Constructing differential categories and deconstructing categories of games

Jim Laird[a,1], Giulio Manzonetto[b,2], Guy McCusker[a]

[a] *Department of Computer Science, University of Bath, Bath, BA2 7AY, UK*
[b] *Radboud University, Intelligent Systems, Nijmegen, The Netherlands*

Abstract

Differential categories were introduced by Blute, Cockett and Seely to axiomatize categorically Ehrhard and Regnier's syntactic differential operator. We present an abstract construction that takes a symmetric monoidal category and yields a differential category, and show how this construction may be applied to categories of games. In one instance, we recover the category previously used to give a fully abstract model of a nondeterministic imperative language. The construction exposes the differential structure already present in this model, and shows how the differential combinator may be encoded in the imperative language. A second instance corresponds to a new differential cartesian category of games. We give a model of a simply-typed resource calculus, Resource PCF, in this category and show that it possesses the finite definability property. Comparison with a semantics based on Bucciarelli, Ehrhard and Manzonetto's relational model reveals that the latter also possesses this property and is fully abstract.

Keywords: differential categories, game semantics, full abstraction

1. Introduction

An important aim in studying higher-order computation is to understand and control the way resources are used. One way to do this is by studying calculi designed to capture resource usage, and their denotational models.

Email addresses: `jiml@cs.bath.ac.uk` (Jim Laird), `g.manzonetto@cs.ru.nl` (Giulio Manzonetto), `g.a.mccusker@bath.ac.uk` (Guy McCusker)

[1]Research supported in part by UK EPSRC grant EP/HO23097.
[2]Research supported in part by NWO Project 612.000.936 CALMOC.

Two such calculi — the *differential λ-calculus* [1] of Ehrhard and Regnier and the *resource calculus* introduced by Tranquilli [2], inspired by the work of Boudol [3], are fundamentally related at the semantic level [4]: both may be interpreted using the notion of *differential category* introduced by Blute, Cockett and Seely [5]. In this paper, we study these concepts on both abstract and concrete levels. We give a construction of a differential category from any symmetric monoidal category, and use it to investigate the structure of newly discovered differential categories, relate them to existing examples, and to prove full abstraction results for *Resource PCF*, a typed programming language based on the resource calculus.

A potential source of differential categories, although not investigated hitherto, is *game semantics*: resource usage is represented rather explicitly in games and strategies. Indeed, we show that an existing games model of Idealized Algol with non-determinism, introduced by Harmer and McCusker [6] contains a differential Cartesian operator [7], and may therefore be used to interpret Resource PCF, although this interpretation contains non-definable ("junk") finitary elements.

We then present the construction which we shall use to analyze differential categories. Its key step takes a symmetric monoidal category with countable biproducts, embeds it in its *Karoubi envelope* (idempotent splitting) and then constructs the *cofree cocommutative comonoid* on this category (by taking a sum of symmetric tensor powers) and a differential operator on the Kleisli category of the corresponding comonad. Since biproducts may be added to any category by free constructions, we have a way of embedding any symmetric monoidal (closed) category in a Cartesian (closed) differential category.

Although this construction is somewhat elaborate, it provides a useful tool for analyzing and relating more directly presented models. For example, applying it to the terminal (one object, one morphism) SMCC yields the key example of a differential category (and model of resource calculus [4]) based on the finite-multiset comonad on the category of sets and relations. We also show that our differential category of games embeds in one constructed from a simple symmetric monoidal category of games. By refining the strategies in these games to eliminate *history sensitive* behaviour, we obtain a constraint on strategies ($\sim$-closure) in our directly presented model of Resource PCF which corresponds to finite definability. Another useful observation is that any functor of symmetric monoidal categories lifts to one between the differential categories constructed from them. In particular, from the terminal functor we derive a functor from our category of games and $\sim$-closed strategies into the relational model which is shown to be *full.* From this we may deduce that the relational model of Resource PCF is *fully abstract.*

Related Works. This article is an extended version of [8]. We build and analyze differential categories as a semantic framework for Resource PCF, a programming language based on the resource calculus. The resource calculus has been recently studied from a syntactic point of view by Pagani and Tranquilli [9] for confluence results, by Manzonetto and Pagani for separability results [10] and by Pagani and Ronchi della Rocca [11] for results about solvability. This calculus is also strongly linked with the differential λ-calculus [1]; in that context Ehrhard and Regnier studied the relationships between Böhm trees, Taylor expansion and linear head reduction [12, 13].

Monoidal and Cartesian differential categories have been introduced by Blute, Cockett and Seely in [5, 7]. Subsequently Bucciarelli, Ehrhard and Manzonetto proposed the notion of Cartesian closed differential category, proved that such categories are sound models of the simply typed resource calculus [4] and studied a concrete example. Other examples of such categories have been given by Blute, Ehrhard and Tasson in [14] and by Carraro, Ehrhard and Salibra in [15].

In the present paper we provide a general method for turning a symmetric monoidal category into a differential category. Our construction of an exponential modality is a special case of the one proposed by Melliès, Tabareau and Tasson in [16]. Our additional contributions are the observations that, under certain circumstances, the equalizers required to build symmetric tensor powers can be obtained by splitting idempotents, and that the resulting model possesses differential structure.

Outline. In Section 2 we fix some categorical notations and recall the basic definitions concerning monoidal and Cartesian differential categories. In Section 3 we introduce Resource PCF, together with its operational and denotational semantics. Section 4 presents a direct description of Harmer and McCusker's differential category of games $\mathbf{G}^{\otimes}$. Section 5 is devoted to provide the categorical construction for turning any symmetric monoidal category into a differential category. In Section 6 we apply the construction to recover and analyze the category $\mathbf{G}^{\otimes}$, that will be then refined in Section 7. Finally, in Section 8 we use our construction to compare the games model with the relational model, and prove that the relational model is fully abstract for Resource PCF.

2. Differential Categories

Differential categories were introduced by Blute, Cockett and Seely to formalize derivatives categorically. The authors started from monoidal categories [5], then extended the notion to Cartesian ones [7]; a further generalization to Cartesian closed categories has been made in [4] to model differential and resource λ-calculi.

Throughout this paper we will be working with categories whose hom-sets are endowed with the structure of a commutative monoid $(+, 0)$. Terminology for the various kinds of categorical structure we encounter varies widely, the adjective "additive" being particularly overloaded, so we will take care to define all our terminology as we go along.

Let us fix some notation. We write the identity map on an object A as id_A or simply A. Composition is written using infix ; in diagram order. We use $\langle f, g\rangle$ to denote the pairing of maps $f : A \to B$ and $g : A \to C$, and π_0, π_1 for the corresponding projections. In Cartesian closed category we denote the exponential object by $A \Rightarrow B$ and the curry of $f : A \times B \to C$ by $\Lambda(f) : A \to (B \Rightarrow C)$. We write $\Lambda^{-}(-)$ for the inverse of $\Lambda(-)$. We elide all associativity and unit isomorphisms associated with monoidal categories.

2.1. (Monoidal) Differential Categories

Let $\mathbf{C}$ be a commutative-monoid-enriched symmetric monoidal category: this means that it is a symmetric monoidal category, and that composition and tensor preserve the commutative monoid structure on hom-sets, so that

$$(f+g); h = f; h + g; h, \qquad k; (f+g) = k; f + k; g, \qquad f; 0 = 0 = 0; f,$$
$$(f+g) \otimes h = f \otimes h + g \otimes h, \qquad f \otimes 0 = 0.$$

A *coalgebra modality* on $\mathbf{C}$ is a comonad $(!, \delta, \epsilon)$ such that each object $!A$ is equipped with a comonoid structure

$$\Delta_A : !A \longrightarrow !A \otimes !A, \qquad e_A : !A \longrightarrow I$$

In addition to the associativity and unit equations for the comonoid, it should be the case that δ is a morphism of comonoids, that is,

$$\delta_A; e_{!A} = e_A \text{ and } \delta_A; \Delta_{!A} = \Delta_A; \delta_A \otimes \delta_A.$$

Given such a structure, a *differential combinator* is a family of maps $D_{A,B} : \mathbf{C}(!A, B) \to \mathbf{C}(A \otimes !A, B)$, natural in A and B and respecting the commutative monoid structure of the hom-sets, satisfying the following four axioms.

- $D(e_A) = 0$,
- $D(\Delta; f \otimes g) = (A \otimes \Delta); (D(f) \otimes g) + (A \otimes \Delta); \cong; (f \otimes D(g))$ where $f : !A \to B$, $g : !A \to C$ and $\cong$ is the appropriate symmetry map,
- $D(\epsilon_A; f) = (A \otimes e_A); f$, for $f : A \to B$,
- $D(\delta_A; !f; g) = (A \otimes \Delta_A); (D(f) \otimes (\delta_A; !f)); D(g)$ for $f : !A \to B$ and $g : !B \to C$.

Definition 2.1. *A* differential category *is a commutative-monoid-enriched symmetric monoidal category with a coalgebra modality and a differential combinator.*

When the coalgebra modality is a linear exponential comonad, its Kleisli category is a *Cartesian differential category*, whose definition we recall in the next subsection.

2.2. Cartesian (Closed) Differential Categories

A category is *left-additive* if precomposition by any map preserves the commutative monoid structure, that is,

$$f; 0 = 0, \qquad\qquad f; (g + h) = (f; g) + (f; h).$$

In such a category, a map f is called *additive* if postcomposition by f this map also preserves the commutative monoid structure. A *Cartesian left-additive* category is a left-additive category with finite products, such that all projections are additive, and if $f : A \to B$ and $g : A \to C$ are additive, so is $\langle f, g \rangle : A \to B \times C$.

Definition 2.2. *A* Cartesian differential category *is a Cartesian left-additive category equipped with a family of operators $D^\times_{A,B} : \mathbf{C}(A, B) \to \mathbf{C}(A \times A, B)$ which preserve the commutative monoid structure on hom-sets and satisfy:*

- $\langle h + k, v \rangle; D^\times(f) = \langle h, v \rangle; D^\times(f) + \langle k, v \rangle; D^\times(f)$,
- $D^\times(\mathrm{id}_A) = \pi_0$, $D^\times(\pi_0) = \pi_0; \pi_0$ *and* $D^\times(\pi_1) = \pi_0; \pi_1$,
- $D^\times(\langle f, g \rangle) = \langle D^\times(f), D^\times(g) \rangle$,
- $D^\times(f; g) = \langle D^\times(f), \pi_1; f \rangle; D^\times(g)$,

- $\langle g, 0, h, k\rangle; D^\times(D^\times(f)) = \langle g, k\rangle; D^\times(f)$,
- $\langle 0, h, g, k\rangle; D^\times(D^\times(f)) = \langle 0, g, h, k\rangle; D^\times(D^\times(f))$.

Definition 2.3. *A* Cartesian-closed differential category *is a Cartesian differential category with closed structure, such that the operation of currying preserves the commutative monoid structure on hom-sets and for all $f : C \times A \to B$, we have:*

$$D^\times(\Lambda(f)) = \Lambda(\langle \pi_0 \times 0_A, \pi_1 \times \mathrm{id}_A\rangle; D^\times(f)) : C \times C \to (A \Rightarrow B).$$

The leading examples of such categories, studied in [4], are Ehrhard's category of finiteness spaces, and the category **MRel** of "multiset relations", which is the Kleisli category for the finite-multiset comonad on the category **Rel** of sets and relations.

3. Resource PCF and its Models

We now describe *Resource PCF*, a simply typed resource calculus which incorporates the constants of PCF, making it a prototypical resource-sensitive programming language. As the name suggests, the fundamental ideas behind this calculus are those from the resource λ-calculus [2] and PCF [17]. The operational semantics of Resource PCF will be described through linear head reduction [18]; an equivalent presentation more in the style of [9] would also be possible.

3.1. Resource PCF: its Syntax and Operational Semantics

Resource PCF has two syntactic categories: terms, that appear in functional position, and bags, that appear in argument position and represent finite multisets of resources. A resource can be either *linear* (it must be used exactly once) or *reusable* (it can be used *ad libitum*) and in the latter case is decorated with a "!" superscript.

Definition 3.1. *Figure 1(a) gives the grammar generating the set Λ^r of* terms*, the set $\Lambda^{(!)}$ of* resources*, and the set Λ^b of* bags*, together with their typical meta-variables.*

We denote by [] the empty bag, and by $P_1 \uplus P_2$ the union of bags P_1, P_2. Terms of the shape $\mathrm{succ}^n(\mathrm{zero})$ are denoted by $\underline{n}$. The α-conversion and the

Syntax of Resource PCF

Λ^r:	M, N	$::= x \mid \lambda x.M \mid MP \mid \mathrm{ifz}(M, M, M) \mid$ $\mid\ \mathrm{Fix}(M) \mid \mathrm{succ}(M) \mid \mathrm{pred}(M) \mid \mathrm{zero}$	terms
$\Lambda^{(!)}$:	$M^{(!)}, N^{(!)}$	$::= M \mid M^!$	resources
Λ^b:	P	$::= [M_1^{(!)}, \dots, M_n^{(!)}]$	bags

(a) Grammar of terms, resources and bags.

Typing Rules

$$\frac{}{x_1 : A_1, \dots, x_n : A_n \vdash x_i : A_i}\ (\mathrm{var}) \qquad \frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash P : A}{\Gamma \vdash MP : B}\ (\mathrm{app})$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B}\ (\mathrm{lam}) \qquad \frac{}{\Gamma \vdash [] : A}\ ([]) \qquad \frac{\Gamma \vdash M : A \quad \Gamma \vdash P : A}{\Gamma \vdash [M^{(!)}] \uplus P : A}\ (\mathrm{bag})$$

$$\frac{\Gamma \vdash N : \mathtt{nat} \quad \Gamma \vdash M_i : A \quad (\text{for } i = 1, 2)}{\Gamma \vdash \mathrm{ifz}(N, M_1, M_2) : A}\ (\mathrm{ifz}) \qquad \frac{\Gamma \vdash M : A \to A}{\Gamma \vdash \mathrm{Fix}(M) : A}\ (\mathrm{fix})$$

$$\frac{}{\Gamma \vdash \mathrm{zero} : \mathtt{nat}}\ (\mathrm{zero}) \qquad \frac{\Gamma \vdash M : \mathtt{nat}}{\Gamma \vdash \mathrm{pred}(M) : \mathtt{nat}}\ (\mathrm{prec}) \qquad \frac{\Gamma \vdash M : \mathtt{nat}}{\Gamma \vdash \mathrm{succ}(M) : \mathtt{nat}}\ (\mathrm{succ})$$

(b) Typing rules for terms and bags.

Figure 1: Syntax and type system of Resource PCF.

set $\mathrm{fv}(M)$ of free variables of M are defined as usual in λ-calculus. A term M is *closed* if it does not contain any free variable, i.e. $\mathrm{fv}(M) = \emptyset$.

Hereafter we consider terms and bags up to α-equivalence.

Types are generated by the following grammar

$$A, B ::= \mathtt{nat} \mid A \to B.$$

Environments Γ are finite lists $x_1 : A_1; \cdots ; x_n : A_n$ assigning types to variables, so that $x : A \in \Gamma$ and $x : B \in \Gamma$ entails $A = B$. The *domain* of an environment Γ is defined by $\mathrm{dom}(\Gamma) = \{x \mid x : A \in \Gamma\}$.

Typing rules are defined in Figure 1(b), yielding a collection of typed terms-in-context $\Gamma \vdash M : A$. In the rule (lam) we suppose without loss of

Contexts:

$$\begin{array}{rl} C(\!|-|\!) ::= & (\!|-|\!) \mid \lambda x.C \mid CP \mid M([C] \uplus P) \mid M([C^!] \uplus P) \mid \mathrm{Fix}(C) \mid \\ \mid & \mathrm{pred}(C) \mid \mathrm{succ}(C) \mid \mathrm{ifz}(C, M, N) \mid \mathrm{ifz}(M, C, N) \mid \mathrm{ifz}(M, N, C) \end{array}$$

Evaluation contexts:

$$E(\!|-|\!) ::= (\!|-|\!) \mid EP \mid \lambda x.E \mid \mathrm{pred}(E) \mid \mathrm{succ}(E) \mid \mathrm{ifz}(E, M, N)$$

Let contexts:

$$F(\!|-|\!) ::= (\!|-|\!) \mid (\lambda x.F)P$$

(a) Arbitrary contexts, evaluation contexts, let contexts.

Linear head reduction:

$$\begin{array}{ll} \multicolumn{2}{l}{E(\!|F(\!|\lambda x.E'(\!|x|\!)|\!)(P \uplus [N])|\!) \to E(\!|F(\!|\lambda x.E'(\!|N|\!)|\!)P|\!)} \\ \multicolumn{2}{l}{E(\!|F(\!|\lambda x.E'(\!|x|\!)|\!)(P \uplus [N^!])|\!) \to E(\!|F(\!|\lambda x.E'(\!|N|\!)|\!)(P \uplus [N^!])|\!)} \\ \multicolumn{2}{l}{E(\!|F(\!|\lambda x.\underline{n}|\!)[N_1^!, \ldots, N_k^!]|\!) \to E(\!|F(\!|\underline{n}|\!)|\!) \text{ for some } k \geq 0,} \\ E(\!|\mathrm{ifz}(\mathrm{zero}, M, N)|\!) \to E(\!|M|\!) & E(\!|\mathrm{ifz}(\mathrm{succ}(\underline{n}), M, N)|\!) \to E(\!|N|\!) \\ E(\!|\mathrm{pred}(\mathrm{succ}(\underline{n}))|\!) \to E(\!|\underline{n}|\!) & E(\!|\mathrm{Fix}(M)|\!) \to E(\!|M[\mathrm{Fix}(M)^!]|\!) \end{array}$$

(b) Operational semantics via linear head reduction.

Figure 2: Operational semantics of Resource PCF.

generality that $x \notin \mathrm{dom}(\Gamma)$. Notice that the bags are homogeneously typed, that is, all resources in a bag must have the same type.

Operational semantics. The operational semantics is defined in Figure 2 via *linear head reduction*, whose definition requires the notions of let- and evaluation- contexts. The three kinds of contexts we consider are defined in Figure 2(a). In general a *context* is a term having a single occurrence of a "hole" inside (formally, an algebraic variable) denoted by $(\!|-|\!)$. Given a context $C(\!|-|\!)$ and a term M we write $C(\!|M|\!)$ for result of substituting M for the hole $(\!|-|\!)$ in C, possibly with captures of free variables. For example, if $C(\!|-|\!) = \lambda x.(\!|-|\!)$ and $M = \lambda y.x[y]$ then $C(\!|M|\!) = \lambda xy.x[y]$. Arbitrary contexts will be considered only in Theorem 8.4. An *evaluation context* is a context having the hole in head position, while a *let context* is a context having the shape of nested redexes.

Intuitively, linear head reduction exploits evaluation- and let- contexts to fetch the redex corresponding to the variable occurring in linear head position. Then this occurrence is replaced nondeterministically by some resource in the corresponding bag; if the resource is linear it is then removed from the bag, otherwise it is kept to be possibly used again later. When a constant $\underline{n}$ occupies the linear head position, we can get rid of the lambda abstraction and the bag, provided that it only contains reusable resources.

Notice that at every step only one of the rules in Figure 2(b) can be applied, therefore the choice of the redex to contract is deterministic. The reduction is however nondeterministic because of the presence of several resources in the bags.

Definition 3.2. *A closed term M of ground type* converges*, written $M\Downarrow$, if M reduces to $\underline{k}$ for some $k \in \omega$. The term M diverges, written $M\Uparrow$ otherwise.*

We now give some examples of converging and diverging reductions. Remark that in the approach of [2] the "mismatch" between the amount of resources expected by a program and the actual amount present in the bag is modelled by a reduction to 0 (the empty program), while in our system the corresponding reduction is blocked.

Example 3.3. *Let $\mathbf{I} = \lambda z.z$. It is easy to check that the following terms are well typed.*

(i) $(\lambda xy.x[y])[\mathbf{I}][\underline{3}]\Downarrow$. *By applying the first rule in Figure 2(b)*[3] *we get* $(\lambda xy.x[y])[\mathbf{I}][\underline{3}] \to (\lambda xy.\mathbf{I}[y])[][\underline{3}]$. *By applying the first rule again*[4] *this reduces to* $(\lambda xy.(\lambda z.y)[])[][\underline{3}]$ *and then to* $(\lambda xy.(\lambda z.\underline{3})[])[][]$. *We can then apply the third rule three times and get* $\underline{3}$ *as result.*

(ii) $(\lambda xy.y)[\mathbf{I}][\underline{3}^!]\Uparrow$. *Intuitively, this term diverges because the linear resource* $\mathbf{I}$ *cannot be erased. Formally, one can reduce* $(\lambda xy.y)[\mathbf{I}][\underline{3}^!] \to (\lambda xy.\underline{3})[\mathbf{I}][\underline{3}^!] \to (\lambda x.\underline{3})[\mathbf{I}]$ *which is blocked since no rule is applicable.*

(iii) $(\lambda x.x[\underline{3}])[]\Uparrow$. *Indeed the term in functional position is waiting for a resource, while it is applied to the empty bag.*

3.2. Denotational Semantics of Resource PCF

Resource PCF can be interpreted in any cpo-enriched Cartesian closed differential category having a weak natural number object. The interpreta-

[3] With $E(\!|-|\!) = (\!|-|\!)[\underline{3}], F(\!|-|\!) = (\!|-|\!)$ and $E'(\!|-|\!) = \lambda y.(\!|-|\!)[y]$.

[4] With $E(\!|-|\!) = (\lambda xy.(\!|-|\!))[][\underline{3}]$ and $F(\!|-|\!) = E'(\!|-|\!) = (\!|-|\!)$.

tions of the constants and constructors of PCF are standard, leaving only the application, which is treated as follows, as in [4].

In every Cartesian closed differential category it is possible to define an operator $\star$ on morphisms

$$\frac{f : C \times A \to B \quad g : C \to A}{f \star g : C \times A \to B}$$

by setting $f \star g := \langle\langle 0_C, g \circ \pi_0\rangle, \mathrm{id}\rangle; D^\times(f)$.

The operator $f \star g$ can be seen as the semantic counterpart of differential substitution; the type is not modified because $f \star g$ may still depend on A. Intuitively, such an operator is obtained by force-feeding the second argument A of f with *one copy* of the result of g. Indeed $f \star g$ can be decomposed as:

$$\begin{tikzcd}
C \times A \arrow[r, "\langle \pi_0; g, \mathrm{id}_{C\times A}\rangle"] \arrow[drr, "f \star g"'] & A \times (C \times A) \arrow[r, "\langle\langle 0_C, \mathrm{id}_A\rangle, \mathrm{id}_{C\times A}\rangle"] & (C \times A) \times (C \times A) \arrow[d, "D^\times(f)"] \\
& & B
\end{tikzcd}$$

where $\langle\langle 0_C, \mathrm{id}_A\rangle, \mathrm{id}_{C\times A}\rangle; D^\times(f)$ represents the partial derivative of f in its A component (cf. [7, p. 49]).

The interpretation $[\![M]\!]_\Gamma : [\![\Gamma]\!] \to [\![A]\!]$ of $\Gamma \vdash M : A$ is defined as usual, except for the case of application where we set:

$$[\![M[\vec{L}, \vec{N}^!]]\!]_\Gamma = \langle \mathrm{id}, \textstyle\sum_{i=1}^n [\![N_i]\!]_\Gamma\rangle; ((\cdots(\Lambda^-([\![M]\!]_\Gamma) \star [\![L_1]\!]_\Gamma)\cdots) \star [\![L_\ell]\!]_\Gamma).$$

It is easy to check that this definition is independent from the enumeration of the resources in the bag, as expected.

4. A Cartesian Closed Differential Category of Games

In this section we recall the definition of the category of games introduced in [6, 19], and show that it is a Cartesian closed differential category.

4.1. Arenas and Strategies

An *arena* A is a finite bipartite forest over two sets of moves, M_A^P and M_A^O, with edge relation $\vdash$. We say that a move is *enabled* by its parent in the forest, and that root moves are *initial.* A *QA-arena* is an arena equipped with a labelling function that labels each move as a question (Q) or answer (A), such that every answer is the child of a question.

A *justified sequence* is a sequence of moves s, alternately from M_A^O and M_A^P, together with a (partial) *pointer function* such that every occurrence m of a non-initial move in s points to an earlier occurrence n of its parent; in this case we say that n *justifies* m; the transitive closure of this relation is called *hereditary justification*.

Given a justified sequence s, the *player view* $\ulcorner s \urcorner$ is defined as follows.

$$\begin{array}{rcll} \ulcorner \varepsilon \urcorner & = & \varepsilon & \\ \ulcorner s \cdot n \urcorner & = & n & \text{if } n \text{ is an initial O-move} \\ \ulcorner s \cdot n \urcorner & = & \ulcorner s \urcorner \cdot n & \text{if } n \text{ is a P-move} \\ \ulcorner s \cdot m \cdot t \cdot n \urcorner & = & \ulcorner s \urcorner \cdot m \cdot n & \text{if } n \text{ is an O-move justified by } m \end{array}$$

A justified sequence satisfies *P-visibility* if, for every prefix sm with m a P-move, the justifier of m appears in $\ulcorner s \urcorner$. The player view of a sequence that satisfies P-visibility is always itself a justified sequence.

Given a justified sequence s, we say that an answer-move occurrence a *answers* the question occurrence q that justifies it. A justified sequence s satisfies *P-well-bracketing* if, for every prefix $s'a$ with a an answer move by P, the question that a answers is the rightmost O-question in the view $\ulcorner s' \urcorner$; call this the *pending question* at s'.

The notions of *opponent view*, *O-visibility* and *O-well-bracketing* are defined dually.

A justified sequence is *complete* if every question is answered exactly once; we write $\mathsf{comp}(A)$ for the set of complete justified sequences of A.

Lemma 4.1. *If s is a complete justified sequence that satisfies P-visibility (resp. O-visibility), then s satisfies P-well-bracketing (resp. O-well-bracketing).*

Proof. A simple analysis of views shows that, if q is an O-question that is answered when some later O-question q' is pending, then when q' is answered, again a later O-question is pending. There can therefore be no O-question that is answered when it is not pending, because s is finite. $\square$

A sequence is *well-opened* if it contains exactly one initial O-move. A *strategy* for an arena A is a set of complete sequences in which O plays first, satisfying P-visibility (and, by Lemma 4.1, P-bracketing). Given a strategy σ, $\mathsf{wv}(\sigma)$ is the set of sequences in σ that are well-opened and satisfy O-visibility.

Given arenas A and B, we write $A \uplus B$ for the arena arising as the disjoint union of A and B, and $A^\perp$ for the arena A with O and P-moves interchanged.

4.2. Categories of Games

We can define a category $\mathbf{G}$ in which objects are arenas whose roots are all O-moves, and morphisms $A \to B$ are strategies on the arena $A^\perp \uplus B$. Composition of strategies is the usual "parallel composition plus hiding" construction, and identities are copycat strategies. As proved in [6, 19], this category is monoidal closed: disjoint union of arenas gives a tensor product, with unit I the empty arena, and exponentials are given by the arena $A \multimap B$, which consists of the arena B with a copy of $A^\perp$ attached below each initial move; duplication of $A^\perp$ is required to maintain the forest structure.

Every object of $\mathbf{G}$ can be endowed with a comonoid structure as follows. The comultiplication $\Delta_A : A \to A \uplus A$ consists of the copycat strategy which interleaves play in the two copies of A on the right to produce the play on the left, and the counit is the strategy $\{\varepsilon\} : A \to I$. The subcategory of comonoid homomorphisms is a Cartesian closed category $\mathbf{G}^\otimes$. These maps are those whose choice of move at any stage depends only on the current thread, that is, the subsequence of moves hereditarily justified by the initial O-move currently in view; it follows that such strategies are completely determined by the well-opened plays they contain.

The programming language Erratic Idealized Algol (EIA) is an applied typed λ-calculus over base types `comm`, `nat` and `var`, with an appropriate stock of constants making it a higher-order imperative programming language with local state, consisting of variables (of type `var`) in which natural numbers can be stored. Typical constants are those for arithmetic and recursion as in PCF; imperative programming constants such as $\mathtt{assign} : \mathtt{var} \to \mathtt{nat} \to \mathtt{comm}$, which encodes the assignment operator $x := n$, $\mathtt{deref} : \mathtt{var} \to \mathtt{nat}$ which allows variable lookup, and $\mathtt{new} : (\mathtt{var} \to \mathtt{comm}) \to \mathtt{comm}$ which encodes local variable declaration. The constants include an erratic choice operator $\mathtt{or} : \mathtt{comm} \to \mathtt{comm} \to \mathtt{comm}$ which encodes nondeterministic choice. We equip the language with an operational semantics based on the may-converge predicate $\Downarrow$.

As shown in [19], this programming language can be given denotational semantics in the category $\mathbf{G}^\otimes$. The base types are interpreted as simple arenas, for instance $[\![\mathtt{nat}]\!]$ is the arena with one opponent question q enabling ω-many player-answers $0, 1, 2, \ldots$; $[\![\mathtt{comm}]\!]$ is similar, having a single initial opponent question `run` and a single player-answer `done`. The interpretation of the imperative programming constants is as in the standard games model of Idealized Algol from [20], and the erratic choice operator is interpreted by union of strategies.

Theorem 4.2 (Full abstraction [19])**.** *The model of EIA in $\mathbf{G}^{\otimes}$ is sound, and moreover, for any type A:*

- *if s is a complete well-opened play of $[\![A]\!]$ satisfying visibility, there is a closed term M of type A such that $\mathsf{wv}([\![M]\!]) = \{s\}$;*
- *terms $M, N : A$ are contextually equivalent if and only if $\mathsf{wv}([\![M]\!]) = \mathsf{wv}([\![N]\!])$.*

We now exhibit the differential structure that $\mathbf{G}^{\otimes}$ possesses.

Definition 4.3. *Let s be a complete, well-opened play in $A^{\perp} \uplus B$ which contains at least one initial A-move. Say that a complete play s' in $A^{\perp} \uplus A^{\perp} \uplus B$ is a* derivative *of s if*

$$\Delta; \{s'\} = \{s\}$$

(where Δ is the suitable diagonal) and s' contains one initial move in the left occurrence of $A^{\perp}$.

The definition says that s' is obtained from s by labelling one initial $A^{\perp}$-move occurring in s, along with all moves hereditarily justified by it, as appearing in the left-hand occurrence of $A^{\perp}$; the remaining $A^{\perp}$-moves are relabelled to appear in the right-hand occurrence.

Example 4.4. *For instance the following play*

$$\begin{array}{rccc}
s: & \mathbf{N}^{\perp} & \uplus & \mathbf{N} \\
 & & & q \\
 & q & & \\
 & 0 & & \\
 & q & & \\
 & 1 & & \\
 & & & 1
\end{array}$$

has two derivatives:

$$
\begin{array}{lccccc}
s_1' : & \mathbf{N}^\perp & \uplus & \mathbf{N}^\perp & \uplus & \mathbf{N} \\
 & & & & & q \\
 & q & & & & \\
 & 1 & & & & \\
 & & & q & & \\
 & & & 0 & & \\
 & & & & & 1
\end{array}
\qquad\qquad
\begin{array}{lccccc}
s_2' : & \mathbf{N}^\perp & \uplus & \mathbf{N}^\perp & \uplus & \mathbf{N} \\
 & & & & & q \\
 & & & q & & \\
 & & & 1 & & \\
 & q & & & & \\
 & 0 & & & & \\
 & & & & & 1
\end{array}
$$

We then define $D^\times(\sigma)$ as the strategy whose well-opened plays are

$$\{s' \in \mathsf{comp}(A^\perp \uplus A^\perp \uplus B) \mid s' \text{ is a derivative of some well-opened } s \in \sigma\}.$$

We can verify directly that this makes $\mathbf{G}^\otimes$ a Cartesian closed differential category; later we will see that this follows from a general construction. Because of the definability property of the model of EIA, it is reasonable to expect that the differential operator is programmable in EIA, and indeed it is. For terms of type $A \to \mathtt{nat}$ we can define the differential operator as follows (using appropriate syntactic sugar).

$$
\begin{array}{l}
\lambda f : A \to \mathtt{nat}.\lambda a : A.\lambda a' : A. \\
\qquad\qquad\qquad \mathtt{new}\ b := \mathtt{true} \\
\qquad\qquad\qquad \mathtt{new}\ y := f((\mathtt{if}\ b\ \mathtt{then}\ (b := \mathtt{false}; a)\ \mathtt{else}\ a')\ \mathtt{or}\ a')\ \mathtt{in} \\
\qquad\qquad\qquad\qquad \mathtt{if}\ \neg b\ \mathtt{then}\ y\ \mathtt{else}\ \mathtt{diverge}
\end{array}
$$

In any converging execution of this code, the argument a is supplied to f exactly once, though the choice of *which* call to f's argument receives a is made nondeterministically; all other calls to f's argument receive a'.

5. Constructing Differential Categories

We now describe a construction of models of intuitionistic linear logic that are also differential categories (Figure 3). The main ingredient is the construction of a category which possesses a comonad delivering cofree cocommutative comonoids; we begin with this step, before describing some preliminary steps that can readily construct appropriate categories.

5.1. Building differential categories through Karoubi envelope

Let $\mathbf{C}$ be a symmetric monoidal category. For any object A of $\mathbf{C}$, write $A^{\otimes n}$ to denote the n-fold tensor power of A. The *symmetric* tensor power A^n, if it exists, is the equaliser of the diagram

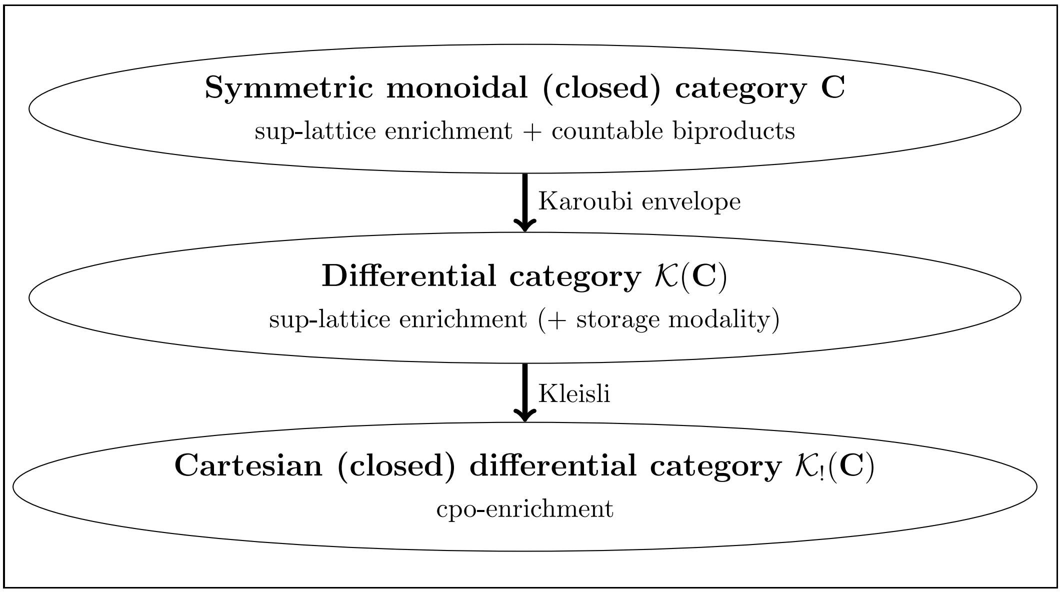

Figure 3: The main construction.

$$A^{\otimes n} \quad \begin{array}{c} \xrightarrow{\hspace{8em}} \\ \vdots \; n! \text{ permutations} \\ \xrightarrow{\hspace{8em}} \end{array} \quad A^{\otimes n}$$

consisting of all $n!$ permutations from $A^{\otimes n}$ to itself.

We begin by establishing circumstances under which the Karoubi envelope (idempotent splitting) $\mathcal{K}(\mathbf{C})$ possesses such equalizers. Recall that $\mathcal{K}(\mathbf{C})$ has as its objects pairs (A, f) where A is an object of $\mathbf{C}$ and $f : A \to A$ is an idempotent, and as its maps $(A, f) \to (B, g)$, those maps $h : A \to B$ from $\mathbf{C}$ such that

$$h = f; h; g$$

equivalently $f; h = h = h; g$. This category inherits the monoidal structure from $\mathbf{C}$.

Suppose that each hom-set in $\mathbf{C}$ is equipped with an n-ary sum operator Σ_n for each natural number n; that is, for any multiset S containing n maps $f_1, \dots, f_n : A \to B$ we can form the sum $\sum_n S$; and that each n-ary operator is commutative and idempotent, and composition preserves these sums.

Lemma 5.1. *Let (A, f) be any object of $\mathcal{K}(\mathbf{C})$, and let G be any subgroup of the group of automorphisms of (A, f) of finite cardinality n. Then the following diagram is an equalizer.*

$$(A, f; \textstyle\sum_n G) \xrightarrow{f; \sum_n G} (A, f) \ \underset{\longrightarrow}{\overset{\longrightarrow}{\vdots \ \textit{every } g \in G}} \ (A, f)$$

Proof. Routine calculation verifies that $f; \sum_n G$ is indeed an idempotent so that $(A, f; \sum_n G)$ is a well-defined object of $\mathcal{K}(\mathbf{C})$, and that $f; \sum_n G$ is a map of the appropriate type. That $f; \sum_n G; g = f; \sum_n G; g'$ for any $g, g' \in G$ follows from the fact that composition preserves sums and that postcomposition with any $g \in G$ maps the set G onto itself, because G is a group. Let $h : (A', f') \to (A, f)$ be any map such that $h; g = h; g'$ for all $g, g' \in G$. Then each $h; g = h$, so $h; f; \sum_n G = h; \sum_n G = \sum_n [h; g \mid g \in G] = \sum_n [h, \ldots, h] = h$. So h factors through $f; \sum_n G$ via h itself, and this factorization is necessarily unique. □

We can now readily construct symmetric tensor powers in $\mathcal{K}(\mathbf{C})$ as follows. Given an object A of $\mathbf{C}$, define $\Theta_{A,n} : A^{\otimes n} \to A^{\otimes n}$ to be the sum of the $n!$ permutation maps, which of course form a group. The above lemma specializes to the following.

Corollary 5.2. *For any object (A, f) of $\mathcal{K}(\mathbf{C})$, the following diagram is an equalizer.*

$$(A^{\otimes n}, f^{\otimes n}; \Theta_{A,n}) \xrightarrow{f^{\otimes n}; \Theta_{A,n}} (A^{\otimes n}, f^{\otimes n}) \ \underset{\longrightarrow}{\overset{\longrightarrow}{\vdots \ n! \textit{ permutations}}} \ (A^{\otimes n}, f^{\otimes n})$$

Moreover, these equalizer diagrams are preserved by tensor products.

One consequence of this is that there are maps $(A, f)^{m+n} \to (A, f)^m \otimes (A, f)^n$ whose underlying maps are given by $f^{\otimes m+n}; \Theta_{A,m+n}$, which are equalizers of the evident subgroup of $m! \times n!$ permutations on $A^{\otimes m+n}$ that map the first m components onto themselves.

These symmetric tensor powers will allow us to construct a coalgebra modality on $\mathcal{K}(\mathbf{C})$ as the free commutative comonoid, provided we have *countable products* and that the monoidal structure distributes over them.

Recall that a *commutative comonoid* in a symmetric monoidal category is an object A together with maps $\Delta : A \to A \otimes A$ and $e : A \to I$ satisfying the obvious commutativity, associativity and unit diagrams; morphisms of comonoids are morphisms between the underlying objects that preserve the comonoid structure. Let $\mathcal{K}^{\otimes}(\mathbf{C})$ be the category of commutative comonoids and comonoid morphisms in $\mathcal{K}(\mathbf{C})$.

Lemma 5.3. *The forgetful functor $U : \mathcal{K}^{\otimes}(\mathbf{C}) \to \mathcal{K}(\mathbf{C})$ has a right adjoint, whose action on objects takes (A, f) to the product $\prod_{n\in\omega}(A, f)^n$, which we call $!(A, f)$.*

Proof. For any m and n, we have the map

$$\pi_{m+n}; f^{\otimes m+n}; \Theta_{A,m+n} : !(A, f) \longrightarrow (A, f)^m \otimes (A, f)^n.$$

Tupling all these gives us a map $!(A, f) \to \prod_{m,n}(A, f)^m \otimes (A, f)^n$, and by distributivity of tensor over product, this gives a map $\Delta : !(A, f) \to !(A, f) \otimes !(A, f)$. We also have the map $\pi_0 : !(A, f) \to I$. It can readily be verified that these maps give $!(A, f)$ the structure of a comonoid. Moreover, it is the free comonoid on (A, f): if (B, g) is any commutative comonoid and $\alpha : (B, g) \to (A, f)$ any morphism, we construct a comonoid morphism $\alpha^\dagger : (B, g) \to !(A, f)$ as follows. The comultiplication $\Delta^n : (B, g) \to (B, g)^{\otimes n}$ equalizes all permutations, so the composition $\Delta^n; \alpha^{\otimes n}$ does too, yielding a map $(B, g) \to (A, f)^n$. Tupling all these maps gives us the required comonoid map $\alpha^\dagger$, and it is easily checked that this is the unique map such that $\alpha^\dagger; \pi_1 = \alpha$. □

Composing these two adjoint functors yields a comonad $(!, \delta, \epsilon)$ on $\mathcal{K}(\mathbf{C})$.

Lemma 5.4. *The comonad $(!, \delta, \epsilon)$ is a coalgebra modality. In fact, it is a linear exponential comonad (also known as a storage modality).*

Proof. Showing that we have a coalgebra modality amounts to demonstrating that δ is a morphism of comonoids. In fact much more is true: it is routine to check that the forgetful functor $U : \mathcal{K}^{\otimes}(\mathbf{C}) \to \mathcal{K}(\mathbf{C})$ satisfies the conditions of Beck's monadicity theorem, which implies that the category of coalgebras is isomorphic to $\mathcal{K}^{\otimes}(\mathbf{C})$. Therefore $(!, \delta, \epsilon)$ is a linear exponential comonad. □

The fact that the cofree commutative comonoid provides a linear exponential comonad is attributed to Lafont. The construction of this comonad along the lines given above follows the recipe in [16]; our main contribution at this stage is the use of the Karoubi envelope and sum structure on hom-sets to generate a category possessing the required equalizers.

Now suppose that, in addition to the n-ary sum operators in each hom-set, $\mathbf{C}$ is enriched over commutative monoids $(+, 0)$. In this situation we can construct a differential operator on $\mathcal{K}(\mathbf{C})$, making it into a differential category.

The differential operator is given by precomposition with the *deriving transformation* $d : (A, f) \otimes !(A, f) \to !(A, f)$ defined as follows. For each n, the map $f^{\otimes n+1}; \Theta_{A,n+1}$ in $\mathbf{C}$ gives us a morphism

$$f^{\otimes n+1}; \Theta_{A,n+1} : (A, f) \otimes (A, f)^n \to (A, f)^{n+1}$$

and hence we obtain maps $\cong; \pi_n; f^{\otimes n+1}; \Theta_{A,n+1} : (A, f) \otimes !(A, f) \to (A, f)^{n+1}$ where $\cong$ is the distributivity map. Tupling all these gives us a morphism $(A, f) \otimes !(A, f) \to \prod_n (A, f)^{n+1}$, and finally pairing this with $0 : (A, f) \otimes !(A, f) \to I$ gives the required map.

We have taken care in the development above to separate the n-ary sums needed to produce the symmetric tensors from the sums provided by commutative monoid enrichment. This is because we have in mind certain models, not considered in the present paper, in which these sums are indeed different (but related); in particular, the commutative monoid structure need not be idempotent while the n-ary sums must be. For the present paper, we focus on a special case of the development above, in which $\mathbf{C}$ is enriched over sup-lattices, that is, *idempotent* commutative monoids with *all* sums. The consequences of our constructions in such cases can be summarized as follows.

Theorem 5.5. *Let* $\mathbf{C}$ *be a sup-lattice-enriched symmetric monoidal category with countable distributive products. With the structure described above,* $\mathcal{K}(\mathbf{C})$ *is a sup-lattice-enriched differential category, and the Kleisli category* $\mathcal{K}_!(\mathbf{C})$ *a cpo-enriched Cartesian differential category. If* $\mathbf{C}$ *is monoidal closed (in the sup-lattice-enriched sense) then* $\mathcal{K}_!(\mathbf{C})$ *is a cpo-enriched Cartesian-closed differential category.*

Proof. That the differential operator satisfies the required equations to make $\mathcal{K}(\mathbf{C})$ a differential category is lengthy but straightforward to check. Sup-lattice enrichment follows directly from that of $\mathbf{C}$. The fact that $\mathcal{K}_!(\mathbf{C})$ is Cartesian differential follows from Proposition 3.2.1 of [7]. The Cartesian closure of $\mathcal{K}_!(\mathbf{C})$ is a well-known fact about linear exponential comonads. For the cpo-enrichment, it is enough to observe that the passage from $\alpha : !(A, f) \to (B, g)$ to $\alpha^\dagger : !(A, f) \to !(B, g)$ preserves directed suprema. $\square$

Even when $\mathbf{C}$ is not monoidal closed, it is still possible to arrive at a Cartesian closed differential category when there are enough exponentials: if $\mathbf{C}$ has all exponentials $A \multimap R$ for some fixed object R, then the full subcategory of $\mathcal{K}_!(\mathbf{C})$ consisting of such R-exponentials is Cartesian closed, and also possesses a weak distributive coproduct structure, the "lifted sum" $\Sigma_{i\in I}A_i = (\prod_{i\in I}(!A_i \multimap R)) \multimap R$. In particular, $\prod_{n\in\omega} R \multimap R$ is a weak natural numbers object.

Remark. As is well known, in a commutative-monoid-enriched category, every product $A \times B$ is in fact a *biproduct*, that is, it is also a coproduct and the canonical map

$$[\langle \mathrm{id}_A, 0\rangle, \langle 0, \mathrm{id}_B\rangle] : A \oplus B \to A \times B$$

is an isomorphism. Similarly, every coproduct is a biproduct; and terminal and initial objects coincide. In the presence of sup-lattice enrichment, this extends to infinite products, so our countable products are in fact countable biproducts.

5.2. Sup-lattice enrichment and biproducts via free constructions

To apply the construction above, we need a sup-lattice enriched symmetric monoidal category with countable distributive biproducts. Such categories can readily be manufactured via a series of free constructions.

Beginning with a symmetric monoidal category, one can construct its *sup-lattice-completion* as the category with the same objects, but whose maps $A \to B$ are sets of maps in the original category (cf. [21] VIII.2 exercise 5). This is a sup-lattice enriched category, with joins of maps given by unions, and monoidal structure inherited from the original category; closed structure is also inherited, if it exists. We denote the sup-lattice completion of a category $\mathbf{C}$ by $\mathbf{C}^+$.

Given a sup-lattice-enriched symmetric monoidal category, its *biproduct completion* (cf. [21] VIII.2 exercise 6) has as objects indexed sets $\{A_i \mid i \in I\}$ of objects in the original category, and as morphisms $\{A_i \mid i \in I\} \to \{B_j \mid j \in J\}$ *matrices* of morphisms, that is, for each i, j, a morphism $A_i \to B_j$. Composition is (potentially infinite) matrix multiplication; the infinite sums required for composition are the reason we require sup-lattice enrichment. The biproduct of a set of objects is given by the disjoint union of families. We write $\mathbf{BP}(\mathbf{C})$ for the biproduct completion of a category $\mathbf{C}$.

We will be interested in some categories which arise by performing these two constructions in sequence. Given a category $\mathbf{C}$, we denote by $\mathbf{FamRel}(\mathbf{C})$ the category whose objects are families $\{A_i \mid i \in I\}$ of objects of $\mathbf{C}$, and whose morphisms $\{A_i \mid i \in I\} \to \{B_j \mid j \in J\}$ are given by sets of triples

$$\begin{tikzcd}
 & \mathbf{BP}(\mathbf{1}^+) \cong \mathbf{FamRel}(\mathbf{1}) \cong \mathbf{Rel} \arrow[dr, "\text{Karoubi envelope}"] & \\
\mathbf{1}^+ \arrow[ur, "\text{Biproduct comp.}"] & & \mathcal{K}(\mathbf{Rel}) \arrow[d, "\text{Kleisli}"] \\
\mathbf{1} \arrow[u, "\text{sup-lattice comp.}"] \arrow[uur, bend right, "\mathbf{FamRel}(-)"'] & & \mathbf{MRel} \hookrightarrow \mathcal{K}_!(\mathbf{Rel})
\end{tikzcd}$$

Figure 4: Reconstructing **MRel** from the terminal SMCC **1**.

(i, j, f) where $i \in I$, $j \in J$ and $f : A_i \to B_j$ in $\mathbf{C}$. Note that for a given i and j there may be no such triples in a morphism, or one, or many. It is easy to check that $\mathbf{FamRel}(\mathbf{C})$ is isomorphic to the category $\mathbf{BP}(\mathbf{C}^+)$.

Example 5.6. *A simple but central example begins with the terminal category* $\mathbf{1}$ *(Figure 4). Indeed* $\mathbf{FamRel}(\mathbf{1})$ *is the category* $\mathbf{Rel}$ *of sets and relations. On the image of* $\mathbf{Rel}$ *in* $\mathcal{K}(\mathbf{Rel})$*,* $!$ *is the finite-multiset comonad, and we therefore find* $\mathbf{MRel}$ *embedded in* $\mathcal{K}_!(\mathbf{Rel})$ *as a sub-Cartesian-differential-category.*

6. Analysis of $\mathbf{G}^\otimes$

In this section we apply some of the constructions developed above to reconstruct $\mathbf{G}^\otimes$ and discover its differential structure as an instance of our construction. We begin by defining a new category $\mathbf{EG}$ of *exhausting games*, to which we will apply our constructions, eventually arriving at a Cartesian closed differential category containing $\mathbf{G}^\otimes$ as a subcategory. We then introduce a refined category of exhausting games, $\mathbf{EG}_\sim$, and again apply our constructions to obtain a model of Resource PCF with the finite definability property.

Given a finite arena A, a *path* is a non-repeating enumeration of all moves, respecting the order given by the edge relation in the arena — that is, a traversal of the forest — such that the first move is by O and moves alternate polarity thereafter. Note that every move in a path has a unique justifier earlier in the path. An *exhausting strategy* on A is a set of even-length paths that satisfy P-visibility; if A has an odd number of moves, the only strategy is the empty set.

Definition 6.1. *The category* **EG** *has finite O-rooted arenas as objects and exhausting strategies on $A^\perp \uplus B$ as maps from A to B, with composition and identities as usual.*

Again, disjoint union of arenas gives a monoidal structure; and if B has a single root, then the arena $A \multimap B$ is an exponential, so **EG** has all R-exponentials, where R is the arena with a single move belonging to O.

It is clear that **EG** is sup-lattice enriched: unions of strategies are strategies, and composition preserves unions. We may therefore form its biproduct completion to obtain the structure we require to construct a differential category as in Subsection 5.2. We write $\mathcal{K}(\mathbf{BP}(\mathbf{EG}))$ for the differential category so constructed, and $\mathcal{K}_!(\mathbf{BP}(\mathbf{EG}))$ for its Kleisli category, which is a Cartesian differential category. The full subcategory of R-exponentials is Cartesian closed and has a weak natural numbers object. We shall now show how to recover $\mathbf{G}^\otimes$ as a subcategory of this.

Let A be any QA-arena, and consider a non-repeating (justified) sequence s of pairs (a, n) where a is a move of A and n is a natural number. Taking left projection on such a sequence gives a justified sequence in A, which we call $\hat{s}$; we say that s is a tagging of $\hat{s}$, and write $\mathsf{tcomp}(A)$ for the set of all taggings of complete well-opened plays in A.

Let $s \in \mathsf{tcomp}(A)$ be a tagging of the complete play $\hat{s}$. We can define an arena $\|s\|$ whose moves are the elements of s and whose edge relation is precisely the justification structure of s. Thus s becomes a path of $\|s\|$. If t is a tagged sequence such that $\hat{t} = \hat{s}$, we can define a morphism $\phi_{s,t} : \|s\| \to \|t\|$ by

$$\phi_{s,t} = \{u \in \|s\|^\perp \uplus \|t\| \mid \hat{u} \in \mathrm{id}_A\}.$$

Note that it is possible for distinct taggings s, t of $\hat{s}$ to give rise to the same arena $\|s\| = \|t\|$. In this case, $\phi_{s,t}$ is an idempotent on this arena. Similarly, given taggings s_1, s_2, s_3 of the same complete play $\hat{s}$, we have

$$\phi_{s_1,s_2}; \phi_{s_2,s_3} = \phi_{s_1,s_3}.$$

Let A^* be the set of arenas $\{\|s\| \mid s \in \mathsf{tcomp}(A)\}$. Considering this set as a (self-indexed) family, it is an object on the biproduct completion of **EG**. We define an endomorphism $\phi_A : A^* \to A^*$ in $\mathbf{BP}(\mathbf{EG})$ as the "matrix" with entries given by

$$(\phi_A)_{\|s\|,\|t\|} = \begin{cases} \phi_{s',t'}, & \text{if there exist } s', t' \text{ such that } \|s\| = \|s'\|, \|t\| = \|t'\| \text{ and } \hat{s'} = \hat{t'} \\ \emptyset, & \text{otherwise.} \end{cases}$$

By the above considerations, this is well-defined and idempotent, so (A^*, ϕ_A) is an object of $\mathcal{K}(\mathbf{BP}(\mathbf{EG}))$.

We now define a functor from $\mathbf{G}$ to $\mathcal{K}(\mathbf{BP}(\mathbf{EG}))$ whose action on objects takes an arena A to $!(A^*, \phi_A)$. In order to describe the action on morphisms, we first analyse what a $f : !(A^*, \phi_A) \to !(B^*, \phi_B)$ looks like.

Since $!(A^*, \phi_A) = \bigoplus_{n\in\omega}(A^*, \phi_A)^n$, such a map is given by a collection of maps

$$f_{m,n} : (A^*, \phi_A)^m \to (B^*, \phi_B)^n$$

for $m, n \in \omega$. Each $f_{m,n}$ consists of a matrix of morphisms of type

$$\|s_1\| \otimes \cdots \otimes \|s_m\| \to \|t_1\| \otimes \cdots \otimes \|t_n\|$$

in $\mathbf{G}$, where each $s_i \in \mathsf{tcomp}(A)$ and $t_j \in \mathsf{tcomp}(B)$; and this matrix must be invariant under composition with ϕ_A, ϕ_B, and permutations of the tensors.

We can therefore regard a map $f : !(A^*, \phi_A) \to !(B^*, \phi_B)$ as a set of complete plays in arenas as above, closed under permutation and retagging. That is, suppose f contains some play

$$u \in \|s_1\| \otimes \cdots \otimes \|s_m\| \to \|t_1\| \otimes \cdots \otimes \|t_n\|.$$

Let s'_i and t'_j be retaggings of the s_i and t_j, and α and β permutations of $\{1, \ldots, m\}$ and $\{1, \ldots, n\}$ respectively. Permuting and retagging moves of u yields a play

$$u' \in \|s'_{\alpha(1)}\| \otimes \cdots \otimes \|s'_{\alpha(m)}\| \to \|t'_{\beta(1)}\| \otimes \cdots \otimes \|t'_{\beta(n)}\|$$

and the invariance of f under composition with idempotents says exactly that this u' is also in f.

Given a play u in such an f, untagging each component yields a play $\hat{u}$ of $(A^{\otimes m})^\perp \uplus B^{\otimes n}$ which contains exactly one initial move in each A- and each B-component. Identifying these components (essentially a further untagging) yields a play in $A^\perp \uplus B$ with m initial A-moves and n initial B-moves. By abuse of notation we will also write $\hat{u}$ for this further untagging. We are now able to define our functor from $\mathbf{G}$ to $\mathcal{K}(\mathbf{BP}(\mathbf{EG}))$:

Definition 6.2. *The functor $\Phi : \mathbf{G} \to \mathcal{K}(\mathbf{BP}(\mathbf{EG}))$ maps an arena A to $!(A^*, \phi_A)$ and a morphism $\sigma : A \to B$ to the map $\Phi(\sigma)$ whose plays in the arena $\|s_1\| \otimes \cdots \otimes \|s_m\| \to \|t_1\| \otimes \cdots \otimes \|t_n\|$ are those u such that $\hat{u} \in \sigma$.*

Lemma 6.3. *Φ is a well-defined, full and faithful functor $\mathbf{G} \to \mathcal{K}(\mathbf{BP}(\mathbf{EG}))$.*

Proof. For well-definedness, we must show that $\Phi(\sigma)$ is indeed closed under permutation and retagging; but by definition $\Phi(\sigma)$ includes all possible taggings and permutations of the plays of σ. Likewise, the identity on $!(A^*, \phi_A)$ consists of the appropriate idempotent, which contains all plays that untag to plays of the identity, which is by definition $\Phi(\mathrm{id}_A)$. It is straightforward to verify that Φ preserves composition.

To see that Φ is faithful, note that if $\sigma, \tau : A \to B$ with $\sigma \neq \tau$, then (wlog) σ contains a complete play $u \notin \tau$. But then any tagging of u lies in $\Phi(\sigma)$ and not in $\Phi(\tau)$, so $\Phi(\sigma) \neq \Phi(\tau)$.

Finally, to see that Φ is full, let $f : !(A^*, \phi_A) \to !(B^*, \phi_B)$ be any map. Untagging all of the plays in f gives a set of complete plays in $A^\perp \uplus B$ which is therefore a morphism $\sigma : A \to B$ in $\mathbf{G}$. We claim that $f = \Phi(\sigma)$. By definition $\Phi(\sigma)$ contains all plays that untag to plays in σ, so $f \subseteq \Phi(\sigma)$. Now consider any $u \in \Phi(\sigma)$. By definition of σ, the untagging $\hat{u} \in \sigma$ is equal to $\hat{v}$ for some $v \in f$. There must be a retagging of v that yields u. But f is closed under retaggings, so $u \in f$ and $\Phi(\sigma) \subseteq f$ as required. □

Given arenas A and B, the object $((A \uplus B)^*, \phi_{A \uplus B})$ is the product of (A^*, ϕ_A) and (B^*, ϕ_B) in $\mathcal{K}(\mathbf{BP}(\mathbf{EG}))$. Since ! is a linear exponential comonad, we have the "Girard isomorphism" between $\Phi(A \uplus B) = !((A^*, \phi_A) \times (B^*, \phi_B))$ and $!(A^*, \phi_A) \otimes !(B^*, \phi_B) = \Phi(A) \otimes \Phi(B)$. It is routine to verify the following:

Lemma 6.4. *The above isomorphism makes Φ into a strong monoidal functor.*

We now show that Φ restricts to a functor $\mathbf{G}^\otimes \to \mathcal{K}_!(\mathbf{BP}(\mathbf{EG}))$. Recall that $\mathbf{G}^\otimes$ is defined as the subcategory of $\mathbf{G}$ consisting of the comonoid homomorphisms, and that $\mathcal{K}_!(\mathbf{BP}(\mathbf{EG}))$ is the category of comonoid homomorphisms on free !-coalgebras in $\mathcal{K}(\mathbf{BP}(\mathbf{EG}))$.

Proposition 6.5. *The functor Φ preserves and reflects comonoid homomorphisms. That is, a map σ is a comonoid homomorphism in $\mathbf{EG}$ if and only if $\Phi(\sigma)$ is a comonoid homomorphism in $\mathcal{K}(\mathbf{BP}(\mathbf{EG}))$. Φ therefore restricts to a full and faithful product-preserving functor from $\mathbf{G}^\otimes$ to $\mathcal{K}_!(\mathbf{BP}(\mathbf{EG}))$.*

Proof. We write $\Delta_A : A \to A \uplus A$ for the comonoid multiplication in $\mathbf{EG}$. Straightforward calculation establishes that the composite

$$!(A^*, \phi_A) = \Phi(A) \xrightarrow{\Phi(\Delta_A)} \Phi(A \uplus A) \xrightarrow{\cong} \Phi(A) \otimes \Phi(A) = !(A^*, \phi_A) \otimes !(A^*, \phi_A)$$

is the comultiplication map of $!(A^*, \phi_A)$.

Let $\sigma : A \to B$ be a map in **EG**. Observe that the outer rectangle of the diagram

$$\begin{tikzcd}
\Phi(A) \arrow[rr, "\Phi(\sigma)"] \arrow[d, "\Phi(\Delta_A)"'] & & \Phi(B) \arrow[d, "\Phi(\Delta_B)"] \\
\Phi(A \uplus A) \arrow[rr, "\Phi(\sigma \uplus \sigma)"] \arrow[d, "\cong"'] & & \Phi(B \uplus B) \arrow[d, "\cong"] \\
\Phi(A) \otimes \Phi(A) \arrow[rr, "\Phi(\sigma) \otimes \Phi(\sigma)"'] & & \Phi(B) \otimes \Phi(B)
\end{tikzcd}$$

commutes if and only if the upper square commutes (the lower square commutes by naturality of the monoidal isomorphism). Therefore σ preserves comultiplication if and only if $\Phi(\sigma)$ does. A similar argument shows that σ preserves the counit of the comonoid structure if and only if $\Phi(\sigma)$ does. $\square$

7. Some Refined Categories of Games

The functor Φ exhibits $\mathbf{G}^{\otimes}$ as a full subcategory of the Cartesian differential category $\mathcal{K}(\mathbf{BP}(\mathbf{EG}))$. Thus $\mathbf{G}^{\otimes}$ is rich enough to interpret Resource PCF. Indeed, it is richer, as the model of EIA demonstrates, and for this reason the model of Resource PCF in $\mathbf{G}^{\otimes}$ is far from being fully abstract. We now develop a model of Resource PCF which possesses the finite definability property, by applying our general construction to a refined category of exhausting games.

Let A be any arena. We define an equivalence relation $\sim$ on the paths of A as the smallest equivalence relation such that

$$s \cdot o \cdot p \cdot o' \cdot p' \cdot t \sim s \cdot o' \cdot p' \cdot o \cdot p \cdot t$$

where o, o' are O-moves and p, p' are P-moves. We call a path *safe* if, whenever $s = s' \cdot o \cdot p \cdot o' \cdot p' \cdot t$ and o justifies p', p justifies o'. The $\sim$ relation captures a notion of causal independence similar to that of Melliès [22], and allows us to refine our games model to obtain definability for Resource PCF.

A $\sim$-*strategy* σ on an arena A is a set of safe paths that is $\sim$-closed, that is, if $s \in \sigma$ and $s \sim t$ then $t \in \sigma$. A $\sim$-strategy σ is *deterministic* if it is non-empty, and the longest common prefix of any $s, t \in \sigma$ has even length.

Lemma 7.1. *If σ is a $\sim$-strategy and $s \in \sigma$ then s satisfies P-visibility.*

Proof. Let $s = s_1 \cdot p \cdot s_2$, for some P-move p. By permuting moves we can find $s_1' \cdot p \cdot s_2' \sim s$ such that $s_1' = \ulcorner s_1 \urcorner$. The justifier of p must appear in $s_1' = \ulcorner s_1 \urcorner$. $\square$

Given a path s in an arena A, write $\widetilde{s}$ for the equivalence class of s under $\sim$.

Lemma 7.2. *For any safe path s of A, $\widetilde{s}$ is a deterministic $\sim$-strategy, and any deterministic $\sim$-strategy is of the form $\widetilde{s}$ for some safe path s.*

Proof. That each $\widetilde{s}$ is a deterministic $\sim$-strategy is clear: it is a $\sim$-closed set of safe paths by definition, and deterministic because if $s \sim t$ then the successor of every O-move in s is the same as that of the same move in t.

Conversely, let σ be a deterministic $\sim$-strategy. We will show that σ is *history-free*, that is, that there is a function f from the O-moves of A to the P-moves such that for every $s = s_1 \cdot o \cdot p \cdot s_2$ in σ, $p = f(o)$. It then follows that σ is the $\sim$-closure of any $s \in \sigma$, that is, any safe path of the form $o_1 \cdot f(o_1) \cdots o_n \cdot f(o_n)$ Note that such a function f must necessarily be a bijection.

We show simultaneously that every O-move has the same successor in every $s \in \sigma$ and that every P-move has the same predecessor in every $s \in \sigma$, by induction on the position of the earliest occurrence of a move in any $s \in \sigma$.

The only moves that can appear as the first move are the roots of A. Let o be any root. If $s = s_1 \cdot o \cdot p \cdot s_2 \in \sigma$ and also $t = t_1 \cdot o \cdot p' \cdot t_2 \in \sigma$, by $\sim$-closure we also have $o \cdot p \cdot s_1 s_2 \in \sigma$ and $o \cdot p' \cdot t_1 t_2 \in \sigma$, so by determinacy $p = p'$. Similarly, the only moves that can appear as the second move are the immediate descendants of the root. Let p be such a move, appearing as the successor of some root o in some $s \in \sigma$. By the above, p is the successor of o in *every* $s \in \sigma$, and therefore cannot appear as the successor of a different move.

Now consider any non-root O-move o and any path $s \in \sigma$. By definition of $\sim$, s is $\sim$-related to a sequence of the form

$$o_1 \cdot p_1 \cdots o_k \cdot p_k \cdot o_{k+1} \cdots$$

where $o = o_{k+1}$, each o_i is the predecessor of p_i in s, and each p_i enables o_{i+1}; that is to say, the prefix of this sequence up to o has the form of a P-view. We claim that this P-view is the same in every $s \in \sigma$.

The predecessor p_k of o in this view is its ancestor in A, which is unique and therefore the same in each s. Since p_k must occur before o in every s, the inductive hypothesis applies, so p_k's predecessor o_k is the same in each s. Applying this argument repeatedly shows that the view is the same in every s.

Therefore every $s \in \sigma$ is related by $\sim$ to a sequence with the same prefix up to and including o. Since σ is deterministic, the successor of o must be the same in every case, as required. That each P-move has a unique predecessor is established in the same way. $\square$

We can now build two categories:

- $\mathbf{EG}_\sim$ has O-rooted arenas as objects and deterministic $\sim$-strategies on $A^\perp \uplus B$ as maps $A \to B$;
- $\mathbf{EG}_\sim^+$ has the same objects, but its maps $A \to B$ are arbitrary $\sim$-strategies on $A^\perp \uplus B$.

$\mathbf{EG}_\sim^+$ is therefore the subcategory of $\mathbf{EG}$ consisting of $\sim$-closed strategies.

We are ready to construct a Cartesian differential category, starting with $\mathbf{EG}_\sim$. The first step is to take its sup-lattice completion; a consequence of Lemma 7.2 is that this is exactly $\mathbf{EG}_\sim^+$, justifying our choice of nomenclature.

Lemma 7.3. $\mathbf{EG}_\sim^+$ *is the sup-lattice completion of* $\mathbf{EG}_\sim$*.*

Nevertheless, it is convenient to take the first two steps of the construction together, working with $\mathbf{FamRel}(\mathbf{EG}_\sim)$.

Our construction gives us a comonad ! on $\mathcal{K}(\mathbf{FamRel}(\mathbf{EG}_\sim))$, such that the Kleisli category $\mathcal{K}_!(\mathbf{FamRel}(\mathbf{EG}_\sim))$ is a Cartesian differential category. Though $\mathbf{EG}_\sim$ is not monoidal closed, it has all R-exponentials, so the full subcategory of $\mathcal{K}_!(\mathbf{FamRel}(\mathbf{EG}_\sim))$ comprising the arenas with a single root is a Cartesian closed differential category.

As before, we may give a direct definition of a category of games which is a sub-Cartesian-closed-differential-category of this one. Let $\mathbf{G}_\sim$ be the subcategory of $\mathbf{G}$ consisting of $\sim$-closed strategies. Again taking the subcategory of comonoid homomorphisms, we arrive at a Cartesian closed differential category $\mathbf{G}_\sim^\otimes$. Just as in Section 6, we can define a full and faithful functor from

$\mathbf{G}^{\otimes}_{\sim}$ into $\mathcal{K}_!(\mathbf{FamRel}(\mathbf{EG}_{\sim}))$ which preserves the Cartesian closed differential structure.

8. Analysing Models of Resource PCF

Our constructions show that each of $\mathcal{K}_!(\mathbf{FamRel}(\mathbf{EG}_{\sim}))$, $\mathcal{K}_!(\mathbf{BP}(\mathbf{EG}))$ and $\mathcal{K}_!(\mathbf{FamRel}(\mathbf{1}))$ is a cpo-enriched differential Cartesian category with enough exponentials to interpret Resource PCF; and indeed we have identified full subcategories $\mathbf{G}^{\otimes}$, $\mathbf{G}^{\otimes}_{\sim}$ and $\mathbf{MRel}$ which are cpo-enriched Cartesian closed differential categories containing all the objects needed to interpret Resource PCF soundly. However, for $\mathbf{G}^{\otimes}_{\sim}$ and $\mathbf{MRel}$, there is more to be said.

Consider those arenas for which there exists a Q/A-labelling such that every question enables a unique answer — this is a constraint on the shapes of the trees, rather than additional structure. We write $\mathbf{EG}^{QA}_{\sim}$ for the full subcategory of $\mathbf{EG}_{\sim}$ consisting of such arenas, and note that $\mathbf{G}^{\otimes}_{\sim}$ embeds in $\mathcal{K}_!(\mathbf{FamRel}(\mathbf{EG}^{QA}_{\sim}))$ by construction.

Lemma 8.1. *For every such arena, the set of safe paths is non-empty.*

Proof. We construct a safe path by induction on partial paths. Begin with any root node. Having constructed a partial path s, consider the pending question in s. If it enables any questions that do not appear in s, extend s with one of them. Otherwise, extend s with the unique answer of the pending question. If there is no pending question, extend s with any question enabled by one of the answers in the P-view of s, if one exists, or an unplayed root node, if one exists. If no such moves exist, s is a safe path. $\square$

Corollary 8.2. *The unique functor $\top : \mathbf{EG}^{QA}_{\sim} \to \mathbf{1}$ is full. (This amounts to the fact that the set of safe paths of $A^{\perp} \uplus B$ is non-empty.)*

This full functor extends through our constructions to a full functor from $\mathcal{K}_!(\mathbf{FamRel}(\mathbf{EG}^{QA}_{\sim}))$ to $\mathcal{K}_!(\mathbf{FamRel}(\mathbf{1}))$. Moreover, the only idempotents we make use of in the Karoubi envelope have the form $\sum_{f \in G} f$ where G is some group of automorphisms. In the case of $\mathbf{Rel}$, these idempotents are equivalence relations, and an object $(A, \simeq)$ in $\mathcal{K}(\mathbf{Rel})$ is isomorphic to $(A/\simeq, =)$. The part of the Karoubi envelope that is used in our constructions is therefore equivalent to $\mathbf{Rel}$ itself, with the comonad being the usual finite-multiset comonad, and the Kleisli category being $\mathbf{MRel}$. We therefore obtain a full functor from $\mathbf{G}^{\otimes}_{\sim}$ to $\mathbf{MRel}$ which preserves all the relevant structure.

This functor may be described concretely as follows. Given a complete justified sequence s on a QA-arena, write $|s|$ for the underlying multiset of moves of s, partially ordered by the justification relation. The functor sends an arena A to the set of all such pomsets, which we call the *positions* of A. If s is a well-opened complete justified sequence on $A^\perp \uplus B$, $|s|$ is a pair consisting of a multiset of positions of A and a position of B. The functor sends a map $A \to B$ to the set of positions of its sequences. This is essentially the "time-forgetting" map of [23], which here is functorial because of $\sim$-closure.

Theorem 8.3. *The models of Resource PCF in* $\mathbf{G}^{\otimes}_{\sim}$ *and* **MRel** *have the finite definability property: every finite element of the model is the denotation of some term of Resource PCF.*

Proof. For $\mathbf{G}^{\otimes}_{\sim}$, we show that the $\sim$-closure of any finite set of complete plays is the denotation of some term of Resource PCF. By $\sim$-closure, we can restrict our attention to plays satisfying visibility and bracketing. The argument is then a straightforward induction on the length of such plays, following the steps in the definability proof for the innocent strategy model of PCF [24].

Definability for **MRel** follows from the fullness of the positional collapse of $\mathbf{G}^{\otimes}_{\sim}$ onto **MRel**. □

Theorem 8.4. *The model of Resource PCF in* **MRel** *is fully abstract, that is, for any terms* M *and* N, $[\![M]\!] \subseteq [\![N]\!]$ *if and only if whenever* $C(\!|M|\!) \Downarrow$ *for some context* C, *we also have* $C(\!|N|\!) \Downarrow$.

Proof. As usual, soundness of the model together with the monotonicity of all the constructs in its semantics if $[\![M]\!] \subseteq [\![N]\!]$ and $C(\!|M|\!) \Downarrow$, we also have $C(\!|N|\!) \Downarrow$. For the converse, let M and N be closed terms of type A such that $[\![M]\!] \not\subseteq [\![N]\!]$. There must be some $a \in [\![M]\!] \setminus [\![N]\!]$. By finite definability, the relation $\{(a, 0)\} : A \to \mathtt{nat}$ is the denotation of some term $x : A \vdash C(\!|x|\!) : \mathtt{nat}$. Therefore $[\![C(\!|M|\!)]\!] = [\![\mathrm{zero}]\!]$ while $[\![C(\!|N|\!)]\!] = \emptyset$, so $C(\!|M|\!) \Downarrow$ but $C(\!|N|\!) \not\Downarrow$.

We remark that the above argument does not transfer to $\mathbf{G}^{\otimes}_{\sim}$, because the game-semantic equivalent of $\{(a, 0)\}$ would not in general be $\sim$-closed, and indeed this model is not fully abstract, for similar reasons to the failure of full abstraction of the innocent strategy model of PCF [24]. For instance, the model contains strategies for both left-to-right and right-to-left evaluation of the addition function, but no $\sim$-closed strategy (and no Resource PCF context) can distinguish these. □

9. Conclusions

We have presented a construction of differential categories, along with several examples. The construction allows us not only to construct new differential categories, but also to recover some previously known ones, and further, to analyse the structure and properties they possess; in particular, a full abstraction result for the relational model of Resource PCF follows directly from a definability result on the games model. There are several refinements and other directions in which this work can be developed.

9.1. Further examples of differential categories

The relational model allows unconstrained creation and destruction of resources: for any set A, and element $a \in A$ there are relations $\underline{a} : I \to A = \{(*, a)\}$ and $\overline{a} : A \to I = \{(a, *)\}$. Thus the definability and full abstraction results for the relational model depend on the ability to write programs in Resource PCF which can consume any inessential resources by testing their arguments and ignoring the results, which is less straightforward in the differential λ-calculus itself.

A simple refinement of the relational model starts from the observation that there is a simple construction of a differential category from any commutative monoid, $(M, \odot, e)$. Applying the **FamRel** construction to M (as a discrete symmetric monoidal category) yields a category in which:

- Objects are pairs (X, f), where X is a set and $f : X \to M$ is a function determining a "weight" in M for each element in X.
- Morphisms from (X, f) to (Y, g) are relations $R : X \to Y$ such that $(x, y) \in R$ implies $f(x) = g(y)$.

The symmetric monoidal structure on $\mathbf{FamRel}(M)$ is given by $(X, f) \otimes (Y, g) = (X \times Y, \lambda\langle x, y\rangle.f(x) \odot g(y))$. $\mathbf{FamRel}(M)$ has symmetric tensor powers: $(A, f)^n = (A^n, f_n)$, where A^n is the set of unordered n-tuples over A and $f_n\{x_1, \ldots, x_n\} = \bigodot_{i=1}^n f(x_i)$. Since the union of disjoint sets is a biproduct (as in **Rel**) the free cocommutative comonoid on (A, f) is the set of the finite multisets on A, with the weight of each multiset being the sum of the weight of its members.

If M is an *abelian group* then $\mathbf{FamRel}(M)$ is compact closed, and hence symmetric monoidal closed, with $(A, f) \multimap (B, g) = (A \times B, \lambda\langle x, y\rangle.f(x)^{-1} \odot g(y))$.

Weighting with values in M allows the creation and destruction of resources to be controlled — the relations $\underline{a} : I \to A$ and $\overline{a} : A \to I$ are morphisms only if a has weight zero. If we take M to be the integers (the free abelian group), so that objects in $\mathbf{FamRel}(Int)$ are signed multisets, we can capture one aspect of our games models — the quantity of resources (moves) to be consumed — but not the order in which this occurs, via the strong monoidal functor from $\mathbf{EG}$ into Int which sends each arena A to the difference between the numbers of Opponent and Player moves in A: the existence of an even length path on $A \to B$ entails that the O/P differences of A and B are equal. This lifts to a functor of differential categories: since any arena for which there exists a Q/A labelling has a difference of zero, for $\mathbf{G}^{\otimes}$, and $\mathbf{G}^{\otimes}_{\sim}$ this is equivalent to projection into $\mathbf{MRel}$.

It would be interesting to explore examples of this kind in more detail, to develop a range of models that explicitly capture resource usage in various ways and understand the connections between them.

9.2. Further directions

For the examples in this paper, our construction has made the simplifying assumption that the formal sums used to define symmetric tensor powers in the Karoubi envelope are given by the same commutative monoid enrichment required for a differential category, although the properties required of them are distinct: idempotency in the first case, and associativity (and a unit) in the second. As we have hinted, we are interested in models in which they are not the same, although in any category enriched with both such structures, the idempotent n-ary sum must be a *division by* n of the commutative monoid, in a formal sense. We believe this allows the *Taylor expansion* of morphisms in a differential category to be made precise, and leads to some more "quantitative" models of resource use. It also suggests an extension of the syntax of the differential λ-calculus to include such a notion of division.

In this paper, we have confined our attention to the interpretation of typed differentials: specifically, resource PCF. Further work is required to investigate the requirements on a symmetric monoidal category to construct a model of the untyped differential λ-calculus, and the relationship of that model to the underlying category. In particular, the reflexive objects in our categories of games remain to be studied.

Evidently, there is a wide variety of symmetric monoidal categories to which our construction of free cocommutative comonoids and differential categories may be applied: we have barely scratched the surface. Studying some of these should bring further understanding of the differential operator, and its potential applications: for example, SMCs with analytical properties such as metric spaces. One intriguing observation is that the construction of the Fock space of a Hilbert space, and its creation and annihilation operators appears to correspond to our construction of the free cocommutative comonoid, and its differential operator. Can the syntax and semantics of the differential λ-calculus be used to describe quantum mechanics and its relationship with computation?

References

[1] T. Ehrhard, L. Regnier, The differential lambda-calculus, Theor. Comput. Sci. 309 (2003) 1–41.

[2] P. Tranquilli, Intuitionistic differential nets and lambda-calculus, Theoretical Computer Science In Press, Corrected Proof (2010) –.

[3] G. Boudol, The lambda-calculus with multiplicities (abstract), in: E. Best (Ed.), CONCUR, volume 715 of *Lecture Notes in Computer Science*, Springer, 1993, pp. 1–6.

[4] A. Bucciarelli, T. Ehrhard, G. Manzonetto, Categorical models for simply typed resource calculi, Electronic Notes in Theoretical Computer Science 265 (2010) 213 – 230. Proceedings of the 26th Conference on the Mathematical Foundations of Programming Semantics (MFPS 2010).

[5] R. F. Blute, J. R. B. Cockett, R. A. G. Seely, Differential categories, Mathematical. Structures in Comp. Sci. 16 (2006) 1049–1083.

[6] R. Harmer, G. McCusker, A fully abstract game semantics for finite nondeterminism, in: Proceedings, Fourteenth Annual IEEE Symposium on Logic in Computer Science, IEEE Computer Society Press, 1999, pp. 422 – 430.

[7] R. F. Blute, J. R. B. Cockett, R. A. G. Seely, Cartesian differential categories, Theory and Applications of Categories 22 (2009) 622–672.

[8] J. Laird, G. Manzonetto, G. McCusker, Constructing differential categories and deconstructing categories of games, in: L. Aceto, M. Henzinger, J. Sgall (Eds.), ICALP (2), volume 6756 of *Lecture Notes in Computer Science*, Springer, 2011, pp. 186–197.

[9] M. Pagani, P. Tranquilli, Parallel reduction in resource lambda-calculus, in: Z. Hu (Ed.), Programming Languages and Systems, 7th Asian Symposium, APLAS 2009. Proceedings, volume 5904 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 226–242.

[10] G. Manzonetto, M. Pagani, Böhm's theorem for resource lambda calculus through Taylor expansion, in: C.-H. L. Ong (Ed.), TLCA, volume 6690 of *Lecture Notes in Computer Science*, Springer, 2011, pp. 153–168.

[11] M. Pagani, S. R. D. Rocca, Linearity, non-determinism and solvability, Fundam. Inform. 103 (2010) 173–202.

[12] T. Ehrhard, L. Regnier, Uniformity and the Taylor expansion of ordinary lambda-terms, Theor. Comput. Sci. 403 (2008) 347–372.

[13] T. Ehrhard, L. Regnier, Böhm trees, Krivine's machine and the Taylor expansion of lambda-terms, in: A. Beckmann, U. Berger, B. Löwe, J. V. Tucker (Eds.), CiE, volume 3988 of *Lecture Notes in Computer Science*, Springer, 2006, pp. 186–197.

[14] R. Blute, T. Ehrhard, C. Tasson, A convenient differential category, CoRR abs/1006.3140 (2010).

[15] A. Carraro, T. Ehrhard, A. Salibra, Exponentials with infinite multiplicities, in: A. Dawar, H. Veith (Eds.), CSL, volume 6247 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 170–184.

[16] P.-A. Melliès, N. Tabareau, C. Tasson, An explicit formula for the free exponential modality of linear logic, in: S. Albers, A. Marchetti-Spaccamela, Y. Matias, S. Nikoletseas, W. Thomas (Eds.), Automata, Languages and Programming, volume 5556 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2009, pp. 247–260.

[17] G. D. Plotkin, LCF considered as a programming language, Theor. Comput. Sci. 5 (1977) 223–255.

[18] V. Danos, L. Regnier, Head Linear Reduction, Technical Report, Université Paris 7 and Université Aix-Marseille 2, 2004.

[19] R. Harmer, Games and full abstraction for nondeterministic languages, Ph.D. thesis, University of London, 1999.

[20] S. Abramsky, G. McCusker, Linearity, sharing and state: a fully abstract game semantics for Idealized Algol with active expressions, in: P. W. O'Hearn, R. D. Tennent (Eds.), Algol-like Languages, Birkhaüser, 1997, pp. 297–329 of volume 2.

[21] S. Mac Lane, Categories for the Working Mathematician, Springer-Verlag, Berlin, 1971.

[22] P.-A. Melliès, Asynchronous games 2: the true concurrency of innocence, Theor. Comput. Sci. 358 (2006) 200–228.

[23] P. Baillot, V. Danos, T. Ehrhard, L. Regnier, Timeless games, in: M. Nielsen, W. Thomas (Eds.), Computer Science Logic: 11th International Workshop Proceedings, Lecture Notes in Computer Science, Springer-Verlag, 1998, pp. 56–77.

[24] J. M. E. Hyland, C.-H. L. Ong, On full abstraction for PCF: I, II and III, Information and Computation 162 (2000) 285–408.