



# Cartesian closed 2-categories and permutation equivalence in higher-order rewriting

Tom Hirschowitz

► **To cite this version:**

Tom Hirschowitz. Cartesian closed 2-categories and permutation equivalence in higher-order rewriting. Logical Methods in Computer Science, Logical Methods in Computer Science Association, 2013, 9 (3), pp.10. .

**HAL Id: hal-00540205**

**<https://hal.archives-ouvertes.fr/hal-00540205v2>**

Submitted on 29 Jan 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# CARTESIAN CLOSED 2-CATEGORIES AND PERMUTATION EQUIVALENCE IN HIGHER-ORDER REWRITING

TOM HIRSCHOWITZ

ABSTRACT. We propose a semantics for permutation equivalence in higher-order rewriting. This semantics takes place in cartesian closed 2-categories, and is proved sound and complete.

## 1. INTRODUCTION

It is known since the end of the 80's that 2-categories with finite products provide a semantics for term rewriting [3]. *Higher-order rewriting* [10, 17, 14, 15] is a framework for specifying rewrite systems on terms with variable binding. Many results from standard term rewriting have been generalised to higher-order rewriting, notably normalisation or confluence results. An important tool for confluence results is the notion of *permutation equivalence*, which was generalised to the higher-order case by Bruggink [1]. He defines a calculus of *proof terms* for specifying reductions in a higher-order rewrite system.

We here propose a categorical semantics for a variant of this calculus, in terms of *cartesian closed 2-categories*. We first define *cartesian closed 2-signatures*, which generalise higher-order rewrite systems, and organise them into a category  $\text{Sig}$ . We then construct an adjunction

$$(1.1) \quad \text{Sig} \begin{array}{c} \xrightarrow{\mathcal{H}} \\ \perp \\ \xleftarrow{\mathcal{W}} \end{array} 2\text{CCCat},$$

where  $2\text{CCCat}$  is the category of small cartesian closed 2-categories. From a given higher-order rewrite system  $S$ , the functor  $\mathcal{H}$  constructs a cartesian closed 2-category, whose 2-cells are Bruggink's proof terms modulo permutation equivalence, which we prove is the free cartesian closed 2-category generated by  $S$ .

We review a number of examples and non-examples, and sketch an extension to deal with the latter.

**Related work.** Our cartesian closed 2-signatures are a 2-dimensional refinement of *cartesian closed sketches* [16, 4, 9]. Bruggink's calculus of permutation equivalence is close in spirit to Hilken's 2-categorical semantics of the simply-typed  $\lambda$ -calculus [7], but technically different and generalised to arbitrary higher-order rewrite systems. Capriotti [2] proposes a semantics of so-called *flat* permutation equivalence in sesquicategories. More related work is discussed in Section 4.2.

---

1991 *Mathematics Subject Classification.* D.1.1;D.3.1;F.3.2;F.4.1;

*Key words and phrases.* Cartesian closed 2-categories, lambda calculus, higher-order rewriting, combinatory reduction systems, categorical semantics.

This work has been partially funded by the French ANR projet blanc "Curry Howard pour la Concurrence" CHOCO BLAN07-1.189926.

## 2. CARTESIAN CLOSED SIGNATURES AND CATEGORIES

We start by recalling the well-known, or at least folklore, adjunction between what we here call (*cartesian closed*) *1-signatures* and cartesian closed categories.

For any set  $X$ , define *types* over  $X$  by the grammar:

$$A, B, \dots \in \mathcal{L}_0(X) \quad ::= \quad x \mid 1 \mid A \times B \mid B^A,$$

with  $x \in X$ .

**Proposition 1.**  $\mathcal{L}_0$  defines a monad on  $\mathbf{Set}$ .

Let the set of *sequents* over a set  $X$  be  $\mathcal{S}_0(X) = \mathcal{L}_0(X)^* \times \mathcal{L}_0(X)$ , i.e., sequents are pairs of a list of types and a type. The assignment  $X \mapsto \mathcal{S}_0(X)$  extends to an endofunctor on  $\mathbf{Set}$ .

**Definition 1.** A *1-signature* consists of a set  $X_0$  of sorts, and an  $\mathcal{S}_0(X_0)$ -indexed set  $X_1$  of operations, or equivalently a map  $X_1 \rightarrow \mathcal{S}_0(X_0)$ .

A *morphism of 1-signatures*  $(X_0, X_1) \rightarrow (Y_0, Y_1)$  is a pair  $(f_0, f_1)$  where  $f_i: X_i \rightarrow Y_i$  such that

$$\begin{array}{ccc} X_1 & \xrightarrow{f_1} & Y_1 \\ \downarrow & & \downarrow \\ \mathcal{S}_0(X_0) & \xrightarrow{\mathcal{S}_0(f_0)} & \mathcal{S}_0(Y_0) \end{array}$$

commutes. Morphisms compose in the obvious way, and we have:

**Proposition 2.** *Composition of morphisms is associative and unital, and hence 1-signatures and their morphisms form a category  $\mathbf{Sig}_1$ .*

There is a well-known adjunction

$$\begin{array}{ccc} & \mathcal{H}_1 & \\ & \curvearrowright & \\ \mathbf{Sig}_1 & \xleftrightarrow{\perp} & \mathbf{CCCat} \\ & \curvearrowleft & \\ & \mathcal{W}_1 & \end{array}$$

between 1-signatures and the category  $\mathbf{CCCat}$  of small cartesian closed categories (with chosen structure) and (strict) cartesian closed functors.

The functor  $\mathcal{W}_1$  sends a cartesian closed category  $\mathcal{C}$  to the signature with sorts  $\mathcal{C}_0$ , its set of objects, and with operations  $A_1, \dots, A_n \rightarrow A$  the set  $\mathcal{C}(\llbracket A_1 \times \dots \times A_n \rrbracket, \llbracket A \rrbracket)$ , where  $\llbracket - \rrbracket$  denotes the function  $\mathcal{L}_0(\mathcal{C}_0) \rightarrow \mathcal{C}_0$  defined by induction:

$$(2.1) \quad \begin{aligned} \llbracket c \rrbracket &= c & c \in \mathcal{C}_0 \\ \llbracket 1 \rrbracket &= 1 \\ \llbracket A \times B \rrbracket &= \llbracket A \rrbracket \times \llbracket B \rrbracket \\ \llbracket B^A \rrbracket &= \llbracket B \rrbracket^{\llbracket A \rrbracket}. \end{aligned}$$

Conversely, given a 1-signature  $X$ , consider the simply-typed  $\lambda$ -calculus with base types in  $X_0$  and constants in  $X_1$ . Terms modulo  $\beta\eta$  form a category  $\mathcal{H}_1(X)$  with objects all types over  $X_0$  and morphisms  $A \rightarrow B$  all terms of type  $B$  with one free variable of type  $A$ .

A less often formulated observation, which is useful to us, is that the adjunction  $\mathcal{H}_1 \dashv \mathcal{W}_1$  decomposes into two adjunctions

$$\begin{array}{ccccc}
& & \mathcal{L}_1 & & \mathcal{F}_1 \\
& \swarrow & \xrightarrow{\quad} & \searrow & \xrightarrow{\quad} \\
\text{Sig}_1 & \xrightarrow{\quad} & \mathcal{L}_1\text{-Alg} & \xrightarrow{\quad} & \text{CCCat}, \\
& \nwarrow & \xleftarrow{\quad} & \swarrow & \xleftarrow{\quad} \\
& & \mathcal{U}_1 & & \mathcal{V}_1
\end{array}$$

where  $\mathcal{L}_1\text{-Alg}$  is the category of algebras for the monad  $\mathcal{L}_1$  defined as follows (and  $\mathcal{L}_1$  is shorthand for the functor  $X \mapsto (\mathcal{L}_1(X), \mu)$ ).

For any 1-signature  $X$ , let  $\mathcal{L}_1(X)$  denote the 1-signature with

- as sorts the set  $X_0$ , and
- as operations  $\Gamma \vdash A$  the  $\lambda$ -terms  $\Gamma \vdash M : A$ , modulo  $\beta\eta$ .

$\mathcal{L}_1$  extends to an endofunctor on  $\text{Sig}_1$ , whose action on morphisms of 1-signatures  $X \xrightarrow{f} Y$  substitutes constants  $c \in X_1$  with  $f_1(c)$ . We obtain

**Proposition 3.**  $\mathcal{L}_1$  is a monad on  $\text{Sig}_1$ .

The functor  $\mathcal{V}_1$  sends any cartesian closed category  $\mathcal{C}$  to the  $\mathcal{L}_1$ -algebras  $(\mathcal{C}_0, \mathcal{C}_1)$  defined as follows. First,  $\mathcal{C}_0$  is the set of objects of  $\mathcal{C}$ . It has a canonical  $\mathcal{L}_0$ -algebra structure, say  $h_0 : \mathcal{L}_0(\mathcal{C}_0) \rightarrow \mathcal{C}_0$ , obtained by interpreting type constructors in  $\mathcal{C}$  as in (2.1). Extending this to contexts  $G$  by  $h_0(G) = \prod_i h_0(G_i)$ , let the operations in  $\mathcal{C}_1(G, A)$  be the 1-cells in  $\mathcal{C}(h_0(G), h_0(A))$ . Beware: the domain and codomain of such an operation are really  $G$  and  $A$ , not  $h_0(G)$  and  $h_0(A)$ . Similarly, interpreting the  $\lambda$ -calculus in  $\mathcal{C}$ , the 1-signature  $(\mathcal{C}_0, \mathcal{C}_1)$  has a canonical  $\mathcal{L}_1$ -algebra structure, say  $h_1 : \mathcal{L}_1(\mathcal{C}_0, \mathcal{C}_1) \rightarrow (\mathcal{C}_0, \mathcal{C}_1)$ :

$$\begin{aligned}
h_1(G \vdash x_i : G_i) &= \pi_i \\
h_1(G \vdash () : 1) &= ! \\
h_1(G \vdash c(M_1, \dots, M_n)) &= c \circ (h_1(M_1), \dots, h_1(M_n)) \\
h_1(G \vdash \lambda x : A. M : B^A) &= \varphi(h_1(G, x : A \vdash M : B)) \\
h_1(G \vdash MN : B) &= ev \circ (h_1(M), h_1(N)) \\
h_1(G \vdash (M, N) : A \times B) &= (h_1(M), h_1(N)) \\
h_1(G \vdash \pi M : A) &= \pi \circ M \\
h_1(G \vdash \pi' M : A) &= \pi' \circ M,
\end{aligned}$$

where  $!$  is the unique morphism  $h_0(G) \rightarrow 1$ ,  $\varphi$  is the bijection  $\mathcal{C}(h_0(G, A), h_0(B)) \cong \mathcal{C}(h_0(G), h_0(B^A))$ , and  $ev$  is the structure morphism  $h_0(B^A \times A) \rightarrow h_0(B)$ .

$\mathcal{L}_1$ -algebras are much like cartesian closed categories whose objects are freely generated by their set of sorts. A perhaps useful analogy here is with multicategories  $\mathcal{M}$ , seen as being close to monoidal categories whose objects are freely generated by those of  $\mathcal{M}$  by tensor and unit. Here, the functor  $\mathcal{F}_1$  sends any  $\mathcal{L}_1$ -algebra  $(X, h)$  to the cartesian closed category with

- objects the types over  $X_0$ , i.e.,  $\mathcal{L}_0(X_0)$ ,
- morphisms  $A \rightarrow B$  the set of operations in  $X_1(A, B)$ .

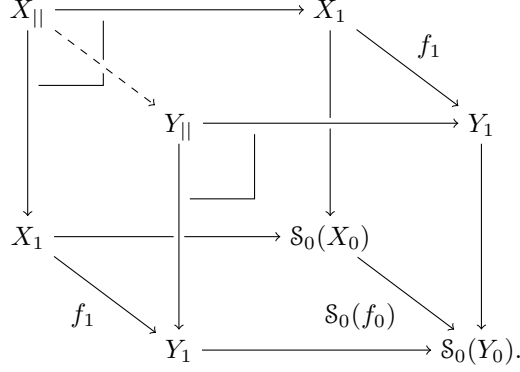
This canonically forms a cartesian closed category, with structure induced by the  $\mathcal{L}_1$ -algebra structure. We define it in more detail in dimension 2 in Section 7.2.

### 3. CARTESIAN CLOSED 2-SIGNATURES

Given a 1-signature  $X$ , let  $X_{||}$  denote the set of pairs of parallel operations, i.e., pairs of operations  $M, N$  above the same sequent. Otherwise said,  $X_{||}$  is the pullback

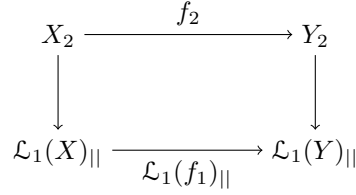
$$\begin{array}{ccc}
X_{||} & \xrightarrow{\quad} & X_1 \\
\downarrow & \lrcorner & \downarrow \\
X_1 & \xrightarrow{\quad} & \mathfrak{S}_0(X_0).
\end{array}$$

Any morphism  $f: X \rightarrow Y$  of 1-signatures yields a function  $f_{||}: X_{||} \rightarrow Y_{||}$ , via the dashed arrow (obtained by universal property of pullback) in



**Definition 2.** A 2-signature consists of a 1-signature  $X$ , plus a set  $X_2$  of reduction rules with a function  $X_2 \rightarrow \mathcal{L}_1(X)_{||}$ .

A morphism of 2-signatures  $(X, X_2) \rightarrow (Y, Y_2)$  is a pair  $(f, f_2)$  where  $f: X \rightarrow Y$  is a morphism of 1-signatures and  $f_2: X_2 \rightarrow Y_2$  makes the diagram



commute. We obtain:

**Proposition 4.** Composition of morphisms is associative and unital, and hence 2-signatures and their morphisms form a category  $\text{Sig}$ .

#### 4. EXAMPLES

**4.1. Higher-order rewrite systems.** The prime example of a 2-signature is that for the pure  $\lambda$ -calculus: it has a sort  $t$  and operations

$$a: t \times t \rightarrow t \qquad \ell: t^t \rightarrow t,$$

with a reduction rule  $\beta$  above the pair

$$x: t^t, y: t \vdash a(\ell(x), y), x(y): t$$

in  $\mathcal{L}_1(\{t\}, \{\ell, a\})_{||}$ . Categorically, this will yield a 2-cell

$$\begin{array}{ccc} & t \times t & \\ \ell \times t \nearrow & \Downarrow \beta & \searrow a \\ t^t \times t & & t. \\ & \text{ev} & \end{array}$$

This is an example of a *higher-order rewrite system* in the sense of Nipkow [14]. Nipkow's definition is formally different, but his higher-order rewrite systems are in bijection with 2-signatures  $h: X_2 \rightarrow \mathcal{L}_1(X)_{||}$  such that for all rules  $r \in X_2$ , letting  $(\Gamma \vdash M, N: A) = h(r)$ :

- $M$  is not a variable,
- $A$  is a sort,
- each variable occurring in  $\Gamma$  occurs free in  $M$ .

These restrictions help formulating and proving decidability problems on higher-order rewrite systems, whose extension to our setting we leave open.

Let us now anticipate over our main results below and state our soundness and completeness theorem. Given a higher-order rewrite system  $X$ , i.e., a 2-signature satisfying the above conditions, let  $\mathcal{R}(X)$  be the following locally-preordered 2-category. It has:

- objects are types in  $\mathcal{L}_0(X_0)$ ;
- morphisms  $A \rightarrow B$  are  $\lambda$ -terms in  $\mathcal{L}_1(X)(A \vdash B)$ , modulo  $\beta\eta$ ;
- given two parallel morphisms  $M$  and  $N$ , there is one 2-cell  $M \rightarrow N$  exactly when there is a sequence of reductions  $M \rightarrow^* N$  in the usual sense [14].

**Proposition 5.**  $\mathcal{R}(X)$  is 2-cartesian closed.

$\mathcal{R}(X)$  and  $\mathcal{H}(X)$  have the same objects and morphisms. But because our inference rules for forming reductions are the same as deduction rules for proving the existence of a reduction in the usual sense, we may send any reduction  $P: M \rightarrow N$  to the unique reduction  $M \rightarrow N$  in  $\mathcal{R}(X)$ .

**Theorem 1** (Soundness and completeness). *This defines an identity-on-objects, identity-on-morphisms, locally full cartesian closed 2-functor  $\mathcal{R}(X) \xrightarrow{1} \mathcal{H}(X)$ .*

**4.2. Theories with binding.** Understanding reduction rules as equations, it is easy to define the free cartesian closed category generated by a 2-signature. This yields an adjunction

$$(4.1) \quad \text{Sig} \begin{array}{c} \xrightarrow{\mathcal{H}'} \\ \perp \\ \xleftarrow{\mathcal{W}'} \end{array} \text{CCCat.}$$

This adjunction provides a categorical semantics for theories with binding, which is more general than other approaches by Fiore and Hur [6], Hirschowitz and Maggesi [8], and Zsidó [18], and which is in line with Lambek's seminal paper [11].

If I understand correctly, the motivation for Fiore and Hur's subtle approach is the will to explain the  $\lambda$ -calculus by strictly less than itself. The present framework does not obey this specification, and instead tends to view the  $\lambda$ -calculus as a universal (parameterised) theory with binding.

We end this section by giving a formal construction of the adjunction (4.1). Cartesian closed categories form a full, reflective subcategory of  $2\text{CCCat}$ , via the functor  $\mathcal{J}: 2\text{CCCat} \rightarrow \text{CCCat}$  sending a cartesian closed 2-category  $\mathcal{C}$  to the cartesian closed category with:

- objects those of  $\mathcal{C}$ ,
- morphisms those of  $\mathcal{C}$ , modulo the congruence generated by  $f \sim g$  iff there exists a 2-cell  $f \rightarrow g$ .

Here,  $\mathcal{J}(\mathcal{C})$  is thought of as the free cartesian closed 2-category with trivial 2-cells (i.e., 0 or 1). The desired adjunction is obtained by composing the adjunctions

$$\text{Sig} \begin{array}{c} \xrightarrow{\mathcal{H}} \\ \perp \\ \xleftarrow{\mathcal{W}} \end{array} 2\text{CCCat} \begin{array}{c} \xrightarrow{\mathcal{J}} \\ \perp \\ \xleftarrow{\quad} \end{array} \text{CCCat.}$$

**4.3. Non-examples.** Non-examples are given by calculi whose reduction semantics is defined on terms modulo a so-called *structural congruence*, e.g., CCS [12], or the  $\pi$ -calculus [5, 13].

For example, consider the CCS term  $(a \mid 0) \mid \bar{a}$ . In CCS, it is *structurally* equivalent to  $(a \mid \bar{a}) \mid 0$ , which then reduces to  $0 \mid 0$ .

In order to account for this, we would have to consider a 2-signature with reduction rules for structural congruence, here  $(M_1 \mid M_2) \mid M_3 \rightarrow M_1 \mid (M_2 \mid M_3)$  for associativity, and  $M \mid N \rightarrow N \mid M$  for commutativity. But then, these reductions count as proper reductions, which departs from the desired computational behaviour. For example, the term  $a \mid a$  has an infinite reduction sequence, using commutativity.

Anticipating the development in the next sections, a potential solution is to extend 2-signatures to *2-theories*. For any 2-signature  $X$ , let  $X_{\parallel}$  denote the set of pairs of reduction rules  $r, s$  with a common type  $G \vdash M \rightarrow N : A$ . A 2-theory is a 2-signature  $X$ , together with a set of equations between parallel reductions, i.e., a subset  $X_3$  of  $\mathcal{L}(X)_{\parallel}$ .

The main adjunction announced above (1.1) extends to an adjunction between 2-theories and cartesian closed 2-categories. Using equations, we may specify that any reduction  $M \rightarrow M$  using only structural rules be the identity on  $M$ , and consider the computational behaviour of a 2-category to consist of its non-invertible morphisms, as proposed by Hilken [7]. A question is whether for a given calculus this can be done with finitely many equations.

## 5. A 2-LAMBDA-CALCULUS

We now begin the construction of Adjunction (1.1). We start in this section by defining a monad  $\mathcal{L}$  on  $\text{Sig}$ , which we will use to factor Adjunction (1.1) as

$$\text{Sig} \begin{array}{c} \xrightarrow{\mathcal{L}} \\ \perp \\ \xleftarrow{\mathcal{U}} \end{array} \mathcal{L}\text{-Alg} \begin{array}{c} \xrightarrow{\mathcal{F}} \\ \perp \\ \xleftarrow{\mathcal{V}} \end{array} 2\text{CCCat},$$

where:

- $\mathcal{L}\text{-Alg}$  is the category of  $\mathcal{L}$ -algebras,
- $\mathcal{L} : \text{Sig} \rightarrow \mathcal{L}\text{-Alg}$  is a shortcut for  $X \mapsto (\mathcal{L}^2 X \xrightarrow{\mu} \mathcal{L} X)$ ,
- $\mathcal{U}(\mathcal{L} X \xrightarrow{h} X) = X$ ,
- $2\text{CCCat}$  is the category of cartesian closed 2-categories, which we define in Section 6.

The left-hand adjunction holds by  $\mathcal{L}$  being a monad, thus we concentrate in Section 7 on establishing the right-hand one.

But for now, let us define the monad  $\mathcal{L}$ .

**5.1. Syntax.** Given a 2-signature  $X = ((X_0, X_1), h : X_2 \rightarrow \mathcal{L}_1(X)_{\parallel})$  (actually  $\mathcal{L}_1(X)$  is  $\mathcal{L}_1(X_0, X_1)$ ), we construct a new 2-signature  $\mathcal{L}(X)$ , whose reduction rules represent reduction sequences in the “higher-order rewrite system” defined by  $X$ , modulo permutation equivalence. The 2-signature  $\mathcal{L}(X)$  has the same base 1-signature  $(X_0, X_1)$ , and as reduction rules the terms of a  $2\lambda$ -calculus (in the sense of Hilken [7]) modulo permutation equivalence, which we now define.

First, terms, called *reductions*, are defined by induction in Figure 1. The typing judgement has the shape  $\Gamma \vdash P : M \rightarrow N : A$ , where  $A$  is a type in  $\mathcal{L}_0(X_0)$ ,  $\Gamma$  is a list of pairs of a variable and a type, with no variable appearing more than once,  $M$  and  $N$  are terms of type  $\Gamma \vdash A$  modulo  $\beta\eta$ , and  $P$  is a reduction. In the sequel, we often forget the variables in such pairs  $(\Gamma \vdash A)$ , and identify them with sequents in  $\mathcal{S}_0(X_0)$ .

When clear from context, we abbreviate substitutions  $[M_1/x_1, \dots, M_n/x_n]$  by  $[M_1, \dots, M_n]$ . For a context  $G$ ,  $G_i$  denotes its  $i$ th type. Also, for  $(M, N) \in \mathcal{L}_1(X)_{\parallel}$ , we let  $X(M, N)$  be the set of all reduction rules  $r \in X_2$  such that  $h(r) = (M, N)$ . We write  $X(\Gamma \vdash M, N : A)$  to indicate the common type of  $M$  and  $N$ . Similarly,  $X(G \vdash A)$  denotes the set of operations in  $X_1$  above  $G \vdash A$ .

$$\begin{array}{c}
\dfrac{\dots \quad \Gamma \vdash P_i : M_i \rightarrow N_i : G_i \quad \dots}{\Gamma \vdash r(P_1, \dots, P_n) : M[M_1, \dots, M_n] \rightarrow N[N_1, \dots, N_n] : A} \quad (r \in X(G \vdash M, N : A)) \\
\\
\dfrac{\Gamma \vdash P : M_1 \rightarrow M_2 : A \quad \Gamma \vdash Q : M_2 \rightarrow M_3 : A}{\Gamma \vdash P ;_{M_2} Q : M_1 \rightarrow M_3 : A} \quad \Gamma, x : A, \Delta \vdash x : x \rightarrow x : A \\
\\
\Gamma \vdash () : () \rightarrow () : 1 \\
\\
\dfrac{\Gamma \vdash P_1 : M_1 \rightarrow N_1 : G_1 \quad \dots \quad \Gamma \vdash P_n : M_n \rightarrow N_n : G_n}{\Gamma \vdash c(P_1, \dots, P_n) : c(M_1, \dots, M_n) \rightarrow c(N_1, \dots, N_n) : A} \quad (c \in X_1(G \vdash A)) \\
\\
\dfrac{\Gamma, x : A \vdash P : M \rightarrow N : B}{\Gamma \vdash \lambda x : A. P : \lambda x : A. M \rightarrow \lambda x : A. N : B^A} \\
\\
\dfrac{\Gamma \vdash P : M \rightarrow M' : B^A \quad \Gamma \vdash Q : N \rightarrow N' : A}{\Gamma \vdash PQ : MN \rightarrow M'N' : B} \\
\\
\dfrac{\Gamma \vdash P : M \rightarrow M' : A \quad \Gamma \vdash Q : N \rightarrow N' : B}{\Gamma \vdash (P, Q) : (M, N) \rightarrow (M', N') : A \times B} \\
\\
\dfrac{\Gamma \vdash P : M \rightarrow N : A \times B}{\Gamma \vdash \pi_{A,B} P : \pi_{A,B} M \rightarrow \pi_{A,B} N : A} \quad \dfrac{\Gamma \vdash P : M \rightarrow N : A \times B}{\Gamma \vdash \pi'_{A,B} P : \pi'_{A,B} M \rightarrow \pi'_{A,B} N : B}
\end{array}$$

FIGURE 1. Reductions

5.2. **Substitution.** Next, we define substitution, which has “type”

$$(5.1) \quad \dfrac{\Gamma \vdash Q : N \rightarrow N' : \Delta \quad \Delta \vdash P : M \rightarrow M' : A}{\Gamma \vdash P[Q] : M[N] \rightarrow M'[N'] : A},$$

i.e., given a reduction  $P$  and a tuple of reductions  $Q$ , it produces a reduction of the indicated type, which we denote  $P[Q]$ . Here, we denote by  $\Gamma \vdash Q : N \rightarrow N' : \Delta$  a tuple of reductions  $\Gamma \vdash Q_i : N_i \rightarrow N'_i : \Delta_i$ , for  $1 \leq i \leq |\Delta|$ .

The definition is a bit tricky:

- first we define *left whiskering*, which has “type”

$$\dfrac{\Gamma \vdash Q : N \rightarrow N' : \Delta \quad \Delta \vdash M : A}{\Gamma \vdash M[Q] : M[N] \rightarrow M'[N'] : A},$$

- then we define *right whiskering*, which has “type”

$$\dfrac{\Gamma \vdash N : \Delta \quad \Delta \vdash P : M \rightarrow M' : A}{\Gamma \vdash P[N] : M[N] \rightarrow M'[N] : A},$$

(where  $N$  denotes a tuple),

- then we define substitution by

$$P[Q] = (P[N] ;_{M'[N]} M'[Q]).$$

There is of course another legitimate definition, namely

$$P[Q] = (M[Q] ;_{M[N']} P[N']).$$

The two will be equated by permutation equivalence in the next section.



Left whiskering is defined by induction, with  $\Delta = (x_1 : A_1, \dots, x_n : A_n)$  and  $Q = (Q_1, \dots, Q_n)$ , by:

$$\begin{aligned}
() [Q] &= () \\
x_i [Q] &= Q_i \\
c(M_1, \dots, M_p) [Q] &= c(M_1 [Q], \dots, M_p [Q]) \\
(\lambda x : B. M) [Q] &= \lambda x : B. (M [Q, x]) \quad (\text{for } x \notin \text{dom}(\Delta)) \\
(MN) [Q] &= (M [Q] N [Q]) \\
(M, N) [Q] &= (M [Q], N [Q]) \\
(\pi_{A,B} M) [Q] &= \pi_{A,B} (M [Q]) \\
(\pi'_{A,B} M) [Q] &= \pi'_{A,B} (M [Q]).
\end{aligned}$$

Right whiskering is defined by induction, with  $\Delta = (x_1 : A_1, \dots, x_n : A_n)$  and  $N = (N_1, \dots, N_n)$ , by:

$$\begin{aligned}
(r(P_1, \dots, P_p)) [N] &= r(P_1 [N], \dots, P_p [N]) \\
(P_1 ;_{M''} P_2) [N] &= (P_1 [N] ;_{M'' [N]} P_2 [N]) \\
() [N] &= () \\
x_i [N] &= N_i \\
c(P_1, \dots, P_p) [N] &= c(P_1 [N], \dots, P_p [N]) \\
(\lambda x : B. P') [N] &= \lambda x : B. (P' [N, x]) \quad (\text{for } x \notin \text{dom}(\Delta)) \\
(P_1 P_2) [N] &= (P_1 [N] P_2 [N]) \\
(P_1, P_2) [N] &= (P_1 [N], P_2 [N]) \\
(\pi_{A,B} P') [N] &= \pi_{A,B} (P' [N]) \\
(\pi'_{A,B} P') [N] &= \pi'_{A,B} (P' [N]).
\end{aligned}$$

**Definition 3.** Let  $P[Q] = (P[N] ;_{M'[N]} M'[Q])$ .

**Proposition 6.** Given reductions  $P$  and  $Q$  as above, the capture-avoiding substitution  $P[Q]$  is a well-typed reduction  $\Gamma \vdash P[Q] : M[N] \rightarrow M'[N'] : A$ .

Similarly, there is a weakening operation with “type”

$$\frac{\Gamma \vdash P : M \rightarrow N : A}{\Gamma, x : B \vdash P : M \rightarrow N : A.} \quad (x \notin \Gamma)$$

**5.3. Permutation equivalence.** We now define *permutation equivalence* on reductions, by the equations in Figures 3 and 4, in Appendix A. The *congruence* rules in Figure 3 are bureaucratic: they just say that permutation equivalence is a congruence. The *category* rules make reductions of a given type  $\Gamma \vdash A$  into a category. In Figure 4, the *beta and eta* rules mirror the term-level beta and eta rules. Finally, the *lifting* rules lift composition of reductions towards toplevel.

So,  $\mathcal{L}(X)$  has sorts  $X_0$ , operations  $X_1$ , and as reduction rules in  $\mathcal{L}(X)(G \vdash M, N : A)$  all reductions  $G \vdash P : M \rightarrow N : A$ , modulo the equations.

This easily extends to:

**Proposition 7.**  $\mathcal{L}$  is a functor  $\text{Sig} \rightarrow \text{Sig}$ .

Now, consider  $\mathcal{L}\mathcal{L}(X)$ . We define a mapping  $\mu_X : \mathcal{L}\mathcal{L}(X) \rightarrow \mathcal{L}(X)$ , by induction on reductions. The typing rule for reduction rules specialises to:

$$\frac{(R \in \mathcal{L}(X)(G \vdash M, N : A)) \quad \Gamma \vdash P_1 : M_1 \rightarrow N_1 : G_1 \quad \dots \quad \Gamma \vdash P_n : M_n \rightarrow N_n : G_n}{\Gamma \vdash R(P_1, \dots, P_n) : M[M_1, \dots, M_n] \rightarrow N[N_1, \dots, N_n] : A}$$

We set  $\mu(R(P_1, \dots, P_n)) = R[\mu(P_1), \dots, \mu(P_n)]$ . The other cases just propagate the substitution:

$$\begin{aligned}
P ; Q &\mapsto \mu(P) ; \mu(Q) \\
x &\mapsto x \\
() &\mapsto () \\
c(P_1, \dots, P_n) &\mapsto c(\mu(P_1), \dots, \mu(P_n)) \\
\lambda x : A. P &\mapsto \lambda x : A. \mu(P) \\
PQ &\mapsto \mu(P)\mu(Q) \\
(P, Q) &\mapsto (\mu(P), \mu(Q)) \\
\pi P &\mapsto \pi(\mu(P)) \\
\pi' P &\mapsto \pi'(\mu(P)).
\end{aligned}$$

**Lemma 1.** *This defines a natural transformation  $\mu : \mathcal{L}^2 \rightarrow \mathcal{L}$ , which makes the diagram*

$$\begin{array}{ccc}
\mathcal{L}^3 & \xrightarrow{\mathcal{L}\mu} & \mathcal{L}^2 \\
\mu\mathcal{L} \downarrow & & \downarrow \mu \\
\mathcal{L}^2 & \xrightarrow{\mu} & \mathcal{L}
\end{array}$$

*commute.*

Similarly, there is a natural transformation  $\eta : id \rightarrow \mathcal{L}$ , sending each  $r \in X(G \vdash M, N : A)$  to the reduction  $G \vdash r(x_1, \dots, x_n) : M \rightarrow N : A$ , and we have:

**Lemma 2.** *The diagram*

$$\begin{array}{ccccc}
& & \eta\mathcal{L} & & \\
& & \rightarrow & & \\
\mathcal{L} & & \mathcal{L}^2 & \xleftarrow{\mathcal{L}\eta} & \mathcal{L} \\
& \searrow & \downarrow \mu & \swarrow & \\
& & \mathcal{L} & & 
\end{array}$$

*commutes.*

**Corollary 1.**  *$(\mathcal{L}, \mu, \eta)$  is a monad on  $\text{Sig}$ .*

A crucial result is:

**Proposition 8.** *For all  $\Gamma \vdash Q : N \rightarrow N' : \Delta$  and  $\Delta \vdash P : M \rightarrow M' : A$ , we have:*

$$\Gamma \vdash P[Q] \equiv (M[Q] ;_{M[N']} P[N']) : M[N] \rightarrow M'[N'] : A.$$

*Proof.* We proceed by induction on  $P$ . Most cases are bureaucratic. Consider for instance  $P = c(P_1, \dots, P_p)$ . Then, by definition:

$$P[Q] = (c(P_1[N], \dots, P_p[N]) ;_{c(M'_1[N], \dots, M'_p[N])} c(M'_1[Q], \dots, M'_p[Q])).$$

By the third lifting rule, this is  $\equiv$ -related to

$$c(P_1[N] ;_{M'_1[N]} M'_1[Q], \dots, P_p[N] ;_{M'_p[N]} M'_p[Q]).$$

By  $p$  applications of the induction hypothesis, we obtain

$$c(M_1[Q] ;_{M_1[N']} P_1[N'], \dots, M_p[Q] ;_{M_p[N']} P_p[N']),$$

which by lifting again yields the desired result:

$$c(M_1[Q], \dots, M_p[Q]) ;_{c(M_1[N'], \dots, M_p[N'])} c(P_1[N'], \dots, P_p[N']).$$

The case where something actually happens is  $P = r(P_1, \dots, P_p)$ , with  $r \in X(G \vdash M_0, M'_0 : A)$  and each  $\Delta \vdash P_i : M_i \rightarrow M'_i : G_i$ . Then, the left-hand side is

$$r(P_1[N], \dots, P_p[N]) ;_{M_0[M_1, \dots, M_p][N]} M'_0[M'_1, \dots, M'_p][Q].$$

By lifting, omitting indices of vertical compositions, we have

$$r(P_1[N], \dots, P_p[N]) \equiv r(M_1[N], \dots, M_p[N]); M'_0[P_1[N], \dots, P_p[N]].$$

Observing that  $M'_0[M'_1, \dots, M'_p][Q] = M'_0[M'_1[Q], \dots, M'_p[Q]]$ , the whole is  $\equiv$ -related to

$$\begin{aligned} & r(M_1[N], \dots, M_p[N]); \\ & M'_0[P_1[N], \dots, P_p[N]]; \\ & M'_0[M'_1[Q], \dots, M'_p[Q]], \end{aligned}$$

i.e., by lifting (inductively):

$$\begin{aligned} & r(M_1[N], \dots, M_p[N]); \\ & M'_0[(P_1[N]; M'_1[Q]), \dots, (P_p[N]; M'_1[Q])]. \end{aligned}$$

This is by induction hypothesis  $\equiv$ -related to

$$\begin{aligned} & r(M_1[N], \dots, M_p[N]); \\ & M'_0[(M_1[Q]; P_1[N']), \dots, (M_1[Q]; P_p[N'])], \end{aligned}$$

i.e., by lifting again to

$$\begin{aligned} & r(M_1[N], \dots, M_p[N]); \\ & M'_0[M_1[Q], \dots, M_1[Q]]; \\ & M'_0[P_1[N'], \dots, P_p[N']]. \end{aligned}$$

The second lifting rule then yields

$$\begin{aligned} & r(M_1[Q], \dots, M_p[Q]); \\ & M'_0[P_1[N'], \dots, P_p[N']]. \end{aligned}$$

And then the first lifting rule yields

$$\begin{aligned} & M_0[M_1[Q], \dots, M_1[Q]]; \\ & r(M_1[N'], \dots, M_p[N']); \\ & M'_0[P_1[N'], \dots, P_p[N']], \end{aligned}$$

so, by the second lifting rule again:

$$\begin{aligned} & M_0[M_1[Q], \dots, M_1[Q]]; \\ & r(P_1[N'], \dots, P_p[N']), \end{aligned}$$

i.e., the right-hand side. □

## 6. CARTESIAN CLOSED 2-CATEGORIES

**6.1. Definition.** In a 2-category  $\mathcal{C}$ , a diagram  $A \xleftarrow{p} C \xrightarrow{q} B$  is a *product diagram* iff for all object  $D$ , the induced functor

$$\mathcal{C}(D, C) \xrightarrow{\Delta} \mathcal{C}(D, C) \times \mathcal{C}(D, C) \xrightarrow{\mathcal{C}(D, p) \times \mathcal{C}(D, q)} \mathcal{C}(D, A) \times \mathcal{C}(D, B)$$

is an isomorphism. Because this family of functors is 2-natural in  $D$ , the inverse functors will also be 2-natural.

Similarly, an object  $1$  of  $\mathcal{C}$  is *terminal* iff for all  $D$  the unique functor

$$\mathcal{C}(D, 1) \xrightarrow{!} 1$$

is an isomorphism (where the right-hand  $1$  is the terminal category).

**Definition 4.** A 2-category with finite products, or *fp 2-category*, is a 2-category  $\mathcal{C}$ , equipped with a terminal object and a 2-functor

$$\mathcal{C} \times \mathcal{C} \xrightarrow{\times} \mathcal{C},$$

plus, for all  $A$  and  $B$ , a product diagram

$$A \xleftarrow{p} A \times B \xrightarrow{q} B.$$

In such an fp 2-category  $\mathcal{C}$ , given objects  $A$  and  $B$ , an *exponential* for them is a pair of an object  $B^A$  and a morphism  $ev: A \times B^A \rightarrow B$ , such that for all  $D$ , the functor

$$\begin{array}{ccc}
 & \mathcal{C}(A, A) \times \mathcal{C}(D, B^A) & \xrightarrow{\times} & \mathcal{C}(A \times D, A \times B^A) \\
 (id_{A^!}, id) \nearrow & & & \searrow \mathcal{C}(A \times D, ev) \\
 \mathcal{C}(D, B^A) & & & \mathcal{C}(A \times D, B)
 \end{array}$$

is an isomorphism. As above, because this family of functors is 2-natural in  $D$ , the inverse functors will also be 2-natural.

**Definition 5.** A cartesian closed 2-category, or *cartesian closed 2-category*, is an fp 2-category, equipped with a choice of exponentials for all pairs of objects. The category  $2\text{CCCat}$  has cartesian closed 2-categories as objects, and strictly structure-preserving functors between them as morphisms.

## 7. MAIN ADJUNCTION

**7.1. Right adjoint.** Given a cartesian closed 2-category  $\mathcal{C}$ , define  $\mathcal{V}(\mathcal{C}) = (\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2)$  as follows. First, let as in Section 2  $(\mathcal{C}_0, \mathcal{C}_1) = \mathcal{V}_1(\mathcal{C})$ , and recall the canonical  $\mathcal{L}_0$  and  $\mathcal{L}_1$ -algebra structures  $h_0$  and  $h_1$ . Let then the reduction rules in  $\mathcal{C}_2(G \vdash M, N : A)$  be the 2-cells in  $\mathcal{C}(h_0(G), h_0(A))(h_1(M), h_1(N))$ , abbreviated to  $\mathcal{C}(G, A)(M, N)$  in the sequel.

This signature  $\mathcal{V}\mathcal{C}$  has a canonical  $\mathcal{L}$ -algebra structure  $h_2: \mathcal{L}(\mathcal{V}\mathcal{C}) \rightarrow \mathcal{V}\mathcal{C}$ , which we define by induction over terms in Figure 2. In the case for  $\lambda$ ,  $\varphi$  denotes the structure isomorphism  $\mathcal{C}((\prod \Gamma) \times A, B) \cong \mathcal{C}(\prod \Gamma, B^A)$ .

In order for the definition to make sense as a morphism  $\mathcal{L}(\mathcal{V}\mathcal{C}) \rightarrow \mathcal{V}\mathcal{C}$ , we have to check its compatibility with the equations. We have first:

**Lemma 3.** For all  $\Delta \vdash Q : N \rightarrow N' : \Gamma$  and  $\Gamma \vdash P : M \rightarrow M' : A$  in  $\mathcal{L}(\mathcal{V}\mathcal{C})$ ,

$$\begin{array}{ccc}
 \Delta & \begin{array}{c} \xrightarrow{M[N]} \\ \Downarrow h_2(P[Q]) \\ \xrightarrow{M'[N']} \end{array} & A \\
 & = & \Delta \begin{array}{c} \xrightarrow{N} \\ \Downarrow h_2(Q) \\ \xrightarrow{N'} \end{array} \Gamma \begin{array}{c} \xrightarrow{M} \\ \Downarrow h_2(P) \\ \xrightarrow{M'} \end{array} A.
 \end{array}$$

*Proof.* By induction on  $P$  and the axioms for cartesian closed 2-categories.  $\square$

**Lemma 4.** Any two equated reductions are mapped to the same 2-cell in  $\mathcal{C}$ .

*Proof.* We proceed by induction on the proof of the considered equation. The rules of Figure 3 hold because, in  $\mathcal{C}$ , vertical composition is associative and unital, and equality is a congruence. The beta rule is less easy, so we spell it out.

The left-hand reduction is interpreted in  $\mathcal{C}$  as

$$\begin{array}{ccc}
 & (\varphi M, N) & \\
 & \curvearrowright & \\
 \prod \Gamma & \Downarrow (\varphi P, Q) & B^A \times A \xrightarrow{ev} B \\
 & \curvearrowleft & \\
 & (\varphi M', N') & 
 \end{array}$$

which is equal to

$$\begin{aligned}
& (G \vdash x_i : x_i \rightarrow x_i : G_i) \mapsto (id_{\pi_i} : \pi_i \rightarrow \pi_i : \prod G \rightarrow G_i) \\
& (G \vdash () : () \rightarrow () : 1) \mapsto (id_! : ! \rightarrow ! : \prod G \rightarrow 1) \\
& (\Gamma \vdash c(P_1, \dots, P_n) : c(M_1, \dots, M_n) \rightarrow c(N_1, \dots, N_n) : A) \mapsto \\
& \quad (M_1, \dots, M_n) \\
& \quad \prod \Gamma \begin{array}{c} \curvearrowright \\ \Downarrow P \\ \curvearrowleft \end{array} \prod G \xrightarrow{c} A \quad (c \in \mathcal{C}_1(G, A), P = (P_1, \dots, P_n)) \\
& \quad (N_1, \dots, N_n) \\
& (\Gamma \vdash r(P_1, \dots, P_n) : M[M_1, \dots, M_n] \rightarrow N[N_1, \dots, N_n] : A) \mapsto \\
& \quad (M_1, \dots, M_n) \\
& \quad \prod \Gamma \begin{array}{c} \curvearrowright \\ \Downarrow P \\ \curvearrowleft \end{array} \prod G \begin{array}{c} \xrightarrow{M} \\ \Downarrow r \\ \xrightarrow{N} \end{array} A \quad (P = (P_1, \dots, P_n)) \\
& \quad (N_1, \dots, N_n) \\
& (G \vdash P ;_{M_2} Q : M_1 \rightarrow M_3 : A) \mapsto \prod G \begin{array}{c} \xrightarrow{M_1} \\ \Downarrow P \\ \Downarrow Q \\ \xrightarrow{M_3} \end{array} A \\
& (\Gamma \vdash \lambda x : A. P : \lambda x : A. M \rightarrow \lambda x : A. N : B^A) \mapsto \varphi(P : M \rightarrow N : (\prod \Gamma) \times A \rightarrow B) \\
& \quad (M, N) \\
& (\Gamma \vdash PQ : MN \rightarrow M'N' : B) \mapsto \prod \Gamma \begin{array}{c} \curvearrowright \\ \Downarrow (P, Q) \\ \curvearrowleft \end{array} B^A \times A \xrightarrow{ev} B \\
& \quad (M', N') \\
& (\Gamma \vdash (P, Q) : (M, N) \rightarrow (M', N') : A \times B) \mapsto \prod \Gamma \begin{array}{c} \curvearrowright \\ \Downarrow (P, Q) \\ \curvearrowleft \end{array} A \times B \\
& \quad (M', N') \\
& (\Gamma \vdash \pi_{A,B} P : \pi_{A,B} M \rightarrow \pi_{A,B} N : A) \mapsto \prod \Gamma \begin{array}{c} \xrightarrow{M} \\ \Downarrow P \\ \xrightarrow{N} \end{array} A \times B \xrightarrow{\pi} A \\
& (\Gamma \vdash \pi'_{A,B} P : \pi'_{A,B} M \rightarrow \pi'_{A,B} N : B) \mapsto \prod \Gamma \begin{array}{c} \xrightarrow{M} \\ \Downarrow P \\ \xrightarrow{N} \end{array} A \times B \xrightarrow{\pi'} B
\end{aligned}$$

FIGURE 2. The  $\mathcal{L}$ -algebra structure on  $\mathcal{V}(\mathcal{C})$

$$\begin{array}{ccccc}
& & (id, N) & & \varphi M \times A \\
& \curvearrowright & \rightarrow & \curvearrowright & \\
\Pi \Gamma & & \Downarrow (id, Q) & & \Downarrow \varphi P \times A \\
& \curvearrowleft & \rightarrow & \curvearrowleft & \\
& & (id, N') & & \varphi M' \times A
\end{array}
\quad
\begin{array}{c}
B^A \times A \xrightarrow{ev} B
\end{array}$$

which is in turn equal (by cartesian closedness of  $\mathcal{C}$ ) to:

$$\begin{array}{ccccc}
& & (id, N) & & M \\
& \curvearrowright & \rightarrow & \curvearrowright & \\
\Pi \Gamma & & \Downarrow (id, Q) & & \Downarrow P \\
& \curvearrowleft & \rightarrow & \curvearrowleft & \\
& & (id, N') & & M'
\end{array}
\quad
\begin{array}{c}
B
\end{array}$$

and hence to the right-hand side of the equation by Lemma 3. The other beta and eta rules similarly hold by the properties of products, internal homs, and terminal object in  $\mathcal{C}$ .

The lifting rules hold by (particular cases of) the interchange law in  $\mathcal{C}$  and functoriality of the structural isomorphisms

$$\mathcal{C}(A \times B, C) \cong \mathcal{C}(B, C^A) \quad \text{and} \quad \mathcal{C}(C, A \times B) \cong \mathcal{C}(C, A) \times \mathcal{C}(C, B),$$

which concludes the proof.  $\square$

This assignment extends to cartesian closed functors and we have:

**Proposition 9.**  $\mathcal{V}$  is a functor  $2\text{CCCat} \rightarrow \text{Sig}$ .

**7.2. Left adjoint.** Given an  $\mathcal{L}$ -algebra  $h: \mathcal{L}(X) \rightarrow X$ , we now construct a cartesian closed 2-category  $\mathcal{F}(X, h)$ . It has:

- objects the types in  $\mathcal{L}_0(X_0)$ ;
- 1-cells  $A \rightarrow B$  the terms in  $\mathcal{L}_1(X_0, X_1)(A, B)$ ;
- 2-cells  $M \rightarrow N: A \rightarrow B$  the reduction rules in  $X_2(M, N)$ .

We then must define the cartesian closed 2-category structure, and we start with the 2-category structure. Composition of 1-cells  $A \xrightarrow{M} B \xrightarrow{N} C$  is defined to be  $A \xrightarrow{N[M]} C$ . Vertical composition of 2-cells

$$\begin{array}{ccc}
& M_1 & \\
& \curvearrowright & \\
A & \xrightarrow{M_2} & B \\
& \curvearrowleft & \\
& M_3 & \\
& \Downarrow \alpha & \\
& \Downarrow \beta & \\
& \curvearrowright & \\
& M_3 & \\
& \curvearrowleft & \\
& M_1 & 
\end{array}$$

is given by  $h(\eta(\alpha);_{M_2} \eta(\beta))$ .

Horizontal composition of 2-cells

$$(7.1) \quad \begin{array}{ccccc}
& & M & & N \\
& \curvearrowright & \rightarrow & \curvearrowright & \\
A & & \Downarrow \alpha & & \Downarrow \beta \\
& \curvearrowleft & \rightarrow & \curvearrowleft & \\
& & M' & & N'
\end{array}
\quad
\begin{array}{c}
C
\end{array}$$

is obtained as  $h(\beta(\eta(\alpha)))$ .

To show that this yields a 2-category structure, the only non obvious point is the interchange law. We deal with it using the following series of results. First, consider the *left whiskering*

$$\begin{array}{ccccc}
& & M & & \\
& \swarrow & \downarrow \alpha & \searrow & \\
A & \xrightarrow{\quad} & B & \xrightarrow{N} & C \\
& \searrow & \downarrow \alpha & \swarrow & \\
& & M' & & 
\end{array}$$

of a 2-cell  $\alpha$  by a 1-cell  $N$ , i.e., the composition  $id_N \circ \alpha = h((h(N))(\eta(\alpha)))$ .

**Lemma 5.** *We have:  $h((h(N))(\eta(\alpha))) = h(N[\eta(\alpha)])$ .*

*Proof.* Indeed, consider the term  $N(\eta(\eta(\alpha)))$  in  $\mathcal{L}(\mathcal{L}(X))$ . Its images by  $h \circ \mathcal{L}(h)$  and  $h \circ \mu$  coincide, and are respectively  $h((h(N))(\eta(\alpha)))$ , i.e.,  $id_N \circ \alpha$ , and  $h(N[\eta(\alpha)])$ .  $\square$

Similarly, consider the *right whiskering*

$$\begin{array}{ccccc}
& & & N & \\
& & & \downarrow \gamma & \\
A & \xrightarrow{M} & B & \xrightarrow{\quad} & C \\
& & & \downarrow \gamma & \\
& & & N' & 
\end{array}$$

of a 2-cell  $\gamma$  by a 1-cell  $M$ , i.e., the composition  $\gamma \circ id_N = h(\gamma(\eta(h(M))))$ .

**Lemma 6.** *We have:  $h(\gamma(\eta(h(M)))) = h(\gamma(M))$ .*

*Proof.* Consider  $(\eta\gamma)(\eta M)$  in  $\mathcal{L}(\mathcal{L}(X))$ . Its images by  $h \circ \mathcal{L}(h)$  and  $h \circ \mu$  coincide, and are respectively  $h(\gamma(\eta(h(M))))$  and  $h(\gamma(M))$ .  $\square$

Now, we prove that the two sensible ways of mimicking horizontal composition using whiskering coincide with actual horizontal composition:

**Lemma 7.** *For any cells as in (7.1),*

$$(\beta \circ id_M) ; (id_{N'} \circ \alpha) = \beta \circ \alpha = (id_N \circ \alpha) ; (\beta \circ id_{M'}).$$

*Proof.* Consider first the reduction  $\eta(\beta(M)) ; \eta(N'[\eta(\alpha)])$  in  $\mathcal{L}(\mathcal{L}(X))$ . Taking  $h \circ \mathcal{L}(h)$  and  $h \circ \mu$  as above respectively yields

- $h(\eta(h(\beta(M)))) ; \eta(h(N'[\eta(\alpha)]))$ , and
- $h(\beta(M) ; N'[\eta(\alpha)]) = h(\beta(\eta(\alpha)))$ ,

hence the left-hand equality. Then consider  $\eta(N[\eta(\alpha)]) ; \eta(\gamma(M'))$ . Evaluating as before yields the right-hand equality.  $\square$

Finally, consider any configuration like:

$$\begin{array}{ccccc}
& & M & & \\
& \swarrow & \downarrow \alpha & \searrow & \\
A & \xrightarrow{M'} & B & \xrightarrow{N} & C \\
& \searrow & \downarrow \beta & \swarrow & \\
& & M'' & & 
\end{array}$$

**Lemma 8.** *We have  $(id_N \circ \alpha) ; (id_N \circ \beta) = id_N \circ (\alpha ; \beta)$ .*

*Proof.* Consider  $\eta(N[\eta(\alpha)]) ; \eta(N[\eta(\beta)])$ . Evaluating yields equality of

- $h(\eta(h(N[\eta(\alpha)]))) ; \eta(h(N[\eta(\beta)]))$ , i.e., the left-hand side, and
- $h(N[\eta(\alpha)] ; N[\eta(\beta)])$ , i.e.,  $h(N[\eta(\alpha) ; \eta(\beta)])$  by lifting.

But now consider  $N[\eta(\alpha) ; \eta(\beta)]$ . Evaluating yields equality of

- $h(N[\eta(\alpha) ; \eta(\beta)])$ , as above, and
- $h(N[\eta(h(\eta(\alpha) ; \eta(\beta)))])$ , i.e.,  $h(N[\eta(\alpha) ; \beta])$  (where  $\alpha ; \beta$  denotes vertical composition in our candidate 2-category), i.e., the right-hand side.  $\square$

**Lemma 9.** *The interchange law holds, i.e., for all reduction rules as in*

$$\begin{array}{ccc}
& M_1 & N_1 \\
A \begin{array}{c} \curvearrowright \\ \xrightarrow{M_2} \\ \downarrow \alpha \\ \xrightarrow{M_3} \\ \curvearrowleft \end{array} & B & \begin{array}{c} \curvearrowright \\ \xrightarrow{N_2} \\ \downarrow \gamma \\ \xrightarrow{N_3} \\ \curvearrowleft \end{array} C, \\
& M_3 & N_3 \\
& \downarrow \beta & \downarrow \theta
\end{array}$$

we have

$$(\gamma; \theta) \circ (\alpha; \beta) = (\gamma \circ \alpha); (\theta \circ \beta).$$

*Proof.* By the previous results, we have

$$\begin{aligned}
& (\gamma; \theta) \circ (\alpha; \beta) \\
&= ((\gamma; \theta) \circ M_1); (N_3 \circ (\alpha; \beta)) \\
&= (\gamma \circ M_1); (\theta \circ M_1); (N_3 \circ \alpha); (N_3 \circ \beta) \\
&= (\gamma \circ M_1); (N_2 \circ \alpha); (\theta \circ M_2); (N_3 \circ \beta) \\
&= (\gamma \circ \alpha); (\theta \circ \beta).
\end{aligned}$$

□

Now, let us show cartesian closedness. We have a bijection of hom-sets  $\mathcal{L}_1(X)(C \vdash A \times B) \cong \mathcal{L}_1(X)(C \vdash A) \times \mathcal{L}_1(X)(C \vdash B)$ , given by

$$\begin{aligned}
\mathcal{L}_1(X)(C \vdash A \times B) &\rightarrow \mathcal{L}_1(X)(C \vdash A) \times \mathcal{L}_1(X)(C \vdash B) \\
M &\mapsto \pi M, \pi' M
\end{aligned}$$

and

$$\begin{aligned}
\mathcal{L}_1(X)(C \vdash A) \times \mathcal{L}_1(X)(C \vdash B) &\rightarrow \mathcal{L}_1(X)(C \vdash A \times B) \\
M, N &\mapsto (M, N).
\end{aligned}$$

These are mutually inverse thanks to the beta and eta rules for products in the simply-typed  $\lambda$ -calculus.

On 2-hom-sets, we have

$$\begin{aligned}
\mathcal{L}(X)(C \vdash M, N: A \times B) &\rightarrow \mathcal{L}(X)(C \vdash \pi M, \pi N: A) \times \mathcal{L}(X)(C \vdash \pi' M, \pi' N: B) \\
P &\mapsto \pi P, \pi' P
\end{aligned}$$

and (omitting  $C$ )

$$\begin{aligned}
\mathcal{L}(X)(M_1, N_1: A) \times \mathcal{L}(X)(M_2, N_2: B) &\rightarrow \mathcal{L}(X)((M_1, M_2), (N_1, N_2): A \times B) \\
P_1, P_2 &\mapsto (P_1, P_2),
\end{aligned}$$

which are mutually inverse thanks to the beta and eta rules for products in Figure 4. We use these to define the desired isomorphism  $(u, v)$

$$X_2(C \vdash M, N: A \times B) \cong X_2(C \vdash \pi M, \pi N: A) \times X_2(C \vdash \pi' M, \pi' N: B),$$

as in the diagrams

$$\begin{array}{ccc}
X_2(M, N) & \xrightarrow{u} & X_2(\pi M, \pi N) \times X_2(\pi' M, \pi' N) \\
\eta \downarrow & & \uparrow h \times h \\
\mathcal{L}(X)(M, N) & \xrightarrow{\cong} & \mathcal{L}(X)(\pi M, \pi N) \times \mathcal{L}(X)(\pi' M, \pi' N)
\end{array}$$

and

$$\begin{array}{ccc}
X_2(\pi M, \pi N) \times X_2(\pi' M, \pi' N) & \xrightarrow{v} & X_2(M, N) \\
\eta \times \eta \downarrow & & \uparrow h \\
\mathcal{L}(X)(\pi M, \pi N) \times \mathcal{L}(X)(\pi' M, \pi' N) & \xrightarrow{\cong} & \mathcal{L}(X)(M, N).
\end{array}$$



Starting from  $r \in X_2(M, N)$ , we obtain

$$v(u(r)) = h(\eta(h(\pi(\eta(r))))), \eta(h(\pi'(\eta(r))))).$$

But consider  $(\eta(\pi\eta(r)), \eta(\pi'\eta(r)))$  in  $\mathcal{L}(\mathcal{L}X)$ ; its images by  $h \circ \mathcal{L}h$  and  $h \circ \mu$  are respectively:

- $h(\eta(h(\pi(\eta r))), \eta(h(\pi'(\eta r))))$ , and
- $h(\pi\eta(r), \pi'\eta(r))$ , i.e.,  $h(\eta(r))$ , i.e.,  $r$ ,

which must be equal because  $h$  is an  $\mathcal{L}$ -algebra, hence  $v \circ u = id$ .

Conversely, starting from  $(r, s) \in X_2(M_1, M_2) \times X_2(N_1, N_2)$ , we obtain the pair with components

$$h(\pi(\eta(h(\eta(r), \eta(s)))))) \quad \text{and} \quad h(\pi'(\eta(h(\eta(r), \eta(s))))).$$

Considering  $\pi(\eta(\eta(r), \eta(s))) \in \mathcal{L}(\mathcal{L}(X))$ , its images by  $h \circ \mathcal{L}(h)$  and  $h \circ \mu$  are respectively:

- $h(\pi(\eta(h(\eta(r), \eta(s))))))$ , and
- $h(\pi(\eta(r), \eta(s))) = h(\eta(r)) = r$ .

As above, they must be equal, and by symmetry the second component is  $s$ , and we have proved  $u \circ v = id$ . Similar reasoning for the terminal object and internal homs leads to:

**Proposition 10.** *This yields a cartesian closed 2-category structure on  $\mathcal{C}$ .*

This extends to morphisms of  $\mathcal{L}$ -algebras, so we have constructed a functor  $\mathcal{F}: \mathcal{L}\text{-Alg} \rightarrow 2\text{CCCat}$ .

**7.3. Adjunction.** Consider any  $\mathcal{L}$ -algebra  $(X, h)$ . What does  $(Y, k) = \mathcal{V}(\mathcal{F}(X, h))$  look like? Sorts in  $Y_0$  are types in  $\mathcal{L}_0(X_0)$ . Operations  $Y_1(G \vdash A)$  are terms in  $\mathcal{L}_1(X_0, X_1)(\prod G \vdash A)$ . Reduction rules in  $Y_2(G \vdash M, N: B)$  are reductions in  $\mathcal{L}(X)(\prod G \vdash M', N': B)$ , where  $M' = M[\pi_1 x/x_1, \dots, \pi_n x/x_n]$  (and similarly for  $N'$ ).

Let  $\eta_X$  send:

- each sort  $\iota \in X_0$  to the type  $\iota \in \mathcal{L}_0(X_0)$ ,
- each operation  $c \in X(G \vdash A)$  to the term  $c(\pi_1 x, \dots, \pi_n x)$ , and
- each reduction rule  $r \in X_2(G \vdash M, N: A)$  to the reduction  $x: \prod G \vdash r(\pi_1 x, \dots, \pi_n x): M' \rightarrow N': A$ .

**Theorem 2.** *This  $\eta$  is a natural transformation which is the unit of an adjunction*

$$\mathcal{L}\text{-Alg} \begin{array}{c} \xrightarrow{\mathcal{F}} \\ \perp \\ \xleftarrow{\mathcal{V}} \end{array} 2\text{CCCat}.$$

*Proof.* Consider any morphism  $f: (X, h) \rightarrow \mathcal{V}(\mathcal{C})$ , and let  $(Y, k) = \mathcal{V}(\mathcal{F}(X, h))$  and  $\mathcal{V}(\mathcal{C}) = (\mathcal{C}_0, \mathcal{C}_1, h_2: \mathcal{C}_2 \rightarrow \mathcal{C}_1)$ . We now define a uniquely determined cartesian closed functor  $f': \mathcal{F}(X, h) \rightarrow \mathcal{C}$  making the triangle

$$\begin{array}{ccc} X & \xrightarrow{\eta_X} & \mathcal{V}(\mathcal{F}(X)) \\ & \searrow f & \downarrow \mathcal{V}(f') \\ & & \mathcal{V}(\mathcal{C}) \end{array}$$

commute.

On objects, it is determined by induction: on sorts by  $f_0$ , and on type constructors by the requirement that  $f'$  be cartesian closed. On morphisms, it is similarly determined by  $f_1$  and  $f'$  being cartesian closed. On 2-cells, define  $f'$  to be  $f_2: X_2(A \vdash M, N: B) \rightarrow \mathcal{C}(f'(A), f'(B))(f'(M), f'(N))$ , which is also the only possible choice from  $f$ .

We thus only have to show that  $f'$  is cartesian closed, which follows by  $f$  being a morphism of  $\mathcal{L}$ -algebras. For example, to show that binary products of reductions are preserved, consider  $r \in X_2(C \vdash M_1, M_2: A)$  and  $s \in X_2(C \vdash N_1, N_2: B)$ . Their product in  $\mathcal{F}(X)$  is obtained by considering the atomic reductions  $x: C \vdash r(x): M_1 \rightarrow M_2: A$  and  $x: C \vdash s(x): N_1 \rightarrow N_2: B$  and taking  $h(r(x), s(x))$ , which is sent by  $f_2$  to  $f_2(h(r(x), s(x)))$ . But, because  $f$  is a morphism of  $\mathcal{L}$ -algebras, this is the same as  $h_2((f_2(r))(x), (f_2(s))(y))$ , which is by definition (i.e., Figure 2) the product  $(f_2(r), f_2(s))$  in  $\mathcal{C}$ .  $\square$

## REFERENCES

- [1] H. J. Sander Bruggink. *Equivalence of reductions in higher-order rewriting*. PhD thesis, Utrecht University, 2008.
- [2] Paolo Capriotti. Concurrent semantics with variable binding. Master’s thesis, University of Pisa, 2009.
- [3] Andrea Corradini, Fabio Gadducci, and Ugo Montanari. Relating two categorial models of term rewriting. In Jieh Hsiang, editor, *RTA*, volume 914 of *Lecture Notes in Computer Science*, pages 225–240. Springer, 1995.
- [4] Thierry Despeyroux and André Hirschowitz. Principles for functional abstract syntax. Draft, 1995.
- [5] Uffe Engberg and Mogens Nielsen. A calculus of communicating systems with label passing. Technical Report PB-208, Aarhus University, 1986.
- [6] Marcelo P. Fiore. Second-order and dependently-sorted abstract syntax. In *LICS*, pages 57–68. IEEE Computer Society, 2008.
- [7] Barney P. Hilken. Towards a proof theory of rewriting: The simply typed  $2\lambda$ -calculus. *Theor. Comput. Sci.*, 170(1-2):407–444, 1996.
- [8] André Hirschowitz and Marco Maggesi. Modules over monads and linearity. In *WoLLIC ’07*, volume 4576 of *LNCS*. Springer, 2007.
- [9] Yoshiki Kinoshita, John Power, and Makoto Takeyama. Sketches. *Journal of Pure and Applied Algebra*, 143(1-3), 1999.
- [10] Jan W. Klop. *Combinatory Reduction Systems*. PhD thesis, CWI, Amsterdam, 1980.
- [11] Joachim Lambek. From  $\lambda$ -calculus to cartesian closed categories. Academic Press, 1980.
- [12] Robin Milner. *A Calculus of Communicating Systems*, volume 92 of *LNCS*. Springer, 1980.
- [13] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, I/II. *Information and Computation*, 100(1):1–77, 1992.
- [14] Tobias Nipkow. Higher-order critical pairs. In *LICS*, pages 342–349. IEEE Computer Society, 1991.
- [15] Vincent van Oostrom and Femke van Raamsdonk. Comparing combinatory reduction systems and higher-order rewrite systems. In Jan Heering, Karl Meinke, Bernhard Möller, and Tobias Nipkow, editors, *HOA*, volume 816 of *Lecture Notes in Computer Science*, pages 276–304. Springer, 1993.
- [16] Charles Wells. A generalization of the concept of sketch. *Theor. Comput. Sci.*, 70(1):159–178, 1990.
- [17] David A. Wolfram. *The Clausal Theory of Types*. Number 21 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1993.
- [18] Julianna Zsidó. *Typed Abstract Syntax*. PhD thesis, Université de Nice-Sophia Antipolis, 2010.

## APPENDIX A. EQUATIONS ON REDUCTIONS

*Current address:* CNRS, Université de Savoie

Congruence	
$\frac{\Gamma \vdash P : M \rightarrow N : A}{\Gamma \vdash P \equiv P : M \rightarrow N : A}$	$\frac{\Gamma \vdash P \equiv Q : M \rightarrow N : A}{\Gamma \vdash Q \equiv P : M \rightarrow N : A}$
$\frac{\Gamma \vdash P_1 \equiv P_2 : M \rightarrow N : A \quad \Gamma \vdash P_2 \equiv P_3 : M \rightarrow N : A}{\Gamma \vdash P_1 \equiv P_3 : M \rightarrow N : A}$	
$\frac{\Gamma \vdash P \equiv P' : M_1 \rightarrow M_2 : A \quad \Gamma \vdash Q \equiv Q' : M_2 \rightarrow M_3 : A}{\Gamma \vdash (P ;_{M_2} Q) \equiv (P' ;_{M_2} Q') : M_1 \rightarrow M_3 : A}$	
$(r \in X(G \vdash M, N : A))$	
$\frac{\Gamma \vdash P_1 \equiv Q_1 : M_1 \rightarrow N_1 : G_1 \quad \dots \quad \Gamma \vdash P_n \equiv Q_n : M_n \rightarrow N_n : G_n}{\Gamma \vdash r(P_1, \dots, P_n) \equiv r(Q_1, \dots, Q_n) : M[M_1, \dots, M_n] \rightarrow N[N_1, \dots, N_n] : A}$	
$(c \in X(G \vdash A))$	
$\frac{\Gamma \vdash P_1 \equiv Q_1 : M_1 \rightarrow N_1 : G_1 \quad \dots \quad \Gamma \vdash P_n \equiv Q_n : M_n \rightarrow N_n : G_n}{\Gamma \vdash c(P_1, \dots, P_n) \equiv c(Q_1, \dots, Q_n) : c(M_1, \dots, M_n) \rightarrow c(N_1, \dots, N_n) : A}$	
$\frac{\Gamma, x : A \vdash P \equiv Q : M \rightarrow N : B}{\Gamma \vdash (\lambda x : A. P) \equiv (\lambda x : A. Q) : \lambda x : A. M \rightarrow \lambda x : A. N : B^A}$	
$\frac{\Gamma \vdash P \equiv P' : M \rightarrow M' : B^A \quad \Gamma \vdash Q \equiv Q' : N \rightarrow N' : A}{\Gamma \vdash (PQ) \equiv (P'Q') : MN \rightarrow M'N' : B}$	
$\frac{\Gamma \vdash P \equiv P' : M \rightarrow M' : A \quad \Gamma \vdash Q \equiv Q' : N \rightarrow N' : B}{\Gamma \vdash (P, Q) \equiv (P', Q') : (M, N) \rightarrow (M', N') : A \times B}$	
$\frac{\Gamma \vdash P \equiv Q : M \rightarrow N : A \times B}{\Gamma \vdash (\pi_{A,B} P) \equiv (\pi_{A,B} Q) : \pi_{A,B} M \rightarrow \pi_{A,B} N : A}$	
$\frac{\Gamma \vdash P \equiv Q : M \rightarrow N : A \times B}{\Gamma \vdash (\pi'_{A,B} P) \equiv (\pi'_{A,B} Q) : \pi'_{A,B} M \rightarrow \pi'_{A,B} N : A}$	
Category	
$\frac{\Gamma \vdash P_1 : M_1 \rightarrow M_2 : A \quad \Gamma \vdash P_2 : M_2 \rightarrow M_3 : A \quad \Gamma \vdash P_3 : M_3 \rightarrow M_4 : A}{\Gamma \vdash (P_1 ;_{M_2} (P_2 ;_{M_3} P_3)) \equiv ((P_1 ;_{M_2} P_2) ;_{M_3} P_3) : M_1 \rightarrow M_4 : A}$	
$\frac{\Gamma \vdash P : M \rightarrow N : A}{\Gamma \vdash (P ;_N N) \equiv P : M \rightarrow N : A}$	$\frac{\Gamma \vdash P : M \rightarrow N : A}{\Gamma \vdash (M ;_M P) \equiv P : M \rightarrow N : A}$

FIGURE 3. Equations on reductions (Congruence and category)

Beta and eta	
$\frac{\Gamma, x : A \vdash P : M \rightarrow M' : B \quad \Gamma \vdash Q : N \rightarrow N' : A}{\Gamma \vdash ((\lambda x : A. P)Q) \equiv P[Q/x] : (\lambda x : A. M)N \rightarrow M'[N'/x] : B}$	
$\frac{\Gamma \vdash P \equiv M \rightarrow N : B^A}{\Gamma \vdash P \equiv \lambda x : A. (Px) : M \rightarrow N : B^A} \quad (x \notin \Gamma)$	
$\frac{\Gamma \vdash P : M_1 \rightarrow M_2 : A \quad \Gamma \vdash Q : N_1 \rightarrow N_2 : B}{\Gamma \vdash \pi(P, Q) \equiv P : \pi(M_1, N_1) \rightarrow M_2 : A}$	
$\frac{\Gamma \vdash P : M_1 \rightarrow M_2 : A \quad \Gamma \vdash Q : N_1 \rightarrow N_2 : B}{\Gamma \vdash \pi'(P, Q) \equiv P : \pi'(M_1, N_1) \rightarrow N_2 : A}$	
$\frac{\Gamma \vdash P : (M_1, N_1) \rightarrow (M_2, N_2) : A \times B}{\Gamma \vdash P \equiv (\pi P, \pi' P) : (M_1, N_1) \rightarrow (M_2, N_2) : A \times B}$	$\frac{\Gamma \vdash P : M \rightarrow N : 1}{\Gamma \vdash P \equiv () : M \rightarrow N : 1}$
Lifting	
$\frac{(r \in X(\Gamma \vdash \langle M_1, M_2 \rangle : A)) \quad \Delta \vdash P : N_1 \rightarrow N_2 : \Gamma \quad \Delta \vdash Q : N_2 \rightarrow N_3 : \Gamma}{\Gamma \vdash r(P ;_{N_2} Q) \equiv M_1[P] ;_{M_1[N_2]} r(Q) : M_1[N_1] \rightarrow M_2[N_3] : A}$	
$\frac{(r \in X(\Gamma \vdash \langle M_1, M_2 \rangle : A)) \quad \Delta \vdash P : N_1 \rightarrow N_2 : \Gamma \quad \Delta \vdash Q : N_2 \rightarrow N_3 : \Gamma}{\Gamma \vdash r(P ;_{N_2} Q) \equiv r(P) ;_{M_2[N_2]} M_2[Q] : M_1[N_1] \rightarrow M_2[N_3] : A}$	
$\frac{\Gamma \vdash P : M_1 \rightarrow M_2 : G \quad \Gamma \vdash Q : M_2 \rightarrow M_3 : G}{\Gamma \vdash (c(P ;_{M_2} Q)) \equiv (c(P) ;_{c(M_2)} c(Q)) : M_1 \rightarrow M_3 : A} \quad (c \in X(G \vdash A))$	
$\frac{\Gamma, x : A \vdash P : M_1 \rightarrow M_2 : B \quad \Gamma, x : A \vdash Q : M_2 \rightarrow M_3 : B}{\Gamma \vdash (\lambda x : A. (P ;_{M_2} Q)) \equiv ((\lambda x : A. P) ;_{\lambda x : A. M_2} (\lambda x : A. Q)) : \lambda x : A. M_1 \rightarrow \lambda x : A. M_3 : B^A}$	
$\frac{\Gamma \vdash P : M_1 \rightarrow M_2 : B^A \quad \Gamma \vdash P' : M_2 \rightarrow M_3 : B^A \quad \Gamma \vdash Q : N_1 \rightarrow N_2 : A \quad \Gamma \vdash Q' : N_2 \rightarrow N_3 : A}{\Gamma \vdash ((P ;_{M_2} P')(Q ;_{N_2} Q')) \equiv ((PQ) ;_{M_2 N_2} (P'Q')) : M_1 N_1 \rightarrow M_3 N_3 : B}$	
$\frac{\Gamma \vdash P : M_1 \rightarrow M_2 : A \quad \Gamma \vdash P' : M_2 \rightarrow M_3 : A \quad \Gamma \vdash Q : N_1 \rightarrow N_2 : B \quad \Gamma \vdash Q' : N_2 \rightarrow N_3 : B}{\Gamma \vdash ((P ;_{M_2} P'), (Q ;_{N_2} Q')) \equiv ((P, Q) ;_{(M_2, N_2)} (P', Q')) : (M_1, N_1) \rightarrow (M_3, N_3) : A \times B}$	
$\frac{\Gamma \vdash P : M_1 \rightarrow M_2 : A \times B \quad \Gamma \vdash Q : M_2 \rightarrow M_3 : A \times B}{\Gamma \vdash (\pi_{A,B}(P ;_{M_2} Q)) \equiv (\pi_{A,B} P ;_{\pi_{A,B} M_2} \pi_{A,B} Q) : M_1 \rightarrow M_3 : A}$	
$\frac{\Gamma \vdash P : M_1 \rightarrow M_2 : A \times B \quad \Gamma \vdash Q : M_2 \rightarrow M_3 : A \times B}{\Gamma \vdash (\pi'_{A,B}(P ;_{M_2} Q)) \equiv (\pi'_{A,B} P ;_{\pi'_{A,B} M_2} \pi'_{A,B} Q) : M_1 \rightarrow M_3 : B}$	

FIGURE 4. Equations on reductions (beta-eta and lifting)