma_1, \vec{a}) \times !U(\Delta, \Gamma_1).$$

Again, by Yoneda (Lemma 20),

$$\cong \int^{\Gamma_1\in !U^n} [\![T^!]\!]_{\vec{x}}(\Gamma_1, \langle a\rangle) \times !U(\Delta, \Gamma_1).$$

Then, by applying Yoneda one more time on the context Γ and by definition of the denotation of a box we can conclude. For what concerns the size, simply notice that $\mathrm{s}(\tilde{\pi}) = \mathrm{s}([f]\tilde{\pi})$.

The abstraction case follows from the *i.h.* immediately.

We do the application, the box case being similar to it. If $S = QR$ then

$$Sub_{S,x,T}(\Delta, a) = \int^{\Gamma_1,\Gamma_2}\int^{\vec{a}} [\![QR]\!]_{\vec{x}\oplus\langle x\rangle}(\Gamma_1 \oplus \langle \vec{a}\rangle, a) \times [\![T^!]\!]_{\vec{x}}(\Gamma_2, \vec{a}) \times !U^n(\Delta, \Gamma_1 \otimes \Gamma_2).$$

We develop $[\![QR]\!]_{\vec{x}\oplus\langle x\rangle}(\Gamma_1 \oplus \langle\vec{a}\rangle, a)$:

$$[\![QR]\!]_{\vec{x}\oplus\langle x\rangle}(\Gamma_1 \oplus \langle\vec{a}\rangle, a) = \int^{\Gamma_1'\oplus\langle\vec{a}_1\rangle,\Gamma_2'\oplus\langle\vec{a}_2\rangle}\int^{\vec{b}} [\![Q]\!]_{\vec{x}\oplus\langle x\rangle}(\Gamma_1' \oplus \langle\vec{a}_1\rangle, \iota(\vec{b}, a)) \\ \times [\![R]\!]_{\vec{x}\oplus\langle x\rangle}(\Gamma_2' \oplus \langle\vec{a}_2\rangle, \vec{b}) \times !U^n(\Gamma_1 \oplus \langle\vec{a}\rangle, \Gamma_1' \oplus \langle\vec{a}_1\rangle \otimes \Gamma_2' \oplus \langle\vec{a}_2\rangle).$$

By the structure of the product category, we have

$$[\![QR]\!]_{\vec{x}\oplus\langle x\rangle}(\Gamma_1 \oplus \langle\vec{a}\rangle, a) = \int^{\Gamma_1'\Gamma_2'}\int^{\vec{a}_1,\vec{a}_2}\int^{\vec{b}} [\![Q]\!]_{\vec{x}\oplus\langle x\rangle}(\Gamma_1' \oplus \langle\vec{a}_1\rangle, \iota(\vec{b}, a)) \\ \times [\![R]\!]_{\vec{x}\oplus\langle x\rangle}(\Gamma_2' \oplus \langle\vec{a}_2\rangle, \vec{b}) \times !U^n(\Gamma_1, \Gamma_1' \otimes \Gamma_2') \times !U(\vec{a}, \vec{a}_1 \oplus \vec{a}_2).$$

We apply Yoneda (Lemma 20) on Γ_1 and on $\vec{a}$ and we get

$$Sub_{S,x,T}(\Delta, a) \cong \int^{\Gamma_1'\Gamma_2',\Gamma_2}\int^{\vec{b},\vec{a}_1,\vec{a}_2} [\![Q]\!]_{\vec{x}\oplus\langle x\rangle}(\Gamma_1' \oplus \langle\vec{a}_1\rangle, \iota(\vec{b}, a)) \times [\![R]\!]_{\vec{x}\oplus\langle x\rangle}(\Gamma_2' \oplus \langle\vec{a}_2\rangle, \vec{b}) \\ \times [\![T^!]\!]_{\vec{x}}(\Gamma_2, \vec{a}_1 \oplus \vec{a}_2) \times !U^n(\Delta, (\Gamma_1' \otimes \Gamma_2') \otimes \Gamma_2).$$

By a simple inspection of the definition of the denotation of a box, we can rewrite it as

$$\cong \int^{\Gamma_i',\Gamma_{\vec{a}_i},\Gamma_2}\int^{\vec{b},\vec{a}_1,\vec{a}_2} [\![Q]\!]_{\vec{x}\oplus\langle x\rangle}(\Gamma_1' \oplus \langle\vec{a}_1\rangle, \iota(\vec{b}, a)) \times [\![R]\!]_{\vec{x}\oplus\langle x\rangle}(\Gamma_2' \oplus \langle\vec{a}_2\rangle, \vec{b}) \times [\![T^!]\!]_{\vec{x}}(\Gamma_{\vec{a}_1}, \vec{a}_1) \\ \times [\![T^!]\!]_{\vec{x}}(\Gamma_{\vec{a}_2}, \vec{a}_2) \times !U^n(, \Gamma_2, \bigotimes \Gamma_{\vec{a}_1} \otimes \bigotimes \Gamma_{\vec{a}_2}) \times !U^n(\Delta, (\Gamma_1' \otimes \Gamma_2') \otimes \Gamma_2).$$

Where, if we set $\vec{a}_i = \langle a_{i,1}, \dots, a_{i,k_i}\rangle$, $[\![T]\!]_{\vec{x}}(\Gamma_{\vec{a}_i}, \vec{a}_i) = \prod_{j\in k_i} [\![T]\!]_{\vec{x}}(\Gamma_{i,j}, a_{i,j})$ and $\bigotimes \Gamma_{\vec{a}_i} = \bigotimes_{j\in k_i} \Gamma_{i,j}$ with $i \in \{1, 2\}$. We apply Yoneda on Γ_2

$$\cong \int^{\Gamma_i',\Gamma_{\vec{a}_i}}\int^{\vec{b},\vec{a}_1,\vec{a}_2} [\![Q]\!]_{\vec{x}\oplus\langle x\rangle}(\Gamma_1' \oplus \langle\vec{a}_1\rangle, \iota(\vec{b}, a)) \times [\![R]\!]_{\vec{x}\oplus\langle x\rangle}(\Gamma_2' \oplus \langle\vec{a}_2\rangle, \vec{b}) \times [\![T^!]\!]_{\vec{x}}(\Gamma_{\vec{a}_1}, \vec{a}_1) \\ \times [\![T^!]\!]_{\vec{x}}(\Gamma_{\vec{a}_2}, \vec{a}_2) \times !U^n(\Delta, (\Gamma_1' \otimes \Gamma_2') \otimes (\bigotimes \Gamma_{\vec{a}_1} \otimes \bigotimes \Gamma_{\vec{a}_2})).$$

Now, by the *symmetry* of the tensor product $\otimes$ and by the fact that functors preserves isomorphisms, we get

$$\cong \int^{\Gamma_i',\Gamma_{\vec{a}_i}}\int^{\vec{b},\vec{a}_1,\vec{a}_2} [\![Q]\!]_{\vec{x}\oplus\langle x\rangle}(\Gamma_1' \oplus \langle\vec{a}_1\rangle, \iota(\vec{b}, a)) \times [\![R]\!]_{\vec{x}\oplus\langle x\rangle}(\Gamma_2' \oplus \langle\vec{a}_2\rangle, \vec{b}) \times [\![T^!]\!]_{\vec{x}}(\Gamma_{\vec{a}_1}, \vec{a}_1) \\ \times [\![T^!]\!]_{\vec{x}}(\Gamma_{\vec{a}_2}, \vec{a}_2) \times !U^n(, \Gamma_2, \bigotimes \Gamma_{\vec{a}_1} \otimes \bigotimes \Gamma_{\vec{a}_2}) \times !U^n(\Delta, ((\Gamma_1' \otimes \Gamma_{\vec{a}_1}) \otimes (\Gamma_2' \otimes \Gamma_{\vec{a}_2})).$$

Now, if we apply Yoneda twice to $\Gamma_i' \otimes \Gamma_{\vec{a}_i}$, we get

$$\cong \int^{\Gamma_i',\Gamma_{\vec{a}_i},\Delta_i}\int^{\vec{b},\vec{a}_1,\vec{a}_2} [\![Q]\!]_{\vec{x}\oplus\langle x\rangle}(\Gamma_1 \oplus \langle\vec{a}_1\rangle, \iota(\vec{b}, a)) \times [\![R]\!]_{\vec{x}\oplus\langle x\rangle}(\Gamma_2' \oplus \langle\vec{a}_2\rangle, \vec{b}) \times [\![T^!]\!]_{\vec{x}}(\Gamma_{\vec{a}_1}', \vec{a}_1) \\ \times [\![T^!]\!]_{\vec{x}}(\Gamma_{\vec{a}_2}, \vec{a}_2) \times !U^n(\Delta, \Delta_1 \otimes \Delta_2) \otimes !U^n(\Delta_1, \Gamma_1' \otimes \bigotimes \Gamma_{\vec{a}_1}) \otimes !U^n(\Delta_2, \Gamma_2' \otimes \bigotimes \Gamma_{\vec{a}_2}).$$

By co-continuity and commutativity, and by applying Yoneda (Lemma 20) twice, we have

$$\cong \int^{\vec{b}}\int^{\Gamma_1',\Gamma_{\vec{a}_1},\Delta_1,\Phi_1}\int^{\vec{a}_1} [\![Q]\!]_{\vec{x}\oplus\langle x\rangle}(\Gamma_1' \oplus \langle\vec{a}_1\rangle, \iota(\vec{b}, a)) \times [\![T^!]\!]_{\vec{x}}(\Phi_1, \vec{a}_1) \\ \times !U^n(\Phi_1, \bigotimes \Gamma_{\vec{a}_1}') \times U^n(\Delta_1, \Gamma_1' \otimes \Phi_1) \\ \times \int^{\Gamma_2',\Gamma_{\vec{a}_2},\Delta_2,\Phi_2}\int^{\vec{a}_2} [\![R]\!]_{\vec{x}\oplus\langle x\rangle}(\Gamma_2' \oplus \langle\vec{a}_2\rangle, \vec{b}) \times [\![T^!]\!]_{\vec{x}}(\Phi_2, \vec{a}_2) \\ \times !U^n(\Phi_2, \bigotimes \Gamma_{\vec{a}_2}') \times U^n(\Delta_2, \Gamma_2' \otimes \Phi_2) \times !U^n(\Delta, \Delta_1 \otimes \Delta_2).$$

By definition, the former coend is just

$$\int^{\vec{b}} \int^{\Delta_1, \Delta_2} Sub_{Q,x,T}(\Delta_1, \iota(\vec{b}, a)) \times Sub_{R,x,T}(\Delta_2, \vec{b}) \times {!U^n}(\Delta, \Delta_1 \otimes \Delta_2).$$

We remark that, forgetting the equivalence relation, the built isomorphism

$$Sub_{S,x,T}(\Delta, a) \cong \int^{\vec{b}} Sub_{Q,x,T}(\Delta_1, \iota(\vec{b}, a)) \times Sub_{R,x,T}(\Delta_2, \vec{b}) \times {!U^n}(\Delta, \Delta_1 \otimes \Delta_2)$$

consists of the following map

$$\langle \vec{a}, \langle \vec{b}, \langle \Gamma_1, \Gamma_2, \langle \langle \Gamma'_1 \oplus \langle \vec{a}_1 \rangle, \Gamma'_2 \oplus \langle \vec{a}_2 \rangle, \langle \alpha_1, \alpha_2, \eta_1 \rangle \rangle, \langle \langle \vec{\Gamma} = \langle \Gamma_{2,1}, \ldots, \Gamma_{2,\mathsf{len}(\vec{a})} \rangle, \vec{\beta} = \langle \beta_1, \ldots, \beta_{\mathsf{len}(\vec{a})} \rangle, \eta_2 \rangle \rangle \rangle, \theta \rangle \rangle \rangle \mapsto$$

$$\langle \vec{b}, \langle \Gamma'_1 \otimes \bigotimes \Gamma_{\vec{a}_1}, \alpha_1, \langle \vec{\beta}_{\vec{a}_1}, 1_{\bigotimes \Gamma_{\vec{a}_1}} \rangle, 1_{\Gamma'_1 \otimes \bigotimes \Gamma_{\vec{a}_1}} \rangle, \langle \Gamma'_2 \otimes \bigotimes \Gamma_{\vec{a}_2}, \alpha_2, \langle \vec{\beta}_{\vec{a}_2}, 1_{\bigotimes \Gamma_{\vec{a}_2}} \rangle, 1_{\Gamma'_2 \otimes \bigotimes \Gamma_{\vec{a}_2}} \rangle,$$

$$((\eta_1 \otimes (\sigma^\star \circ \eta_2)) \circ \theta) \circ \tau \rangle$$

where

- $\theta : \Delta \to \Gamma_1 \otimes \Gamma_2, \alpha_1 \in [\![Q]\!]_{\vec{x}}(\Gamma'_1 \oplus \langle \vec{a}_1 \rangle, \iota(\vec{b}, a))$ and $\alpha_2 \in [\![R]\!]_{\vec{x}}(\Gamma'_2 \oplus \langle \vec{a}_2 \rangle, \vec{b})$;
- $\langle \eta_1, f = \langle \sigma, \vec{f} \rangle \rangle : \Gamma_1 \oplus \langle \vec{a} \rangle \to \Gamma'_1 \oplus \langle \vec{a}_1 \rangle \otimes \Gamma'_2 \oplus \langle \vec{a}_2 \rangle$ and $\eta_2 : \Gamma_2 \to \bigotimes \vec{\Gamma}, \vec{\beta} \in [\![T]\!]_{\vec{x}}(\Gamma_2, \vec{a})$;
- $[f]\vec{\Gamma} = \Gamma_{\vec{a}_1} \otimes \Gamma_{\vec{a}_2}$ and $[f]\vec{\beta} = \vec{\beta}_{\vec{a}_1} \oplus \vec{\beta}_{\vec{a}_2}$;
- $\tau : (\Gamma'_1 \otimes \Gamma'_2) \otimes (\bigotimes \Gamma_{\vec{a}_1} \otimes \bigotimes \Gamma_{\vec{a}_2}) \to (\Gamma'_1 \otimes \bigotimes \Gamma_{\vec{a}_1}) \otimes (\Gamma'_2 \otimes \bigotimes \Gamma_{\vec{a}_2})$ is the obvious symmetry.

By definition, we have (for $S = QR$)

$$[\![S\{T/x\}]\!]_{\vec{x}}(\Delta, a) = \int^{\vec{b}} \int^{\Delta_1, \Delta_2} [\![Q\{T/x\}]\!]_{\vec{x}}(\Delta_1, \iota(\vec{b}, a)) \times [\![R\{T/x\}]\!]_{\vec{x}}(\Delta_2, \vec{b}) \times {!U^n}(\Delta, \Delta_1 \otimes \Delta_2).$$

By *i.h.*, we get two isomorphisms $[\![Q\{T/x\}]\!]_{\vec{x}}(\Delta_1, \iota(\vec{b}, a)) \cong Sub_{Q,x,T}(\Delta_1, \iota(\vec{b}, a))$ and $[\![R\{T/x\}]\!]_{\vec{x}}(\Delta_2, \vec{b}) \cong Sub_{R,x,T}(\Delta_2, \vec{b})$. We have our isomorphism, since isomorphisms are preserved by products and coends. Then we can conclude, since morphism actions do not change size of points and we have $\mathrm{s}\left(\vec{\beta}\right) = \mathrm{s}\left(\vec{\beta}_{\vec{a}_1}\right) + \mathrm{s}\left(\vec{\beta}_{\vec{a}_2}\right)$. ◀

Failure of Subject Reduction with Subtyping for Polyadic Terms. In Section 1 (see Footnote 2), we mentioned that polyadic terms [41, 48] fail dramatically subject reduction for intersection type distributors. Here we show a counterexample. We recall the definition of linear polyadic calculus [41] in the framework of bang calculus.

$$p, q ::= x \mid \lambda \langle x_1, \ldots, x_k \rangle.p \mid pq \mid \langle p_1, \ldots, p_k \rangle \mid \bot$$

Terms are taken up to α-equivalence and up to linearity with respect to $\bot$ (*i.e.*, $\lambda \vec{x}.\bot = p\langle \bot \rangle = \bot$, etc.[8]). The reduction $\to_{\mathsf{p}}$ is the contextual closure of the following base case:

$$(\lambda \vec{x}.p)\vec{q} \mapsto_{\mathsf{p}} \begin{cases} p\{\vec{q}/\vec{x}\} & \text{if } \mathsf{len}(\vec{q}) = \mathsf{len}(\vec{x}) \\ \bot & \text{otherwise.} \end{cases}$$

Since we want to link a calculus of approximants to intersection type distributors, the first thing to check is that the calculus satisfies subject reduction and expansion within our system

[8] This is slightly different from the original definition of [41], but being up to linearity simplify calculations.

$\mathcal{R}_{\to}$. Let $\zeta = \langle \vec{x}_1, \ldots, \vec{x}_n \rangle$ and $\Delta = \langle \vec{a}_1, \ldots, \vec{a}_n \rangle$. We write $\zeta : \Delta$ for $\vec{x}_1 : \vec{a}_1, \ldots, \vec{x}_n : \vec{a}_n$. We give the following naive type assignment:

$$\frac{f : a' \to a}{\langle\rangle : \langle\rangle, \ldots, \langle x \rangle : \langle a' \rangle, \ldots, \langle\rangle : \langle\rangle \vdash x : a} \qquad \frac{(\zeta_i : \Gamma_i \vdash q_i)_{i=1}^k \qquad \eta : \Delta \to \bigotimes_{i=1}^k \Gamma_i}{[\eta](\bigotimes_{i=1}^k \zeta_i) : \Delta \vdash \langle q_1, \ldots, q_k \rangle : \langle a_1, \ldots, a_k \rangle}$$

$$\frac{\zeta \oplus \langle \vec{x} \rangle : \Delta \oplus \langle \vec{a} \rangle \vdash p : a}{\zeta : \Delta \vdash \lambda \vec{x}.p : \vec{a} \Rightarrow a} \qquad \frac{\zeta_0 : \Gamma_0 \vdash p : \vec{a} \Rightarrow a \qquad \zeta_1 : \Gamma_1 \vdash q \qquad \eta : \Delta \to \Gamma_0 \otimes \Gamma_1}{[\eta](\zeta_0 \otimes \zeta_1) : \Delta \vdash pq : a}$$

where in the application case the left action $[\eta]\zeta$ means only that the positions of variables in ζ are rearranged in accordance with the permutation induced by the morphism η. This is reasonable and necessary, since the morphism η can in general rearrange the position of types. This means that if $\zeta = \langle \vec{x}_1, \ldots, \vec{x}_n \rangle$ and $\eta = \langle \langle \sigma_1, \vec{f_1} \rangle, \ldots, \langle \sigma_n, \vec{f_n} \rangle \rangle$ then $[\eta]\zeta = \langle [\sigma_1]\vec{x}_1, \ldots, [\sigma_n]\vec{x}_n \rangle$ where $[\sigma]\langle x_1, \ldots, x_k \rangle = \langle x_{\sigma(1)}, \ldots, x_{\sigma(k)} \rangle$ is just the left action of the symmetry group. It is easy to see that $\perp$ is not typable in the type system above.

▶ **Example 21.** We present a counter-example for the subject reduction of the former system. Take the polyadic term $p = (\lambda x.x\langle \lambda\langle\rangle.y_1\langle\rangle, \lambda\langle f \rangle.y_2\langle f \rangle \rangle)\langle \lambda\langle z_1, z_2 \rangle.z_1\langle z_2 \rangle \rangle$. This term clearly reduces to $\perp$, but it is typable in the former type system. Let $\pi =$

$$\dfrac{\dfrac{\dfrac{g : b' \to b}{\langle x \rangle : \langle b' \rangle, \langle\rangle \vdash x : b} \qquad \langle\rangle : \langle\rangle, \langle y_1 \rangle : \langle \langle\rangle \Rightarrow a \rangle \vdash \lambda\langle\rangle.y_1\langle\rangle : \langle\rangle \Rightarrow a \qquad \langle\rangle : \langle\rangle, \langle y_1 \rangle : \langle \langle c \rangle \Rightarrow a \rangle \vdash \lambda\langle f \rangle.y_1\langle f \rangle : \langle c \rangle \Rightarrow a}{\langle x \rangle : \langle b' \rangle, \langle y_1, y_2 \rangle : \langle \langle\rangle \Rightarrow a, \langle c \rangle \Rightarrow a \rangle \vdash x\langle \lambda\langle\rangle.y_1\langle\rangle, \lambda\langle f \rangle.y_2\langle f \rangle \rangle : a}}{\langle y_1, y_2 \rangle : \langle \langle\rangle \Rightarrow a, \langle c \rangle \Rightarrow a \rangle \vdash \lambda\langle x \rangle.x\langle \lambda\langle\rangle.y_1\langle\rangle, \lambda\langle f \rangle.y_2\langle f \rangle \rangle : \langle b' \rangle \Rightarrow a}$$

Where $c = \langle\rangle \Rightarrow a$ and $b' = \langle \langle c \rangle \Rightarrow a, \langle\rangle \Rightarrow a \rangle \Rightarrow a$ and $b = \langle \langle\rangle \Rightarrow a, \langle c \rangle \Rightarrow a \rangle \Rightarrow a$ the morphism g being of the shape $\langle \sigma, 1_{\langle\rangle \Rightarrow a}, 1_{\langle a \rangle \Rightarrow a} \rangle \Rightarrow 1$ with sigma being the obvious permutation. Consider $\rho =$

$$\dfrac{\dfrac{\langle z_1 \rangle : \langle \langle c \rangle \Rightarrow a \rangle \vdash z_1 : \langle c \rangle \Rightarrow a \qquad \langle z_2 \rangle : \langle c \rangle \vdash z_2 : c}{\langle z_1, z_2 \rangle : \langle \langle c \rangle \Rightarrow a, c \rangle \vdash z_1\langle z_2 \rangle : a}}{\vdash \lambda\langle z_1, z_2 \rangle.z_1\langle z_2 \rangle : \langle \langle c \rangle \Rightarrow a, c \rangle \Rightarrow a}$$

Now take $\pi' =$

$$\dfrac{\begin{array}{c}\pi \\ \vdots \\ \langle y_1, y_2 \rangle : \langle \langle\rangle \Rightarrow a, \langle a \rangle \Rightarrow a \rangle \vdash \lambda\langle x \rangle.x\langle \lambda\langle\rangle.y_1\langle\rangle, \lambda\langle f \rangle.y_2\langle f \rangle \rangle : \langle b' \rangle \Rightarrow a\end{array} \qquad \begin{array}{c}\rho \\ \vdots \\ \vdash \lambda\langle z_1, z_2 \rangle.z_1\langle z_2 \rangle : b'\end{array}}{\langle y_1, y_2 \rangle : \langle \langle\rangle \Rightarrow a, \langle a \rangle \Rightarrow a \rangle \vdash p : a}$$

The term p reduces to $\perp$. Indeed,

$$\begin{aligned} p &= (\lambda x.x\langle \lambda\langle\rangle.y_1\langle\rangle, \lambda\langle f \rangle.y_2\langle f \rangle \rangle)\langle \lambda\langle z_1, z_2 \rangle.z_1\langle z_2 \rangle \rangle \\ &\to_{\mathsf{p}} (\lambda\langle z_1, z_2 \rangle.z_1\langle z_2 \rangle)\langle \lambda\langle\rangle.y_1\langle\rangle, \lambda\langle f \rangle.y_2\langle f \rangle \rangle \\ &\to_{\mathsf{p}} (\lambda\langle\rangle.y_1\langle\rangle)\langle \lambda\langle f \rangle.y_2\langle f \rangle \rangle \ \to_{\mathsf{p}} \ \perp \end{aligned}$$

Therefore, $p \to_{\mathsf{p}}^* \perp$ and p is typable, while $\perp$ it is not. The problem relies completely in the variable rule: the subtyping feature of the system is not detected by the syntax of the standard polyadic calculus. If we want to find an appropriate term language for our system, whose elements are also approximants of ordinary bang terms, we need to take seriously the qualitative information produced by the subtyping.