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

TOM HIRSCHOWITZ

CNRS, Université de Savoie
e-mail address: tom.hirschowitz@univ-savoie.fr

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

Cartesian closed categories provide semantics for equational theories with variable binding [12, 4]. On the other hand, 2-categories with finite products provide semantics for term rewriting [3]. The present paper shows that cartesian closed 2-categories provide semantics for term rewriting with variable binding, as embodied by Bruggink’s generalisation [1] of permutation equivalence [16, Chapter 8] to higher-order rewriting [11, 19, 15, 17].

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

\[
\begin{array}{c}
\text{Sig} \\
\downarrow \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \ Quad
Related work. Our cartesian closed 2-signatures may be seen as a 2-dimensional refinement of cartesian closed sketches \[18, 5, 10\]. Bruggink’s calculus of permutation equivalence is close in spirit to Hilken’s 2-categorical semantics of the simply-typed \(\lambda\)-calculus \[8\], 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.

2. Cartesian closed signatures and categories

We start by recalling the well-known adjunction \[12, 4\] between what we here call (cartesian closed) 1-signatures and cartesian closed categories.

For any set \(X\), define types over \(X\) by the grammar:

\[
A, B, \ldots \in L_0(X) := x \mid 1 \mid A \times B \mid B^A,
\]

with \(x \in X\).

**Proposition 2.1.** \(L_0\) defines a monad on \(\text{Set}\).

Let the set of sequents over a set \(X\) be \(S_0(X) = L_0(X)^\star \times L_0(X)\), i.e., sequents are pairs of a list of types and a type. The assignment \(X \mapsto S_0(X)\) extends to an endofunctor on \(\text{Set}\).

**Definition 2.2.** A 1-signature consists of a set \(X_0\) of sorts, and an \(S_0(X_0)\)-indexed set \(X_1\) of operations, or equivalently a map \(X_1 \to S_0(X_0)\).

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

\[
\begin{aligned}
X_1 &\xrightarrow{f_1} Y_1 \\
S_0(X_0) &\xrightarrow{S_0(f_0)} S_0(Y_0)
\end{aligned}
\]

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

**Proposition 2.3.** Composition of morphisms is associative and unital, and hence 1-signatures and their morphisms form a category \(\text{Sig}_1\).

There is a well-known adjunction

\[
\text{Sig}_1 \quad \dashv \quad \text{CCCat}
\]

between 1-signatures and the category \(\text{CCCat}\) of small cartesian closed categories (with chosen structure) and (strict) cartesian closed functors, i.e., functors \(F : C \to D\) preserving binary products, projections, and the terminal object on the nose, and such that, for all objects \(A, B \in C\), currying

\[
F(B^A) \times F(A) = F(B^A \times A) \xrightarrow{F(\text{ev}_{A,B})} F(B)
\]

yields an identity.
The functor $\mathcal{W}_1$ maps any cartesian closed category $\mathcal{C}$ to the signature with sorts $\mathcal{C}_0$, its set of objects, and with operations $A_1, \ldots, A_n \to A$ the set $\mathcal{C}([A_1 \times \ldots \times A_n], [A])$, where $[-]$ denotes the function $\mathcal{L}_0(\mathcal{C}_0) \to \mathcal{C}_0$ defined by induction:

\[
\begin{align*}
\llbracket c \rrbracket &= c & & c \in \mathcal{C}_0 \\
\llbracket 1 \rrbracket &= 1 \\
\llbracket A \times B \rrbracket &= \llbracket A \rrbracket \times \llbracket B \rrbracket \\
\llbracket B^A \rrbracket &= \llbracket B \rrbracket^{\llbracket A \rrbracket}.
\end{align*}
\]

Conversely, given a 1-signature $X$, consider the simply-typed $\lambda$-calculus with base types in $X_0$ and constants in $X_1$. I.e., for any $c \in X_1(G, A)$ and terms $\Gamma \vdash M_i : A_i$, for all $1 \leq i \leq n$, where $G = (A_1, \ldots, A_n)$, we have a term $\Gamma \vdash c(M_1, \ldots, M_n) : A$, representing the application of the constant $c$ to $M_1, \ldots, M_n$. We use special parentheses to avoid ambiguity with term application. Terms modulo $\beta \eta$ form a cartesian closed category $\mathcal{H}_1(X)$ with objects all types over $X_0$ and morphisms $A \to B$ 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 $\mathcal{H}_1 \to \mathcal{W}_1$ decomposes into two adjunctions

\[
\text{Sig}_1 \overset{\mathcal{L}_1}{\longrightarrow} \text{L}_1\text{-Alg} \overset{\mathcal{F}_1}{\longrightarrow} \text{CCCat},
\]

as follows.

Consider first the endofunctor $\mathcal{L}_1$ on $\text{Sig}_1$ defined on objects by mapping any 1-signature $X$ to the 1-signature with

- as sorts the set $X_0$, and
- as operations $\Gamma \vdash A$ the $\lambda$-terms $\Gamma \vdash M : A$, with base types in $X_0$ and constants in $X_1$, as sketched above, modulo $\beta \eta$.

On morphisms of 1-signatures $X \to Y$, let $\mathcal{L}_1(f)$ substitute constants $c \in X_1$ with $f_1(c)$. We obtain

**Proposition 2.4.** $\mathcal{L}_1$ is a monad on $\text{Sig}_1$, with unit and multiplication, say $\eta$ and $\mu$.

Let now $\text{L}_1\text{-Alg}$ be the category of algebras for the monad $\mathcal{L}_1$ and $\mathcal{K}_1$ be the ‘free algebra’ functor $X \mapsto (\mathcal{L}_1(X), \mu_X)$.

The functor $\mathcal{V}_1$ maps any cartesian closed category $\mathcal{C}$ to the $\text{L}_1\text{-algebra}$ with base 1-signature $(\mathcal{C}_0, \mathcal{C}_1)$, defined as follows. First, $\mathcal{C}_0$ is the set of objects of $\mathcal{C}$. It has a canonical $\mathcal{L}_0\text{-algebra}$ structure, say $h_0 : \mathcal{L}_0(\mathcal{C}_0) \to \mathcal{C}_0$, 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 morphisms 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 $\mathcal{L}_1\text{-algebra}$ structure, say $h_1 : \mathcal{L}_1(\mathcal{C}_0, \mathcal{C}_1) \to (\mathcal{C}_0, \mathcal{C}_1)$:

\[
\begin{align*}
\pi_i & : \mathcal{L}_1(\mathcal{C}_0, \mathcal{C}_1) \\
\eta & : \mathcal{L}_1(\mathcal{C}_0, \mathcal{C}_1) \\
h_1(G \vdash x_i : G_i) & = \pi_i \\
h_1(G \vdash (\cdot) : \bot) & = ! \\
h_1(G \vdash c(M_1, \ldots, M_n) : A) & = c \circ (h_1(M_1), \ldots, h_1(M_n)) \\
h_1(G \vdash \lambda x : A.M : B^A) & = \varphi(h_1(G, x : A \vdash M : B)) \\
h_1(G \vdash MN : B) & = ev \circ (h_1(M), h_1(N)) \\
h_1(G \vdash (M, N) : A \times B) & = \langle h_1(M), h_1(N) \rangle \\
h_1(G \vdash \pi M : A) & = \pi \circ h_1(M) \\
h_1(G \vdash \pi' M : A) & = \pi' \circ h_1(M),
\end{align*}
\]
where \(!\) is the unique morphism \(h_0(G) \to 1\), \(\varphi\) 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 \(h_0(B^A \times A) \to h_0(B)\).

\(\mathcal{L}_1\)-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., \(L_0(X_0)\),
- morphisms \(A \to B\) 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 \(X_\parallel\) denote the set of pairs of parallel operations, i.e., pairs of operations \(M, N\) over the same sequent. Otherwise said, \(X_\parallel\) is the pullback

\[
\begin{array}{c}
X_\parallel \\
\downarrow \\
X_1 \\
\downarrow \\
S_0(X_0).
\end{array}
\]

Any morphism \(f:X \to Y\) of 1-signatures yields a map \(f_\parallel:X_\parallel \to Y_\parallel\), via the dashed arrow (obtained by universal property of pullback) in

\[
\begin{array}{c}
X_\parallel \\
\downarrow \\
X_1 \\
\downarrow \\
S_0(X_0) \\
\downarrow \\
Y_1 \\
\downarrow \\
Y_\parallel \\
\downarrow \\
S_0(Y_0).
\end{array}
\]

**Definition 3.1.** A 2-signature consists of a 1-signature \(X\), plus a set \(X_2\) of reduction rules with a map \(X_2 \to \mathcal{L}_1(X_\parallel)\).

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

\[
\begin{array}{c}
X_2 \\
\downarrow \\
\mathcal{L}_1(X_\parallel) \\
\downarrow \\
\mathcal{L}_1(Y_\parallel).
\end{array}
\]

commute. We obtain:

**Proposition 3.2.** 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
\[
a : (t, t \vdash t) \quad \ell : (t^t \vdash t),
\]
with a reduction rule \(\beta\) over the pair \(x : t^t, y : t \vdash a(\ell(x), y), x(y) : t\) in \(L_1(\{t\}, \{\ell, a\})\). Categorically, this will yield a 2-cell
\[
t \times t \xrightarrow{\text{ev}} a \xrightarrow{\beta} t.
\]

This is an example of a higher-order rewrite system in the sense of Nipkow [15]. Nipkow’s definition is formally different, but his higher-order rewrite systems are in bijection with 2-signatures \(h : X_2 \to L_1(X)\) 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 dealing with decidability problems on higher-order rewrite systems, whose extension to our setting we leave open.

Let us now anticipate over Adjunction (1.1) and 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 \(R(X)\) be the following locally-preordered 2-category. It has:
- objects are types in \(L_0(X_0)\);
- morphisms \(A \to B\) are \(\lambda\)-terms in \(L_1(X)(A \vdash B)\), modulo \(\beta\eta\);
- given two parallel morphisms \(M\) and \(N\), there is one 2-cell \(M \to N\) exactly when there is a sequence of reductions \(M \to^* N\) in the usual sense [15].

**Proposition 4.1.** \(R(X)\) is 2-cartesian closed.

\(R(X)\) and \(H(X)\) 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 map any reduction \(P : M \to N\) to the unique reduction \(M \to N\) in \(R(X)\). Conversely, any standard reduction step has a proof, which provides a reduction \(P\). We have proved:

**Theorem 4.2** (Soundness and completeness). There exists an identity-on-objects, identity-on-morphisms, locally full cartesian closed 2-functor \(H(X) \xrightarrow{\dagger} R(X)\).

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 the adjunction
\[
\text{Sig} \xrightarrow{\text{CCCat}} \xleftarrow{\text{H'}} (4.1)
\]
recalled above.
This adjunction provides a categorical semantics for theories with binding, which is more general than other approaches by Fiore and Hur [7], Hirschowitz and Maggesi [9], and Zsidó [20].

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

We end this section by giving a formal construction of Adjunction (4.1). Cartesian closed categories form a full, reflective subcategory of 2CCCat, via the functor \( J:2CCCat \rightarrow CCCat \) mapping any cartesian closed 2-category \( C \) to the cartesian closed category with:

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

Here, \( J(C) \) is thought of as the free locally discrete cartesian closed 2-category. Adjunction (4.1) is obtained by composing the adjunctions \( \Sigma_g/bot \rightarrow 2CCCat/bot \rightarrow CCCat \).

### 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 [13], or the π-calculus [6, 14].

For example, consider the CCS term \( (a/|div)\cdot alt0/|div)\cdot alt0\cdot a/|div\cdot alt0\cdot a \). In CCS, it is structurally equivalent to \( (a/|div)\cdot alt0/|div)\cdot alt0\cdot a \), which then reduces to \( 0/|div\cdot alt0\cdot 0 \).

In order to account for this, we would have to consider a 2-signature with reduction rules for structural congruence, here \( (M_1/|div)\cdot alt0 \cdot M_2/|div)\cdot alt0 \cdot M_3 \rightarrow M_1/|div)\cdot alt0 \cdot M_2/|div)\cdot alt0 \cdot M_3 \) for associativity, and \( M/|div)\cdot alt0 \cdot N \rightarrow N/|div)\cdot alt0 \cdot M \) for commutativity. But then, these reductions count as proper reductions, which departs from the desired computational behaviour. For example, the term \( a/|div\cdot alt0\cdot a \) 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 \( X/\| \) 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 \( L(X)/\| \) (where \( L \) is defined in Section 5).

Another possibility would be to define 2-theories to consist of a pair of sets over \( L_1(X)/\| \), one for structural equations, and the other for proper reduction rules.

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 2-cells, as proposed by Hilken [8]. 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

\[
\begin{array}{ccc}
\text{Sig} & \xrightarrow{\mathcal{K}} & \mathcal{L}\text{-Alg} \\
& \mathcal{U} \downarrow & \mathcal{V} \\
& & \text{2CCCat,}
\end{array}
\]

where:
- $\mathcal{L}\text{-Alg}$ is the category of $\mathcal{L}$-algebras,
- $\mathcal{K} : \text{Sig} \to \mathcal{L}\text{-Alg}$ maps any $X$ to the free $\mathcal{L}$-algebra ($\mathcal{L}^2 X \xrightarrow{\mathcal{L}} \mathcal{L} X$),
- $\mathcal{U} (\mathcal{L} X \xrightarrow{h} X) = X$,
- 2CCCat 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 $X = ((X_0, X_1), a : X_2 \to \mathcal{L}_1 (X))$ (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 [8]) modulo permutation equivalence, which we now define.

First, terms, called reductions, are defined by induction in Figure 4. The typing judgement has the shape $\Gamma \vdash P : M \to N : A$, 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 $\mathcal{S}_0 (X_0)$.

Remark 5.1. For any $\Gamma \vdash M : A$, we have a reduction $\Gamma \vdash M : M \to M : A$.

When clear from context, we abbreviate substitutions $[M_1 / x_1, \ldots, M_n / x_n]$ of terms by $[M_1, \ldots, M_n]$. For a context $G$, $G_1$ denotes its $i$th type. Also, for $(M, N) \in \mathcal{L}_1 (X)$, we let $X(\mathcal{L}_1 (M, N))$ be the set of all reduction rules $r \in X_2$ such that $a(r) = (M, N)$. We write $X(\mathcal{L}_1 (M, N) : A)$ to indicate the common type of $M$ and $N$. Similarly, $X(G) \vdash A$ denotes the set of operations in $X_1$ above $G \vdash A$.

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

\[
\frac{\Gamma \vdash Q : N \to N'; \Delta \quad \Delta \vdash P : M \to M' : A}{\Gamma \vdash P [Q] : M[N] \to M'[N'] : A},
\]

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

First, observe that we have a form of weakening: for any reduction $\Gamma \vdash P : M \to N : A$ and $x \notin \Gamma$, we also have $\Gamma, x : B \vdash P : M \to N : A$. We use this implicitly in the following.

The definition of substitution is a bit tricky:
There is of course another legitimate definition, namely

$$\Gamma \vdash P : M_1 \rightarrow M_2 : A$$

$$\Gamma \vdash Q : M_2 \rightarrow M_3 : A$$

$$\Gamma, x : A, \Delta \vdash x : x \rightarrow A$$

$$\Gamma \vdash ( : ( \rightarrow ) : 1$$

$$\Gamma \vdash P_1 : M_1 \rightarrow N_1 : G_1$$

$$\Gamma \vdash P_n : M_n \rightarrow N_n : G_n$$

$$\Gamma \vdash c(P_1, \ldots, P_n) : c(M_1, \ldots, M_n) \rightarrow c(N_1, \ldots, N_n) : A$$

$$\Gamma, x : A, \Delta \vdash P : M \rightarrow N : B$$

$$\Gamma \vdash \lambda x : A. P : \lambda x : A. M \rightarrow \lambda x : A. N : B^A$$

$$\Gamma \vdash P : M \rightarrow M' : B^A$$

$$\Gamma \vdash Q : N \rightarrow N' : A$$

$$\Gamma \vdash (P, Q) : (M, N) \rightarrow (M', N') : A \times B$$

$$\Gamma \vdash \pi_{A,B}P : \pi_{A,B}M \rightarrow \pi_{A,B}N : A$$

$$\Gamma \vdash \pi'_{A,B}P : \pi'_{A,B}M \rightarrow \pi'_{A,B}N : B$$

Figure 1: Reductions

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

$$\Gamma \vdash Q : N \rightarrow N' : \Delta$$

$$\Gamma \vdash M[Q] : M[N] \rightarrow M'[N'] : A;$$

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

$$\Gamma \vdash N : \Delta$$

$$\Gamma \vdash P : M \rightarrow M' : A$$

$$\Gamma \vdash P[N] : M[N] \rightarrow M'[N] : A;$$

(where $N$ denotes a tuple);

- then we define substitution by $P[Q] = (P[Q] : M'[Q]).$

There is of course another legitimate definition, namely $M[Q] : M[N'] P[N']$. The two will be equated by permutation equivalence in the next section.

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

\[
\begin{align*}
()[Q] &= () \\
x_i[Q] &= Q_i \\
c[M_1, \ldots, M_p][Q] &= c[M_1[Q], \ldots, M_p[Q]] \\
(\lambda x : B. M)[Q] &= \lambda x : B. (M[Q], x) \quad \text{for } x \notin \text{dom}(\Delta) \\
(MN)[Q] &= (M[Q]N[Q]) \\
(M, N)[Q] &= (M[Q], N[Q]) \\
(\pi_{A,B}M)[Q] &= \pi_{A,B}(M[Q]) \\
(\pi'_{A,B}M)[Q] &= \pi'_{A,B}(M[Q])
\end{align*}
\]
Right whiskering is defined inductively, with \( \Delta = (x_1:A_1, \ldots, x_n:A_n) \) and \( N = (N_1, \ldots, N_n) \), by:

\[
\begin{align*}
(r \llbracket P_1, \ldots, P_p \rrbracket)[N] & = r \llbracket P_1[N], \ldots, P_p[N] \rrbracket \\
(P_1 \cdot M \cdot P_2)[N] & = (P_1[N] \cdot M[N][N] \cdot P_2[N]) \\
()[N] & = () \\
x_1[N] & = N_1 \\
c[P_1, \ldots, P_p][N] & = c[P_1[N], \ldots, P_p[N]] \\
(\lambda x:B.P)[N] & = \lambda x:B.(P'[N,x]) \quad \text{(for } x \notin \text{dom}(\Delta)) \\
(P_1, P_2)[N] & = (P_1[N], P_2[N]) \\
(\pi_{A,B}P')[N] & = \pi_{A,B}(P'[N]) \\
(\pi'_{A,B}P')[N] & = \pi'_{A,B}(P'[N]).
\end{align*}
\]

**Definition 5.2.** Let \( P'[Q] = (P[N] ; M'[N] ; M'[Q]). \)

**Proposition 5.3.** Given reductions \( P \) and \( Q \) as above, \( P'[Q] \) is a well-typed reduction \( \Gamma \vdash P'[Q] : M[N] \rightarrow M'[N'] : A \).

### 5.3. Permutation equivalence

We now define permutation equivalence on reductions, by the equations in Figures 3, 4 and 5 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 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 5.4.** \( \mathcal{L} \) is a functor \( \text{Sig} \rightarrow \text{Sig} \).

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

\[
\begin{align*}
\Gamma & \vdash R \llbracket P_1, \ldots, P_n \rrbracket : M[M_1, \ldots, M_n] \rightarrow N[N_1, \ldots, N_n] : A \\
\Gamma & \vdash R \llbracket P_1, \ldots, P_n \rrbracket : M[M_1, \ldots, M_n] \rightarrow N[N_1, \ldots, N_n] : A
\end{align*}
\]

We set \( \mu(R \llbracket P_1, \ldots, P_n \rrbracket) = R(\mu(P_1), \ldots, \mu(P_n)) \). The other cases just propagate the substitution:

\[
\begin{align*}
P : Q & \mapsto \mu(P) : \mu(Q) \\
x & \mapsto x \\
() & \mapsto () \\
c[P_1, \ldots, P_n] & \mapsto c[\mu(P_1), \ldots, \mu(P_n)] \\
\lambda x : A.P & \mapsto \lambda x : A.\mu(P) \\
PQ & \mapsto \mu(P)\mu(Q) \\
(P, Q) & \mapsto (\mu(P), \mu(Q)) \\
\pi P & \mapsto \pi(\mu(P)) \\
\pi' P & \mapsto \pi'(\mu(P)).
\end{align*}
\]

**Lemma 5.5.** This defines a natural transformation \( \mu : \mathcal{L}^2 \rightarrow \mathcal{L} \), which makes the diagram
Similarly, there is a natural transformation \( \eta : id \to \mathcal{L} \), mapping each \( r \in X(G \vdash M, N : A) \) to the reduction \( G \vdash r \langle x_1, \ldots, x_n \rangle : M \to N : A \), and we have:

**Lemma 5.6.** The diagram

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

commutes.

A crucial result is:

**Corollary 5.7.** \((\mathcal{L}, \mu, \eta)\) is a monad on \(\text{Sig}\).

**Proposition 5.8.** For all \( \Gamma \vdash Q : N \to N' : \Delta \) and \( \Delta \vdash P : M \to M' : A \), we have:

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

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

\[ P[Q] = (c \langle P_1[N], \ldots, P_p[N] \rangle ; c \langle M'_1[N], \ldots, M'_p[N] \rangle) \equiv c \langle M'_1[Q], \ldots, M'_p[Q] \rangle. \]

By the third lifting rule, this is permutation equivalent to

\[ c \langle P_1[N] ;_{M'_1[N]} M'_1[Q], \ldots, P_p[N] ;_{M'_p[N]} M'_p[Q] \rangle. \]

By \( p \) applications of the induction hypothesis, we obtain

\[ c \langle M_1[Q] ;_{M_1[N']} P_1[N'], \ldots, M_p[Q] ;_{M_p[N']} P_p[N'] \rangle, \]

which by lifting again yields the desired result:

\[ c \langle M_1[Q], \ldots, M_p[Q] ;_{M_1[N'], \ldots, M_p[N']} c \langle P_1[N'], \ldots, P_p[N'] \rangle. \]

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

\[ r \langle P_1[N], \ldots, P_p[N] \rangle ;_{M'_0[M'_1, \ldots, M'_p][N]} M'_0[M'_1, \ldots, M'_p][Q]. \]

By lifting, omitting indices of vertical compositions and implicitly using the category rules, we have

\[ r \langle P_1[N], \ldots, P_p[N] \rangle \equiv r \langle M_1[N], \ldots, M_p[N] \rangle ;_{M'_0[P_1[N], \ldots, P_p[N]]} M'_0[M'_1[Q], \ldots, M'_p[Q]], \]

the whole is permutation equivalent to

\[ r \langle M_1[N], \ldots, M_p[N] \rangle ;_{M'_0[P_1[N], \ldots, P_p[N]]} M'_0[M'_1[Q], \ldots, M'_p[Q]], \]
i.e., by lifting (inductively):
\[
\begin{align*}
&\langle M_1[N], \ldots, M_p[N] \rangle; \\
&\langle M'_0[(P_1[N]; M'_1[Q]), \ldots, (P_p[N_; M'_p[Q])] \rangle.
\end{align*}
\]

By induction hypothesis, this is permutation equivalent to
\[
\begin{align*}
&\langle M_1[N], \ldots, M_p[N] \rangle; \\
&\langle M'_0[(M_1[Q]; P_1[N']), \ldots, (M_p[Q]; P_p[N'])] \rangle,
\end{align*}
\]
i.e., by lifting again to
\[
\begin{align*}
&\langle M_1[N], \ldots, M_p[N] \rangle; \\
&\langle M'_0[M_1[Q], \ldots, M_p[Q]]; \\
&\langle M'_0[P_1[N'], \ldots, P_p[N']] \rangle.
\end{align*}
\]

The second lifting rule then yields
\[
\begin{align*}
&\langle M_1[q], \ldots, M_p[q] \rangle; \\
&\langle M'_0[M_1[q], \ldots, M_p[q]]; \\
&\langle M'_0[P_1[N'], \ldots, P_p[N']] \rangle,
\end{align*}
\]
and hence
\[
\begin{align*}
&M_0[M_1[q], \ldots, M_p[q]]; \\
&M_0[M_1[q], \ldots, M_p[q]]; \\
&r\langle P_1[N'], \ldots, P_p[N'] \rangle,
\end{align*}
\]
so, by the second lifting rule again:
\[
\begin{align*}
&M_0[M_1[q], \ldots, M_p[q]]; \\
&r\langle P_1[N'], \ldots, P_p[N'] \rangle,
\end{align*}
\]
i.e., the right-hand side.

6. Cartesian closed 2-categories

6.1. **Definition.** In a 2-category \( C \), a diagram \( A \xleftarrow{p} C \xrightarrow{q} B \) is a *product diagram* iff for all object \( D \), the induced functor
\[
C(D, C) \xrightarrow{(C(D,p), C(D,q))} C(D, A) \times C(D, B)
\]
is an isomorphism of categories. Because this family of functors is 2-natural in \( D \), the inverse functors will also be 2-natural.

Similarly, an object \( 1 \) of \( C \) is *terminal* iff for all \( D \) the unique functor
\[
C(D, 1) \xrightarrow{1} 1
\]
is an isomorphism (where the right-hand 1 is the terminal category).

**Definition 6.1.** A 2-category with finite products, or fp 2-category, is a 2-category \( C \), equipped with a terminal object and a 2-functor
\[
\begin{align*}
C \times C \xrightarrow{x} C,
\end{align*}
\]
plus, for all \( A \) and \( B \), a product diagram
\[
\begin{align*}
A \xleftarrow{p} A \times B \xrightarrow{q} B.
\end{align*}
\]
In such an fp 2-category \( C \), given objects \( A \) and \( B \), an *exponential* for them is a pair of an object \( B^A \) and a morphism \( ev: A \times B^A \to B \), such that for all \( D \), the functor
\[
\begin{align*}
C(D, B^A) \xrightarrow{(C(D, p), C(D, 1))} C(D, A) \times C(D, B)
\end{align*}
\]
is an isomorphism. As above, because this family of functors is 2-natural in $D$, the inverse functors will also be 2-natural.

**Definition 6.2.** 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 $2CCCat$ has cartesian closed 2-categories as objects, and strictly structure-preserving functors between them as morphisms.

We observe in particular that this implies preservation of projections and evaluation morphisms.

7. Main adjunction

7.1. **Right adjoint.** Given a cartesian closed 2-category $C$, define $\mathcal{V}(C) = (C_0, C_1, C_2)$ as follows. First, let as in Section 2 ($C_0, C_1) = \mathcal{V}_1(C)$, and recall the canonical $\mathcal{L}_0$ and $\mathcal{L}_1$-algebra structures $h_0$ and $h_1$. Let then the reduction rules in $C_2(G \vdash M, N : A)$ be the 2-cells in $C(h_0(G), h_0(A))(h_1(M), h_1(N))$, abbreviated to $C(G, A)(M, N)$ in the sequel.

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

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

**Lemma 7.1.** For all $\Delta \vdash Q : N \to N' : \Gamma$ and $\Gamma \vdash P : M \to M' : A$ in $\mathcal{L}(\mathcal{V}C)$,

$$
\Delta \vdash \begin{array}{c}
\left\langle \psi M, N \right\rangle \\
M[N]
\end{array}
\begin{array}{c}
\left\langle \psi M, N \right\rangle \\
M'[N']
\end{array}
\begin{array}{c}
P(P[Q])
\end{array}
\begin{array}{c}
P(P[Q])
\end{array}
A
= \Delta \vdash \begin{array}{c}
\left\langle \psi M, N \right\rangle \\
M[N]
\end{array}
\begin{array}{c}
\left\langle \psi M, N \right\rangle \\
M'[N']
\end{array}
\begin{array}{c}
\psi_2(P[Q])
\end{array}
\begin{array}{c}
\psi_2(P[Q])
\end{array}
\begin{array}{c}
\Gamma
\end{array}
\begin{array}{c}
\Gamma
\end{array}
\begin{array}{c}
\psi_2(P)
\end{array}
\begin{array}{c}
\psi_2(P)
\end{array}
\begin{array}{c}
A.
\end{array}
$$

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

**Lemma 7.2.** Any two equated reductions are mapped to the same 2-cell in $C$.

**Proof.** We proceed by induction on the proof of the considered equation. The congruence and category rules of Figures 3 and 4 hold because, in $C$, 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 $C$ as

$$
\prod \Gamma \begin{array}{c}
\left\langle \psi M, N \right\rangle \\
\left\langle \psi M', N' \right\rangle
\end{array}
\begin{array}{c}
\left\langle \psi P, Q \right\rangle \\
\left\langle \psi P, Q \right\rangle
\end{array}
B^A \times A \xrightarrow{ev} B
$$

which is equal to
Figure 2: The \( \mathcal{L} \)-algebra structure on \( \mathcal{V}(\mathcal{C}) \)
which is in turn equal (by cartesian closedness of \( \mathcal{C} \)) to:

\[
\prod \Gamma \langle \text{id},N \rangle \prod \Gamma \times A \xrightarrow{\varphi M \times A} B \times A \xrightarrow{ev} B
\]

and hence to the right-hand side of the equation by Lemma \([7.1]\). 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 7.3.** \( \mathcal{V} \) is a functor \( \mathbf{2CCCat} \rightarrow \mathbf{Sig} \).

### 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 \( A \xrightarrow{N[M]} C \). Identities are given by variables, as usual. Vertical composition of 2-cells

\[
A \xrightarrow{M_1} B \xrightarrow{M_2 \varphi \alpha} B \xrightarrow{M_3 \varphi \beta} C
\]

is given by \( h(\eta(\alpha); M_2, \eta(\beta)) \).

Horizontal composition of 2-cells

\[
A \xrightarrow{M} B \xrightarrow{N \varphi \beta} C
\]

(7.1)

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

The identity at \( A \xrightarrow{M} B \) is \( h(M) \).
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**

\[
A \xymatrix{ & B \ar[r]^N & C \ar[l]_*[r]_\alpha^M \ar[l]_{M'}_*[l]_\beta}
\]

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

**Lemma 7.4.** We have: \(h((h(N)(\eta(\alpha)))) = h(N[\eta(\alpha)])\).

**Proof.** Indeed, consider the term \(N[\eta(\alpha)]\) 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 \(h(N[\eta(\alpha)])\). \(\Box\)

Similarly, consider the **right whiskering**

\[
A \xymatrix{ & B \ar[r]^N & C \ar[l]_*[r]_\alpha^M \ar[l]_{M'}_*[l]_\beta}
\]

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

**Lemma 7.5.** We have: \(h(\gamma(\eta(h(M)))) = h(\gamma(\eta(M)))\).

**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 \(h(\gamma(\eta(M)))\). \(\Box\)

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

**Lemma 7.6.** For any cells as in \ref{111},

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

**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 \(\eta(N[\eta(\alpha)]) ; \eta(\beta(M'))\). Evaluating as before yields the right-hand equality. \(\Box\)

Furthermore, consider any configuration like:

\[
A \xymatrix{ & B \ar[r]^N & C \ar[l]_*[r]_\alpha^M \ar[l]_{M'}_*[l]_\beta}
\]

**Lemma 7.7.** 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 $N[\eta(\eta(\alpha); \eta(\beta))]$. Evaluating yields equality of

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

Finally, by a similar argument, we have:

**Lemma 7.8.** For any $A \rightarrow M \rightarrow B \rightarrow C$, we have

\[
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
A \quad M \\
\end{array}
\end{array}
\end{array}
\]

we have $(\alpha \circ id_M); (\beta \circ id_M) = (\alpha; \beta) \circ id_M$.

**Lemma 7.9.** The interchange law holds, i.e., for all reduction rules as in $A \rightarrow M_1 \rightarrow M_2 \rightarrow \rightarrow B \rightarrow N_1 \rightarrow N_2 \rightarrow C$, we have

\[
(\gamma; \theta) \circ (\alpha; \beta) = (\gamma \circ \alpha); (\theta \circ \beta).
\]

**Proof.** By the previous results, we have

\[
(\gamma; \theta) \circ (\alpha; \beta) = ((\gamma; \theta) \circ M_1); (N_3 \circ (\alpha; \beta)) = (\gamma \circ M_1); (\theta \circ M_1); (N_3 \circ \alpha); (N_3 \circ \beta) = (\gamma \circ M_1); (\theta \circ M_2); (N_3 \circ \beta) = (\gamma \circ \alpha); (\theta \circ \beta).
\]

Now, let us show cartesian closedness. We have a bijection of hom-sets $L_1(X)(C \vdash A \times B) \approx L_1(X)(C \vdash A) \times L_1(X)(C \vdash B)$, given by

\[
L_1(X)(C \vdash A \times B) \rightarrow L_1(X)(C \vdash A) \times L_1(X)(C \vdash B)
\]

\[
M \mapsto \pi M, \pi' M
\]

and

\[
L_1(X)(C \vdash A) \times L_1(X)(C \vdash B) \rightarrow L_1(X)(C \vdash A \times B)
\]

\[
M, N \mapsto (M, N).
\]

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

\[
L(X)(C \vdash M, N; A \times B) \rightarrow L(X)(C \vdash \pi M, \pi N; A) \times L(X)(C \vdash \pi' M, \pi' N; B)
\]

\[
P \mapsto \pi P, \pi' P
\]

and (omitting $C$)

\[
L(X)(M_1, N_1; A) \times L(X)(M_2, N_2; B) \rightarrow L(X)((M_1, M_2), (N_1, N_2); A \times B)
\]

\[
P_1, P_2 \mapsto (P_1, P_2),
\]
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
\[
\begin{array}{ccc}
X_2(M, N) & \xrightarrow{u} & X_2(\pi M, \pi N) \times X_2(\pi' M, \pi' N) \\
\downarrow{\eta} & & \downarrow{h \times h} \\
\mathcal{L}(X)(M, N) & \xrightarrow{\eta \times \eta} & \mathcal{L}(X)(\pi M, \pi N) \times \mathcal{L}(X)(\pi' M, \pi' N)
\end{array}
\]
and
\[
\begin{array}{ccc}
X_2(\pi M, \pi N) \times X_2(\pi' M, \pi' N) & \xrightarrow{v} & X_2(M, N) \\
\downarrow{\eta \times \eta} & & \downarrow{h} \\
\mathcal{L}(X)(\pi M, \pi N) \times \mathcal{L}(X)(\pi' M, \pi' N) & \xrightarrow{\eta} & \mathcal{L}(X)(M, N).
\end{array}
\]

Starting from \(r \in X_2(M, N)\), we obtain
\[
v(u(r)) = h(\eta(\pi(\eta(r)))) , \eta(\pi'(\eta(r))).
\]
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 \(h \circ \mathcal{L}\mu\) are respectively:
- \(h(\eta(\pi(\eta(r)))), \eta(\pi'(\eta(r))))\), 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
\[
h(\pi(\eta(\pi(\eta(r), \eta(s)))))) \quad \text{and} \quad h(\pi'(\eta(\eta(\eta(r), \eta(s))))).
\]
Considering \(\pi(\eta(\eta(r), \eta(s))) \in \mathcal{L}(\mathcal{L}X)\), its images by \(h \circ \mathcal{L}(h)\) and \(h \circ \mathcal{L}\mu\) 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 7.10.** This yields a cartesian closed 2-category structure on \(\mathcal{L}\).

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

### 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 \(\mathcal{L}_1(X_0, X_1)(\mu(\prod G) \vdash \mu(A))\), where \(\mu\) denotes the monad multiplication for \(\mathcal{L}_0\). Reduction rules in \(Y_2(G \vdash M, N; B)\) are reduction rules in \(X_2(\mu(\prod G) \vdash M', N'; \mu(B))\), where \(M' = M[\pi_1 x/x_1, \ldots, \pi_n x/x_n]\) (and similarly for \(N'\)).

Let \(\eta_X^c\) map:
- each sort \(t \in X_0\) to the type \(t \in \mathcal{L}_0(X_0)\) (this is the monad unit for \(\mathcal{L}_0\)),
- each operation \(c \in X(G \vdash A)\) to the term \(c[\pi_1 x, \ldots, \pi_n x]\), and
• each reduction rule \( r \in X_2(G \vdash M, N : A) \) to the reduction rule
\[
h(r \langle \pi_1 x, \ldots, \pi_n x \rangle) \in X_2(\prod G \vdash M', N' : A).
\]
(Thanks to the fact that \( \mu(L_0(\eta)(A)) = A \).)

**Theorem 7.11.** This \( \eta^C \) is a natural transformation which is the unit of an adjunction

\[
\begin{array}{ccc}
\mathcal{L}\text{-Alg} & \xrightarrow{\mathcal{F}} & \text{2CCCat} \\
\rotatebox{90}{$\eta^C$} & \downarrow & \rotatebox{90}{$\mathcal{V}$} \\
\mathcal{X} & \xrightarrow{f} & \mathcal{V}((\mathcal{X})
\end{array}
\]

**Proof.** Consider any morphism \( f : (X, h) \to \mathcal{V}(\mathcal{C}) \), and let \( \mathcal{X} = \mathcal{F}(X, h) \), \( (Y, k) = \mathcal{V}(\mathcal{X}) \), and \( \mathcal{V}(\mathcal{C}) = (C_0, C_1, h_2 : C_2 \to C_1) \). We now define a uniquely determined cartesian closed functor \( f' : \mathcal{X} \to \mathcal{C} \) making the triangle

\[
\begin{array}{ccc}
X & \xrightarrow{\eta^C_X} & \mathcal{V}(\mathcal{X}) \\
\rotatebox{90}{$f$} & \downarrow \mathcal{V}(f') & \downarrow \mathcal{V}(f') \\
& \mathcal{V}(\mathcal{C}) & \\
\end{array}
\]

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 \( f_2 : X_2(A \vdash M, N : B) \to \mathcal{C}(f'(A), f'(B))(f'(M), f'(N)) \), which is also the only possible choice from \( f \).

This indeed makes the above triangle commute, because any \( r \in X_2(G \vdash M, N : A) \) is first mapped to \( h(r \langle \pi_1 x, \ldots, \pi_n x \rangle) \in X_2(\prod G \vdash M', N' : A) \), and then to
\[
h(r \langle \pi_1 x, \ldots, \pi_n x \rangle)
\]
in \( \mathcal{C} \), which, because \( f \) is a morphism of \( \mathcal{L} \)-algebras, is equal to
\[
h_2((f_2(r)) \langle \pi_1 x, \ldots, \pi_n x \rangle),
\]
i.e., to \( f_2(r) \).

It thus remains to show that \( f' \) is cartesian closed, which follows by \( f \) being a morphism of \( \mathcal{L} \)-algebras. For example, to show that binary pairings 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 \( x : C \vdash r \langle x \rangle : M_1 \to M_2 : A \) and \( x : C \vdash s \langle x \rangle : N_1 \to N_2 : B \) and taking \( h(r \langle x \rangle, s \langle x \rangle) \), which is mapped by \( f_2 \) to \( f_2(h(r \langle x \rangle, s \langle x \rangle)) \). But, because \( f \) is a morphism of \( \mathcal{L} \)-algebras, this is the same as \( h_2((f_2(r)) \langle x \rangle, (f_2(s)) \langle x \rangle) \), which is by definition (i.e., Figure 2) the pairing \( (f_2(r), f_2(s)) \) in \( \mathcal{C} \).

**Acknowledgements.** Thanks to Nicolas Tabareau for useful feedback, and to Aurore Alcolei for reviving the subject after a few years.
References


Figure 3: Equations on reductions (Congruence)
### Category

\[\Gamma \vdash P_1 : M_1 \rightarrow M_2 : A \quad \Gamma \vdash P_2 : M_2 \rightarrow M_3 : A \quad \Gamma \vdash P_3 : M_3 \rightarrow M_4 : A\]

\[\Gamma \vdash (P_1 ; M_2 (P_2 ; M_3 P_3)) \equiv ((P_1 ; M_2 P_2) ; M_3 P_3) : M_1 \rightarrow M_4 : A\]

\[\Gamma \vdash P : M \rightarrow N : A \quad \Gamma \vdash (P ; N) \equiv P : M \rightarrow N : A\]

### Beta and Eta

\[\Gamma, x : A \vdash P : M \rightarrow M' : B \quad \Gamma \vdash Q : N \rightarrow N' : A\]

\[\Gamma \vdash ((\lambda x : A.P)Q) \equiv P[Q/x] : (\lambda x : A.M)N \rightarrow M'[N'/x] : B\]

\[\Gamma \vdash \pi(P, Q) \equiv \pi(M_1, N_1) \rightarrow M_2 : A\]

### Figure 4: Equations on reductions (Category and Beta-Eta)
Figure 5: Equations on reductions (Lifting)