# Coherence by Normalization for Linear Multicategorical Structures

**Federico Olimpieri**
School of Mathematics, University of Leeds, UK

## Abstract

We establish a formal correspondence between resource calculi and appropriate linear multicategories. We consider the cases of (symmetric) representable, symmetric closed and autonomous multicategories. For all these structures, we prove that morphisms of the corresponding free constructions can be presented by means of typed resource terms, up to a reduction relation and a structural equivalence. Thanks to the linearity of the calculi, we can prove strong normalization of the reduction by combinatorial methods, defining appropriate decreasing measures. From this, we achieve a general coherence result: morphisms that live in the free multicategorical structures are the same whenever the normal forms of the associated terms are equal. As further application, we obtain syntactic proofs of Mac Lane's coherence theorems for (symmetric) monoidal categories.

**2012 ACM Subject Classification** Theory of computation → Type theory; Theory of computation → Categorical semantics

**Keywords and phrases** Coherence, Linear Multicategories, Resource Calculi, Normalization

**Digital Object Identifier** 10.4230/LIPIcs.CSL.2024.43

**Related Version** *Full Version*: https://arxiv.org/abs/2302.05755

**Funding** *Federico Olimpieri*: This work was supported by the US Air Force Office for Scientific Research under award number FA9550-21-1-0007.

## 1 Introduction

The basis of the celebrated *Curry-Howard-Lambek correspondence* is that logical systems, typed $\lambda$-calculi and appropriate categorical constructions are different presentations of the same mathematical structure. An important consequence of the correspondence is that we can give *syntactical presentations* of categories, that can be exploited to prove general results by means of elementary methods, such as induction. At the same time, we can use categorical methods to obtain a more modular and clean design of programming languages. The classic example is given by simply typed $\lambda$-calculi and cartesian closed categories [22]. The idea is well-known: morphisms in free cartesian closed categories over sets are identified with equivalence classes of $\lambda$-terms up to $\beta\eta$-equality. Another important setting is the *linear* one, where we consider *monoidal* categories instead of cartesian ones. In this case, *linear logic* [11] enters the scene: symmetric monoidal closed categories correspond to *linear* $\lambda$-calculi. Computationally, this is a huge restriction, since linear terms can neither copy nor delete their inputs during computation. A refinement of this picture can be obtained by switching from categories to *multicategories* [21]. These structures were indeed first introduced by Lambek to achieve a categorical framework formally closer to typed calculi/proof systems. Morphisms of multicategories can have multiple sources $f : a_1, \dots, a_n \to a$, recalling the structure of a *type judgment* $x_1 : a_1, \dots, x_n : a_n \vdash f : a$.

We are interested in establishing a Curry-Howard-Lambek style correspondence for appropriate linear multicategories and then employ it to obtain *coherence results.* When we deal with complex structures such as tensor products, it becomes crucial to have a *decision process* to establish whether two arrows are equal. This is called a *coherence* problem. The


32nd EACSL Annual Conference on Computer Science Logic (CSL 2024).
Editors: Aniello Murano and Alexandra Silva; Article No. 43; pp. 43:1–43:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

main example is Mac Lane's original result [24], which states that *all structural diagrams* in monoidal categories commute. If one considers more complex structures, the class of commutative diagrams is normally more restrictive. In the case of closed monoidal categories, Kelly and Mac Lane [17] associated *graphs* to structural morphisms, obtaining the following coherence result: two structural arrows between appropriate objects[1] are equal whenever their graph is the same. We aim to achieve coherence results for linear multicategories, building on Lambek's and Mints [28] intuition that coherence problems can be rephrased in the language of proof theory and obtained by exploiting appropriate notions of *normalization* for proofs/terms [21]. We do so by establishing a formal connection between *resource* calculi and *linear* multicategorical structures.

**Main Results.** We study free multicategorical constructions for (symmetric) representable and closed structures. Representability consists of the multicategorical monoidal structure [12]. We prove that free linear multicategories built on appropriate signatures can be presented by means of typed resource calculi, where morphisms correspond to equivalence classes of terms up to a certain equivalence. We handle the tensor product *via pattern-matching*, presented as a syntactic *explicit substitution.* The definition of our type systems is given in *natural deduction* style: we have introduction and elimination rules for each type constructor. Our work is conceptually inspired by an "*adjoint functors* point-of-view". A basic fact of the classic Curry-Howard-Lambek correspondence is that $\beta\eta$-equality can be expressed by means of the *unit* ($\eta$) and the *counit* ($\beta$) of the adjunction between products and arrow types. We generalize this observation to the multicategorical setting, thus introducing an appropriate reduction relation that corresponds to the representable structure. Indeed, a fundamental aspect of our work consists of the in depth study of resource terms rewriting. We introduce confluent and strongly normalizing reductions, that express the appropriate equalities. In order to do so, we exploit *action-at-distance* to define our operational semantics, that has proven to be a successful approach to calculi with explicit substitution [18, 1, 2]. An important feature of this approach is to distinguish the operational semantics, defined by action-at-distance, from a notion of *structural equivalence*, that deals with commutations of explicit substitution with the other syntactic constructors. This approach overcomes the classic difficulties of rewriting systems with explicit substitution, allowing us to obtain confluence and strong normalization in an elegant way. In this way, we get a general *coherence result*: two structural morphisms of linear multicategories are equal whenever the *normal forms* of their associated terms are equal. In the context of (symmetric) representable multicategories, we apply this result to obtain a *syntactic proof* of stronger coherence theorems, that can be seen as multicategorical versions of the classic MacLane coherence theorems for (symmetric) monoidal categories [24]. The coherence theorem for representable multicategories was already proved in [12]. We give an alternative type-theoretic proof for it. To our knowledge, the other coherence results that we present are new. Moreover, exploiting the equivalence between monoidal categories and representable multicategories established by Hermida [12], we are able to obtain the original Mac Lane's results as corollaries of our coherence theorems.

**Related Work.** Building on Lambek's original ideas, several researchers have advocated the use of multicategories to model computational structures. Hyland [14] proposed to rebuild the theory of pure $\lambda$-calculus by means of *cartesian operads*, that is one-object cartesian multicategories. The idea of seeing resource calculi as multicategories was first

[1] A restriction on the type of morphisms is needed due to the presence of the monoidal unit.

employed by Mazza *et al.* [26, 25]. We build on their approach, showing that these calculi correspond to appropriate *universal constructions*, namely free linear multicategories. The first resource calculus has been introduced by Boudol [6]. A similar construction was also independently considered by Kfoury [19]. Resource terms have gained special interest thanks to the definition by Ehrhard and Regnier of the *Taylor expansion* for $\lambda$-terms [9]. From this perspective, the resource calculus is a *theory of approximation of programs* and has been successfully exploited to study the computational properties of $\lambda$-terms [3, 35, 26, 30]. Our syntax is very close to the one of *polyadic calculi* or *rigid resource calculi* [26, 34]. We need to extend the standard operational semantics, adding an $\eta$-reduction and a reduction for explicit substitution. Our $\eta$-reduction is built from an expansion rule instead of a contraction, since $\eta$-expansion naturally fits the adjoint point-of-view, corresponding the the unit of the considered adjunction. In dealing with the technical rewriting issues, we follow [28, 16, 8], obtaining a terminating $\eta$-reduction. As already discussed, we handle the explicit substitution following Accattoli and Kesner methodology [18, 2, 1].

The calculi we present are also strongly related to *intuitionistic linear logic* [20, 4]. It is well-known that resource calculi can be seen as fragments of ILL [26, 25]. While ILL is presented *via* sequent calculus, we chose a natural deduction setting, this latter being directly connected to the "adjoint functors" point-of-view. Accattoli and Kesner approach to explicit substitution allows us to bypass the cumbersome commutation rules needed for ILL rewriting. Moreover, resource calculi are closer to the multicategorical definitions (their constructors being *unbiased* [23], *i.e.*, $k$-ary). Our handling of symmetries is also more canonical and explicit. We use the properties of *shuffle permutations*, in a way similar to Hasegawa [29] and Shulman [33], also inspired by our ongoing work on *bicategorical semantics* [27]. In this way, the type system is *syntax directed* and we are able to prove that, given a term, there exists at most one type derivation for it. The pioneering work of Mints [28] is very close to our perspective. Mints introduced a linear $\lambda$-calculus to study the coherence problem of closed category by the means of normalization. We build on that approach, extending it to several different structures and to the multicategorical setting.

Shulman's type theory for (symmetric) monoidal categories [33] does not employ explicit substitutions, being able to handle tensors in way similar to what happens with standard product types. Our proposal differs considerably from Shulman's, both in purpose and in implementation. While Shulman's goal is to start from the categorical structure and define a "practical" type theory to make computations, ours consists of establishing a formal correspondence between two *independent* worlds: resource calculi and linear multicategories and then employ it to prove results about the categorical structure.

Proof-theoretic methods to establish coherence results have been widely exploited and studied also in recent times, see for instance [7, 36]. *Graphical approaches* to monoidal structures [31] have been widely developed. Particularly interesting for our work are the Kelly-Mac Lane graphs [17], This approach has been extended *via* linear logic, thanks to the notion of *proof-net* [5, 13]. However, the handling of monoidal units needs extra care from this perspective, while the terms calculi approach can account for them without any particular complication.

## 2 Preliminaries

We introduce some concepts, notations and conventions that we will use in the rest of the paper.

**Integers, Permutations and Lists.** For $n \in \mathbb{N}$, we set $[n] = \{1, \ldots, n\}$ and we denote by $S_n$ the symmetric group of order $n$. The elements of $S_n$ are permutations, that we identify with bijections $[n] \cong [n]$. Given $\sigma, \tau \in S_n$, we denote by $\sigma \circ \tau$ their composition. Given $\sigma \in S_n, \tau \in S_m$ we denote by $\sigma \oplus \tau : [n+m] \cong [n+m]$ the evident induced permutation. We now introduce the notion of shuffle permutation, that is crucial to obtain canonical type derivations for resource terms with permutations (Proposition 32).

▶ **Definition 1** (Shuffles). *Let $n_1, \ldots, n_k \in \mathbb{N}$ with $n = \sum_{i=1}^{k} n_i$. A $(n_1, \ldots, n_k)$-*shuffle *is a bijection $\sigma : \sum_{i=1}^{k} [n_i] \cong [n]$ such that the composite $[n_i] \hookrightarrow \sum_{i=1}^{k} [n_i] \cong [n]$ is monotone for all $i \in [k]$. We denote the set of all $(n_1, \ldots, n_k)$-shuffles as $\mathsf{shu}(n_1, \ldots, n_k)$.*

The relevant result on shuffles is the following, that induces canonical decomposition of arbitrary permutations over sums of integers.

▶ **Lemma 2.** *Every permutation $\sigma \in S_{\sum_{i=1}^{k} n_i}$ can be canonically decomposed as $\tau_0 \circ (\bigoplus_{i=1}^{k} \tau_i)$ with $\tau_0 \in \mathsf{shu}(n_1, \ldots, n_k)$ and $\tau_i \in S_{n_i}$ for $i \in [k]$.*

Given a set $A$ and a list of its elements $\gamma = a_1, \ldots, a_k$ and $\sigma \in S_k$ we set $\gamma \cdot \sigma = a_{\sigma(1)}, \ldots, a_{\sigma(k)}$ for the symmetric group right action. We write $\mathsf{len}(\gamma)$ for its length. We denote the *stabilisers* for this action as $\mathsf{Stab}(\gamma) = \{\sigma \in S_k \mid \gamma \cdot \sigma = \gamma\}$. Given lists $\gamma_1, \ldots, \gamma_k$, we set $\mathsf{shu}(\gamma_1, \ldots, \gamma_k) = \mathsf{shu}(\mathsf{len}(\gamma_1), \ldots, \mathsf{len}(\gamma_k))$.

**Multicategories.** Multicategories constitute the main object of our work. A multicategory is a multigraph that comes equipped with an appropriate composition operation.

▶ **Definition 3.** *A* multigraph *$\mathcal{G}$ is given by the following data:*

- *A collection of* nodes *$\mathcal{G}_0 \ni a, b, c \ldots$*
- *For every $a_1, \ldots, a_n, b \in \mathcal{G}_0$, a collection of* multiarrows *$\mathcal{G}(a_1, \ldots, a_n; b) \ni s, t, u \ldots$*

*We denote by $arr(\mathcal{G})$ the set of all multiarrows of $\mathcal{G}$.*

▶ **Definition 4.** *A* multicategory *is a multigraph $\mathcal{G}$ equipped with the following additional structure:*

- *A* composition operation *$- \circ \langle -, \ldots, - \rangle : \mathcal{G}(a_1, \ldots, a_n; b) \times \prod_{i=1}^{n} \mathcal{G}(\gamma_i, a_i) \to \mathcal{G}(\gamma_1, \ldots, \gamma_n; a)$.*
- identities *$id_a \in \mathcal{G}(a, a)$.*

*The former data is subjected to evident associativity and identity axioms. We call* objects *the nodes of $\mathcal{G}$ and* morphisms *its multiarrows.*

A multicategory can be equipped with structure. We now introduce the notions of *symmetric*, *closed* and *representable* multicategories.

▶ **Definition 5.** *A multicategory $\mathcal{M}$ is* symmetric *if, for $\sigma \in S_k$ we have a family of bijections $- \cdot \sigma : \mathcal{M}(\gamma, a_1, \ldots, a_k; a) \cong \mathcal{M}(\gamma, a_{\sigma(1)}, \ldots, a_{\sigma(k)}; a)$ that satisfies additional axioms [23].*

▶ **Definition 6.** *A (right)* closed *structure for a multicategory $\mathcal{M}$ is given by a family of objects $(a_1 \otimes \cdots \otimes a_k) \multimap a \in \mathcal{M}$ and arrows $ev_{a_1, \ldots, a_k, a} : a_1, \ldots, a_k, (a_1 \otimes \cdots \otimes a_k) \multimap a \to a$ , for $a_1, \ldots, a_k, a \in \mathcal{M}$, such that the maps*

$$ev \circ \langle -, id_{a_1}, \ldots, id_{a_k} \rangle : \mathcal{M}(\gamma; (a_1 \otimes \cdots \otimes a_k) \multimap a) \to \mathcal{M}(\gamma, a_1, \ldots, a_k; a)$$

*induce a bijection, multinatural in $\gamma$ and natural in $a$. We write $\lambda(-)$ to denote the inverses to these maps.*

▶ **Definition 7.** *A* representable *structure for a multicategory $\mathcal{M}$ is given by a family of objects $(a_1 \otimes \cdots \otimes a_k) \in \mathcal{M}$ and arrows $\mathsf{re}_{a_1,\ldots,a_k} : a_1, \ldots, a_k \to (a_1 \otimes \cdots \otimes a_k)$, for $a_1, \ldots, a_k \in \mathcal{M}$, such that he maps*

$$- \circ \langle id_\gamma, \mathsf{re}, id_\delta \rangle : \mathcal{M}(\gamma, (a_1 \otimes \cdots \otimes a_k), \delta; a) \to \mathcal{M}(\gamma, a_1, \ldots, a_k, \delta; a)$$

*induce a bijection, multinatural in $\gamma, \delta$ and natural in $a$. We write $\mathsf{let}(-)$ to denote the inverses to these maps.*

We use the name *autonomous multicategories* to denote symmetric representable closed multicategories. We have categories of representable multicategories ($\mathsf{RepM}$), symmetric representable multicategories ($\mathsf{RepsM}$), closed multicategories ($\mathsf{ClosedM}$) and autonomous multicategories ($\mathsf{autoM}$), whose morphisms are functors that preserve the structure on the nose.

**Signatures.** We introduce signatures for the structures we consider.

▶ **Definition 8.** *A* representable signature *is a pair $\langle \mathsf{At}, \mathcal{R} \rangle$ where $\mathsf{At}$ is a set of atoms $\mathsf{At}$ and $\mathcal{R}$ is a multigraph with nodes generated by the following inductive grammar:*

$$\mathcal{R}_0 \ni a ::= o \in \mathsf{At} \mid (a_1 \otimes \cdots \otimes a_k) \qquad (k \in \mathbb{N}).$$

▶ **Definition 9.** *A* closed signature $\mathcal{L}$ *is a pair $\langle \mathsf{At}, \mathcal{L} \rangle$ where $\mathsf{At}$ is a set of atoms $\mathsf{At}$ and $\mathcal{L}$ is a multigraph with with nodes generated by the following inductive grammar:*

$$\mathcal{L}_0 \ni a ::= o \in \mathsf{At} \mid (a_1 \otimes \cdots \otimes a_k) \multimap a \qquad (k \in \mathbb{N}).$$

▶ **Definition 10.** *An* autonomous signature *is a pair $\langle \mathsf{At}, \mathcal{H} \rangle$ where $\mathsf{At}$ is a set of atoms $\mathsf{At}$ and $\mathcal{H}$ is a multigraph with nodes generated by the following inductive grammar:*

$$\mathcal{H}_0 \ni a ::= o \in \mathsf{At} \mid (a_1 \otimes \cdots \otimes a_k) \mid (a_1 \otimes \cdots \otimes a_k) \multimap a \qquad (k \in \mathbb{N}).$$

A signature is *discrete* whenever the collections of multiarrows are empty. We shall often identify a signature with its graph. There are categories $\mathsf{ClosedSig}, \mathsf{RepSig}$ and $\mathsf{AutoSig}$ for, respectively, closed, representable and autonomous signatures. We have forgetful functors from the categories $\mathsf{ClosedM}, \mathsf{RepM}$ and $\mathsf{autoM}$, which we denote by $(\overline{-})$. One of the main goals of this paper is to build the left adjoints to those functors *via* appropriate resource calculi.

**Monoidal Categories vs Representable Multicategories.** In order to transport coherence results from (symmetric) representable multicategories to ordinary (symmetric) monoidal categories, we shall employ an equivalence result due to Hermida [12, Theorem 9.8]. Let $\mathsf{Mon}$ be the category of monoidal categories and lax monoidal functors.

▶ **Theorem 11** ([12]). *There is an equivalence of categories* $\mathsf{RepM} \underset{\mathsf{mon}(-)}{\overset{\mathsf{rep}(-)}{\rightleftarrows}} \mathsf{Mon}$, with $\simeq$ between the arrows.

The representable structure of a monoidal category $(\mathbb{M}, \otimes_{\mathbb{M}}, 1)$ is given by $(a_1 \otimes_{\mathbb{M}} \cdots \otimes_{\mathbb{M}} a_k) = (a_1) \otimes_{\mathbb{M}} (a_2 \otimes_{\mathbb{M}} (\cdots \otimes_{\mathbb{M}} a_k) \ldots)$. Then composition needs a choice of structural isomorphisms of $\mathbb{M}$ to be properly defined [12, Definition 9.2][2]. The former equivalence can be extended to the symmetric case in the natural way.

[2] If we assume Mac Lane's Coherence Theorem, the choice is unique. However, we shall not do so, since we are going to exploit Theorem 11 to *transport* an appropriate coherence theorem on representable multicategories to ordinary monoidal categories, thus obtaining the Mac Lane's result as corollary.

$$\frac{f \in \mathcal{R}(a_1, \ldots, a_n; b) \qquad \gamma_1 \vdash s_1 : a_1 \ldots \gamma_n \vdash s_n : a_n}{(\gamma_1, \ldots, \gamma_n) \vdash f(s_1, \ldots, s_n) : b} \qquad \frac{\gamma_1 \vdash s_1 : a_1 \ldots \gamma_k \vdash s_k : a_k}{\gamma_1, \ldots, \gamma_k \vdash \langle s_1, \ldots, s_k \rangle : (a_1 \otimes \cdots \otimes a_k)}$$

$$\frac{a \in \mathcal{R}_0}{x : a \vdash x : a} \qquad \frac{\gamma \vdash s : (a_1 \otimes \cdots \otimes a_k) \qquad \delta, x_1 : a_1, \ldots, x_k : a_k, \delta' \vdash t : b}{\delta, \gamma, \delta' \vdash t[x_1^{a_1}, \ldots, x_k^{a_k} := s] : b}$$

$$\mathtt{C} ::= [\cdot] \mid \langle s_1, \ldots, \mathtt{C}, \ldots, s_k \rangle \mid \mathtt{C}[\vec{x} := t] \mid s[\vec{x} := \mathtt{C}] \mid f(s_1, \ldots, \mathtt{C}, \ldots, s_k).$$
$$\mathtt{E} ::= [\cdot] \mid \langle s_1, \ldots, \mathtt{E}, \ldots, s_k \rangle \mid \mathtt{E}[\vec{x} := s] \mid s[\vec{x} := \mathtt{E}] \quad (\mathtt{E} \neq [\cdot]) \mid f(s_1, \ldots, \mathtt{E}, \ldots, s_k).$$
$$\mathtt{L} ::= [\cdot] \mid \mathtt{L}[\vec{x} := t].$$

**Figure 1** Representable Type System on a signature $\mathcal{R}$ and contexts with one hole. Types are the elements of $\mathcal{R}_0$.

**Notations and Conventions.** Given a set of terms $A$ and a reduction relation $\rightarrow_\epsilon \subseteq A \times A$, we denote respectively as $\twoheadrightarrow_\epsilon$ and $\rightarrow_\epsilon^*$ its transitive closure and its transitive and reflexive closure. We denote by $=_\epsilon \subseteq A \times A$ the smallest equivalence relation generated by $\rightarrow_\epsilon$. For a confluent reduction, we denote by $\mathtt{nf}(s)_\epsilon$ the normal form of $s$, if it exists. Given an equivalence relation $\mathtt{e} \subseteq A \times A$, and $s \in A$, we denote by $[s]_\mathtt{e}$ the corresponding equivalence class. We will often drop the annotation and just write $[s]$. We fix a countable set of variables $\mathcal{V}$, that we will use to define each calculi. Terms are always considered up to renaming of bound variables. Given terms $s, t_1, \ldots, t_k$ and variables $x_1, \ldots, x_k$ we write $s\{t_1, \ldots, t_k/x_1, \ldots, x_k\}$ to denote capture-avoiding substitutions. We often use the abbreviation $s\{\vec{t}/\vec{x}\}$. To define reduction relations, we rely on appropriate notions of *contexts with one hole*. Given a context with hole $\mathtt{C}$ and a term $s$ we write $\mathtt{C}[s]$ for the capture-allowing substitution of the holes of $\mathtt{C}$ by $s$. The *size* of a term $\mathtt{size}(s)$ is the number of syntactic constructors appearing in its body. The calculi we shall introduce are typed *à la Church*, but we will constantly keep the typing implicit, to improve readability. Given $\gamma \vdash s : a$ we write $\mathtt{C}[\delta \vdash p : b] = s$ meaning that $\mathtt{C}[p] = s$ and the type derivation of $\gamma \vdash p : b$ contains a subderivation with conclusion $\delta \vdash p : b$. Given a typing judgment $x_1 : a_1, \ldots, x_n : a_n \vdash s : a$ we shall consider variables appearing in the typing context as bound and we will work up to renaming of those variables. We write $\pi \triangleright \gamma \vdash s : a$ meaning that $\pi$ is a type derivation of conclusion $\gamma \vdash s : a$. For any typing rule with multiple typing contexts, we assume those contexts to be disjoint.

## 3 A Resource Calculus for Representable Multicategories

We present our calculus for representable multicategories. We begin by introducing its syntax and typing, then we discuss its operational semantics. We prove confluence and strong normalization for its reduction. We show that equivalence classes of terms modulo reduction and a notion of *structural equivalence* define the morphisms of free representable multicategories over a signature. As an application of this result, we give a proof of the coherence theorem for representable multicategories.

**Representable Terms.** Let $\mathcal{R}$ be a representable signature. The *representable resource terms* over $\mathcal{R}$ are defined by the following inductive grammar:

$$\Lambda_{\mathsf{rep}}(\mathcal{R}) \ni s, t ::= x \in \mathcal{V} \mid \langle s_1, \ldots, s_k \rangle \mid s[x_1^{a_1}, \ldots, x_k^{a_k} := t] \mid f(s_1, \ldots, s_k)$$

for $k \in \mathbb{N}$ and $f \in \mathrm{arr}(\mathcal{R}), a_i \in \mathcal{R}$. A term of the shape $\langle s_1, \dots, s_k \rangle$ is called a *list*. A term of the shape $s[x_1, \dots, x_k := t]$ is called an *(explicit) substitution*. Variables under the scope of an explicit substitution are bound. Given a term $s$, we denote by $\mathsf{ST}(s)$ the set of its *subterms* defined in the natural way.

▶ Remark 12. Our calculus follows the linear logic tradition of modelling the tensor product structure by means of a $\mathsf{let}$ constructor [4]. We opted for the syntactic choice of an explicit substitution $s[x_1, \dots, x_k := t]$, which stands for the more verbose $\mathsf{let}$ expression, $\mathsf{let}\ \langle x_{,1}, \dots, x_k \rangle := t\ \mathsf{in}\ s$. Terms of the shape $f(s_1, \dots, s_k)$ are needed to capture the multiarrows induced by the signature $\mathcal{R}$.

Typing and contexts with hole for representable terms is defined in Figure 1. A context is *atomic* when it contains just atomic types. We define the following subset of terms $\mathsf{LT} = \{\mathsf{L}[\langle s_1, \dots, s_k \rangle] \mid$ for some context $\mathsf{L}$ and terms $s_i\}$.

▶ Remark 13. The condition about disjoint contexts grants *linearity*. A term is *linear* when each variable appears at most once in its body. It is easy to check that, by construction, all typed terms are linear. Moreover, given $\gamma \vdash s : a$, the context $\gamma$ is *relevant*, meaning that it contains just the free variables of $s$.

A type of the shape $(a_1 \otimes \cdots \otimes a_k)$ is called a $k$-ary *tensor product*. We use a vector notation to refer to arbitrary tensors, *eg.*, $\vec{a}, \vec{b} \dots$ If $k = 0$, the type $()$ is also called the *unit*. We set $\Lambda_{\mathsf{rep}}(\mathcal{R})(a_1, \dots, a_n; a) = \{s \mid x_1 : a_1, \dots, x_n : a_n \vdash_{\mathsf{rep}} s : a$ for some $x_i \in \mathsf{fv}(s).\}$. We observe that, given a representable term $\gamma \vdash s : a$, there exists a unique type derivation for it.

**Terms Under Reduction.** We now introduce the reduction relation for representable terms. This relation consists of the union of two different subreductions: $\beta$ and $\eta$ reductions, defined in Figure 2. The *structural equivalence* on terms is defined as the smallest congruence on terms generated by the rule of Figure 2. We assume that the context $\mathtt{C}$ does not bind any variable of $t$.

▶ Remark 14. Our $\eta$-reduction consists of a restricted version of the standard notion of $\eta$-expansion, the restriction is needed to achieve strong normalization. We build on a well-established tradition in term rewriting [28, 16, 8]. Unrestricted $\eta$ is trivially non-terminating. Indeed, for $x : (a \otimes b) \vdash x : (a \otimes b)$ we have the non-terminating chain $x \to_\eta \langle x, y \rangle[x, y := z] \to_\eta \langle x, y \rangle[x, y := \langle v, w \rangle[v, w := z]] \to_\eta \dots$ Hence, we need to forbid $\eta$-reduction on the right side of a substitution term, that is exactly what the restricted $\eta$-contexts do. Moreover, there is also a problem of interaction between $\eta$ and $\beta$. Consider $s = \langle x, y \rangle$ well-typed, then we can produce the non-terminating chain $s \to_\eta \langle v, w \rangle[v, w := \langle x, y \rangle] \to_\beta s \to_\eta \dots$ For this reason, the root-step of $\eta$ has to be restricted too. The presence of a substitution context in the $\beta$-rule is an action-at-distance [2], that allows to "free" possible blocked redexes, as the following one $x[x := (\langle y \rangle[y := z])]$. In this way, we can bypass traditional commutation rules and retrieve good rewriting properties. Structural equivalence intuitively says that explicit substitutions can "freely travel" in the body of a term.

We prove that typings are preserved under reduction and structural equivalence.

▶ **Proposition 15** (Subject Reduction and Equivalence). *Let $s \to_{\mathsf{rep}} s'$ or $s \equiv s'$ with $\gamma \vdash s : a$. then $\gamma \vdash s' : a$.*

**Proof.** The proof is by induction on $s \to_{\mathsf{rep}} s'$ and $s \equiv s'$ and exploits an appropriate substitution lemma. ◀

We now prove that the structural equivalence is a strong bisimulation for the reduction $\to_{\mathsf{rep}}$ . Intuitively, this means that the equivalence does not affect terms rewriting.

▶ **Proposition 16.** *If $s' \equiv s$ and $s \to_{\mathsf{rep}} t$ there exists a term $t'$ s.t. $t' \equiv t$ and $s' \to_{\mathsf{rep}} t'$.*

We show that we can associate appropriate measures to terms that decrease under reduction. For $\beta$, we just consider the size of terms. For $\eta$, we build on Mints approach [28]. We define the *size* of a type by induction: $\mathtt{size}\,(o) = 0, \mathtt{size}\,(\langle a_1, \ldots, a_k\rangle) = 1 + \sum \mathtt{size}\,(a_i)$. Given $\gamma \vdash s : a$ we define a set of typed subterms of $s$: $\mathsf{EST}(s) = \{\delta \vdash p : a \mid p \in \mathsf{ST}(s) \setminus \mathsf{LT}$ s.t. $\mathtt{E}[\delta \vdash p : a] = s$ for some context $\mathtt{E}\}$. We set $\eta(s) = \sum_{\delta \vdash p:a \in \mathsf{EST}(s)} \mathtt{size}\,(a)$.

▶ Remark 17. The size of terms decreases under $\beta$-reduction as a consequence of *linearity*. Redexes cannot be copied nor deleted under reduction, since the substitution is linear. This fact is trivially false for standard $\lambda$-calculi, where the size of terms can possibly grow during computation. The intuition behind the $\eta$ measure is that we are counting all subterms of $s$ on which we could perform the $\eta$-reduction. The restrictions on the shape of $p \in \mathsf{EST}(s)$ is indeed directly derived from the ones on $\eta$-reduction.

▶ **Proposition 18.** *The following statements hold. If $s \to_\beta s'$ then $\mathtt{size}\,(s') < \mathtt{size}\,(s)$; if $s \to_\eta s'$ then $\eta(s') < \eta(s)$.*

▶ **Proposition 19.** *The reductions $\to_\beta$ and $\to_\eta$ are separately strongly normalizing and confluent.*

**Proof.** Strong normalization is a corollary of the former proposition. For confluence, first one proves local confluence by induction and then apply Newman's Lemma. ◀

We want to extend the result of separate strong normalization and confluence to the whole $\to_{\mathsf{rep}}$-reduction. To do so, we prove that $\beta$ and $\eta$ suitably *commutes*.

▶ **Proposition 20.** *If $s \to_\beta^* t \to_\eta^* t'$ there exists $s'$ s.t. $s \to_\eta^* s'$ and $s' \to_\beta^* t'$.*

▶ **Theorem 21.** *The reduction $\to_{\mathsf{rep}}$ is confluent and strongly normalizing.*

**Proof.** Strong normalization is achieved by observing that any infinite reduction chain of $\to_{\mathsf{rep}}$ would trigger, by Proposition 20, an infinite reduction chain for $\eta$, that is strongly normalizing. Confluence is achieved by first proving local confluence and then by applying Newman's Lemma. ◀

Given $s \in \Lambda_{\mathsf{reps}}(\mathcal{R})(\gamma; a)$, we denote by $\mathsf{nf}(s)$ its unique normal form. As a corollary of subject reduction, we get that $\mathsf{nf}(s) \in \Lambda_{\mathsf{reps}}(\mathcal{R})(\gamma; a)$. We shall now present an inductive characterization of $\to_{\mathsf{rep}}$-normal terms for the case where $\mathcal{R}$ is a *discrete* signature.

▶ **Definition 22.** *Consider the following set, inductively defined:*

$$\mathsf{nf}(\Lambda_{\mathsf{rep}}(\mathcal{R})) \ni s ::= v[\vec{x}_1 := x_1] \ldots [\vec{x}_n := x_n] \qquad v ::= \langle v_1, \ldots, v_k\rangle \mid x$$

*where $k, n \in \mathbb{N}, \gamma \vdash p : o$ with $o$ being an atomic type and $\delta \vdash v : a$ with $\delta$ being atomic.*

▶ **Proposition 23.** *A term $s \in \Lambda_{\mathsf{rep}}(\mathcal{R})$ is a normal form for $\to_{\mathsf{rep}}$ iff there exists $s' \in \mathsf{nf}(\Lambda_{\mathsf{rep}}(\mathcal{R}))$ s.t. $s \equiv s'$.*

**Proof.** ($\Rightarrow$) By induction on $s$. If $s = x$ then $s \in \mathsf{nf}(\Lambda_{\mathsf{rep}}(\mathcal{R}))$. If $\langle s_1, \ldots, s_k\rangle$ then $s_i$ are normal form. Then we apply the IH and get $s'_i \in \mathsf{nf}(\Lambda_{\mathsf{rep}}(\mathcal{R}))$ s.t. $s_i \equiv s'_i$. By definition $s'_i = v_i[\vec{x}_{i,1} := x_{i,1}] \ldots [\vec{x}_{i,1} := x_{i,n_i}]$. We then set $s' = \langle v_1, \ldots, v_k\rangle[\vec{x}_{1,1} := x_{1,n_1}] \ldots [\vec{x}_{k,1} := x_{k,n_k}]$. If $s = p[\vec{x} := q]$ we have that $p$ is a normal form and $q$ is a $\beta$-normal form. We reason by cases on $q$. If $q$ does not have $\eta$-redexes, we apply the IH and conclude in a way similar to the list case. If $q$ has $\eta$-redexes, since $s$ is $\beta$ normal we have that $q \notin \mathsf{LT}$. We can prove that $q = x[\vec{x}_1 := q_1] \ldots [\vec{x}_1 := q_n]$ with $q_i$ hereditarely of the same shape. Hence we conclude by pushing out all the substitutions from left to right. ◀

$$\beta \textit{ Root step: } \quad s[x_1^{a_1}, \ldots, x_k^{a_k} := \mathsf{L}[\langle t_1, \ldots, t_k \rangle]] \to_\beta \mathsf{L}[s\{t_1, \ldots, t_k / x_1, \ldots, x_k\}].$$

$$\eta \textit{ Root step: } \quad s \to_\eta \vec{x}[\vec{x}^{\vec{a}} := s] \qquad \text{where } \vec{x} \text{ fresh }, \ \gamma \vdash s : \vec{a}, \ s \notin \mathsf{LT}.$$

$$\textit{Contextual extensions: } \quad \frac{s \to_\beta s'}{\mathsf{C}[s] \to_\beta \mathsf{C}[s']} \qquad \frac{s \to_\eta s'}{\mathsf{E}[s] \to_\eta \mathsf{E}[s']} \qquad (\to_{\mathsf{rep}} = \to_\beta \cup \to_\eta).$$

$$\textit{Structural equivalence: } \quad \mathsf{C}[s[\vec{x} := t]] \equiv \mathsf{C}[s][\vec{x} := t] \qquad \vec{x} \notin \mathsf{fv}(\mathsf{C}).$$

**Figure 2** Representable reduction relations and structural equivalence.

**Free Representable Multicategories.** Let $\mathcal{R}$ be a representable signature. First, we define a multicategory $\mathsf{RM}(\mathcal{R})$ by setting $\mathrm{ob}(\mathsf{RM}(\mathcal{R})) = \mathcal{R}_0$ and $\mathsf{RM}(\mathcal{R})(\gamma; a) = \Lambda_{\mathsf{rep}}(\mathcal{R})(\gamma; a)/\sim$ where $\sim = (\equiv \cup =_{\mathsf{rep}})$. Composition is given by substitution, identities are given by variables. The operation is well-defined on equivalence classes and satisfies associativity, identity axioms. We also have that if $s \sim s'$, then $\mathsf{nf}(s) \equiv \mathsf{nf}(s')$. We denote by $\eta_{\mathcal{R}} : \mathcal{R} \to \overline{\mathsf{RM}(\mathcal{R})}$ the evident inclusion.

▶ **Proposition 24** (Representability). *We have a bijection* $\mathsf{RM}(\mathcal{R})(\gamma, (a_1 \otimes \cdots \otimes a_k), \delta; a) \cong \mathsf{RM}(\mathcal{R})(\gamma, a_1, \ldots, a_k, \delta; a)$ *multinatural in* $\gamma, \delta$ *and natural in* $a$*, induced by the map* $[s] \mapsto [s\{\langle x_1, \ldots, x_k \rangle / x\}]$.

**Proof.** Naturality follows from basic properties of substitution. Inverses are given by the maps $(-)[\vec{x} := x] : \mathsf{RM}(\mathcal{R})(\gamma, a_1, \ldots, a_k, \delta; a) \to \mathsf{RM}(\mathcal{R})(\gamma, (a_1 \otimes \cdots \otimes a_k), \delta; a)$. ◀

▶ **Definition 25.** *Let* $\mathcal{R}$ *be a representable signature and* $\mathsf{S}$ *be a representable multicategory. Let* $i : \mathcal{R} \to \overline{\mathsf{S}}$ *be a map of representable signatures. We define a family of maps* $\mathsf{RT}(i)_{\gamma,a} : \Lambda_{\mathsf{rep}}(\mathcal{R})(\gamma; a) \to \mathsf{S}(i(\gamma); i(a))$ *by induction as follows:*

$$\mathsf{RT}(i)_{a,a}(x) = id_{i(a)} \qquad \mathsf{RT}(i)_{\gamma_1,\ldots,\gamma_k,(a_1 \otimes \cdots \otimes a_k)}(\langle s_1, \ldots, s_k \rangle) = \bigotimes_{i=1}^{k} \mathsf{RT}(i)_{\gamma_i, a_i}(s_i)$$

$$\mathsf{RT}(i)_{\delta_1,\gamma,\delta_2,a}(s[x_1, \ldots, x_k := t]) = \mathsf{let}(\mathsf{RT}(i)_{\delta_1,a_1,\ldots,a_k,\delta_2,a}(s)) \circ \langle id_{\delta_1}, \mathsf{RT}(i)_{\gamma,(a_1 \otimes \cdots \otimes a_k)}(t), id_{\delta_2} \rangle$$
$$\mathsf{RT}(i)_{\gamma_1,\ldots,\gamma_n,a}(f(s_1, \ldots, s_n)) = i(f) \circ \langle \mathsf{RT}(i)(s_1), \ldots, \mathsf{RT}(i)(s_n) \rangle.$$

▶ **Theorem 26** (Free Construction). *Let* $\mathsf{S}$ *be a a representable multicategory and* $i : \mathcal{R} \to \overline{\mathsf{S}}$ *a map of representable signatures. There exists a unique representable functor* $i^* : \mathsf{RM}(\mathcal{R}) \to \mathsf{S}$ *such that* $i = \overline{i^*} \circ \eta_{\mathcal{R}}$.

**Proof.** The functor is defined exploiting Definition 25. ◀

**Coherence Result.** We fix a *discrete* representable signature $\mathcal{R}$. We show that if $s, t \in \mathsf{RM}(\mathcal{R})(\gamma; a)$, then $s = t$. Our proof strongly relies on the characterization of normal forms given in Proposition 23.

▶ **Lemma 27.** *Let* $\gamma, \gamma'$ *be atomic contexts. If there exists a type* $a$ *and normal terms* $s, s'$ *such that* $s, s' \in \Lambda_{\mathsf{rep}}(\mathcal{R})(\gamma; a)$ *then* $\gamma = \gamma'$ *and* $s \equiv s'$.

▶ **Theorem 28.** *Let* $s, s'$ *be normal terms s.t.* $s, s' \in \Lambda_{\mathsf{rep}}(\mathcal{R})(\gamma; a)$*, then* $s \equiv s'$.

**Proof.** By Proposition 23, $s \equiv t = (v[\vec{x}_1 := x_1] \ldots [\vec{x}_p := x_p])$ and $s' \equiv t' = (v'[\vec{y}_1 := x'_1] \ldots [\vec{y}_p := x'_{p'}])$. We prove that $t \equiv t'$ by induction on $p \in \mathbb{N}$. If $p = 0$ then $t$ is either a list or a variable. We proceed by cases. If $t = x$ then $\gamma = o$ and $a = o$ for some atomic type $o$. By the former lemma we have that $t \equiv t'$. If $t = \langle v_1, \ldots, v_k \rangle$ the result is again a corollary of the former lemma since, by Definition 22, $\gamma$ is atomic. If $p = n + 1$ then $t = v[\vec{x}_1 := x_1] \ldots [\vec{x}_{n+1} := x_{n+1}]$ and, by definition of typing we have

$$\frac{x_{n+1} : \vec{a} \vdash x_{n+1} : \vec{a} \qquad \delta_1, \vec{x}_{n+1} : \vec{a}, \delta_2 \vdash v[\vec{x}_1 := x_1] \ldots [\vec{x}_n := x_n] : a}{\delta_1, x_{n+1} : \vec{a}, \delta_2 \vdash s : a}$$

with $\gamma = \delta_1, x_{n+1} : \vec{a}, \delta_2$. Since $t' \in \mathsf{nf}(\Lambda_{\mathsf{rep}})(\gamma; a)$, there exists $i \in \mathbb{N}$ such that $t' = v'[\vec{y}_1 := x'_1] \ldots [\vec{y}_i := x'_i] \ldots [\vec{y}_p := x'_{p'}]$ and $x'_i = x_{n+1}$. By *structural equivalence* we have that $t' \equiv v'[\vec{y}_1 := x'_1] \ldots [\vec{y}_p := x'_{p'}] \ldots [\vec{y}_i := x_i]$. By definition of typing and by the hypothesis we have that

$$\frac{x'_i : \vec{a} \vdash x'_i : \vec{a} \qquad \delta_1, \vec{x}_i : \vec{a}, \delta_2 \vdash v'[\vec{y}_1 := x'_1] \ldots [\vec{y}_{p'} := x'_{p'}] : a}{\delta_1, x_{n+1} : \vec{a}, \delta_2 \vdash s' : a}$$

By IH we have that $v[\vec{x}_1 := x_1] \ldots [\vec{x}_p := x_n] \equiv v'[\vec{y}_1 := x'_1] \ldots [\vec{y}_p := x'_{p'}]$ . We can then conclude that $t \equiv t'$, by structural equivalence. ◀

▶ **Theorem 29** (Coherence for Representable Multicategories). *Let* $[s], [t] \in \mathsf{RM}(\mathcal{R})(\gamma; a)$. *Then* $[s] = [t]$.

▶ **Theorem 30** (Coherence for Monoidal Categories). *All diagrams in the free monoidal category on a set commute.*

**Proof.** Corollary of the former theorem and Theorem 11, by noticing that $\mathsf{mon}(\mathsf{RM}(\mathcal{R}))$ is the free monoidal category on the underlying set of $\mathcal{R}$. ◀

## 4 A Resource Calculus for Symmetric Representable Multicategories

The symmetric representable terms have exactly the same syntax and operational semantics as the representable ones. We first extend the type system in order to account for symmetries. We then study the free constructions establishing an appropriate coherence result.

The typing is defined in Figure 3. It is easy to see that the representable type system consists of a fragment of the symmetric one, where we just consider identity permutations. We write $\gamma \vdash_{\mathsf{srep}} s : a$ when we need to specify that the type judgment refers to the symmetric representable type system. We set $\Lambda_{\mathsf{reps}}(\mathcal{R})(a_1, \ldots, a_n; a) = \{s \mid x_1 : a_1, \ldots, x_n : a_n \vdash_{\mathsf{srep}} s : a \text{ for some } x_i \in \mathsf{fv}(s).\}$.

▶ Remark 31. The role of permutations in the type system of Figure 3 deserves some commentary. Instead of having an independent permutation rule, variables in contexts can be permuted only when contexts have to be merged. In this way, the system is *syntax directed*. The limitation to the choice of *shuffle permutation* is needed to get uniqueness of type derivations for terms. Indeed, consider $s = \langle \langle x, y \rangle, z \rangle$. If we allow the choice of arbitrary permutations, we could build the following derivations:

$$\frac{\dfrac{x : a \vdash x : a \quad y : b \vdash y : b \quad \sigma}{y : b, x : a \vdash \langle x, y \rangle : (a \otimes b)} \qquad z : a \vdash z : a \quad id}{y : b, x : a, z : a \vdash s : ((a \otimes b) \otimes a)} \qquad \frac{\dfrac{x : a \vdash x : a \quad y : b \vdash y : b \quad id}{x : a, y : b \vdash \langle x, y \rangle : (a \otimes b)} \qquad z : a \vdash z : a \quad \sigma \oplus id}{y : b, x : a, z : a \vdash s : ((a \otimes b) \otimes a)}$$

where $\sigma$ is the swap. Thanks to the shuffle limitation, only the one on the left is allowed.

▶ **Proposition 32** (Canonicity of Typing). *If* $\pi \triangleright \gamma \vdash s : a$ *and* $\pi' \triangleright \gamma \vdash s : a'$ *then* $a = a'$ *and* $\pi = \pi'$.

$$\frac{a \in \mathcal{R}_0}{x : a \vdash x : a} \qquad \frac{\gamma_1 \vdash s_1 : a_1 \dots \gamma_k \vdash s_k : a_k \qquad \sigma \in \mathsf{shu}(\gamma_1, \dots, \gamma_k)}{(\gamma_1, \dots, \gamma_k) \cdot \sigma \vdash \langle s_1, \dots, s_k \rangle : (a_1 \otimes \dots \otimes a_k)}$$

$$\frac{\gamma \vdash s : (a_1 \otimes \dots \otimes a_k) \qquad \delta, x_1 : a_1, \dots, x_k : a_k, \delta' \vdash t : b \qquad \sigma \in \mathsf{shu}(\delta, \gamma', \delta')}{(\delta, \gamma, \delta') \cdot \sigma \vdash t[x_1^{a_1}, \dots, x_k^{a_k} := s] : b}$$

**Figure 3** Symmetric Representable Type System on a signature $\mathcal{R}$. We omit the case $f(\vec{s})$.

**Proof.** By induction on $s$. In the cases where a merging of type contexts happens, such as the list case, we rely on the properties of shuffle permutations and on the fact that type contexts are repetitions-free. Hence, the action of non-identity permutations on contexts is always fixedpoint-free. ◀

▶ **Proposition 33.** *The following rule is admissible:* $\dfrac{\gamma \vdash s : a \qquad \sigma \in S_k}{\gamma \cdot \sigma \vdash s : a}$.

**Proof.** Easy induction on the structure of $s$, exploiting Lemma 2. ◀

The reduction relation is the same as the representable one, that we know to be strongly normalizing and confluent. We also have preservation of typing under reduction and structural equivalence. Given $s \in \Lambda_{\mathsf{reps}}(\mathcal{R})(\gamma; a)$, we denote by $\mathsf{nf}(s)$ its unique normal form. As a corollary of subject reduction, we get that $\mathsf{nf}(s) \in \Lambda_{\mathsf{reps}}(\mathcal{R})(\gamma; a)$.

**Free Symmetric Representable Multicategories.** We now characterize the free symmetric representable construction. Given a representable signature $\mathcal{R}$, we define a multicategory by setting $\mathrm{ob}(\mathsf{SRM}(\mathcal{R})) = \mathcal{R}_0$ and $\mathsf{SRM}(\mathcal{R})(\gamma; a) = \Lambda_{\mathsf{reps}}(\mathcal{R})(\gamma; a)/(\equiv \cup =_{\mathsf{rep}})$. Composition is given by substitution, identities are given by variables. The operation is well-defined on equivalence classes and satisfies associativity, identity axioms. One can prove that $\mathsf{SRM}(\mathcal{R})$ is representable, by repeating the argument given for Proposition 24. The proof that $\mathsf{SRM}(\mathcal{R})$ is symmetric is a direct corollary of Proposition 33:

▶ **Proposition 34** (Symmetry)**.** *We have that* $\mathcal{M}(\gamma, a_1, \dots, a_k; a) = \mathcal{M}(\gamma; a_{\sigma(1)}, \dots, a_{\sigma(k)}; a)$.

▶ **Example 35.** An interesting example of structural equivalence is the following. Let $s = \langle\rangle[- := x][- := y]$ and $s' = \langle\rangle[- := y][- := x]$, with $s, s' \in \Lambda_{\mathsf{reps}}(\mathcal{R})((), (); ())$. We have that $\langle\rangle[- := x][- := y] \equiv \langle\rangle[- := y][- := x]$, with $x : (), y : () \vdash s : ()$ and $y : (), x : () \vdash s' : ()$. This is the way our syntax validates the fact that permutations of the unit type *collapse* to the identity permutation, since $s$ corresponds to the identity permutation, while $s'$ to the swapping of $x$ with $y$.

▶ **Definition 36.** *Let $\mathcal{R}$ be a representable signature and $\mathsf{S}$ be a symmetric representable multicategory. Let $i : \mathcal{R} \to \overline{\mathsf{S}}$ be a map of representable signatures. We define a family of maps $\mathsf{RT}(i)_{\gamma,a} : \Lambda_{\mathsf{reps}}(\mathcal{R})(\gamma; a) \to \mathsf{S}(i(\gamma); i(a))$ by induction as follows:*

$$\mathsf{RT}(i)_{a,a}(x) = id_{i(a)} \qquad \mathsf{RT}(i)_{(\gamma_1,\dots,\gamma_k)\cdot\sigma,\langle a_1,\dots,a_k\rangle}(\langle s_1, \dots, s_k\rangle) = \left(\bigotimes_{j=1}^{k} \mathsf{RT}(i)_{\gamma_j,a_j}(s_j)\right) \circ \sigma$$

$$\mathsf{RT}(i)_{\delta_1,\gamma,\delta_2\cdot\sigma,a}(s[x_1^{a_1}, \dots, x_k^{a_k} := t]) = ((\mathsf{RT}(i)_{\delta_1,a_1,\dots,a_k,a}(s))^* \circ \langle id_{\delta_1}, \mathsf{RT}(i)_{\gamma,\vec{a}}(t), id_{\delta_2}\rangle) \circ \sigma.$$

▶ **Theorem 37** (Free Construction). *Let* $\mathsf{S}$ *be a a symmetric representable multicategory and* $i : \mathcal{R} \to \overline{\mathsf{S}}$ *a map of representable signatures. There exists a unique symmetric representable functor* $i^*$ *such that* $i = \overline{i^*} \circ \eta_{\mathcal{R}}$.

**Coherence Result.** Fix a discrete signature $\mathcal{R}$. We shall prove that morphisms in $\mathsf{SRM}(\mathcal{R})$ can by characterized by means of appropriate permutations of their typing context. This will lead the following coherence result for symmetric representable multicategories: two morphisms in $\mathsf{SRM}(\mathcal{R})$ are equal whenever their *underlying permutations* are the same.

We start by defining the *strictification* of a representable type $\mathsf{strict}(a)$, by induction as follows: $\mathsf{strict}(o) = o$, $\mathsf{strict}((a_1 \otimes \cdots \otimes a_k)) = \mathsf{strict}(a_1), \ldots, \mathsf{strict}(a_k)$. $\mathsf{strict}(a)$ is the list of atoms that appear in the type $a$. We extend the strictification to contexts in the natural way. Let $s \in \mathsf{nf}(\Lambda_{\mathsf{reps}}(\mathcal{R}))(\gamma, a)$ and $\sigma \in \mathsf{Stab}(\mathsf{strict}(\gamma))$. We define the *right action* of $\sigma$ on $s$, $s^\sigma$ by induction as follows:

$$x^{id} = x \qquad \langle s_1, \ldots, s_k \rangle^{\sigma \circ (\bigoplus_{i=1}^k \sigma_i)} = \langle s_1^{\sigma_1}, \ldots, s_k^{\sigma_k} \rangle \cdot \sigma$$

$$(s[\vec{x}_1 := x_1] \ldots [\vec{x}_n := x_n])^\sigma = (s^\sigma)[\sigma(\vec{x}_1) := x_1] \ldots [\sigma(\vec{x}_n) := x_n]$$

where $\sigma(x_1, \ldots, x_k)$ stands for the image of $x_1, \ldots, x_k$ along the permutation $\sigma$.

▶ **Theorem 38.** *Let* $s \in \mathsf{nf}(\Lambda_{\mathsf{reps}}(\mathcal{R}))(\gamma, a)$. *There exists a unique* $\sigma \in \mathsf{Stab}((\mathsf{strict}(\gamma))$ *and a unique non-symmetric representable normal term* $t$ *such that* $s = t^\sigma$.

**Proof.** By induction on $s$, exploiting Proposition 32. If $s = x$ the result is immediate. If $s = \langle s_1, \ldots, s_k \rangle$ with $\gamma = (\gamma_1, \ldots, \gamma_k) \cdot \sigma \vdash \langle s_1, \ldots, s_k \rangle : (a_1 \otimes \cdots \otimes a_k)$ and $\gamma$ being atomic, by IH we have unique $\sigma_1, \ldots, \sigma_k \in \mathsf{St}(\mathsf{strict}(\gamma_i))$ and $t_1, \ldots, t_k \in \mathsf{nf}(\Lambda_{\mathsf{rep}}(A))$ s.t. $s_i = t_i^{\sigma_i}$ for $i \in [k]$. Then, by definition, $s = \langle t_1, \ldots, t_k \rangle^{\sigma \circ (\sigma_1 \otimes \cdots \otimes \sigma_k)}$. Uniqueness derives by Proposition 32. If $s = p[\vec{x}_1 := x_1] \ldots [\vec{x}_n := x_n]$, By IH there exists unique $\sigma$ and $t$ s.t. $p = t^\sigma$. Then we can conclude by the fact that the action of non-identity permutations on variables is fixedpoint-free. ◀

Let $s \in \mathsf{nf}(\Lambda_{\mathsf{reps}}(\mathcal{R}))(\gamma; a)$. We denote by $\mathsf{sym}(s)$ the unique permutation given by the former theorem. Given $s \in \Lambda_{\mathsf{reps}} A(\gamma; a)$ we set $\mathsf{sym}(s) = \mathsf{sym}(\mathsf{nf}(s))$. This definition is clearly coherent with the quotient on terms performed in the free construction.

▶ **Theorem 39.** *Let* $s, s' \in \mathsf{nf}(\Lambda_{\mathsf{reps}}(\mathcal{R}))(\gamma; a)$. *If* $\mathsf{sym}(s) = \mathsf{sym}(s')$ *then* $s \equiv s'$.

▶ **Theorem 40** (Coherence). *Let* $[s], [s'] \in \mathsf{SRM}(\mathcal{A})(\gamma; a)$. *If* $\mathsf{sym}([s]) = \mathsf{sym}[s']$ *then* $[s] = [s']$.

▶ **Theorem 41** (Coherence for Symmetric Monoidal Categories). *Two morphisms in the free symmetric monoidal categories are equal if their underlying permutations are equal.*

**Proof.** Corollary of Theorems 11 and 40. ◀

## 5 A Resource Calculus for Symmetric Closed Multicategories

We consider the case of symmetric closed multicategories, which is orthogonal to the representable structures we introduced in the previous sections. This calculus corresponds to the resource version of linear $\lambda$-calculus, where we have unbiased $k$-ary $\lambda$-abstraction and (linear) application. We begin by defining the terms and their typings, then proceed to introducing their operational semantics. We conclude by characterizing the free construction *via* well-typed equivalence classes of terms.

$$\frac{a \in \mathcal{L}_0}{x : a \vdash x : a} \qquad \frac{\gamma, x_1 : a_1, \ldots, x_k : a_k \vdash s : b}{\gamma \vdash \lambda\langle x_1^{a_1}, \ldots, x_k^{a_k}\rangle.s : (a_1 \otimes \cdots \otimes a_k) \multimap b}$$

$$\frac{\gamma_0 \vdash s : (a_1 \otimes \cdots \otimes a_k) \multimap b \qquad \gamma_1 \vdash t_1 : a_1 \ldots \gamma_k \vdash t_k : a_k \qquad \sigma \in \mathsf{shu}(\gamma_0, \ldots, \gamma_k)}{(\gamma, \delta) \cdot \sigma \vdash s\langle t_1, \ldots, t_k\rangle : b}$$

$\mathtt{C} ::= \mathtt{C} ::= [\cdot] \mid s\langle s_1, \ldots, \mathtt{C}, \ldots, s_k\rangle \mid \mathtt{C}\langle s_1, \ldots, s_k\rangle \mid \lambda\langle x_1, \ldots, x_k\rangle.\mathtt{C} \mid f(s_1, \ldots, \mathtt{C}, \ldots, s_k)$.

$\mathtt{E} ::= [\cdot] \mid s\langle s_1, \ldots, \mathtt{E}, \ldots, s_k\rangle \mid \mathtt{E}\langle s_1, \ldots, s_k\rangle \quad (\mathtt{E} \neq [\cdot]) \mid \lambda\langle x_1, \ldots, x_k\rangle.\mathtt{E} \mid f(s_1, \ldots, \mathtt{E}, \ldots, s_k)$.

**Figure 4** Symmetric closed type system on a signature $\mathcal{L}$ and contexts with one hole. Types are the elements of $\mathcal{L}_0$. We omit the case of $f(\vec{s})$.

$\beta$ *Root-Step*: $(\lambda\langle x_1^{a_1}, \ldots, x_k^{a_k}\rangle.s)\langle t_1, \ldots, t_k\rangle \to_\beta s\{t_1, \ldots, t_k/x_1, \ldots, x_k\}$.

$\eta$ *Root-Step*: $s \to_\eta \lambda\vec{x}^{\vec{a}}.(s\vec{x})$ where $\vec{x}$ fresh $, \gamma \vdash s : \vec{a} \multimap a, s \notin \mathsf{AT}$.

*Contextual extensions*: $\dfrac{s \to_\beta s'}{\mathtt{C}[s] \to_\beta \mathtt{C}[s']}$ $\quad \dfrac{s \to_\eta s'}{\mathtt{E}[s] \to_\eta \mathtt{E}[s']}$ $\quad (\to_{\mathsf{sc}} = \to_\beta \cup \to_\eta)$.

**Figure 5** Symmetric closed reduction relations.

**Symmetric Closed Resource Terms.** Let $\mathcal{L}$ be a closed signature. The *symmetric closed resource terms* on $\mathcal{L}$ are defined by the following inductive grammar:

$$\Lambda_{\mathsf{sc}}(\mathcal{L}) \ni s ::= x \in \mathcal{V} \mid \lambda\langle x_1^{a_1}, \ldots, x_k^{a_k}\rangle.s \mid s\langle s_1, \ldots, s_k\rangle \mid f(s_1, \ldots, s_k)$$

for $k \in \mathbb{N}$ and $f \in \mathsf{arr}(\mathcal{L}), a_i \in \mathcal{L}$. A term of the shape $s\langle s_1, \ldots, s_k\rangle$ is called a *(k-linear) application*. A term of the shape $\lambda\langle x_1, \ldots, x_k\rangle.s$ is called a *(k-linear) $\lambda$-abstraction*. Variables under the scope of a $\lambda$-abstraction are bound. We define the following subset of terms $\mathsf{AT} = \{\mathtt{L}[\lambda\vec{x}.t] \mid$ for some substitution context $\mathtt{L}$ and term $t.\}$. Typing and contexts with hole are defined in Figure 4. Given a term $\gamma \vdash s : a$, there exists a unique type derivation for it.

**Terms under Reduction.** The reduction relation is defined in Figure 5.

▶ Remark 42. The definition of the $\beta$-reduction follows the standard choices for resource calculi. The novel technicality is the restriction of the $\eta$-reduction, that is justified again by the goal of obtaining a strongly normalizing reduction. Indeed, $\eta$-reduction is again not normalizing. The situation recalls what happens in the standard $\lambda$-calculus and we deal with it adapting to our framework the restrictions introduced in [28, 16].

To study the rewriting, we adapt the method introduced for representable terms. We first prove that typing is preserved under reduction. Then, we introduce a measure that decreases under $\eta$. We define the *size* of a type by induction: $\mathtt{size}\,(o) = 0, \mathtt{size}\,(\langle a_1, \ldots, a_k\rangle \multimap a) = 1 + \sum \mathtt{size}\,(a_i) + \mathtt{size}\,(a)$. Given $\gamma \vdash s : a$ we define a set of typed subterms of $s$: $\mathsf{EST}(s) = \{\delta \vdash p : a \mid p \in \mathsf{ST}(s) \setminus \mathsf{AT}$ s.t. $\mathtt{E}[\delta \vdash p : a] = s$ for some context $\mathtt{E}\}$. We set $\eta(s) = \sum_{\delta \vdash p : a \in \mathsf{EST}(s)} \mathtt{size}\,(a)$. The proof of strong normalization and confluence is completely symmetrical to the representable case. Given $s \in \Lambda_{\mathsf{sc}}(\mathcal{R})(\gamma; a)$, we denote by $\mathsf{nf}(s)$ its unique normal form. As a corollary of subject reduction, we get that $\mathsf{nf}(s) \in \Lambda_{\mathsf{sc}}(\mathcal{R})(\gamma; a)$.

**Free Symmetric Closed Multicategories.** Let $\mathcal{L}$ be a closed signature, we define a multicategory $\mathsf{SCM}(\mathcal{L})$ by setting $\mathrm{ob}(\mathsf{SCM}(\mathcal{L})) = \mathcal{L}_0$ and $\mathsf{SCM}(\mathcal{L})(\gamma; a) = \Lambda_{\mathsf{sc}}(\mathcal{L})(\gamma; a)/\sim$ where $\sim\, = \,=_{\mathsf{sc}}$. Composition is given by substitution, identities are given by variables. The operation is well-defined equivalence classes and satisfies associativity and identity axioms. We also have that if $s \sim s'$, then $\mathsf{nf}(s) = \mathsf{nf}(s')$. We denote by $\eta_{\mathcal{L}} : \mathcal{L} \to \overline{\mathsf{SCM}(\mathcal{L})}$ the evident inclusion. One can prove that $\mathsf{SCM}(\mathcal{R})$ is symmetric, by repeating the argument given in the previous section. This multicategory is also *closed*:

▶ **Theorem 43.** *We have a bijection* $\mathsf{SCM}(\mathcal{L})(\gamma; \langle a_1, \dots, a_k\rangle \multimap a) \cong \mathsf{SCM}(\mathcal{L})(\gamma, a_1, \dots, a_k; a)$ *natural in $a$ and multinatural in $\gamma$, induced by the maps* $[s] \mapsto [s\langle x_1, \dots, x_k\rangle]$.

**Proof.** Naturality derives from basic properties of substitution. Inverses are given by the maps $[s] \mapsto [\lambda\langle x_1, \dots, x_k\rangle.s]$. ◀

▶ **Definition 44.** *Let* $\mathsf{E}$ *be a symmetric closed multicategory and let* $i : \mathcal{L} \to \overline{\mathsf{E}}$ *be a map of closed signatures. We define a family of maps* $\mathsf{RT}_{\gamma,a} : \Lambda_{\mathsf{sc}}(\mathcal{L})(\gamma, a) \to \mathsf{E}(i(\gamma), i(a))$ *by induction as follows:*

$$\mathsf{RT}_{a,a}(x) = 1_{i(a)} \qquad \mathsf{RT}_{\gamma,\vec{a}\multimap a}(\lambda\vec{x}.s) = \lambda(\mathsf{RT}_{\gamma,\vec{a},a}(s))$$

$$\mathsf{RT}_{(\gamma_0,\dots,\gamma_k),a}(s\langle t_1, \dots, t_k\rangle) = (ev \circ \langle \mathsf{RT}_{\gamma_0,\langle a_1,\dots,a_k\rangle\multimap a}(s), \mathsf{RT}_{\gamma_1,a_1}(t_1), \dots, \mathsf{RT}_{\gamma_k,a_1}(t_k)\rangle) \cdot \sigma.$$

▶ **Theorem 45** (Free Construction)**.** *Let* $\mathsf{S}$ *be a a symmetric closed multicategory and* $i : \mathcal{L} \to \overline{\mathsf{S}}$ *a map of representable signatures. There exists a unique symmetric closed functor* $i^* : \mathsf{SCM}(\mathcal{L}) \to \mathsf{S}$ *such that* $\overline{i^*} \circ \eta_{\mathcal{L}} = i$.

▶ **Theorem 46** (Coherence)**.** *Let* $[s], [s'] \in \mathsf{SCM}(\mathcal{R})(\gamma; a)$*. Then* $[s] = [s']$ *iff* $\mathsf{nf}([s]) \equiv \mathsf{nf}([s'])$.

## 6 A Resource Calculus for Autonomous Multicategories

In this section we present our calculus for autonomous multicategories. These structures bring together representability, symmetry and closure. For this reason, the calculus we will present is a proper extension of the ones we introduced before. Again, we follow the same pattern of Sections 3 and 5, first introducing the typing, then studying the operational semantics and finally characterizing the free constructions.

**Autonomous Terms.** Let $\mathcal{A}$ be an autonomous signature. The *autonomous resource terms* on $\mathcal{A}$ are defined by the following inductive grammar:

$$\Lambda_{\mathsf{aut}}(\mathcal{A}) \ni s, t ::= x \mid \lambda\langle x_1^{a_1}, \dots, x_k^{a_k}\rangle.s \mid st \mid \langle s_1, \dots, s_k\rangle \mid s[x_1^{a_1}, \dots, x_k^{a_k} := t] \mid f(s_1, \dots, s_k)$$

for $k \in \mathbb{N}$ and $f \in \mathsf{arr}(\mathcal{A}), a_i \in \mathcal{A}$. Variables under the scope of a $\lambda$-abstraction and of a substitution are bound. The typing is given in Figure 6. The calculi introduced in the previous sections can be seen as subsystems of the autonomous one.

Given a subterm $p$ of $s$ we write $\mathsf{ty}(p)_s$ for the type of $p$ in the type derivation of $s$. The mapping is functional as corollary of the former proposition.

$$\frac{a \in \mathcal{A}_0}{x : a \vdash x : a} \qquad \frac{\gamma_1 \vdash s_1 : a_1 \ldots \gamma_k \vdash s_k : a_k \qquad \sigma \in \mathsf{shu}(\gamma_1, \ldots, \gamma_k)}{(\gamma_1, \ldots, \gamma_k) \cdot \sigma \vdash \langle s_1, \ldots, s_k \rangle : (a_1 \otimes \cdots \otimes a_k)}$$

$$\frac{\gamma, x_1 : a_1, \ldots, x_k : a_k \vdash s : b}{\gamma \vdash \lambda \langle x_1^{a_1}, \ldots, x_k^{a_k} \rangle . s : (a_1 \otimes \cdots \otimes a_k) \multimap b} \qquad \frac{\gamma \vdash s : \vec{a} \multimap b \qquad \delta \vdash t : \vec{a} \qquad \sigma \in \mathsf{shu}(\gamma, \delta)}{(\gamma, \delta) \cdot \sigma \vdash st : b}$$

$$\frac{\gamma \vdash s : (a_1 \otimes \cdots \otimes a_k) \qquad \delta_1, x_1 : a_1, \ldots, x_k : a_k, \delta_2 \vdash t : b \qquad \sigma \in \mathsf{shu}(\gamma, \delta_1, \delta_2)}{(\gamma, \delta_1, \delta_2) \cdot \sigma \vdash t[x_1^{a_1}, \ldots, x_k^{a_k} := s] : b}$$

**Figure 6** Autonomous type system on a signature $\mathcal{A}$. We omit the case of $f(\vec{s})$.

**Terms under Reduction.** The reduction relation $\to_{\mathsf{aut}}$, together with its subreductions $\beta$ and $\eta$ are defined by putting together the reductions $\to_{\mathsf{rep}}$ (Figure 2) and $\to_{\mathsf{sc}}$ (Figure 5). The same happens with structural equivalence. The reduction satisfies subject reduction, strong normalization and confluence. The proofs build on the results of the previous sections. As decreasing measures, we use the size of a term for $\beta$-reduction and the sum of the two $\eta$ measures we defined in the previous sections for $\eta$-reduction.

**Free Autonomous Multicategories.** Let $\mathcal{A}$ be an autonomous signature, we define a multicategory $\mathsf{AUT}(\mathcal{A})$ by setting $\mathrm{ob}(\mathsf{AUT}(\mathcal{A})) = \mathcal{A}_0$ and $\mathsf{AUT}(\mathcal{A})(\gamma; a) = \Lambda_{\mathsf{aut}}(\mathcal{A})(\gamma; a)/\sim$ where $\sim$ is the equivalence $\equiv \cup =_{\mathsf{aut}}$ . Composition is given by substitution, identities are given by variables. The operation is well-defined on equivalence classes and satisfies associativity and identity axioms. We also have that if $s \sim s'$ then $\mathsf{nf}(s) \equiv \mathsf{nf}(s')$. We denote by $\eta_{\mathcal{A}} : \mathcal{A} \to \overline{\mathsf{AUT}(\mathcal{A})}$ the evident inclusion. One can prove that this multicategory is symmetric, representable and closed by importing the proofs given in the previous sections.

▶ **Definition 47.** *Let* $\mathsf{S}$ *be an autonomous multicategory and let* $i : \mathcal{A} \to \overline{\mathsf{S}}$ *be a map of autonomous signatures. We define a family of maps* $\mathsf{RT}_{\gamma,a} : \Lambda_{\mathsf{aut}}(\mathcal{A})(\gamma, a) \to \mathsf{E}(i(\gamma), i(a))$ *by induction, extending Definitions 36 and 44 in the natural way.*

▶ **Theorem 48** (Free Construction)**.** *Let* $\mathsf{S}$ *be a an autonomous multicategory and* $i : \mathcal{A} \to \overline{\mathsf{S}}$ *a map of autonomous signatures. There exists a unique autonomous functor* $i^* : \mathsf{AUT}(\mathcal{A}) \to \mathsf{S}$ *such that* $\overline{i^*} \circ \eta_{\mathcal{A}} = i$.

▶ **Theorem 49** (Coherence)**.** *Let* $[s], [s'] \in \mathsf{AUT}(\mathcal{R})(\gamma; a)$. *Then* $[s] = [s']$ *iff* $\mathsf{nf}([s]) \equiv \mathsf{nf}([s'])$.

## 7 Conclusion

We established a formal correspondence between resource calculi and appropriate linear multicategories, providing coherence theorems by means of normalization. As future work, we consider two possible perspectives. It is tempting to parameterize our construction over the choice of allowed *structural rules* on typing contexts. For instance, while the choice of permutations (*i.e.*, symmetries) gives *linear* structures, the choice of arbitrary functions between indexes would give *cartesian* structures. In this way, we would achieve a general method to produce type theories for appropriate *algebraic theories*, in the sense of [15]. For this, the the perspective on multicategories of [32] will be a starting point. Another perspective is the passage to the *second dimension*, following the path of [10]. In this way, the rewriting of terms would become visible in the multicategorical structure itself. Coherence by normalization could then be upgraded to a method of *coherence by standardization*, exploiting a rewriting relation on reduction paths.

## References


1 Beniamino Accattoli. Exponentials as substitutions and the cost of cut elimination in linear logic. In Christel Baier and Dana Fisman, editors, *LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2–5, 2022*, pages 49:1–49:15. ACM, 2022. doi:10.1145/3531130.3532445.

2 Beniamino Accattoli, Eduardo Bonelli, Delia Kesner, and Carlos Lombardi. A nonstandard standardization theorem. In Suresh Jagannathan and Peter Sewell, editors, *The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20-21, 2014*, pages 659–670. ACM, 2014. doi:10.1145/2535838.2535886.

3 Davide Barbarossa and Giulio Manzonetto. Taylor subsumes scott, berry, kahn and plotkin. *Proc. ACM Program. Lang.*, 4(POPL):1:1–1:23, 2020. doi:10.1145/3371069.

4 Nick Benton, Gavin Bierman, Valeria de Paiva, and Martin Hyland. A term calculus for intuitionistic linear logic. In *Proceedings of the International Conference on Typed Lambda Calculi and Applications (TLCA)*. Springer, January 1993.

5 R.F. Blute, J.R.B. Cockett, R.A.G. Seely, and T.H. Trimble. Natural deduction and coherence for weakly distributive categories. *Journal of Pure and Applied Algebra*, 113(3):229–296, 1996. doi:10.1016/0022-4049(95)00159-X.

6 Gérard Boudol. The lambda-calculus with multiplicities. In Eike Best, editor, *CONCUR'93*, pages 1–6, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

7 Pierre-Louis Curien, Richard Garner, and Martin Hofmann. Revisiting the categorical interpretation of dependent type theory. *Theoretical Computer Science*, 546:99–119, 2014. Models of Interaction: Essays in Honour of Glynn Winskel. doi:10.1016/j.tcs.2014.03.003.

8 Roberto Di Cosmo and Delia Kesner. Combining algebraic rewriting, extensional lambda calculi, and fixpoints. *Theoretical Computer Science*, 169(2):201–220, 1996. doi:10.1016/S0304-3975(96)00121-1.

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

10 Marcelo Fiore and Philip Saville. Coherence and normalisation-by-evaluation for bicategorical cartesian closed structure. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, pages 425–439. ACM, 2020. doi:10.1145/3373718.3394769.

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

12 Claudio Hermida. Representable multicategories. *Advances in Mathematics*, 151(2):164–225, 2000. doi:10.1006/aima.1999.1877.

13 Dominic J.D. Hughes. Simple free star-autonomous categories and full coherence. *Journal of Pure and Applied Algebra*, 216(11):2386–2410, 2012. doi:10.1016/j.jpaa.2012.03.020.

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

15 J.M.E. Hyland. Elements of a theory of algebraic theories. *Theoretical Computer Science*, 546:132–144, 2014. Models of Interaction: Essays in Honour of Glynn Winskel. doi:10.1016/j.tcs.2014.03.005.

16 C. Barry Jay and Neil Ghani. The virtues of eta-expansion. *Journal of Functional Programming*, 5(2):135–154, 1995. doi:10.1017/S0956796800001301.

17 G.M. Kelly and S. Maclane. Coherence in closed categories. *Journal of Pure and Applied Algebra*, 1(1):97–140, 1971. doi:10.1016/0022-4049(71)90013-2.

18 Delia Kesner. The theory of calculi with explicit substitutions revisited. In Jacques Duparc and Thomas A. Henzinger, editors, *Computer Science Logic*, pages 238–252, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

**19** AJ Kfoury. A linearization of the lambda-calculus and consequences. *Journal of Logic and Computation*, 10(3):411–436, 2000. doi:10.1093/logcom/10.3.411.

**20** Yves Lafont. *Logiques, Categories & Machines: Implantation de Langages de Programmation guidée par la Logique Catégorique*. Phd thesis, Université Paris VII, 1988.

**21** Joachim Lambek. Deductive systems and categories ii. standard constructions and closed categories. In Peter J. Hilton, editor, *Category Theory, Homology Theory and their Applications I*, pages 76–122, Berlin, Heidelberg, 1969. Springer Berlin Heidelberg.

**22** Joachim Lambek and Philip J. Scott. *Introduction to Higher Order Categorical Logic*. Cambridge University Press, USA, 1986.

**23** Tom Leinster. Higher operads, higher categories, 2003. arXiv:math/0305049.

**24** Saunders Maclane. Natural associativity and commutativity. *Rice Institute Pamphlet – Rice University Studies*, 49:28–46, 1963.

**25** Damiano Mazza. *Polyadic Approximations in Logic and Computation. Habilitation* thesis, Université Paris 13, 2017.

**26** Damiano Mazza, Luc Pellissier, and Pierre Vial. Polyadic approximations, fibrations and intersection types. *PACMPL*, 2018. doi:10.1145/3158094.

**27** Paul-André Melliès, Federico Olimpieri, and Lionel Vaux Auclair. An explicit construction of the homotopy span model of differential linear logic.

**28** G. E. Mints. Closed categories and the theory of proofs. *Journal of Soviet Mathematics*, 1981. doi:10.1007/BF01404107.

**29** Yo Ohta and Masahito Hasegawa. A terminating and confluent linear lambda calculus. In Frank Pfenning, editor, *Term Rewriting and Applications, 17th International Conference, RTA 2006, Seattle, WA, USA, August 12-14, 2006, Proceedings*, volume 4098 of *Lecture Notes in Computer Science*, pages 166–180. Springer, 2006. doi:10.1007/11805618_13.

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

**31** P. Selinger. *A Survey of Graphical Languages for Monoidal Categories*, pages 289–355. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. doi:10.1007/978-3-642-12821-9_4.

**32** Michael Shulman. Categorical logic from a categorical point of view. Draft for AARMS Summer School 2016, 2016. URL: https://mikeshulman.github.io/catlog/catlog.pdf.

**33** Michael Shulman. A practical type theory for symmetric monoidal categories. *Theory and Applications of Categories*, 37(25):863–907, 2021.

**34** 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, 2017. doi:10.1109/LICS.2017.8005093.

**35** Lionel Vaux. Taylor expansion, $\beta$-reduction and normalization. In *Ccomputer Science Logic 2017*, 2017. doi:10.4230/LIPIcs.CSL.2017.39.

**36** Niccolò Veltri. Coherence via focusing for symmetric skew monoidal categories. In Alexandra Silva, Renata Wassermann, and Ruy J. G. B. de Queiroz, editors, *Logic, Language, Information, and Computation – 27th International Workshop, WoLLIC 2021, Virtual Event, October 5-8, 2021, Proceedings*, volume 13038 of *Lecture Notes in Computer Science*, pages 184–200. Springer, 2021. doi:10.1007/978-3-030-88853-4_12.