Mathematical Models of Computational and Combinatorial Structures

(Invited Address)

Marcelo P. Fiore[*]

Computer Laboratory,
University of Cambridge
Marcelo.Fiore@cl.cam.ac.uk

Abstract. The general aim of this talk is to advocate a combinatorial perspective, together with its methods, in the investigation and study of models of computation structures. This, of course, should be taken in conjunction with the well-established views and methods stemming from algebra, category theory, domain theory, logic, type theory, *etc*. In support of this proposal I will show how such an approach leads to interesting connections between various areas of computer science and mathematics; concentrating on one such example in some detail. Specifically, I will consider the line of my research involving denotational models of the pi calculus and algebraic theories with variable-binding operators, indicating how the abstract mathematical structure underlying these models fits with that of Joyal's combinatorial species of structures. This analysis suggests both the unification and generalisation of models, and in the latter vein I will introduce *generalised species of structures* and their calculus. These generalised species encompass and generalise various of the notions of species used in combinatorics. Furthermore, they have a rich mathematical structure (akin to models of Girard's linear logic) that can be described purely within Lawvere's generalised logic. Indeed, I will present and treat the cartesian closed structure, the linear structure, the differential structure, *etc.* of generalised species axiomatically in this mathematical framework. As an upshot, I will observe that the setting allows for interpretations of computational calculi (like the lambda calculus, both typed and untyped; the recently introduced differential lambda calculus of Ehrhard and Regnier; *etc.*) that can be directly seen as translations into a more basic elementary calculus of interacting agents that compute by communicating and operating upon structured data.

Prologue

The process of understanding often unveils structure; and this, in turn, entails deeper understanding. In formal investigations, structure is articulated in mathematical terms. Mathematical structure typically plays a clarifying organisational role providing new insight and leading to new results. Ultimately theories are built; and then specialised,

[*] Research supported by an EPSRC Advanced Research Fellowship.

generalised, or unified. It is to this general context that the present work belongs. From a specific viewpoint, however, it is part of a research programme approaching computation structure from a combinatorial perspective. By this I broadly mean the transport of ideas, methodology, techniques, questions, *etc.* between combinatorics and computer science; in particular, in regarding data type structure as combinatorial structure, and vice versa.

As an example of what such a combinatorial view intends, I will show what the notion of *bijective proof* of combinatorial identities entails on data type structure. A bijective proof of $A = B$ consists in presenting combinatorial structures $\mathcal{A}$ and $\mathcal{B}$ that are respectively counted by A and by B together with a bijection $\mathcal{A} \cong \mathcal{B}$. The notion of bijective proof thus is nothing but that of *isomorphism of structure*, which in this view is given a fundamental role (as it is the case in many other areas of mathematics). In transporting this to the context of computation theory, for instance, one may be interested in bijections that are computable, primitive recursive, feasible, *etc.* In the context of type (or programming language) theory, the notion corresponds to the equivalence of data type structure up to isomorphism as prescribed by terms of the type theory (or programs of the programming language). Such a study has already been considered, though for entirely different reasons, under the broad heading of *type isomorphism*; see, *e.g.*, [9]. Besides applications in computer science, one interest in this subject lies in its connections to various areas of mathematics. Indeed, it is related to Tarski's high school algebra problem in mathematical logic [15], to the word problem in quotient polynomial semirings in computational algebra [16, 14], and to Thompson's groups in group theory [forthcoming joint work with Tom Leinster].

The rest of the paper provides another example of the fruitfulness of the approach advocated here. Specifically, I will first briefly review three diverse mathematical models —respectively suited for modelling name generation, combinatorial structure, and variable binding— highlighting how the various structures in each of them arise in essentially the same manner (Sect. 1). With this basis, I will then present a generalisation of the second model, putting it in the light of models of computation structures (Sect. 2). This yields connections to other areas of computer science and mathematics, and opens new perspectives for research (Sect. 3).

1 Some Computational and Combinatorial Structures

In this section I discuss in retrospective three mathematical models of computation structures in the chronological order in which I got familiar with them during my research. These are: denotational models of the pi calculus [17, 19] (Subsect. 1.1), Joyal's combinatorial species of structures [25, 26] (Subsect. 1.2), and algebraic models of equational theories with variable-binding operators [18, 19, 13] (Subsect. 1.3).[1] My intention here is not to treat any of them in full detail, but rather to give an outline of the

[1] Readers not familiar with the pi calculus [38] can safely skip Subsect. 1.1, or read it after Subsect. 1.3. Readers only interested in the combinatorial aspects can safely restrict their attention to Subsect. 1.2.

most relevant structures present in each model in such a way as to make explicit and apparent the similarities that run through them all.

1.1 Denotational Models of the Pi Calculus

The main ingredient leading to the construction of denotational models of the pi calculus [17, 47] was recognising that its feature allowing for the dynamic generation of names required the traditional denotational models to be indexed (or parameterised) by the set of the known names of a process. Naturally, and in the vein of previous models of store [45, 42] (see also [39–§ 4.1.4]), this was formalised by considering models in functor categories; that is, mathematical universes $\mathcal{S}^{\mathcal{V}}$ of $\mathcal{V}$-variable $\mathcal{S}$-objects (functors $\mathcal{V} \longrightarrow \mathcal{S}$) and $\mathcal{V}$-variable $\mathcal{S}$-maps (natural transformations) between them. In this context, $\mathcal{S}$ provides a basic model of denotations; whilst $\mathcal{V}$ gives an appropriate model of variation (see, *e.g.*, [30]). In the example at hand, $\mathcal{S}$ is a suitable category of domains (or simply the category of sets, if considering the finite pi calculus) and $\mathcal{V}$ is the category $\mathbf{I}$ of finite sets (or finite subsets of a fixed infinite countable set of names) and injections. Thus, a model $\mathsf{P} \in \mathcal{S}^{\mathbf{I}}$ consists of a series of actions

$$\_\,[=] : \mathsf{P}(\mathsf{U}) \times \mathbf{I}(\mathsf{U}, \mathsf{V}) \longrightarrow \mathsf{P}(\mathsf{V}) \qquad\qquad (\mathsf{U}, \mathsf{V} \in \mathbf{I})$$

for which $\mathsf{p}[\mathrm{id}_{\mathsf{U}}] = \mathsf{p}$ and $\mathsf{p}[\imath][\jmath] = \mathsf{p}[\imath \cdot \jmath]$ for all $\mathsf{p} \in \mathsf{P}(\mathsf{U})$ and $\imath : \mathsf{U} \rightarrowtail \mathsf{V}$, $\jmath : \mathsf{V} \rightarrowtail W$ in $\mathbf{I}$. These actions allow us to regard denotations p parameterised by a set of names U as denotations $\mathsf{p}[\imath]$ parameterised by a set of names V with respect to any injective renaming of names $\imath : \mathsf{U} \rightarrowtail \mathsf{V}$ in a consistent manner.

The question arises as to why the model of variation given by the category $\mathbf{I}$ in this context is the appropriate one. It was already pointed out in [17] that what it is important about $\mathbf{I}$ is its structure; namely, that it is equivalent to the free symmetric (strict) monoidal category with an initial unit on one generator. In this light, the generator stands for a generic name, whilst the tensor product allows for the creation of a new disjoint batch of generic names from two old ones. The role of the symmetry is roughly to render batches of generic names into sets, and the condition on the unit being initial allows for the ability of generating new names. This intuitive view is consistent with that of Needham's pure names [41] (see also [37]); *viz.*, those that can only be tested for equality with other ones, and indeed one can also formally recast the category $\mathbf{I}$ in these terms.

The fundamental mathematical structure of $\mathcal{S}^{\mathbf{I}}$ required for modelling the pi calculus can be now seen to arise in abstract generality. I will show this for $\mathcal{S}$ the category $\mathcal{S}et$ of sets and functions, but similar arguments can be made for other suitable categories.

Let $\mathsf{I}[\mathsf{n}]$ be the free symmetric strict monoidal category with tensor product $\oplus$ and initial unit O on the (name) generator n; it can be explicitly described as the category of finite cardinals and injections (with tensor product given by addition, initial unit by the empty set, and generator by the singleton). Through the Yoneda embedding, the generator provides an object of names $\mathsf{N} = \mathsf{y}(\mathsf{n})$ in $\mathcal{S}et^{\mathsf{I}[\mathsf{n}]}$ and, by Day's tensor construction [8, 23], the symmetric tensor product provides a (multiplication) symmetric tensor product $\widehat{\oplus}$ on $\mathcal{S}et^{\mathsf{I}[\mathsf{n}]}$ given by

$$\mathsf{P} \widehat{\oplus} \mathsf{Q} = \int^{\mathsf{U}_1, \mathsf{U}_2 \in \mathsf{I}[\mathsf{n}]} \mathsf{P}(\mathsf{U}_1) \times \mathsf{Q}(\mathsf{U}_2) \times \mathsf{I}[\mathsf{n}]\,(\mathsf{U}_1 \oplus \mathsf{U}_2, \_\,) \qquad \left(\mathsf{P}, \mathsf{Q} \in \mathcal{S}et^{\mathsf{I}[\mathsf{n}]}\right)$$

with $\mathsf{y}(\mathsf{O})$ as unit. (Note that the translation of this tensor product to $\mathcal{S}et^{\mathbf{I}}$ has the following simple description

$$(\mathsf{P}\widehat{\oplus}\mathsf{Q})(\mathsf{U}) = \textstyle\sum_{(\mathsf{U}_1,\mathsf{U}_2)\in \mathrm{SD}(\mathsf{U})} \mathsf{P}(\mathsf{U}_1) \times \mathsf{Q}(\mathsf{U}_2) \qquad \big(\mathsf{P},\mathsf{Q} \in \mathcal{S}et^{\mathbf{I}}, \mathsf{U} \in \mathbf{I}\big)$$

where the disjoint sum is taken over the set $\mathrm{SD}(\mathsf{U})$ of sub-decompositions of U; *i.e.*, pairs $(\mathsf{U}_1,\mathsf{U}_2)$ such that $\mathsf{U}_1 \cup \mathsf{U}_2 \subseteq \mathsf{U}$ and $\mathsf{U}_1 \cap \mathsf{U}_2 = \emptyset$.) More importantly for the current discussion, we have the following situation (consult App. A)

$$\begin{tikzcd}
\mathsf{I}[\mathsf{n}]^{\circ} \arrow[r, hook, "\mathsf{y}"] \arrow[d, "\_\oplus\mathsf{n}"'] \arrow[dr, phantom, "\overset{\mathrm{Lan}}{\cong}"] & \mathcal{S}et^{\mathsf{I}[\mathsf{n}]} \arrow[d, shift right=6, "\_\widehat{\oplus}\mathsf{N}"'] \arrow[d, shift left=6] \\
\mathsf{I}[\mathsf{n}]^{\circ} \arrow[r, hook, "\mathsf{y}"'] & \mathcal{S}et^{\mathsf{I}[\mathsf{n}]} \arrow[u, "\dashv\ \delta_{\mathsf{n}}\ \dashv" description]
\end{tikzcd}$$

which yields a *name generation* operator $\delta_{\mathsf{n}} = (\_ \oplus \mathsf{n})^*$, arising as closed structure (since $(\_ \oplus \mathsf{n})_! \cong \_\widehat{\oplus}\mathsf{N}$) and simply given by

$$\delta_{\mathsf{n}}\mathsf{P} = \mathsf{P}(\_ \oplus \mathsf{n}) \qquad\qquad (\mathsf{P} \in \mathcal{S}et^{\mathsf{I}[\mathsf{n}]})\ .$$

Thus, a denotation in $\delta_{\mathsf{n}}\mathsf{P}$ is nothing but a denotation in P parameterised by a new generic name.

With the aid of the cartesian closed structure of $\mathcal{S}et^{\mathsf{I}[\mathsf{n}]}$ one can then model the behavioural actions of pi calculus processes: input is modelled by the exponential $(\_)^{\mathsf{N}}$, free output by the product $\mathsf{N} \times (\_)$, and bound output by the name generator $\delta_{\mathsf{n}}(\_)$. For details on both the late and early behaviours consult [19].

1.2 Combinatorial Species of Structures

The theory of combinatorial species was introduced by Joyal in [25]. One of its important features is to provide a mathematical framework in which arguments in enumerative combinatorics based on generating functions acquire structural combinatorial meaning leading to bijective proofs of combinatorial identities. For instance, in [26–Chap. 2], Joyal presents a calculus of species in which structural operations on them (together with their laws) exactly correspond, modulo the process of counting, to the operations in algebras of formal power series; see also [5–Chap. 1 and 2].

The basic notion of species of structures is given by a functor $\mathbf{B} \longrightarrow \mathcal{S}et$, for $\mathbf{B}$ the category of finite sets and bijections. Naturally, the category of species is taken to be the category $\mathcal{S}et^{\mathbf{B}}$. Species P can be equivalently given by a series of symmetric-group actions

$$\_[=] : \mathsf{P}[n] \times \mathfrak{S}_n \longrightarrow \mathsf{P}[n] \qquad\qquad (n \in \mathbb{N})$$

for which $p[\mathrm{id}] = p$ and $p[\sigma][\tau] = p[\sigma \cdot \tau]$ for all $p \in \mathsf{P}[n]$ and $\sigma,\tau \in \mathfrak{S}_n$.

Intuitively, for a species P, the set $\mathsf{P}(\mathsf{U})$ consists of the structures of type P that can be put on the set of tokens U; the action provides the abstract rule of transport of structures, which serves for describing structural equivalence (*i.e.*, when structures are the same except for a permutation of the tokens that constitute them). For instance, the species End with structures on a set of tokens U given by the endofunctions on U and action by conjugation is defined as

$$\begin{aligned} &\mathrm{End}(U) = \mathcal{S}et\,(U,U) && (U \in \mathbf{B}) \\ &\mathrm{End}(\sigma)(f) = \sigma\, f\, \sigma^{-1} && \big(f \in \mathrm{End}(U,U), \sigma \in \mathbf{B}(U,V)\big)\ . \end{aligned}$$

Two endofunctions are structurally equivalent if they are conjugate to each other.

I will now present a repertoire of operations on species: addition, multiplication, differentiation, and composition. In doing so, I will be placing emphasis in how they relate to the other structures of the paper; rather than following the standard combinatorial presentation. Nonetheless this approach is certainly known to experts.

It is important first to focus on the structure of $\mathbf{B}$. It was already pointed out in [25–Subsect. 7.3] that it is equivalent to the free symmetric (strict) monoidal category on one generator, and we will henceforth consider it in this vein. Let $\mathsf{B}[\mathsf{x}]$ be the free symmetric strict monoidal category with tensor product $\oplus$ and unit O on the (token) generator x; it can be explicitly described as the category of finite cardinals and permutations (with tensor product given by addition, unit by the empty set, and generator by the singleton).

Addition. The addition $P + Q$ of combinatorial species P and Q is given by their categorical coproduct:

$$(P+Q)(U) = P(U) + Q(U) \qquad (P, Q \in \mathcal{S}et^{\mathbf{B}}, U \in \mathbf{B})\ .$$

Thus, a structure of type $P + Q$ is either a structure of type P or one of type Q together with the information of which type of structure it is.

Multiplication. By Day's tensor construction [8, 23], the symmetric tensor product on $\mathsf{B}[\mathsf{x}]$ provides a multiplication symmetric tensor product $\cdot$ on $\mathcal{S}et^{\mathsf{B}[\mathsf{x}]}$ given by

$$P \cdot Q = \int^{U_1, U_2 \in \mathsf{B}[\mathsf{x}]} P(U_1) \times Q(U_2) \times \mathsf{B}[\mathsf{x}]\,(U_1 \oplus U_2, \_) \qquad (P, Q \in \mathcal{S}et^{\mathsf{B}[\mathsf{x}]})$$

with unit $\mathsf{y}(\mathsf{O})$. (Note that the translation of this tensor product to $\mathcal{S}et^{\mathbf{B}}$ has the following simple description

$$(P \cdot Q)(U) \cong \sum_{(U_1, U_2) \in D(U)} P(U_1) \times Q(U_2) \qquad (P, Q \in \mathcal{S}et^{\mathbf{B}}, U \in \mathbf{B})$$

where the disjoint sum is taken over the set $D(U)$ of decompositions of U; *i.e.*, pairs (U_1, U_2) such that $U_1 \cup U_2 = U$ and $U_1 \cap U_2 = \emptyset$.)

Thus, to construct a structure of type $P \cdot Q$ on a set of tokens U is to decompose U in sets of tokens U_1 and U_2, and put a structure of type P on U_1 and a structure of type Q on U_2.

Differentiation. We have the following situation (see, *e.g.*, [44])

$$\begin{tikzcd}
\mathsf{B}[\mathsf{x}]^\circ \arrow[r, hook, "\mathsf{y}"] \arrow[d, "\_\oplus \mathsf{x}"'] \arrow[dr, phantom, "\overset{\mathrm{Lan}}{\cong}"] & \mathcal{S}et^{\mathsf{B}[\mathsf{x}]} \arrow[d, shift right=8, "\_\cdot \mathsf{X}"'] \arrow[d, shift left=8] \arrow[d, phantom, "\dashv\ \mathrm{d}/\mathrm{d}\mathsf{x}\ \dashv"] \\
\mathsf{B}[\mathsf{x}]^\circ \arrow[r, hook, "\mathsf{y}"'] & \mathcal{S}et^{\mathsf{B}[\mathsf{x}]} \arrow[u]
\end{tikzcd}$$

which yields a differentiation operator $\mathrm{d}/\mathrm{d}\mathsf{x} = (\_ \oplus \mathsf{x})^*$, arising as closed structure $\big($since $(\_ \oplus \mathsf{x})_! \cong \_ \cdot \mathsf{X}$ for $\mathsf{X} = \mathsf{y}(\mathsf{x})\big)$ and simply given by

$$(\mathrm{d}/\mathrm{d}\mathsf{x})\mathsf{P} = \mathsf{P}(\_ \oplus \mathsf{x}) \qquad\qquad (\mathsf{P} \in \mathit{Set}^{\mathsf{B}[\mathsf{x}]}) \ .$$

It follows that a structure of type $(\mathrm{d}/\mathrm{d}\mathsf{x})\mathsf{P}$ over a set of tokens is nothing but a structure of type P over the set of tokens enlarged with a new generic one.

Composition. Using the universal properties of both $\mathsf{B}[\mathsf{x}]$ and $\mathit{Set}^{\mathsf{B}[\mathsf{x}]}$, we obtain the following situation (consult App. A)

$$\begin{tikzcd}
\mathbf{1} \arrow[r, "\mathsf{x}"] \arrow[dr, "\mathsf{P}"'] \arrow[dr, phantom, shift left=3ex, "\cong"] & \mathsf{B}[\mathsf{x}]^{\circ} \arrow[r, hook, "\mathsf{y}"] \arrow[d, "\mathsf{P}^{\bullet(-)}"'] \arrow[d, phantom, shift left=3ex, "\overset{\mathrm{Lan}}{\cong}"] & \mathit{Set}^{\mathsf{B}[\mathsf{x}]} \arrow[dl, "\_\circ\mathsf{P}"', shift right] \\
& \mathit{Set}^{\mathsf{B}[\mathsf{x}]} \arrow[ur, shift right] \arrow[ur, phantom, shift left=2ex, "\dashv"] &
\end{tikzcd}$$

where

$$\mathsf{P}^{\bullet(\overbrace{\mathsf{X} \oplus \cdots \oplus \mathsf{X}}^{n \text{ times}})} = \underbrace{\mathsf{P} \cdot \ldots \cdot \mathsf{P}}_{n \text{ times}} \qquad\qquad (\mathsf{P} \in \mathit{Set}^{\mathsf{B}[\mathsf{x}]})$$

and

$$(\mathsf{Q} \circ \mathsf{P})(\mathsf{U}) = \int^{\mathsf{T} \in \mathsf{B}[\mathsf{x}]} \mathsf{Q}(\mathsf{T}) \times \mathsf{P}^{\bullet\mathsf{T}}(\mathsf{U}) \qquad\qquad (\mathsf{Q}, \mathsf{P} \in \mathit{Set}^{\mathsf{B}[\mathsf{x}]}) \ .$$

This so-called composition (or substitution) operation $\circ$ on species yields a (highly non-symmetric) monoidal closed structure on $\mathit{Set}^{\mathsf{B}[\mathsf{x}]}$ with unit $\mathsf{X} = \mathsf{y}(\mathsf{x})$ (see also [27]). Translating it to $\mathit{Set}^{\mathbf{B}}$ we obtain, whenever $\mathsf{P}(\emptyset) = \emptyset$, that

$$(\mathsf{Q} \circ \mathsf{P})(\mathsf{U}) \cong \sum_{\mathcal{U} \in \mathrm{Part}(\mathsf{U})} \mathsf{Q}(\mathcal{U}) \times \prod_{u \in \mathcal{U}} \mathsf{P}(u) \qquad (\mathsf{Q}, \mathsf{P} \in \mathit{Set}^{\mathbf{B}}, \mathsf{U} \in \mathbf{B})$$

where the disjoint sum is taken over the set $\mathrm{Part}(\mathsf{U})$ of partitions of U. In words, a structure $q[u_1 \mapsto p_1, \ldots, u_n \mapsto p_n]$ in $(\mathsf{Q}\circ\mathsf{P})(\mathsf{U})$ consists of a partition $\mathcal{U} = \{u_1, \ldots, u_n\}$ of the set of (input) tokens U, together with a structure q of type Q over the set $\mathcal{U}$ of n (place-holder) tokens for the structures p_i $(1 \leq i \leq n)$ in $\mathsf{P}(u_i)$. Monoids for this composition tensor product are known as (symmetric) operads (see, *e.g.*, [48]).

An important part of the theory of species (on which I can only refer the reader to [25] here) is that they can be equivalently seen as analytic endofunctors on Set (of which species are the coefficients) that embody the structure of the formal exponential power series induced by counting. From this point of view, the terminology of composition for the above operation is justified by the fact that it corresponds to the usual composition of functors.

1.3 Algebraic Theories with Variable-Binding Operators

The key to the algebraic treatment of abstract syntax with variable binding is to shift attention away from raw terms and focus on terms in context (or term judgements). This requires taking contexts seriously; considering the operations allowed on them and the structural rules that term judgements have, and building categories of contexts that reflect them. The categories of contexts so obtained provide then models of variation whose structure induces, in the associated universe of variable sets, further structure allowing for algebraic theories with variable-binding operators. These general remarks will become clear after reading the rest of the section, where the approach is exemplified.

The original approach of [18] was conceived for the framework of binding algebras [1] (see also [50]), where term judgements are subject to the admissible rules of weakening, contraction, and permutation. Thus, the natural notion of morphism between contexts is that provided by any renaming of variables. Below I will concentrate on the multi-sorted case where, of course, variable renamings should in addition be well-typed; see [13–Sect. II.1] for further details and a discussion of the syntax and semantics of the simply typed lambda calculus.

Abstractly, the category of contexts is then the free cocartesian category over the set of types. As other such free constructions, it can be explicitly described in two stages. First one considers the category F of mono-sorted (or untyped) contexts given by the free cocartesian category on one generator (with coproduct $+$, initial object 0, and generator 1); *i.e.*, the category of finite cardinals and functions (with coproduct given by addition, initial object by the empty set, and generator by the singleton). Then, for a set of types T, the category of T-sorted contexts is given by the comma construction $\mathsf{F}{\downarrow}\mathsf{T}$ (whose objects are maps $\Gamma : |\Gamma| \longrightarrow \mathsf{T}$ with $|\Gamma| \in \mathsf{F}$ and whose morphisms $\rho : \Gamma \longrightarrow \Gamma'$ are maps $\rho : |\Gamma| \longrightarrow |\Gamma'|$ in F such that $\Gamma = \Gamma'\rho$). The initial object $(0 \longrightarrow \mathsf{T})$ in $\mathsf{F}{\downarrow}\mathsf{T}$ is the empty context, whilst the coproduct

$$(|\Gamma| \xrightarrow{\Gamma} \mathsf{T}) + (|\Gamma'| \xrightarrow{\Gamma'} \mathsf{T}) = (|\Gamma| + |\Gamma'| \xrightarrow{[\Gamma,\Gamma']} \mathsf{T})$$

in $\mathsf{F}{\downarrow}\mathsf{T}$ amounts to the operation of context extension.

It is convenient to define $\mathsf{F}[\mathsf{T}] = (\mathsf{F}{\downarrow}\mathsf{T})^\circ$ and identify $\tau \in \mathsf{T}$ with its image $1 \xrightarrow{\tau} \mathsf{T}$ in $\mathsf{F}[\mathsf{T}]$ under the universal embedding $\mathsf{T} \longrightarrow \mathsf{F}[\mathsf{T}]$ exhibiting $\mathsf{F}[\mathsf{T}]$ as the free cartesian category on T.

The mathematical universe in which to consider algebraic theories is then the category $\widehat{\mathsf{F}[\mathsf{T}]}^{\mathsf{T}}$. Informally, for $\mathsf{P} \in \widehat{\mathsf{F}[\mathsf{T}]}^{\mathsf{T}}$, one thinks of the sets $\{\, \mathsf{P}_\tau(\Gamma) \,\}_{\tau\in\mathsf{T},\Gamma\in\mathsf{F}[\mathsf{T}]}$ as the τ-sorted P-elements in context Γ. As an example consider the object of variables $\mathsf{V} = \{\, \mathsf{V}_\tau \,\}_{\tau\in\mathsf{T}}$ given by $\mathsf{V}_\tau = \mathsf{y}(\tau)$; so that

$$\mathsf{V}_\tau(\Gamma) \cong \{\, x \in |\Gamma| \mid \Gamma(x) = \tau \,\} \qquad (\tau \in \mathsf{T}, \Gamma \in \mathsf{F}{\downarrow}\mathsf{T}) \ .$$

Crucially, noting that $(\_ \times \tau)_! \cong \_ \times \mathsf{V}_\tau$, the operation of context extension induces the situation below

$$\begin{tikzcd} \mathsf{F}[\mathsf{T}] \arrow[r, hook, "\mathsf{y}"] \arrow[d, "\_\times\tau"'] \arrow[dr, phantom, "\overset{\text{Lan}}{\cong}"] & \widehat{\mathsf{F}[\mathsf{T}]} \arrow[d, shift right=4, "\_\times\mathsf{V}_\tau"'] \arrow[d, phantom, shift right=2, "\dashv"] \arrow[d, phantom, shift left=2, "\dashv"] \arrow[d, shift left=4] \\ \mathsf{F}[\mathsf{T}] \arrow[r, hook, "\mathsf{y}"'] & \widehat{\mathsf{F}[\mathsf{T}]} \arrow[u] \end{tikzcd} \qquad (\tau \in \mathsf{T})$$

from which it follows that

$$\mathsf{X}^{\mathsf{V}_\tau} \cong (\_ \times \tau)^*(\mathsf{X}) = \mathsf{X}(\_ + \tau) \qquad (\mathsf{X} \in \widehat{\mathsf{F}[\mathsf{T}]}, \tau \in \mathsf{T}) \ .$$

Thus, the object of variables provides suitable arities for binding operators. Indeed, an operator of arity

$$\big(\tau_1^{(1)},\ldots,\tau_{n_1}^{(1)}\big)\tau_1,\ldots,\big(\tau_1^{(k)},\ldots,\tau_{n_k}^{(k)}\big)\tau_k \longrightarrow (\sigma_1,\ldots,\sigma_m)\sigma$$

that binds variables of type $\tau_1^{(i)},\ldots,\tau_{n_i}^{(i)}$ in terms of type τ_i $(1 \leq i \leq k)$ yielding a term of type σ that binds variables of type σ_j $(1 \leq j \leq m)$ corresponds to a morphism

$$\textstyle\prod_{1\leq i\leq k} \mathsf{P}_{\tau_i}{}^{\prod_{1\leq j\leq n_i} \mathsf{V}_{\tau_j^{(i)}}} \longrightarrow \mathsf{P}_\sigma{}^{\prod_{1\leq \ell\leq m} \mathsf{V}_{\sigma_\ell}} \qquad \big(\mathsf{P}\in\widehat{\mathsf{F}[\mathsf{T}]}^{\mathsf{T}}\big)$$

in $\widehat{\mathsf{F}[\mathsf{T}]}$, that further corresponds to a natural family

$$\textstyle\prod_{1\leq i\leq k} \mathsf{P}_{\tau_i}\big(\_ + \langle\tau_1^{(i)},\ldots,\tau_{n_i}^{(i)}\rangle\big) \longrightarrow \mathsf{P}_\sigma(\_ + \langle\sigma_1,\ldots,\sigma_m\rangle) \qquad \big(\mathsf{P}\in\widehat{\mathsf{F}[\mathsf{T}]}^{\mathsf{T}}\big)$$

associating a tuple of elements of type τ_i $(1 \leq i \leq k)$ in a context extended by new generic variables of type $\tau_j^{(i)}$ $(1 \leq j \leq n_i)$ with an element of type σ in a context extended by new generic variables of type σ_ℓ $(1 \leq \ell \leq m)$.

The framework also allows for the axiomatisation of substitution via an equational theory whose algebras correspond to Lawvere theories (see [18]). In App. B, I briefly discuss single-variable substitution in the context of algebraic theories for binding signatures. Here, as it is of direct concern to us, I will concentrate on the notion of simultaneous substitution, which arises in the same manner as operads do with respect to the composition of species. Indeed, using the universal properties of both $\mathsf{F}[\mathsf{T}]$ and $\widehat{\mathsf{F}[\mathsf{T}]}$, we have the following situation

$$\begin{tikzcd}
\mathsf{T} \arrow[r] \arrow[dr, "\mathsf{P}"'] & \mathsf{F}[\mathsf{T}] \arrow[r, hook, "\mathsf{y}"] \arrow[d, "\mathsf{P}^{\times(-)}"] \arrow[dl, phantom, "\cong"] \arrow[dr, phantom, "\overset{\mathrm{Lan}}{\cong}"] & \widehat{\mathsf{F}[\mathsf{T}]} \arrow[dl, "\_\bullet\mathsf{P}"', shift right] \\
& \widehat{\mathsf{F}[\mathsf{T}]} \arrow[ur, shift right] \arrow[ur, phantom, "\dashv"'] &
\end{tikzcd}$$

where

$$\mathsf{P}^{\times\Delta} = \textstyle\prod_{x\in|\Delta|} \mathsf{P}_{\Delta(x)} \qquad \Big(\mathsf{P}\in\widehat{\mathsf{F}[\mathsf{T}]}^{\mathsf{T}}, \Delta\in\mathsf{F}[\mathsf{T}]\Big)$$

and

$$(\mathsf{X}\bullet\mathsf{P})(\Gamma) = \int^{\Delta\in\mathsf{F}[\mathsf{T}]} \mathsf{X}(\Delta)\times\mathsf{P}^{\times\Delta}(\Gamma) \qquad \Big(\mathsf{X}\in\widehat{\mathsf{F}[\mathsf{T}]}, \mathsf{P}\in\widehat{\mathsf{F}[\mathsf{T}]}^{\mathsf{T}}, \Gamma\in\mathsf{F}{\downarrow}\mathsf{T}\Big)\ .$$

We obtain thus a (highly non-symmetric) *composition* monoidal closed structure $\circ$ on $\widehat{\mathsf{F}[\mathsf{T}]}^{\mathsf{T}}$ given by

$$(\mathsf{Q}\circ\mathsf{P})_\tau = \mathsf{Q}_\tau\bullet\mathsf{P} \qquad \Big(\mathsf{Q},\mathsf{P}\in\widehat{\mathsf{F}[\mathsf{T}]}^{\mathsf{T}}, \tau\in\mathsf{T}\Big)$$

with unit the object of variables V.

Monoids for this composition tensor product correspond to multi-sorted Lawvere theories, and embody the structure of simultaneous substitution. To see this consider the axioms for a multiplication operation $P \circ P \longrightarrow P$ noting that, in elementary terms, $(Q \circ P)_\tau(\Gamma)$ consists of equivalence classes of pairs given by $q \in Q_\tau\langle \tau_1, \ldots, \tau_n \rangle$ together with an assignment $\langle \tau_1 \mapsto p_1, \ldots, \tau_n \mapsto p_n \rangle$ with $p_i \in P_{\tau_i}(\Gamma)$ $(1 \leq i \leq n)$ under the identification

$$q[\rho]\langle \tau_1 \mapsto p_1, \ldots, \tau_m \mapsto p_m \rangle = q\langle \sigma_1 \mapsto p_{\rho 1}, \ldots, \sigma_m \mapsto p_{\rho m} \rangle$$

for all renamings $\rho : \langle \sigma_1, \ldots, \sigma_m \rangle \longrightarrow \langle \tau_1, \ldots, \tau_n \rangle$ in $\mathsf{F} \downarrow \mathsf{T}$, $q \in Q_\tau\langle \sigma_1, \ldots, \sigma_m \rangle$, and $p_i \in P_{\tau_i}(\Gamma)$ $(1 \leq i \leq n)$, where $q[\rho] = Q_\tau(\rho)(q) \in Q_\tau\langle \tau_1, \ldots, \tau_n \rangle$.

Finally, note that one can also consider heterogeneous notions of substitution (for which see [19]) and variations on the theme.

2 The Calculus of Generalised Species of Structures

This is the main section of the paper. In Subsect. 2.1, the notion of generalised species of structures is motivated and introduced. Afterwards, some of the structure of generalised species is presented: addition and multiplication in Subsect. 2.2; differential structure in Subsect. 2.3 and 2.6; identities and composition in Subsect. 2.4; and, the cartesian closed structure in Subsect. 2.5. Finally, Subsect. 2.7 outlines the calculus of these operations.

Somehow following the tradition in combinatorics, my emphasis here is to present generalised species as a calculus; including graphical representations that will hopefully convey the idea behind the various constructions on structures. On the other hand, however, I depart from the traditional combinatorial treatment in that the calculus is axiomatically built on top of the mathematical framework of Lawvere's generalised logic [28] (see Fig. 1 in Subsect. 2.7 for an example). This yields new algebraic proofs, even for the restriction of generalised species to their basic form recalled in Subsect. 1.2. In passing, I will remark on the relationship between the structures of this section and those of the previous one.

Generalised species have other roots in ideas of Martin Hyland and Glynn Winskel; and there is a general abstract theory, that we have developed with them and Nicola Gambino, that accounts for their bicategorical (Subsect. 2.4) and cartesian closed (Subsect. 2.5) structures. This perspective is important, for it further organises the subject (placing it, *e.g.*, in the context of models of Girard's linear logic) and guides its development.

2.1 Generalised Species

Recall that the basic notion of species of structures is given by a functor $\mathbf{B} \longrightarrow \mathcal{S}et$, for $\mathbf{B}$ the category of finite sets and bijections [25, 26]. Recall also that $\mathbf{B}$ is equivalent to the free symmetric (strict) monoidal category on one generator. Thus, writing ! for the free symmetric (strict) monoidal completion, species can be equivalently presented as functors $!\mathbf{1} \longrightarrow \mathcal{S}et$. In the spirit of Subsect. 1.3, it makes sense to consider T-sorted species, for T a set of sorts, as functors $!\mathsf{T} \longrightarrow \mathcal{S}et$; and, even more generally, for a small category $\mathbb{T}$ of sorts and maps between them, define $\mathbb{T}$-sorted species of structures (or simply $\mathbb{T}$-species) as functors $!\mathbb{T} \longrightarrow \mathcal{S}et$.

To be able to visualise these structures we will analyse them in some detail. First, as I have already mentioned, the free symmetric strict monoidal category on one generator $!\mathbf{1}$ (with tensor $+$, unit 0, and generator 1) can be described as the category B of finite cardinals and permutations (with tensor product given by addition, unit by the empty set, and generator by the singleton). This category, as it happens with other such free constructions, induces the free symmetric strict monoidal completion $!\mathbb{T}$ of a category $\mathbb{T}$ by the comma construction $\mathsf{B}\Downarrow\mathbb{T}$ whose objects are maps $\mathsf{T} : |\mathsf{T}| \longrightarrow \mathbb{T}$ with $|\mathsf{T}| \in \mathsf{B}$ and whose morphisms are pairs $(\sigma, \vec{\sigma})$ as on the left below

$$\begin{tikzcd}
{|\mathsf{T}|} \arrow[rr, "\sigma"] \arrow[dr, "\mathsf{T}"'] \arrow[rr, phantom, shift right=4, "\overset{\vec{\sigma}}{\Longrightarrow}"] & & {|\mathsf{T}'|} \arrow[dl, "\mathsf{T}'"] \\
& \mathbb{T} &
\end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small]
(i \in |\mathsf{T}|) & \cdots\ \mathsf{T}_i\ \cdots \arrow[d, "\vec{\sigma}_i"] & & & \cdots\ \mathsf{T}_i\ \cdots\ (i \in |\mathsf{T}|) \arrow[dd, "\vec{\tau}_{\sigma i}\vec{\sigma}_i"] \\
(j \in |\mathsf{T}'|) & \cdots\ \mathsf{T}'_{\sigma i}\ \cdots & \mathsf{T}'_j\ \cdots \arrow[dl, "\vec{\tau}_j"'] & = & \\
& \cdots\ \mathsf{T}''_{\tau j}\ \cdots & & & \cdots\ \mathsf{T}''_{\tau\sigma i}\ \cdots
\end{tikzcd}$$

with $\sigma : |\mathsf{T}| \longrightarrow |\mathsf{T}'|$ in B and $\vec{\sigma} : \mathsf{T} \Longrightarrow \mathsf{T}'\sigma$ in $\mathbb{T}^{\mathsf{T}}$. Morphisms and their composition can be drawn as on the right above. The tensor product in $!\mathbb{T}$ is given by $\mathsf{T} \oplus \mathsf{T}' = [\mathsf{T}, \mathsf{T}'] : |\mathsf{T}| + |\mathsf{T}'| \longrightarrow \mathbb{T}$; that is, roughly as

$$\{\!|\cdots \mathsf{T}_i \cdots \mid i \in |\mathsf{T}|\,|\!\}\oplus\{\!|\cdots \mathsf{T}'_j \cdots \mid j \in |\mathsf{T}'|\,|\!\} = \{\!|\cdots \mathsf{T}_i \cdots \mathsf{T}'_j \cdots \mid i \in |\mathsf{T}|, j \in |\mathsf{T}'|\,|\!\} ,$$

with unit $\mathsf{O} = (0 \longrightarrow \mathbb{T})$. (Note as a remark that for what follows, and in keeping closer to the combinatorial spirit, one can equivalently take $!\mathbb{T}$ to be $\mathbf{B}\Downarrow\mathbb{T}$.)

Henceforth, let $\{\!|\_|\!\} : \mathbb{T} \longrightarrow\, !\mathbb{T}$ be the universal embedding exhibiting $!\mathbb{T}$ as the free symmetric strict monoidal category on $\mathbb{T}$.

It follows that a $\mathbb{T}$-species $\mathsf{P} : !\mathbb{T} \longrightarrow \mathcal{S}\mathit{et}$ describes the structures $\mathsf{P}(\mathsf{T})$ of type P that can be put on bags T of tokens in $\mathbb{T}$ (given by objects in $!\mathbb{T}$) together with compatible rules of transport of structure along $\mathbb{T}$-tagged permutations (given by maps in $!\mathbb{T}$) in the form of actions

$$\_[=] : \mathsf{P}(\mathsf{T}) \times !\mathbb{T}(\mathsf{T}, \mathsf{T}') \longrightarrow \mathsf{P}(\mathsf{T}') \qquad (\mathsf{P} : !\mathbb{T} \longrightarrow \mathcal{S}\mathit{et}, \mathsf{T}, \mathsf{T}' \in !\mathbb{T})$$

for which $p[\mathrm{id}_{\mathsf{T}}] = p$ and $p[\sigma][\tau] = p[\sigma \cdot \tau]$ for all $p \in \mathsf{P}(\mathsf{T})$ and $\sigma : \mathsf{T} \longrightarrow \mathsf{T}'$, $\tau : \mathsf{T}' \longrightarrow \mathsf{T}''$ in $!\mathbb{T}$.

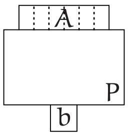
Examples of generalised species in combinatorics abound: permutationals [25, 4] are **CP**-species for **CP** the groupoid of finite cyclic permutations, partitionals [40] are $\mathbf{B}^*$-species for $\mathbf{B}^*$ the groupoid of non-empty finite sets. Further examples are coloured permutationals [34], and species on graphs and digraphs [33].

A fundamental property of the free symmetric (strict) monoidal completion is that it comes equipped with canonical natural coherent equivalences as shown below.

$$\mathbf{1} \underset{\simeq}{\xrightarrow{\mathsf{O}}} !\mathbf{0}\ , \quad !\mathbb{C}_1 \times !\mathbb{C}_2 \underset{\simeq}{\xrightarrow{\otimes}} !(\mathbb{C}_1 + \mathbb{C}_2) : (\mathsf{C}_1, \mathsf{C}_2) \longmapsto\, !\amalg_1(\mathsf{C}_1) \oplus !\amalg_2(\mathsf{C}_2)$$

Thus T-species $!\mathrm{T} \longrightarrow \mathcal{S}\mathit{et}$ are equivalent to functors $\mathbf{B}^{\mathrm{T}} \longrightarrow \mathcal{S}\mathit{et}$, which is the notion of T-sorted species originally introduced by Joyal [25].

Finally, it is important to generalise further; allowing for variable sets of structures. For small categories $\mathbb{A}$ and $\mathbb{B}$, an $(\mathbb{A}, \mathbb{B})$-*species of structures* is defined as a functor $!\mathbb{A} \longrightarrow \widehat{\mathbb{B}}$. The notation $P : \mathbb{A} \;!\!\!\longrightarrow \mathbb{B}$ indicates that P is an $(\mathbb{A}, \mathbb{B})$-species. As before, for such a species P, we have the intuitive reading that $P(A)$ is the $\mathbb{B}^\circ$-variable set of structures of type P on the bag A of tokens in $\mathbb{A}$. However, the definition introduces an asymmetry that naturally leads to think of structures in $P(A)(b)$ as those of type P over a bag A of input tokens (or ports) in $\mathbb{A}$ and (parameterised on) an output token (or port) b in $\mathbb{B}^\circ$. As we will see in Subsect. 2.4 this interpretation is technically correct, and under it structures will be pictorially represented as on the right.



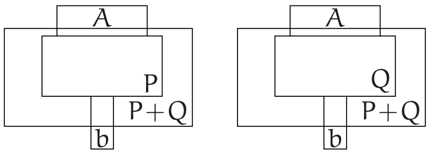
Remark. Below I will be exploiting the fact that species $P : !\mathbb{A} \longrightarrow \widehat{\mathbb{B}}$ are in duality with co-species $P^\perp : \mathbb{B}^\circ \longrightarrow \widehat{!\mathbb{A}^\circ}$ defined as $P^\perp(b)(A) = P(A)(b)$ $(b \in \mathbb{B}^\circ, A \in !\mathbb{A})$.

2.2 Commutative Rig Structure: Addition and Multiplication

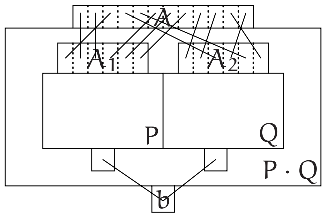
The *zero* species $0 : \mathbb{A} \;!\!\!\longrightarrow \mathbb{B}$ and the *addition* $P + Q : \mathbb{A} \;!\!\!\longrightarrow \mathbb{B}$ of the species $P, Q : \mathbb{A} \;!\!\!\longrightarrow \mathbb{B}$ are defined by

$$0(A)(b) = \emptyset\,, \quad (P+Q)(A)(b) = P(A)(b) + Q(A)(b) \qquad (A \in !\mathbb{A}, b \in \mathbb{B}^\circ)\ .$$

Representations of structures of addition and multiplication type follow. Compare them with the informal description of the addition and multiplication of structures of species given in Subsect. 1.2.



Addition



Multiplication

As in the previous section, Day's tensor construction [8, 23], provides a multiplication symmetric tensor product induced by the free symmetric strict monoidal structure. The *one* species $1 : \mathbb{A} \;!\!\!\longrightarrow \mathbb{B}$ and the *multiplication* $P \cdot Q : \mathbb{A} \;!\!\!\longrightarrow \mathbb{B}$ of the species $P, Q : \mathbb{A} \;!\!\!\longrightarrow \mathbb{B}$ are defined as

$$\begin{aligned} &1(A)(b) = !\mathbb{A}(O, A) \\ &(P \cdot Q)(A)(b) & (A \in !\mathbb{A}, b \in \mathbb{B}^\circ)\ . \\ &\quad = \int^{A_1, A_2 \in !\mathbb{A}} P(A_1)(b) \times Q(A_2)(b) \times !\mathbb{A}(A_1 \oplus A_2, A) \end{aligned}$$

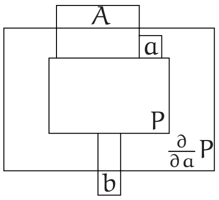
Remark. More succinctly, we have that $P + Q = +\langle P, Q \rangle$ and that $(P \cdot Q)^\perp = \widehat{\oplus}\,\langle P^\perp, Q^\perp \rangle$.

2.3 Differential Structure: Partial Derivatives

For $a \in \mathbb{A}$, the *partial derivative* $\frac{\partial}{\partial a} P : \mathbb{A} \,{}^{!}\!\!\longrightarrow \mathbb{B}$ of the species $P : \mathbb{A} \,{}^{!}\!\!\longrightarrow \mathbb{B}$ is defined as

$$\left(\tfrac{\partial}{\partial a} P\right)(A)(b) = P(A \oplus \{\!\!\{a\}\!\!\})(b) \qquad\qquad (A \in !\mathbb{A}, b \in \mathbb{B}^\circ) \ .$$

Structures of partial-derivative type may be represented as on the left below.



Partial Derivative

$$\begin{tikzcd}
!\mathbb{A}^\circ \arrow[r, hook, "y"] \arrow[d, "- \oplus \{\!\!\{a\}\!\!\}"'] \arrow[dr, phantom, "\overset{\mathrm{Lan}}{\cong}"] & \mathbf{Set}^{!\mathbb{A}} \arrow[d, shift right=6, "- \widehat{\oplus} y\{\!\!\{a\}\!\!\}"'] \arrow[d, phantom, "\dashv \, d/da \, \dashv"] \arrow[d, shift left=6] \\
!\mathbb{A}^\circ \arrow[r, hook, "y"'] & \mathbf{Set}^{!\mathbb{A}} \arrow[u, "d/da" description]
\end{tikzcd}$$

Remark. As in the previous section, the construction of partial derivatives arises from the situation on the right above. Indeed, we have that $(\frac{\partial}{\partial a} P)^\perp = (d/da)\, P^\perp$.

2.4 Bicategorical Structure: Identities and Composition

The *identity* species $\mathrm{I}_\mathbb{C} : \mathbb{C} \,{}^{!}\!\!\longrightarrow \mathbb{C}$ is defined as

$$\mathrm{I}_\mathbb{C}(C)(c) = !\mathbb{C}\left(\{\!\!\{c\}\!\!\}, C\right) \qquad\qquad (C \in !\mathbb{C}, c \in \mathbb{C}^\circ) \ .$$

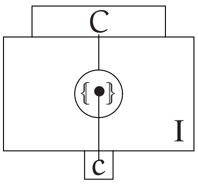
For species $P : \mathbb{A} \,{}^{!}\!\!\longrightarrow \mathbb{B}$ and $Q : \mathbb{B} \,{}^{!}\!\!\longrightarrow \mathbb{C}$, the *composition* $Q \circ P : \mathbb{A} \,{}^{!}\!\!\longrightarrow \mathbb{C}$ is defined as

$$(Q \circ P)(A)(c) = \int^{B \in !\mathbb{B}} Q(B)(c) \times P^\#(A)(B) \qquad\qquad (A \in !\mathbb{A}, c \in \mathbb{C}^\circ)$$

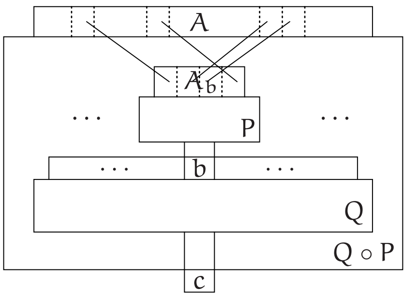
where

$$\begin{aligned} & P^\#(A)(B) \\ & \quad = \int^{X \in (!\mathbb{A})^{|B|}} \left(\textstyle\prod_{k \in |B|} P(X_k)(B_k)\right) \times !\mathbb{A}\left(\textstyle\bigoplus_{k \in |B|} X_k, A\right) \end{aligned} \qquad (A \in !\mathbb{A}, B \in !\mathbb{B}^\circ) \ .$$

One may visualise identities and composition as follows.



Identity



Composition

Remark. Using the universal properties of both $!(\_)$ and $\widehat{(\_)}$, we obtain the following situation

$$\begin{tikzcd}
\mathbb{B}^\circ \arrow[r, "\{\!\!\{-\}\!\!\}"] \arrow[dr, "F"'] \arrow[dr, phantom, bend left, "\cong"] & !\mathbb{B}^\circ \arrow[r, hook, "y"] \arrow[d, "F^{\oplus(-)}"] \arrow[dr, phantom, "\overset{\mathrm{Lan}}{\cong}"] & \mathit{Set}^{!\mathbb{B}} \arrow[dl, "\_ \bullet F"'] \arrow[dl, phantom, bend left, "\dashv"] \\
 & \mathit{Set}^{!\mathbb{A}} \arrow[ur, bend right] &
\end{tikzcd}$$

where $F^{\oplus B} = \widehat{\bigoplus}_{k\in|B|} F(B_k)$. We have that $(Q \circ P)^\perp$ is obtained as $(\_ \bullet P^\perp)\, Q^\perp$.

2.5 Cartesian Closed Structure: Product and Exponentiation

The cartesian closed structure of generalised species is presented.

There is exactly one species $\mathbb{C} \mathrel{!\!\!\longrightarrow} \top$ for $\top = \mathbf{0}$. More generally, for a family $P_i : \mathbb{C} \mathrel{!\!\!\longrightarrow} \mathbb{C}_i$ $(i \in I)$, the *pairing* $\langle P_i\rangle_{i\in I} : \mathbb{C} \mathrel{!\!\!\longrightarrow} \sqcap_{i\in I}\mathbb{C}_i$, where $\sqcap_{i\in I}\mathbb{C}_i = \sum_{i\in I}\mathbb{C}_i$, is defined as

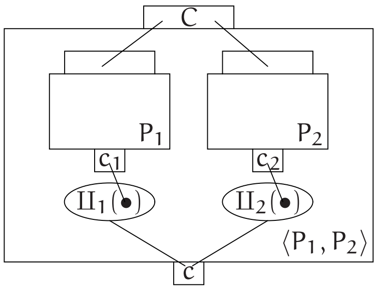
$$\begin{aligned}&\langle P_i\rangle_{i\in I}(C)(c)\\ &\quad = \textstyle\sum_{i\in I}\int^{z\in\mathbb{C}_i} P_i(C)(z) \times (\sqcap_{i\in I}\mathbb{C}_i)\big(c, \amalg_i(z)\big)\end{aligned} \qquad (C \in !\mathbb{C}, c \in \sqcap_{i\in I}\mathbb{C}_i{}^\circ) \ .$$

For $i \in I$, the *projection* $\pi_i : \sqcap_{i\in I}\mathbb{C}_i \mathrel{!\!\!\longrightarrow} \mathbb{C}_i$ is defined as

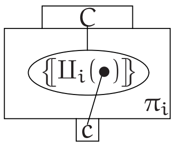
$$\pi_i(C)(c) = !(\sqcap_{i\in I}\mathbb{C}_i)\big(\{\!\!\{\amalg_i(c)\}\!\!\}, C\big) \qquad (C \in !(\sqcap_{i\in I}\mathbb{C}_i), c \in \mathbb{C}_i{}^\circ) \ .$$

From the logical point of view, and using relational notation, $C\,\big[\,\langle P_i\rangle_{i\in I}\,\big]\,c$ is the extent to which there exists $i \in I$ and $c_i \in \mathbb{C}_i$ such that $C\,[P_i]\,c_i$ and c approximates $\amalg_i(c_i)$; whilst $C\,[\pi_i]\,c$ is the extent to which $\{\!\!\{\amalg_i(c)\}\!\!\}$ approximates C.

Pairing and projection may be depicted as follows.



Pairing



Projection

For $P : \mathbb{C} \sqcap \mathbb{A} \mathrel{!\!\!\longrightarrow} \mathbb{B}$, the *abstraction*

$$\lambda_{\mathbb{A}} P : \mathbb{C} \mathrel{!\!\!\longrightarrow} \underline{hom}(\mathbb{A}, \mathbb{B}) \quad \text{where } \underline{hom}(\mathbb{A}, \mathbb{B}) = !\mathbb{A}^\circ \times \mathbb{B}$$

is defined as

$$(\lambda_{\mathbb{A}} P)(C)(A, b) = P(C \otimes A)(b) \qquad (C \in !\mathbb{C}, A \in !\mathbb{A}, b \in \mathbb{B}^\circ) \ ,$$

where recall from Subsect. 2.1 that $C \otimes A = !\amalg_1 C \oplus !\amalg_2 A$. The *evaluation*

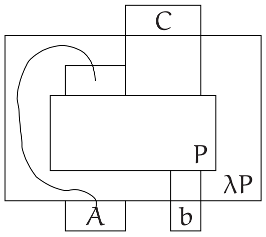
$$\varepsilon_{\mathbb{A},\mathbb{B}} : \underline{hom}(\mathbb{A},\mathbb{B}) \sqcap \mathbb{A} \,{}_{!}\!\!\longrightarrow \mathbb{B}$$

is defined as

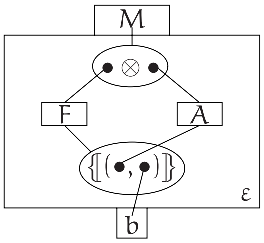
$$\begin{aligned}&\varepsilon_{\mathbb{A},\mathbb{B}}(M)(b) && (M \in !(\underline{hom}(\mathbb{A},\mathbb{B}) \sqcap \mathbb{A}), b \in \mathbb{B}^\circ)\\ &= \int^{F \in !\underline{hom}(\mathbb{A},\mathbb{B}),\, A \in !\mathbb{A}} !\underline{hom}(\mathbb{A},\mathbb{B})(\{\![(A,b)]\!\}, F) \times !\big(\underline{hom}(\mathbb{A},\mathbb{B}) \sqcap \mathbb{A}\big)(F \otimes A, M) \ .\end{aligned}$$

Again from the logical point of view, and using relational notation, we have that $C\,[\lambda_{\mathbb{A}} P]\,(A,b)$ iff $(C \otimes A)\,[P]\,b$; whilst $M\,[\varepsilon_{\mathbb{A},\mathbb{B}}]\,b$ is the extent to which the (step) function $\{\![(A,b)]\!\}$ approximates F, where $M = F \otimes A$ consists of a function F and an argument A.

Schematically, we have the following.



Abstraction



Evaluation

2.6 Higher-Order Differential Structure: Differentiation Operator

For a thorough treatment of differentiation one needs to introduce linear homs. In the current setting they are naturally given by

$$\underline{\ell in}(\mathbb{A},\mathbb{B}) = \mathbb{A}^\circ \times \mathbb{B} \ .$$

With this in place, I can introduce an operator that internalises partial derivatives (and differential application) and satisfies all the basic properties of differentiation.

The *differentiation operator*

$$D_{\mathbb{A},\mathbb{B}} : \underline{hom}(\mathbb{A},\mathbb{B}) \,{}_{!}\!\!\longrightarrow \underline{hom}(\mathbb{A}, \underline{\ell in}(\mathbb{A},\mathbb{B}))$$

is given by

$$D_{\mathbb{A},\mathbb{B}}(F)(A,a,b) = !\underline{hom}(\mathbb{A},\mathbb{B})\big(\{\![(A \oplus \{\![a]\!\}, b)]\!\}, F\big) \quad \begin{array}{l}(F \in !\underline{hom}(\mathbb{A},\mathbb{B}),\\ \ A \in !\mathbb{A}, a \in \mathbb{A}, b \in \mathbb{B}^\circ) \ .\end{array}$$

2.7 Outline of the Calculus

Elsewhere I will give a formal presentation of the calculus of generalised species of structures and indicate how it is justified within the mathematical framework of generalised logic [28]. Here I will just offer an outline.

$$
\begin{aligned}
&\big(\varepsilon \circ \langle \lambda(P) \circ \pi_1, \pi_2 \rangle\big)(D)(b) \qquad\qquad (D \in {!}(\mathbb{C} \sqcap \mathbb{A}), b \in \mathbb{B}^\circ) \\
&\overset{(1)}{\cong} \int^{M \in {!}(\underline{hom}(\mathbb{A},\mathbb{B}) \sqcap \mathbb{A})} \varepsilon(M)(b) \times \langle \lambda(P) \circ \pi_1, \pi_2 \rangle^{\#}(D)(M) \\
&\overset{(2)}{\cong} \int^{M \in {!}(\underline{hom}(\mathbb{A},\mathbb{B}) \sqcap \mathbb{A})} \int^{F \in {!}\underline{hom}(\mathbb{A},\mathbb{B}), A \in {!}\mathbb{A}} {!}\underline{hom}(\mathbb{A},\mathbb{B})\big(\{\![(A,b)]\!\}, F\big) \times {!}(\underline{hom}(\mathbb{A},\mathbb{B}) \sqcap \mathbb{A})\,({!}\amalg_1 F \oplus {!}\amalg_2 A, M) \\
&\qquad\qquad \times \langle \lambda(P) \circ \pi_1, \pi_2 \rangle^{\#}(D)(M) \\
&\overset{(3)}{\cong} \int^{A \in {!}\mathbb{A}} \langle \lambda(P) \circ \pi_1, \pi_2 \rangle^{\#}(D)({!}\amalg_1 \{\![(A,b)]\!\} \oplus {!}\amalg_2 A) \\
&\overset{(4)}{\cong} \int^{A \in {!}\mathbb{A}} \int^{D_1, D_2 \in {!}(\mathbb{C} \sqcap \mathbb{A})} \langle \lambda(P) \circ \pi_1, \pi_2 \rangle^{\#}(D_1)({!}\amalg_1 \{\![(A,b)]\!\}) \times \langle \lambda(P) \circ \pi_1, \pi_2 \rangle^{\#}(D_2)({!}\amalg_2 A) \\
&\qquad\qquad \times {!}(\mathbb{C} \sqcap \mathbb{A})\,(D_1 \oplus D_2, D) \\
&\overset{(5)}{\cong} \int^{A \in {!}\mathbb{A}} \int^{D_1, D_2 \in {!}(\mathbb{C} \sqcap \mathbb{A})} (\lambda(P) \circ \pi_1)^{\#}(D_1)\{\![(A,b)]\!\} \times {\pi_2}^{\#}(D_2)(A) \\
&\qquad\qquad \times {!}(\mathbb{C} \sqcap \mathbb{A})\,(D_1 \oplus D_2, D) \\
&\overset{(6)}{\cong} \int^{A \in {!}\mathbb{A}} \int^{D_1, D_2 \in {!}(\mathbb{C} \sqcap \mathbb{A})} (\lambda(P) \circ \pi_1)(D_1)(A,b) \times {!}(\mathbb{C} \sqcap \mathbb{A})\,({!}\amalg_2 A, D_2) \\
&\qquad\qquad \times {!}(\mathbb{C} \sqcap \mathbb{A})\,(D_1 \oplus D_2, D) \\
&\overset{(7)}{\cong} \int^{A \in {!}\mathbb{A}} \int^{D_1 \in {!}(\mathbb{C} \sqcap \mathbb{A})} \int^{C \in {!}\mathbb{C}} (\lambda P)(C)(A,b) \times {\pi_1}^{\#}(D_1)(C) \times {!}(\mathbb{C} \sqcap \mathbb{A})\,(D_1 \oplus {!}\amalg_2 A, D) \\
&\overset{(8)}{\cong} \int^{A \in {!}\mathbb{A}} \int^{D_1 \in {!}(\mathbb{C} \sqcap \mathbb{A})} \int^{C \in {!}\mathbb{C}} P({!}\amalg_1 C \oplus {!}\amalg_2 A)(b) \times {!}(\mathbb{C} \sqcap \mathbb{A})\,({!}\amalg_1 C, D_1) \\
&\qquad\qquad \times {!}(\mathbb{C} \sqcap \mathbb{A})\,(D_1 \oplus {!}\amalg_2 A, D) \\
&\overset{(9)}{\cong} \int^{A \in {!}\mathbb{A}} \int^{C \in {!}\mathbb{C}} P({!}\amalg_1 C \oplus {!}\amalg_2 A)(b) \times {!}(\mathbb{C} \sqcap \mathbb{A})\,({!}\amalg_1 C \oplus {!}\amalg_2 A, D) \\
&\overset{(10)}{\cong} P(D)(b)
\end{aligned}
$$

Fig. 1. An equational proof of the beta isomorphism $\varepsilon \circ \langle \lambda(P) \circ \pi_1, \pi_2 \rangle \cong P : \mathbb{C} \sqcap \mathbb{A} \mathrel{!\!\longrightarrow} \mathbb{B}$

(1) Definition of composition. (2) Definition of evaluation. (3) Density formula. (4) Law of extensions. (5) Law of extensions and definition of pairing. (6) Law of extensions and definition of projection. (7) Density formula and definition of composition. (8) Law of extensions and definitions of projection and abstraction. (9–10) Density formula and properties of the free symmetric (strict) monoidal completion.

Identities and composition come with canonical natural coherent isomorphisms establishing the unit laws of identities and the associativity of composition. Addition and multiplication yield a commutative rig structure, and commute with pre-composition.

The usual laws of pairing and projection, and of abstraction and evaluation are satisfied up to isomorphism (see Fig. 1 for a proof outline of the beta isomorphism). Thus, the closed structure $\underline{hom}$ comes equipped with internal identities and composition. Also the linear homs $\underline{\ell in}$ come equipped with internal identities and compositions, that actually embed in the closed structure.

Partial derivatives commute between themselves, addition, and multiplication by scalars. Moreover, they satisfy both the Leibniz (or product) and chain rules. For instance, the central reason for which the former holds is that the canonical map

$$(\dagger) \quad \begin{array}{l} \int^{A_1' \in !\mathbb{A}} !\mathbb{A}\left(A_1, A_1' \oplus \{\!\!\{a\}\!\!\}\right) \times !\mathbb{A}(A_1' \oplus A_2, A) \\ \quad + \int^{A_2' \in !\mathbb{A}} !\mathbb{A}\left(A_2, A_2' \oplus \{\!\!\{a\}\!\!\}\right) \times !\mathbb{A}(A_1 \oplus A_2', A) \\ \qquad \xrightarrow{\cong} !\mathbb{A}\left(A_1 \oplus A_2, A \oplus \{\!\!\{a\}\!\!\}\right) \end{array} \qquad \begin{array}{r} (A_1, A_2 \in !\mathbb{A}^\circ, \\ A \in !\mathbb{A}, a \in \mathbb{A}) \end{array}$$

(given by tensoring and composing) is a natural isomorphism. Indeed, the definitions of multiplication and partial derivation yield

$$\begin{array}{ll} \left(\frac{\partial}{\partial a}(P \cdot Q)\right)(A)(b) & \qquad (a \in \mathbb{A}, P, Q : \mathbb{A} \mathrel{!\!\!\longrightarrow} \mathbb{B}, A \in !\mathbb{A}, b \in \mathbb{B}^\circ) \\ \quad = \int^{A_1, A_2 \in !\mathbb{A}} P(A_1)(b) \times Q(A_2)(b) \times !\mathbb{A}\left(A_1 \oplus A_2, A \oplus \{\!\!\{a\}\!\!\}\right) & \end{array}$$

which by (†) above, using various distributivity and commutativity laws, and the density formula, is natural isomorphic to

$$\begin{array}{l} \int^{A_2, A_1' \in !\mathbb{A}} P(A_1' \oplus \{\!\!\{a\}\!\!\})(b) \times Q(A_2)(b) \times !\mathbb{A}(A_1' \oplus A_2, A) \\ \quad + \int^{A_1, A_2' \in !\mathbb{A}} P(A_1)(b) \times Q(A_2' \oplus \{\!\!\{a\}\!\!\})(b) \times !\mathbb{A}(A_1 \oplus A_2', A) \\ \quad = \left(\frac{\partial}{\partial a}(P) \cdot Q + P \cdot \frac{\partial}{\partial a}(Q)\right)(A)(b) \ . \end{array}$$

Further, the differentiation operator, which internalises partial derivation, is a linear operator that is constant on linear maps.

Interestingly, a certain commutation law between abstraction and linear application (used on differentiation) entails the beta rule of the differential lambda calculus of Ehrhard and Regnier [10] as an isomorphism.

3 Concluding Remarks and Research Perspectives

I have drawn a line of investigation concerning models of computational and combinatorial structures. The general common theme of these models is that they live in mathematical universes of variable sets. My presentation here aimed at making explicit and apparent the commonalities amongst the models. In particular, I have placed emphasis in considering the various models of variation as universal constructions; showing how their structure induces relevant further structure on the associated universe of variable sets.

The models touched upon in Sect. 1 and their applications should not be considered in isolation for they are closely related. In this respect, there is a submodel of $\mathcal{S}et^{\mathbf{I}}$, the so-called Schanuel topos (see, *e.g.*, [32, 24]), that occupies an interesting place. Indeed, it has been used both for giving denotational models of dynamically generated names [46, 47] and for modelling and reasoning about abstract syntax with variable binding [20]. Further, it is closely related to the category of species $\mathcal{S}et^{\mathbf{B}}$ [11, 35], which in turn has also been considered as a model of abstract syntax with linear variable-binding [49]. These models are by no means the only relevant for applications, and a fully systematic theory providing, for instance, constructions of models of variation that are guaranteed to properly model specific (classes of) computation structures is not yet in place.

The analysis of Sect. 1 suggests both the unification and generalisation of models, and in the latter vein I motivated and introduced generalised species of structures; see [2, 36, 7] for relevant related work. These generalised species extend various of the notions of species used in combinatorics and also their respective calculi. Indeed, they come equipped with an (heterogeneous) notion of substitution (composition) structuring them into a bicategory, which arises as from models of linear logic by a co-Kleisli construction (see [7–Sect. 9]) and supports linear and cartesian closed structure allowing for a full development of the differential calculus. Further, the setting also provides graph-like models of the lambda calculus, fixed-point operators, *etc*.

As it is the case for the basic notion of species (see [26]), generalised species of structures can be equivalently seen as generalised analytic functors (of which generalised species are the coefficients) between categories of variable sets. From this point of view, the identities and composition defined in Subsect. 2.4 respectively correspond to the usual identities and composition of functors. Interestingly, restricting attention to groupoids (which is the situation considered in combinatorics) there is an intrinsic characterisation of generalised analytic functors that places them in the context of categorical stable domain theory.

It would be important if the aforementioned structure of generalised species gave new applications in combinatorics, or could be used to tackle combinatorial problems.

I have emphasised that the calculus of generalised species can be axiomatically built on top of the mathematical framework of generalised logic. This, besides yielding new algebraic proofs, provides connections with other areas of mathematics and suggests a calculus of enriched generalised species of structures. In particular, enriching over the Sierpinski space places the subject in the context of domain theory.

As for other perspectives, motivated by a conversation with Prakash Panangaden, I was lead to consider the free symmetric (strict) monoidal completion as a symmetric Fock-space construction (see, *e.g.*, [21–Chap. 21]); and indeed, one can introduce the operators of creation and annihilation of particles in the quantum systems that these spaces model and establish their commutation laws. In this line of thought and further motivated by [6, 3], I was considering Feynman diagrams in the context of generalised species when a computational interpretation of my previous calculations became apparent. The outcome of these investigations will be reported elsewhere. Here however I would like to conclude the paper with an informal presentation of three illustrative examples.

1. The density formula

$$\int^{c\in\mathbb{C}} \mathsf{P}(c)\times\mathbb{C}(d,c)\cong\mathsf{P}(d) \qquad\qquad (\mathsf{P}\in\widehat{\mathbb{C}},\ d\in\mathbb{C}^{\circ})$$

amounts to the basic form of action

$$\big(c:\mathbb{C}\big)\big[\ [\mathsf{P}]c\rangle\ ,\ \langle c[\mathbb{C}]d\rangle\ \big]\ \approx\ [\mathsf{P}]d\rangle$$

with the following data flow reading: the agent P with local port c of sort $\mathbb{C}$ bound to the datum d results in the agent P with the datum d.

$$\langle\, D\,[\varepsilon\circ\langle(\lambda P)\circ\pi_1,\pi_2\rangle]\,b\,\rangle \qquad\qquad (D : !(\mathbb{C}\sqcap\mathbb{A}), P : \mathbb{C}\sqcap\mathbb{A}\mathrel{!\!\!\longrightarrow}\mathbb{B}, b : \mathbb{B}^\circ)$$

$$\begin{aligned}
&\overset{(1)}{\approx}\ \big(M : !(\underline{\mathit{hom}}(\mathbb{A},\mathbb{B})\sqcap\mathbb{A})\big)\\
&\qquad \big[\,\langle\, D\,[\langle(\lambda P)\circ\pi_1,\pi_2\rangle^{\#}]\,M\,\rangle\,,\ \langle\, M\,[\varepsilon]\,b\,\rangle\,\big]\\
&\overset{(2)}{\approx}\ \big(M : !(\underline{\mathit{hom}}(\mathbb{A},\mathbb{B})\sqcap\mathbb{A})\big)\\
&\qquad \Big[\,\langle\, D\,[\langle(\lambda P)\circ\pi_1,\pi_2\rangle^{\#}]\,M\,\rangle\,,\\
&\qquad\quad (F : !\underline{\mathit{hom}}(\mathbb{A},\mathbb{B}), A : !\mathbb{A})\\
&\qquad\qquad \big[\,\langle\, M\,[!(\underline{\mathit{hom}}(\mathbb{A},\mathbb{B})\sqcap\mathbb{A})]\,!\amalg_1 F\oplus !\amalg_2 A\,\rangle\,,\ \langle\, F\,[!\underline{\mathit{hom}}(\mathbb{A},\mathbb{B})]\,\{\![(A,b)]\!\}\,\rangle\,\big]\,\Big]\\
&\overset{(3)}{\approx}\ (A : !\mathbb{A})\\
&\qquad \big[\,\langle\, D\,[\langle(\lambda P)\circ\pi_1,\pi_2\rangle^{\#}]\,!\amalg_1\{\![(A,b)]\!\}\oplus !\amalg_2 A\,\rangle\,\big]\\
&\overset{(4)}{\approx}\ (A : !\mathbb{A}, D_1, D_2 : !(\mathbb{C}\sqcap\mathbb{A}))\\
&\qquad \big[\,\langle\, D\,[!(\mathbb{C}\sqcap\mathbb{A})]\,D_1\oplus D_2\,\rangle\,,\\
&\qquad\ \ \langle\, D_1\,[\langle(\lambda P)\circ\pi_1,\pi_2\rangle^{\#}]\,!\amalg_1\{\![(A,b)]\!\}\,\rangle\,,\ \langle\, D_2\,[\langle(\lambda P)\circ\pi_1,\pi_2\rangle^{\#}]\,!\amalg_2 A\,\rangle\,\big]\\
&\overset{(5)}{\approx}\ (A : !\mathbb{A}, D_1, D_2 : !(\mathbb{C}\sqcap\mathbb{A}))\\
&\qquad \big[\,\langle\, D\,[!(\mathbb{C}\sqcap\mathbb{A})]\,D_1\oplus D_2\,\rangle\,,\ \langle\, D_1\,[((\lambda P)\circ\pi_1)^{\#}]\,\{\![(A,b)]\!\}\,\rangle\,,\ \langle\, D_2\,[\pi_2{}^{\#}]\,A\,\rangle\,\big]\\
&\overset{(6)}{\approx}\ (A : !\mathbb{A}, D_1, D_2 : !(\mathbb{C}\sqcap\mathbb{A}))\\
&\qquad \big[\,\langle\, D\,[!(\mathbb{C}\sqcap\mathbb{A})]\,D_1\oplus D_2\,\rangle\,,\ \langle\, D_1\,[(\lambda P)\circ\pi_1]\,(A,b)\,\rangle\,,\ \langle\, D_2\,[!(\mathbb{C}\sqcap\mathbb{A})]\,!\amalg_2 A\,\rangle\,\big]\\
&\overset{(7)}{\approx}\ (A : !\mathbb{A}, D_1 : !(\mathbb{C}\sqcap\mathbb{A}))\\
&\qquad \Big[\,\langle\, D\,[!(\mathbb{C}\sqcap\mathbb{A})]\,D_1\oplus !\amalg_2 A\,\rangle\,,\ (C : !\mathbb{C})\big[\langle\, D_1\,[\pi_1{}^{\#}]\,C\,\rangle, \langle\, C\,[\lambda P]\,(A,b)\,\rangle\big]\,\Big]\\
&\overset{(8)}{\approx}\ (A : !\mathbb{A}, D_1 : !(\mathbb{C}\sqcap\mathbb{A}), C : !\mathbb{C})\\
&\qquad \big[\,\langle\, D\,[!(\mathbb{C}\sqcap\mathbb{A})]\,D_1\oplus !\amalg_2 A\,\rangle\,,\ \langle\, D_1\,[!(\mathbb{C}\sqcap\mathbb{A})]\,!\amalg_1 C\,\rangle\,,\ \langle\, !\amalg_1 C\oplus !\amalg_2 A\,[P]\,b\,\rangle\,\big]\\
&\overset{(9)}{\approx}\ (A : !\mathbb{A}, C : !\mathbb{C})\\
&\qquad \big[\,\langle\, D\,[!(\mathbb{C}\sqcap\mathbb{A})]\,!\amalg_1 C\oplus !\amalg_2 A\,\rangle\,,\ \langle\, !\amalg_1 C\oplus !\amalg_2 A\,[P]\,b\,\rangle\,\big]\\
&\overset{(10)}{\approx}\ \langle\, D\,[P]\,b\,\rangle
\end{aligned}$$

Fig. 2. A computational interpretation of the beta isomorphism

(1) Definition of composition. (2) Definition of evaluation. (3) Laws of data flow. (4) Law of extensions. (5) Law of extensions and definition of pairing. (6) Law of extensions and definition of projection. (7) Law of data flow and definition of composition. (8) Law of extensions and definitions of projection and abstraction. (9–10) Laws of data flow.

2. The isomorphism

$$!(\mathbb{A}\sqcap\mathbb{B})(A'\otimes B', A\otimes B)\cong !\mathbb{A}(A',A)\times !\mathbb{B}(B',B) \qquad \begin{array}{r}(A,A'\in !\mathbb{A},\\ B,B'\in !\mathbb{B})\end{array}$$

amounts to having the law of data flow

$$\langle\, A\otimes B\,[!(\mathbb{A}\sqcap\mathbb{B})]\,A'\otimes B'\,\rangle \ \approx\ \langle\, A\,[!\mathbb{A}]\,A'\,\rangle\,,\ \langle\, B\,[!\mathbb{B}]\,B'\,\rangle$$

establishing that a link between $\mathsf{A} \otimes \mathsf{B}$ and $\mathsf{A}' \otimes \mathsf{B}'$ of type $!(\mathbb{A} \sqcap \mathbb{B})$ amounts to a link of type $!\mathbb{A}$ between A and A' and one of type $!\mathbb{B}$ between B and B'.

3. The computational interpretation of the beta isomorphism in Fig. 1, translated into the informal syntax of agents used in the above two examples, is given in Fig. 2.

References

1. P. Aczel. Frege structures and the notions of proposition, truth and set. In *The Kleene Symposium*, pages 31–60. North-Holland, 1980.
2. J. Baez and J. Dolan. Higher-dimensional algebra III: n-categories and the algebra of opetopes. *Advances in Mathematics*, 135:145–206, 1998.
3. J. Baez and J. Dolan. From finite sets to Feynman diagrams. In B. Engquist and W.Schmid, editors, *Mathematics Unlimited - 2001 and Beyond*, pages 29–50. Springer-Verlag, 2001.
4. F. Bergeron. Une combinatoire du pléthysme. *Journal of Combinatorial Theory (Series A)*, 46:291–305, 1987.
5. F. Bergeron, G. Labelle, and P. Leroux. *Combinatorial species and tree-like structures*, volume 67 of *Encyclopedia of mathematics and its applications*. Cambridge University Press, 1998.
6. R. Blute and P. Panangaden. Proof nets and Feynman diagrams. Available from the second author, 1998.
7. G. Cattani and G. Winskel. Profunctors, open maps and bisimulation. BRICS Report Series RS-04-22, University of Aarhus, 2004.
8. B. Day. On closed categories of functors. In *Reports of the Midwest Category Seminar IV*, volume 137 of *Lecture Notes in Mathematics*, pages 1–38. Springer-Verlag, 1970.
9. R. Di Cosmo. *Isomorphisms of types: from λ-calculus to information retrieval and language design*. Birkhauser, 1995.
10. T. Ehrhard and L. Regnier. The differential lambda calculus. *Theoretical Computer Science*, 309:1–41, 2003.
11. M. Fiore. Notes on combinatorial functors. Draft available electronically, January 2001.
12. M. Fiore. Rough notes on presheaves. Manuscript available electronically, July 2001.
13. M. Fiore. Semantic analysis of normalisation by evaluation for typed lambda calculus. In *Proceedings of the 4th International Conference on Principles and Practice of Declarative Programming (PPDP 2002)*, pages 26–37. ACM Press, 2002.
14. M. Fiore. Isomorphisms of generic recursive polynomial types. In *Proceedings of the 31st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2004)*, pages 77–88. ACM Press, 2004.
15. M. Fiore, R. Di Cosmo, and V. Balat. Remarks on isomorphisms in typed lambda calculi with empty and sum types. In *Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science (LICS'02)*, pages 147–156. IEEE, Computer Society Press, 2002.
16. M. Fiore and T. Leinster. An objective representation of the Gaussian integers. *Journal of Symbolic Computation*, 37(6):707–716, 2004.
17. M. Fiore, E. Moggi, and D. Sangiorgi. A fully-abstract model for the pi-calculus. *Information and Computation*, 178:1–42, 2002. (Extended abstract in *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science (LICS 96)*, pages 43–54, IEEE, Computer Society Press, 1996.).
18. M. Fiore, G. Plotkin, and D. Turi. Abstract syntax and variable binding. In *Proceedings of the 14th Annual IEEE Symposium on Logic in Computer Science (LICS'99)*, pages 193–202. IEEE, Computer Society Press, 1999.

19. M. Fiore and D. Turi. Semantics of name and value passing. In *Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science (LICS'01)*, pages 93–104. IEEE, Computer Society Press, 2001.
20. M. J. Gabbay and A. Pitts. A new approach to abstract syntax with variable binding. *Formal Aspects of Computing*, 13:341–363, 2002. (See also *A new approach to abstract syntax involving binders* in *Proceedings of the 14th Annual IEEE Symposium on Logic in Computer Science (LICS'99)*, pages 214–224, IEEE, Computer Society Press, 1999.).
21. R. Geroch. *Mathematical Physics*. Chicago Lectures in Physics. The University of Chicago Press, 1985.
22. M. Hofmann. Semantical analysis of higher order abstract syntax. In *Proceedings of the 14th Annual IEEE Symposium on Logic in Computer Science (LICS'99)*, pages 204–213. IEEE, Computer Society Press, 1999.
23. G. Im and G. M. Kelly. A universal property of the convolution monoidal structure. *Journal of Pure and Applied Algebra*, 43:75–88, 1986.
24. P. Johnstone. *Sketches of an Elephant: A Topos Theory Compendium*, volume 43 of *Oxford Logic Guides*. Oxford University Press, 2002.
25. A. Joyal. Une theorie combinatoire des séries formelles. *Advances in Mathematics*, 42:1–82, 1981.
26. A. Joyal. Foncteurs analytiques et especès de structures. In *Combinatoire énumérative*, volume 1234 of *Lecture Notes in Mathematics*, pages 126–159. Springer-Verlag, 1986.
27. G. M. Kelly. Clubs and data-type constructors. In *Applications of Categories in Computer Science*, volume 177 of *London Mathematical Society Lecture Notes Series*, pages 163–190. Cambridge University Press, 1992.
28. F. W. Lawvere. Metric spaces, generalized logic and closed categories. *Rend. del Sem. Mat. e Fis. di Milano*, 43:135–166, 1973. (Also in *Reprints in Theory and Applications of Categories*, 1:1–37, 2002.).
29. F. W. Lawvere. Qualitative distinctions between some toposes of generalized graphs. In *Proceedings of the AMS 1987 Symposium on Categories in Computer Science and Logic*, volume 92 of *Contemporary Mathematics*, pages 261–299, 1989.
30. F. W. Lawvere and R. Rosebrugh. *Sets for Mathematics*. Cambridge University Press, 2001.
31. S. Mac Lane. *Categories for the Working Mathematician*. Springer-Verlag, 1971. (Revised edition 1998).
32. S. Mac Lane and I. Moerdijk. *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Universitext. Springer-Verlag, 1992.
33. M. Méndez. Species on digraphs. *Advances in Mathematics*, 123(2):243–275, 1996.
34. M. Méndez and O. Nava. Colored species, c-monoids and plethysm, I. *Journal of Combinatorial Theory (Series A)*, 64:102–129, 1993.
35. M. Menni. About И-quantifiers. *Applied Categorical Structures*, 11(5):421–445, 2003.
36. M. Menni. Symmetric monoidal completions and the exponential principle among labeled combinatorial structures. *Theory and Applications of Categories*, 11:397–419, 2003.
37. R. Milner. What's in a name? In *Computer Systems: Theory, Technology and Applications*, Monographs in Computer Science. Springer-Verlag, 2003.
38. R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, I and II. *Information and Computation*, 100(1):1–77, 1992.
39. E. Moggi. An abstract view of programming languages. LFCS report ECS-LFCS-90-113, University of Edinburgh, 1990.
40. O. Nava and G.-C. Rota. Plethysm, categories and combinatorics. *Advances in Mathematics*, 58:61–88, 1985.
41. R. Needham. *Distributed Systems*, chapter 12, pages 315–328. Addison-Wesley, second edition, 1993.

42. F. Oles. Type algebras, functor categories and block structure. In *Algebraic Methods in Semantics*. Cambridge University Press, 1985.
43. F. Pfenning and C. Elliot. Higher-order abstract syntax. In *Proceedings of the ACM SIGPLAN'88 Symposium on Language Design and Implementation*, 1988.
44. D. Rajan. The adjoints to the derivative functor on species. *Journal of combinatorial theory (Series A)*, 62:93–106, 1993.
45. J. Reynolds. The essence of Algol. In *Proceedings of the International Symposium on Algorithmic Languages*, pages 345–372. North-Holland, 1981.
46. I. Stark. *Names and Higher-Order Functions*. Ph.D. thesis, University of Cambridge, 1994.
47. I. Stark. A fully abstract domain model for the pi-calculus. In *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science (LICS'96)*, pages 36–42. IEEE, Computer Society Press, 1996.
48. R. Street. The role of Michael Batanin's monoidal globular categories. In *Higher Category Theory*, volume 230 of *Contemporary Mathematics*, pages 99–116. A.M.S., 1998.
49. M. Tanaka. Abstract syntax and variable binding for linear binders. In *Proceedings of the 25th International Symposium on Mathematical Foundations of Computer Science (MFCS'00)*, volume 1893 of *Lecture Notes in Computer Science*, pages 670–679. Springer-Verlag, 2000.
50. S. Yong. *A Framework for Binding Operators*. Ph.D. thesis (LFCS report ECS-LFCS-92-207), University of Edinburgh, 1992.

A Fundamental Adjunctions Between Categories of Variable Sets

As it is customary, I write $\widehat{\mathbb{V}}$ for the functor category $\mathcal{S}et^{\mathbb{V}^\circ}$ of so-called $\mathbb{V}^\circ$-variable sets (or presheaves). Recall that there is a universal Yoneda embedding $\mathsf{y} : \mathbb{V} \hookrightarrow \widehat{\mathbb{V}}$ given by $\mathsf{y}(v) = \mathbb{V}(\_\,, v)$.

For small categories $\mathbb{V}$ and $\mathbb{W}$, we have the following important adjoint situations (see, *e.g.*, [29, 12])

$$\begin{tikzcd}
\mathbb{V} \arrow[rr, hook, "\mathsf{y}"] \arrow[dr, "F"'] & \overset{\text{Lan}}{\cong} & \widehat{\mathbb{V}} \arrow[dl, shift right, "F^{\#}"'] \\
& \widehat{\mathbb{W}} \arrow[ur, shift right, "F*"'] \arrow[ur, phantom, "\dashv"] &
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbb{V} \arrow[r, hook, "\mathsf{y}"] \arrow[d, "f"'] \arrow[dr, phantom, "\overset{\text{Lan}}{\cong}"] & \widehat{\mathbb{V}} \arrow[d, shift right=4, "f_!"'] \arrow[d, shift left=4, "\dashv"'] \\
\mathbb{W} \arrow[r, hook, "\mathsf{y}"'] & \widehat{\mathbb{W}} \arrow[u, "\dashv f^* \dashv"']
\end{tikzcd}$$

obtained by left Kan extension, where

$$F^{\#}(P) = \int^{v \in \mathbb{V}} P(v) \times Fv(\_)\ ,\quad F*(Q) = \widehat{\mathbb{W}}(F\_\,, Q) \qquad (P \in \widehat{\mathbb{V}}, Q \in \widehat{\mathbb{W}})$$

and

$$f_!(P) = \int^{v \in \mathbb{V}} P(v) \times \mathbb{W}(\_\,, fv)\ ,\quad f^*(Q) = Q(f\_) \qquad (P \in \widehat{\mathbb{V}}, Q \in \widehat{\mathbb{W}})\ .$$

(See, *e.g.*, [31–Chap. IX] for the above notion $\int$ of coend.)

B Substitution Algebras and Algebraic Theories

A substitution algebra structure on $P = \{ P_\tau \}_{\tau \in T}$ in $\widehat{\mathsf{F}[T]}^{T}$ is given by operators

$$\eta_\tau : \longrightarrow (\tau)\tau\ ,\quad \sigma_{\tau,\tau'} : (\tau)\tau', \tau \longrightarrow \tau' \qquad (\tau, \tau' \in T)\ ,$$

giving rise to morphisms

$$\eta_\tau : 1 \longrightarrow {P_\tau}^{V_\tau} \ , \ \ \sigma_{\tau,\tau'} : {P_{\tau'}}^{V_\tau} \times P_\tau \longrightarrow P_{\tau'} \qquad\qquad (\tau,\tau' \in T)$$

in $\widehat{F[T]}$, subject to the following axioms, where we write $t[x^\tau \mapsto u]_{\tau'}$ as a shorthand for $\sigma_{\tau,\tau'}(\lambda x : V_\tau.t, u)$:

$$\eta_\tau(x)[x^\tau \mapsto u]_\tau = u \qquad\qquad (u : P_\tau)\ ,$$

$$t[x^\tau \mapsto u]_{\tau'} = t \qquad\qquad (t : P_{\tau'})\ ,$$

$$t(x,y)[y^\tau \mapsto \eta_\tau(x)]_{\tau'} = t(x,x) \qquad\qquad (t : {P_{\tau'}}^{V_\tau \times V_\tau}, x : V_\tau)\ ,$$

$$\begin{aligned}&\big(t(y,x)\big[y^{\tau'} \mapsto u(x)\big]_{\tau''}\big)[x^\tau \mapsto v]_{\tau''} && (t : {P_{\tau''}}^{V_{\tau'} \times V_\tau},\\ &= \big(t(y,x)\big[x^\tau \mapsto v\big]_{\tau''}\big)\big[y^{\tau'} \mapsto u(x)[x^\tau \mapsto v]_{\tau'}\big]_{\tau''} && u : {P_{\tau'}}^{V_\tau}, v : P_\tau)\ .\end{aligned}$$

These substitution structures can be incorporated to algebraic theories; see [18] for details. For instance, for the simply typed lambda calculus (see also [13]), where the set of types T is the closure under the arrow type constructor $\Rightarrow$ of a set of base types, this yields substitution algebras $(P, \mathsf{var}, \mathsf{sub})$ with binding operators

$$\begin{array}{lll} \text{(Application)} & \mathsf{app}_{\tau,\tau'} : & \tau{\Rightarrow}\tau', \tau \longrightarrow \tau' \\ \text{(Abstraction)} & \mathsf{abs}_{\tau,\tau'} : & (\tau)\tau' \longrightarrow \tau{\Rightarrow}\tau' \end{array} \qquad (\tau,\tau' \in T)$$

that are required to be compatible in the sense of satisfying the following axioms

$$\begin{aligned}&\big(\mathsf{app}_{\tau',\tau''}\big(t(x), u(x)\big)\big)[x^\tau \mapsto u]_{\tau'} \\ &\quad = \ \mathsf{app}_{\tau',\tau''}\big(t(x)[x^\tau \mapsto u]_{\tau'\Rightarrow\tau''}, u(x)[x^\tau \mapsto u]_{\tau'}\big)\end{aligned} \qquad (t : {P_{\tau'\Rightarrow\tau''}}^{V_\tau}, u : P_\tau)$$

$$\begin{aligned}&\big(\mathsf{abs}_{\tau',\tau''}\big(\lambda y : V_{\tau'}.t(y,x)\big)\big)[x^\tau \mapsto u]_{\tau'\Rightarrow\tau''} \\ &\quad = \ \mathsf{abs}_{\tau',\tau''}\big(\lambda y : V_{\tau'}.t(y,x)[x^\tau \mapsto u]_{\tau''}\big)\end{aligned} \qquad (t : {P_{\tau''}}^{V_{\tau'} \times V_\tau}, u : P_\tau)$$

where $t[x^\tau \mapsto u]_{\tau'}$ stands for $\mathsf{sub}_{\tau,\tau'}(\lambda x : V_\tau.t, u)$.

The initial algebra for this theory can be, of course, described as the simply typed lambda terms (modulo alpha conversion) with the usual capture-avoiding single-variable substitution operation (which in this setting can be shown to arise by structural recursion; again see [18] for details).

Further, beta and eta equality can be easily incorporated as the following axioms:

$$\text{(beta)} \quad \mathsf{app}_{\tau,\tau'}\big(\mathsf{abs}_{\tau,\tau'}(t), u\big) = \mathsf{sub}_{\tau,\tau'}(t,u) \qquad (t : {P_{\tau'}}^{V_\tau}, u : P_\tau)$$

$$\text{(eta)} \quad \mathsf{abs}_{\tau,\tau'}\big(\lambda x : V_\tau.\mathsf{app}_{\tau,\tau'}(t, \mathsf{var}_\tau(x))\big) = t \qquad (t : P_{\tau\Rightarrow\tau'})\ .$$

(Note that the metatheory accounts for the usual side condition required in the eta equality axiom, as in higher-order abstract syntax [43] (see also [22]).)