A type theory for cartesian closed bicategories

(Extended Abstract)

Marcelo Fiore Philip Saville
Department of Computer Science and Technology, University of Cambridge

***Index Terms*—typed lambda calculus, higher category theory, Curry-Howard-Lambek correspondence, cartesian closed bicategories**

***Abstract*—We construct an internal language for cartesian closed bicategories. Precisely, we introduce a type theory modelling the structure of a cartesian closed bicategory and show that its syntactic model satisfies an appropriate universal property, thereby lifting the Curry-Howard-Lambek correspondence to the bicategorical setting. Our approach is principled and practical. Weak substitution structure is constructed using a bicategorification of the notion of abstract clone from universal algebra, and the rules for products and exponentials are synthesised from semantic considerations. The result is a type theory that employs a novel combination of 2-dimensional type theory and explicit substitution, and directly generalises the Simply-Typed Lambda Calculus. This work is the first step in a programme aimed at proving coherence for cartesian closed bicategories.**

I. Introduction

2-categories axiomatise the structures formed by classes of categories, such as the 2-category $\mathbf{Cat}$ of small categories, functors and natural transformations. In such settings the associativity and unit laws of composition hold strictly ('on the nose'). In many situations—in particular where composition is defined by a universal property—these laws only hold up to coherent isomorphism: the resulting structure is that of a *bicategory*. Bicategories are rife in mathematics and theoretical computer science, arising for instance in algebra [1], [2], semantics of computation [3], [4], datatype models [5], [6], categorical logic [7], [8], and categorical algebra [9], [10], [11].

The layers of coherence data required to witness the associativity and unit laws makes calculating in bicategories (and weak n-categories more generally) notoriously difficult. One approach is to reduce bicategorical structure to categorical structure by quotienting, but the loss of intensional information this entails is often unsatisfactory.

There are two main strategies for working with structures defined up to isomorphism. One strategy looks for *coherence theorems* establishing that some class of diagrams always commute. For bicategories and bicategorical limits (bilimits [12]) there are well-known coherence results [13], [14]; however, we know of no analogous result in the literature for closed structure. Another strategy employs a type theory that matches the categorical structure (see *e.g.* [15], [16], [8]); such a system is sometimes called the *internal language* [17] or *internal logic* [18].

In this paper we carry out the internal-language strategy for cartesian closed bicategories, and thereby set up the scene for the coherence strategy to be presented elsewhere. We construct an internal language $\Lambda_{\text{ps}}^{\times,\rightarrow}$ for cartesian closed bicategories (where 'ps' stands for *pseudo*), thus reducing the problem of coherence for cartesian closed bicategories to a property of $\Lambda_{\text{ps}}^{\times,\rightarrow}$. This type theory provides a practical calculus for reasoning in such settings and directly generalises the STLC (Simply-Typed Lambda Calculus) [19].

Our work is motivated by the complexities of calculating in the cartesian closed bicategories of generalised species [7] and of cartesian distributors [9], specifically for their application to higher-dimensional category theory [20]. However, the internal language we present applies to other examples: cartesian closed bicategories also appear in categorical algebra [10] and game semantics [21].

A. 2-dimensional type theories and bicategorical composition

There is a natural connection between 2-categories and rewriting. If objects are types and morphisms are terms, then 2-cells are rewrites between terms. This idea was explored as early as the 1980s in the work of Rydeheard & Stell [22] and Power [23]. For STLC, Simply-Typed Lambda Calculus, Seely [24] suggested that η-expansion and β-contraction may naturally be interpreted as the unit and counit of the adjunction defining (lax) exponentials in a 2-category, an approach followed by Hilken [25] and advocated by Ghani & Jay [26], [27]. More recently, type-theoretic constructions modelling 2-categories with strict cartesian closed structure have been pursued in programming-language theory [28] and proof theory [29] while a directed 2-dimensional type theory in the style of Martin-Löf [30] has been introduced by Licata & Harper [31].

It is crucial to our approach that the equational theory of $\Lambda_{\text{ps}}^{\times,\rightarrow}$ does not identify any more structure than the axioms of cartesian closed bicategories. This entails distinguishing more terms than in STLC. For instance, note that terms such as

$$t[u_1/x_1, u_2/x_2][v/y] \quad \text{and} \quad t[u_1[v/y]/x_1, u_2[v/y]/x_2]$$

are respectively interpreted in a Lambek-style semantics [17] by the equal maps

$$[\![t]\!] \circ \langle [\![u_1]\!], [\![u_2]\!] \rangle \circ [\![v]\!] \quad \text{and} \quad [\![t]\!] \circ \langle [\![u_1]\!] \circ [\![v]\!], [\![u_2]\!] \circ [\![v]\!] \rangle$$

In contrast, in the *bi*categorical setting these composites are only isomorphic.[1] Hence, substitution ought to be associative only up to isomorphism. This places us outside the 2-categorical

[1] This issue is similar to that identified by Curien [32], who attempts to rectify the mismatch between locally cartesian closed categories and Martin-Löf dependent type theory caused by interpreting the (strictly associative) substitution operation as a pullback (associative up to coherent isomorphism).

world of previous work, as well as setting our work apart from type theories in which weak structure is modelled in a strict language, such as Homotopy Type Theory (*c.f.* [33]).

B. The type theory $\Lambda_{\mathrm{ps}}^{\times,\rightarrow}$

We will construct the internal language $\Lambda_{\mathrm{ps}}^{\times,\rightarrow}$ in stages. First we will construct the internal language of bicategories $\Lambda_{\mathrm{ps}}^{\mathrm{b}}$ (Section V) and then the internal language of bicategories with finite products $\Lambda_{\mathrm{ps}}^{\times}$ (Section VII); the type theory $\Lambda_{\mathrm{ps}}^{\times,\rightarrow}$ extends both these systems (Section IX). In each case we construct the syntactic model and prove an appropriate 2-dimensional freeness universal property.

We introduce substitution formally using a version of *explicit substitution* [34], [35]. This syntactic structure and the axioms it is subject to are synthesised from a bicategorification of the abstract clones [36] of universal algebra (Section IV). Abstract clones are a natural bridge between syntactic structure (in the form of intuitionistic type theories or calculi) and semantic structure (in the form of Lawvere theories or cartesian multicategories).

Cartesian closed structure is synthesised from universal arrows at both the global (2-dimensional) and the local (1-dimensional) levels. From this approach we recover versions of the usual $\beta\eta$-laws of STLC, while keeping the rules to a minimum (Sections VII and Section IX). Our formulation points towards similar constructions for tricategories (weak 3-categories [37], [38]) or even ∞-categories.

The type theory $\Lambda_{\mathrm{ps}}^{\times,\rightarrow}$ is, in a precise sense, a language for cartesian closed bicategories. It is capable of being formalised in proof assistants such as Agda [39] and the principled nature of its construction makes it readily amenable to the addition of further structure. We leave such extensions for future work.

II. BICATEGORIES

We recall the definition of bicategory, pseudofunctor and biequivalence. For leisurely introductions consult *e.g.* [1], [2, §9].

Definition II.1 ([1])**.** A *bicategory* $\mathcal{B}$ consists of

- a class of objects $ob(\mathcal{B})$,
- for every $X, Y \in ob(\mathcal{B})$ a *hom-category* $\big(\mathcal{B}(X,Y), \bullet, \mathrm{id}\big)$ with objects *1-cells* $f : X \to Y$ and morphisms *2-cells* $\alpha : f \Rightarrow f' : X \to Y$; composition of 2-cells is called *vertical composition*,
- for every X, Y, $Z \in ob(\mathcal{B})$ an *identity* functor $\mathrm{Id}_X : \mathbb{1} \to \mathcal{B}(X,X)$ and a *horizontal composition* functor $\circ_{X,Y,Z} : \mathcal{B}(Y,Z) \times \mathcal{B}(X,Y) \to \mathcal{B}(X,Z)$,
- invertible 2-cells

$$\begin{aligned}\mathbf{a}_{h,g,f} &: (h \circ g) \circ f \Rightarrow h \circ (g \circ f) : W \to Z \\ \mathbf{l}_f &: \mathrm{Id}_X \circ f \Rightarrow f : W \to X \\ \mathbf{r}_g &: g \circ \mathrm{Id}_X \Rightarrow g : X \to Y\end{aligned}$$

  for every $f : W \to X$, $g : X \to Y$ and $h : Y \to Z$, natural in each of their arguments and satisfying two coherence laws.

The functoriality of horizontal composition gives rise to an *interchange law*: for suitable 2-cells $\tau, \tau', \sigma, \sigma'$ one has $(\tau' \bullet \tau) \circ (\sigma' \bullet \sigma) = (\tau' \circ \sigma') \bullet (\tau \circ \sigma)$.

A morphism of bicategories is called a pseudofunctor (or homomorphism). It is a mapping on objects, 1-cells and 2-cells that preserves horizontal composition up to isomorphism.

Definition II.2 ([1])**.** A *pseudofunctor* $F : \mathcal{B} \to \mathcal{C}$ between bicategories $\mathcal{B}$ and $\mathcal{C}$ consists of

- a mapping $F : ob(\mathcal{B}) \to ob(\mathcal{C})$,
- a functor $F_{X,Y} : \mathcal{B}(X,Y) \to \mathcal{C}(FX, FY)$ for every $X, Y \in ob(\mathcal{B})$,
- an invertible 2-cell $\psi_X : \mathrm{Id}_{FX} \Rightarrow F(\mathrm{Id}_X)$ for every $X \in ob(\mathcal{B})$,
- an invertible 2-cell $\phi_{f,g} : F(f) \circ F(g) \Rightarrow F(f \circ g)$ for every $g : X \to Y$ and $f : Y \to Z$, natural in f and g

subject to three coherence laws. A pseudofunctor for which ψ and ϕ are both the identity is called *strict*.

Example II.3. Every 2-category is a bicategory and every 2-functor is a strict pseudofunctor. A one-object bicategory is equivalently a monoidal category; a monoidal functor is equivalently a pseudofunctor between one-object bicategories.

Bicategorical products and exponentials are defined using the appropriate notion of adjunction, called a biadjunction. For our purposes, the characterisation of biadjoints in terms of biuniversal arrows [40] is most natural (*c.f.* [7]); this is the bicategorical version of the well-known description of adjunctions via universal arrows (*e.g.* [41, Chapter III, §1]). For biuniversal arrows and their relationship to biadjunctions, see *e.g.* [42].

Definition II.4 ([12])**.** Let $F : \mathcal{B} \to \mathcal{C}$ be a pseudofunctor. To give a *right biadjoint* U to F is to give

1) a mapping $U : ob(\mathcal{C}) \to ob(\mathcal{B})$ on objects,
2) a family of 1-cells $(q_C : FUC \to C)_{C \in ob(\mathcal{C})}$,
3) for every $B \in ob(\mathcal{B})$ and $C \in ob(\mathcal{C})$ an adjoint equivalence

$$\mathcal{B}(B, UC) \underset{(-)^\flat}{\overset{q_C \circ F(-)}{\rightleftarrows}} \mathcal{C}(FB, C) \quad (\perp)$$

Morphisms of pseudofunctors are called *pseudonatural transformations* [12] and morphisms of pseudonatural transformations are called *modifications* [1]. Bicategories, pseudofunctors, pseudonatural transformations and modifications organise themselves into a tricategory we denote $\mathbf{Bicat}$.

Example II.5. For every pair of bicategories $\mathcal{B}$ and $\mathcal{C}$ there is a bicategory $\mathrm{Hom}(\mathcal{B}, \mathcal{C})$ of pseudofunctors, pseudonatural transformations and modifications.

Bicategories provide a convenient setting for abstractly describing many categorical concepts (*e.g.* [43], [44]).

Definition II.6. Let $\mathcal{B}$ be a bicategory.

1) An *adjunction* $(A, B, f, g, \eta, \epsilon)$ in $\mathcal{B}$ is a pair of objects (A, B) with arrows $f : A \leftrightarrows B : g$ and 2-cells

$\eta : \mathrm{Id}_A \Rightarrow g \circ f$ and $\epsilon : f \circ g \Rightarrow \mathrm{Id}_B$ subject to two triangle laws.

2) An *equivalence* $(A, B, f, g, \eta, \epsilon)$ in $\mathcal{B}$ is a pair of objects (A, B) with arrows $f : A \leftrightarrows B : g$ and invertible 2-cells $\eta : \mathrm{Id}_A \overset{\cong}{\Rightarrow} g \circ f$ and $\epsilon : f \circ g \overset{\cong}{\Rightarrow} \mathrm{Id}_B$.
3) An *adjoint equivalence* is an adjunction that is also an equivalence.

The appropriate notion of equivalence between bicategories is called biequivalence [45].

Definition II.7. A *biequivalence* between bicategories $\mathcal{B}$ and $\mathcal{C}$ consists of pseudofunctors $F : \mathcal{B} \leftrightarrows \mathcal{C} : G$ with equivalences $G \circ F \simeq \mathrm{id}_{\mathcal{B}}$ and $F \circ G \simeq \mathrm{id}_{\mathcal{C}}$ in the bicategories $\mathrm{Hom}(\mathcal{B}, \mathcal{B})$ and $\mathrm{Hom}(\mathcal{C}, \mathcal{C})$, respectively.

III. SIGNATURES FOR 2-DIMENSIONAL TYPE THEORIES

The STLC with constants is determined by a choice of base types and constant terms (*e.g.* [17]). For constants defined in arbitrary contexts such a choice is determined by a *multigraph*; that is, a set of *nodes* $A_1, \ldots, A_n, B, \ldots$ connected by *multiedges* $[A_1, \ldots, A_n] \to B$. A multigraph consisting solely of *edges* (*i.e.* multiedges of the form $[A] \to B$) is called a *graph*.

Notation III.1. In the following definition, and throughout, we write $A_\bullet$ for a finite sequence $[A_1, \ldots, A_n]$ ($n \in \mathbb{N}$).

Definition III.2. A *2-multigraph* $\mathcal{G}$ is a set $\mathcal{G}_0$ of *nodes* equipped with a graph $\mathcal{G}(A_\bullet; B)$ of *edges* and *surfaces* for every $n \in \mathbb{N}$ and $A_1, \ldots, A_n, B \in \mathcal{G}_0$. A *homomorphism* of 2-multigraphs $h : \mathcal{G} \to \mathcal{G}'$ is a map $h : \mathcal{G}_0 \to \mathcal{G}'_0$ together with functions

$$h_{A_1,\ldots,A_n;B} : \mathcal{G}(A_\bullet; B) \to \mathcal{G}'([hA_1, \ldots, hA_n]; hB)$$
$$h_{f,g} : \mathcal{G}(A_\bullet; B)(f, g) \to \mathcal{G}'([hA_1, \ldots, hA_n]; hB)(hf, hg)$$

for every $n \in \mathbb{N}$, $A_1, \ldots, A_n, B \in \mathcal{G}_0$ and $f, g \in \mathcal{G}(A_\bullet; B)$.

We denote the category of 2-multigraphs by 2-MGrph. The full subcategory 2-Grph of *2-graphs* is formed by restricting to 2-multigraphs $\mathcal{G}$ such that $\mathcal{G}([A_1, \ldots, A_n]; B) = \varnothing$ whenever $n \neq 1$.

IV. SUBSTITUTION STRUCTURE UP TO ISOMORPHISM

The type theory we shall construct has types, terms, rewrites and an explicit substitution operation. It is therefore determined by a 2-multigraph together with a specified substitution structure. Accordingly, to synthesise our language we introduce an intermediate step between 2-multigraphs and bicategories, which we call biclones. These are a bicategorification of the abstract clones of universal algebra [36], which capture a presentation-independent notion of equational theory with substitution.

Definition IV.1. An *S-biclone* $\mathcal{C}$ is a set S of *sorts* equipped with the following for all n, $m \in \mathbb{N}$ and $X_1, \ldots, X_n, Y, Y_1, \ldots, Y_m, Z \in S$:

- a category $\mathcal{C}(X_\bullet; Y)$ with objects *1-cells* $f : X_\bullet \to Y$ and morphisms *2-cells* $\alpha : f \Rightarrow g : X_\bullet \to Y$,
- distinguished *projection* functors $\mathsf{p}_i : \mathbb{1} \to \mathcal{C}(X_\bullet; X_i)$ for $1 \leqslant i \leqslant n$,
- a *substitution* functor

$$\mathsf{sub}_{X_\bullet;Y_\bullet;Z} : \mathcal{C}(Y_\bullet; Z) \times \prod_{j=1}^{m} \mathcal{C}(X_\bullet; Y_j) \to \mathcal{C}(X_\bullet; Z)$$

which we denote by

$$\mathsf{sub}_{X_\bullet;Y_\bullet;Z}\big(f, (g_1, \ldots, g_m)\big) := f[g_1, \ldots, g_m]$$

(we write $t[v_\bullet[w_\bullet]]$ or $t[v_1[w_\bullet], \ldots, v_n[w_\bullet]]$ for the iterated substitution $t[v_1[w_1, \ldots, w_l], \ldots, v_n[w_1, \ldots, w_l]]$, *c.f.* Notation III.1),
- natural families of invertible 2-cells

$$\mathsf{assoc}_{t,u_\bullet,v_\bullet} : t[u_1, \ldots, u_n][v_\bullet] \Rightarrow t[u_1[v_\bullet], \ldots, u_n[v_\bullet]]$$
$$\iota_u : u \Rightarrow u[\mathsf{p}_1, \ldots, \mathsf{p}_n]$$
$$\varrho^{(k)}_{u_1,\ldots,u_n} : \mathsf{p}_k[u_1, \ldots, u_n] \Rightarrow u_k \quad (k = 1, \ldots, n)$$

for every $t \in \mathcal{C}(Y_\bullet, Z)$, $u_j \in \mathcal{C}(X_\bullet, Y_j)$, $v_i \in \mathcal{C}(W_\bullet, X_i)$ and $u \in \mathcal{C}(X_\bullet, Y)$ ($i = 1, \ldots, n$ and $j = 1, \ldots, m$).

This data is subject to two compatibility laws:

$$\begin{array}{ccc} t[u_1, \ldots, u_n] & \xrightarrow{\iota[u_1,\ldots,u_n]} & t[\mathsf{p}_1, \ldots, \mathsf{p}_n][u_1, \ldots, u_n] \\ {\scriptstyle \mathrm{id}}\downarrow & & \downarrow{\scriptstyle \mathsf{assoc}} \\ t[u_1, \ldots, u_n] & \xleftarrow[t[\varrho^{(1)},\ldots,\varrho^{(n)}]]{} & t[\mathsf{p}_1[u_1, \ldots, u_n], \ldots, \mathsf{p}_n[u_1, \ldots, u_n]] \end{array}$$

$$\begin{array}{ccccc} t[u_\bullet][v_\bullet][w_\bullet] & \xrightarrow{\mathsf{assoc}[w_\bullet]} & t[u_\bullet[v_\bullet]][w_\bullet] & \xrightarrow{\mathsf{assoc}} & t[u_\bullet[v_\bullet][w_\bullet]] \\ {\scriptstyle \mathsf{assoc}}\downarrow & & & & \downarrow{\scriptstyle t[\mathsf{assoc},\ldots,\mathsf{assoc}]} \\ t[u_\bullet][v_\bullet[w_\bullet]] & & \xrightarrow[\mathsf{assoc}]{} & & t[u_\bullet[v_\bullet[w_\bullet]]] \end{array}$$

When the set S of sorts is clear we refer to an S-biclone as simply a *biclone*.

Thinking of a bicategory as roughly a 2-category with structure up to isomorphism, one may think of a biclone as roughly a **Cat**-enriched clone with structure up to isomorphism. Indeed, the definition of clone may be generalised to hold in any cartesian category (and even more generally, *e.g.* [46], [47]): if the structural 2-cells ι, ϱ and assoc are all the identity, a biclone is equivalently a *2-clone*, *i.e.* a clone in the cartesian category **Cat**. We have directed the 2-cells to match the definition of a *skew monoidal category* [48]; the definition should therefore generalise to the lax setting (*c.f.* also the *lax bicategories* of Leinster [49, §3.4]).

Every S-biclone $\mathcal{C}$ has an underlying *linear-core* bicategory $\mathcal{C}_{\ell c}$ with objects S and hom-categories $\mathcal{C}_{\ell c}(X, Y) = \mathcal{C}\big([X]; Y\big)$ (*c.f.* [50]). Every 2-multigraph freely induces a sorted biclone, and one may introduce bicategorical substitution structure into a type theory with base types, constant terms and constant rewrites specified by a 2-graph $\mathcal{G}$ by postulating the structure of the free sorted biclone on $\mathcal{G}$ and then restricting it to its underlying linear core. This is the gist of the following section.

$$\frac{}{x_1 : A_1, \dots, x_n : A_n \vdash x_k : A_k} \text{ var } (1 \leqslant k \leqslant n) \qquad \frac{\big(c \in \mathcal{G}(A_1, \dots, A_n; B)\big)}{x_1 : A_1, \dots, x_n : A_n \vdash c(x_1, \dots, x_n) : B} \text{ const}$$

$$\frac{x_1 : A_1, \dots, x_n : A_n \vdash t : B \qquad (\Delta \vdash u_i : A_i)_{i=1,\dots,n}}{\Delta \vdash t\{x_1 \mapsto u_1, \dots, x_n \mapsto u_n\} : B} \text{ horiz-comp}$$

Figure 1. Introduction rules on basic terms

$$\frac{x_1 : A_1, \dots, x_n : A_n \vdash t : B}{\begin{array}{c} x_1 : A_1, \dots, x_n : A_n \vdash \iota_t : t \Rightarrow t\{x_i \mapsto x_i\} : B \\ x_1 : A_1, \dots, x_n : A_n \vdash \iota_t^{-1} : t\{x_i \mapsto x_i\} \Rightarrow t : B \end{array}} \iota\text{-intro}$$

$$\frac{x_1 : A_1, \dots, x_n : A_n \vdash x_k : A_k \qquad (\Delta \vdash u_i : A_i)_{i=1,\dots,n}}{\begin{array}{c} \Delta \vdash \varrho^{(k)}_{u_1,\dots,u_n} : x_k\{x_i \mapsto u_i\} \Rightarrow u_k : A_k \\ \Delta \vdash \varrho^{(-k)}_{u_1,\dots,u_n} : u_k \Rightarrow x_k\{x_i \mapsto u_i\} : A_k \end{array}} \varrho^{(k)}\text{-intro } (1 \leqslant k \leqslant n)$$

$$\frac{y_1 : B_1, \dots, y_n : B_n \vdash t : C \qquad (x_1 : A_1, \dots, x_m : A_m \vdash v_i : B_i)_{i=1,\dots,n} \qquad (\Delta \vdash u_j : A_j)_{j=1,\dots m}}{\begin{array}{c} \Delta \vdash \mathsf{assoc}_{t,v_\bullet,u_\bullet} : t\{y_i \mapsto v_i\}\{x_j \mapsto u_j\} \Rightarrow t\{y_i \mapsto v_i\{x_j \mapsto u_j\}\} : C \\ \Delta \vdash \mathsf{assoc}^{-1}_{t,v_\bullet,u_\bullet} : t\{y_i \mapsto v_i\{x_j \mapsto u_j\}\} \Rightarrow t\{y_i \mapsto v_i\}\{x_j \mapsto u_j\} : C \end{array}} \text{assoc-intro}$$

Figure 2. Introduction rules on structural rewrites

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \mathrm{id}_t : t \Rightarrow t : A} \text{ id-intro} \qquad \frac{\big(\sigma \in \mathcal{G}(A_1, \dots, A_n; B)(c, c')\big)}{x_1 : A_1, \dots, x_n : A_n \vdash \sigma(x_1, \dots, x_n) : c(x_1, \dots, x_n) \Rightarrow c'(x_1, \dots, x_n) : B} \text{ 2-const}$$

Figure 3. Introduction rules on basic rewrites

$$\frac{\Gamma \vdash \tau' : t' \Rightarrow t'' : A \qquad \Gamma \vdash \tau : t \Rightarrow t' : A}{\Gamma \vdash \tau' \bullet \tau : t \Rightarrow t'' : A} \text{ vert-comp}$$

$$\frac{x_1 : A_1, \dots, x_n : A_n \vdash \tau : t \Rightarrow t' : B \qquad (\Delta \vdash \sigma_i : u_i \Rightarrow u_i' : A_i)_{i=1,\dots,n}}{\Delta \vdash \tau\{x_i \mapsto \sigma_i\} : t\{x_i \mapsto u_i\} \Rightarrow t'\{x_i \mapsto u_i'\} : B} \text{ horiz-comp}$$

Figure 4. Constructors on rewrites

Introduction rules for $\Lambda^{\mathrm{b}}_{\mathrm{ps}}$, $\Lambda^{\times}_{\mathrm{ps}}$ and $\Lambda^{\times,\to}_{\mathrm{ps}}$. For $\Lambda^{\mathrm{b}}_{\mathrm{ps}}$ these rules are restricted to *unary* contexts.

V. A TYPE THEORY FOR BICATEGORIES

We follow the tradition of 2-dimensional type theories consisting of types, terms and rewrites, *e.g.* [24], [25], [31], [29]. Following Hilken [25], we consider two forms of judgement. Alongside the usual $\Gamma \vdash t : A$ to indicate 'term t has type A in context Γ', we write $\Gamma \vdash \tau : t \Rightarrow t' : A$ to indicate 'τ is a rewrite from term t of type A to term t' of type A, in context Γ'.

For now we do not want to assume our model has products so we restrict to unary contexts. Base types, constants and rewrites are therefore specified by a 2-graph $\mathcal{G}$. The term introduction rules are collected in Figure 1 and the rewrite introduction rules in Figures 2–4. We denote the language thus defined by $\Lambda^{\mathrm{b}}_{\mathrm{ps}}(\mathcal{G})$.

The terms are variables x, $y, \dots$, constant terms $c(x)$, $c'(x), \dots$ and *explicit substitutions* $t\{x \mapsto u\}$. Thus for every term u and term t with free variable x we postulate a term $t\{x \mapsto u\}$; this is a formal analogue of the term $t[u/x]$ defined by the meta-operation of capture-avoiding substitution (*c.f.* [34], [35]). The variable x is bound by this operation, and we work with terms up to α-equivalence defined in the standard way. Instead of being typed in the empty context, constants are given by the edges of $\mathcal{G}$. The restriction to unary contexts ensures the syntactic model is a bicategory, rather than a biclone. When we construct the type theory for bicategories with finite products we shall allow constants and explicit substitution to be multi-ary.

The grammar for rewrites is synthesised from the free biclone on $\mathcal{G}$. The *structural rewrites* assoc, ι and ϱ witness the three laws of a biclone; we slightly abuse notation by simultaneously introducing these rewrites and their inverses. The *constant rewrites* $\sigma(x)$ are specified by the surfaces of $\mathcal{G}$ and for every term t we have an *identity rewrite* id_t. The explicit substitution operation mirrors that for terms while vertical composition is captured by a binary operation on rewrites (*c.f.* [25], [29]). We make use of the same conventions on binding and α-equivalence as for terms.

Notation V.1. We adopt the following abuses of notation.

1) Writing t for id_t in a rewrite.
2) Writing $t\{x_i \mapsto u_i\}_{i=1}^{n}$ or simply $t\{x_i \mapsto u_i\}$ for $t\{x_1 \mapsto u_1, \dots, x_n \mapsto u_n\}$, and similarly on rewrites.
3) Writing $c\{t_1, \dots, t_n\}$ for the explicit substitution $c(x_1, \dots, x_n)\{x_i \mapsto t_i\}$ for constants c, and similarly on rewrites.

The equational theory $\equiv$ on rewrites is derived directly from the axioms of a biclone; these are collected in Figures 5–8. In this extended abstract, the rules for the invertibility of the structural rewrites and the congruence rules on $\equiv$ are omitted.

The type theory $\Lambda_{\mathrm{ps}}^{\mathrm{b}}(\mathcal{G})$ satisfies the expected well-formedness properties (*c.f.* [51, Chapter 4] for STLC). Uniqueness of typing is obtained by adding type annotations on bound variables, constants and vertical compositions; we omit this extra information for readability.

A. The syntactic model for $\Lambda_{\mathrm{ps}}^{\mathrm{b}}$

We construct the syntactic model for $\Lambda_{\mathrm{ps}}^{\mathrm{b}}(\mathcal{G})$ and prove that it enjoys a 2-dimensional freeness universal property analogous to that of [38, §2.2.1]. Put colloquially, $\Lambda_{\mathrm{ps}}^{\mathrm{b}}(\mathcal{G})$ is the internal language for bicategories with signature a 2-graph $\mathcal{G}$. The bicategorical structure is induced directly from the biclone structure.

Construction V.2. *For any 2-graph $\mathcal{G}$, define a bicategory $\mathcal{T}_{\mathrm{ps}}^{\mathrm{b}}(\mathcal{G})$ as follows. Objects are unary contexts $(x : A)$. The hom-category $\mathcal{T}_{\mathrm{ps}}^{\mathrm{b}}(\mathcal{G})\big((x : A), (y : B)\big)$ has objects α-equivalence classes of derivable terms $(x : A \vdash t : B)$ and morphisms $\alpha{\equiv}$-equivalence classes of rewrites $(x : A \vdash \tau : t \Rightarrow t' : B)$ under vertical composition. Horizontal composition is given by explicit substitution and the identity on $(x : A)$ by the* `var` *rule $(x : A \vdash x : A)$. The structural isomorphisms $\mathbf{l}, \mathbf{r}$ and $\mathbf{a}$ are ϱ, ι^{-1} and* assoc*, respectively.*

Construction V.3. *For any 2-graph $\mathcal{G}$, bicategory $\mathcal{B}$ and 2-graph homomorphism $h : \mathcal{G} \to \mathcal{B}$, the semantics of Figure 13 restricted to $\mathcal{T}_{\mathrm{ps}}^{\mathrm{b}}(\mathcal{G})$ induces a strict pseudofunctor $h^{\#} : \mathcal{T}_{\mathrm{ps}}^{\mathrm{b}}(\mathcal{G}) \to \mathcal{B}$.*

The universal *extension* pseudofunctor $h^{\#}$ cannot be characterised by a strict universal property; for instance, for each type A the bicategory $\mathcal{T}_{\mathrm{ps}}^{\mathrm{b}}(\mathcal{G})$ contains countably many equivalent objects $(x : A)$ for x ranging over variables. To obtain the desired universal property, we restrict to a sub-bicategory in which there is a single variable name.

Construction V.4. *Let $\mathcal{S}_{\mathrm{ps}}^{\mathrm{b}}(\mathcal{G})$ denote the bicategory with objects unary contexts $(x : A)$ for x a* fixed *variable and horizontal and vertical composition operations as in $\mathcal{T}_{\mathrm{ps}}^{\mathrm{b}}(\mathcal{G})$.*

The bicategories $\mathcal{S}_{\mathrm{ps}}^{\mathrm{b}}(\mathcal{G})$ and $\mathcal{T}_{\mathrm{ps}}^{\mathrm{b}}(\mathcal{G})$ are biequivalent, and the restricted model $\mathcal{S}_{\mathrm{ps}}^{\mathrm{b}}(\mathcal{G})$ is free on $\mathcal{G}$ in the following sense.

Theorem V.5. *Let $\mathcal{G}$ be a 2-graph. For any bicategory $\mathcal{B}$ and 2-graph homomorphism $h : \mathcal{G} \to \mathcal{B}$, there exists a unique strict pseudofunctor $h^{\#} : \mathcal{S}_{\mathrm{ps}}^{\mathrm{b}}(\mathcal{G}) \to \mathcal{B}$ such that $h^{\#} \circ \iota = h$, for $\iota : \mathcal{G} \hookrightarrow \mathcal{S}_{\mathrm{ps}}^{\mathrm{b}}(\mathcal{G})$ the inclusion.*

The full model $\mathcal{T}_{\mathrm{ps}}^{\mathrm{b}}(\mathcal{G})$ therefore satisfies the universal property up to biequivalence. Proving the freeness universal property in this way has the benefit of allowing us to establish uniqueness without reasoning about uniqueness of pseudonatural transformations or modifications. It follows that $\Lambda_{\mathrm{ps}}^{\mathrm{b}}$ is an internal language for bicategories. By Example II.3, restricting to a single base type gives rise to an internal language for monoidal categories.

The argument in this section applies with only minor adjustments to biclones. Thus, allowing $\Lambda_{\mathrm{ps}}^{\mathrm{b}}(\mathcal{G})$ to have contexts with fixed variables over a 2-multigraph $\mathcal{G}$ gives rise to the free biclone on $\mathcal{G}$ in the sense of Theorem V.5. This relies on a straightforward generalisation of the notions of pseudofunctor, transformation and modification. Details will be given elsewhere.

VI. FP-BICATEGORIES

We recall the notion of bicategory with finite products, defined as a bilimit [12]. To avoid confusion with the 'cartesian bicategories' of Carboni and Walters [52], [53], we use the term *fp-bicategories*. fp-Bicategories are the objects of a tricategory $\mathbf{fp\text{-}Bicat}$.

It is convenient to work directly with n-ary products ($n \in \mathbb{N}$). This reduces the need to deal with the equivalent objects given by rebracketing binary products. For bicategories $\mathcal{B}_1, \dots, \mathcal{B}_n$ the *product bicategory* $\prod_{i=1}^{n} \mathcal{B}_i$ has objects $(B_1, \dots, B_n) \in \prod_{i=1}^{n} ob(\mathcal{B}_i)$ and structure given pointwise. An *fp-bicategory* is therefore a bicategory $\mathcal{B}$ equipped with a right biadjoint to the diagonal pseudofunctor $\Delta_n : \mathcal{B} \to \mathcal{B}^{\times n} : B \mapsto (B, \dots, B)$ for each $n \in \mathbb{N}$. This unwinds to the following definition.

Definition VI.1. An *fp-bicategory* is a bicategory $\mathcal{B}$ equipped with the following data for every $n \in \mathbb{N}$ and $A_1, \dots, A_n \in \mathcal{B}$:

1) a *product* object $\prod_n(A_1, \dots, A_n)$,
2) *projection* 1-cells $\pi_k : \prod_n(A_1, \dots, A_n) \to A_k$ for $1 \leqslant k \leqslant n$,
3) for every $X \in \mathcal{B}$ an adjoint equivalence

$$\mathcal{B}\big(X, \textstyle\prod_n(A_1, \dots, A_n)\big) \underset{\langle -,\dots,= \rangle}{\overset{(\pi_1 \circ -, \dots, \pi_n \circ -)}{\rightleftarrows}} \perp \textstyle\prod_{i=1}^{n} \mathcal{B}(X, A_i) \qquad (1)$$

We call the right adjoint $\langle -, \dots, = \rangle$ the *n-ary tupling*.

Remark VI.2. One obtains a *lax* n-ary product structure by merely asking for an adjunction in diagram (1).

Example VI.3. Every small fp-bicategory $\big(\mathcal{B}, \Pi_n(-)\big)$ defines an $ob(\mathcal{B})$-biclone $\mathcal{B}_{c\ell}$ by setting $\mathcal{B}_{c\ell}([X_1, \dots, X_n]; Y)$ to be $\mathcal{B}\big(\prod_n(X_1, \dots, X_n), Y\big)$.

Notation VI.4.

1) We write $A_1 \times \cdots \times A_n$ or $\prod_{i=1}^{n} A_i$ for $\prod_n(A_1, \dots, A_n)$ and denote the terminal object $\prod_0()$ by $\mathbf{1}$.

$$\frac{\Gamma \vdash \tau : t \Rightarrow t' : A}{\Gamma \vdash \tau \bullet \mathrm{id}_t \equiv \tau : t \Rightarrow t' : A}\ \bullet\text{-right-unit} \qquad \frac{\Gamma \vdash \tau : t \Rightarrow t' : A}{\Gamma \vdash \tau \equiv \mathrm{id}_{t'} \bullet \tau : t \Rightarrow t' : A}\ \bullet\text{-left-unit}$$

$$\frac{\Gamma \vdash \tau'' : t'' \Rightarrow t''' : A \qquad \Gamma \vdash \tau' : t' \Rightarrow t'' : A \qquad \Gamma \vdash \tau : t \Rightarrow t' : A}{\Gamma \vdash (\tau'' \bullet \tau') \bullet \tau \equiv \tau'' \bullet (\tau' \bullet \tau) : t \Rightarrow t''' : A}\ \bullet\text{-assoc}$$

Figure 5. Categorical structure of vertical composition

$$\frac{x_1 : A_1, \dots, x_n : A_n \vdash t : B \qquad (\Delta \vdash u_i : A_i)_{i=1,\dots,n}}{\Delta \vdash \mathrm{id}_t\{x_i \mapsto u_i\} \equiv \mathrm{id}_{t\{x_i \mapsto u_i\}} : t\{x_i \mapsto u_i\} \Rightarrow t\{x_i \mapsto u_i\} : B}\ \text{id-preservation}$$

$$\frac{\begin{array}{cc} x_1 : A_1, \dots, x_n : A_n \vdash \tau : t \Rightarrow t' : B & (\Delta \vdash \sigma_i : u_i \Rightarrow u_i' : A_i)_{i=1,\dots,n} \\ x_1 : A_1, \dots, x_n : A_n \vdash \tau' : t' \Rightarrow t'' : B & (\Delta \vdash \sigma_i' : u_i' \Rightarrow u_i'' : A_i)_{i=1,\dots,n} \end{array}}{\Delta \vdash \tau'\{x_i \mapsto \sigma_i'\} \bullet \tau\{x_i \mapsto \sigma_i\} \equiv (\tau' \bullet \tau)\{x_i \mapsto \sigma_i' \bullet \sigma_i\} : t\{x_i \mapsto u_i\} \Rightarrow t''\{x_i \mapsto u_i''\} : B}\ \text{interchange}$$

Figure 6. Preservation rules

$$\frac{(\Delta \vdash \sigma_i : u_i \Rightarrow u_i' : A_i)_{i=1,\dots,n}}{\Delta \vdash \varrho^{(k)}_{u_1',\dots,u_n'} \bullet x_k\{x_i \mapsto \sigma_i\} \equiv \sigma_k \bullet \varrho^{(k)}_{u_1,\dots,u_n} : x_k\{x_i \mapsto u_i\} \Rightarrow u_k' : A_k}\ (1 \leqslant k \leqslant n)$$

$$\frac{x_1 : A_1, \dots, x_n : A_n \vdash \tau : t \Rightarrow t' : B}{x_1 : A_1, \dots, x_n : A_n \vdash \iota_{t'} \bullet \tau \equiv \tau\{x_i \mapsto x_i\} \bullet \iota_t : t \Rightarrow t'\{x_i \mapsto x_i\} : B}$$

$$\frac{y_1 : B_1, \dots, y_n : B_n \vdash \tau : t \Rightarrow t' : C \qquad (x_1 : A_1, \dots, x_m : A_m \vdash \sigma_i : v_i \Rightarrow v_i' : B_i)_{i=1,\dots,n} \qquad (\Delta \vdash \mu_j : u_j \Rightarrow u_j' : A_j)_{j=1,\dots m}}{\Delta \vdash \mathsf{assoc}_{t',v_\bullet,u_\bullet} \bullet \tau\{y_i \mapsto \sigma_i\}\{x_j \mapsto \mu_j\} \equiv \tau\{y_i \mapsto \sigma_i\{x_j \mapsto \mu_j\}\} \bullet \mathsf{assoc}_{t,v_\bullet,u_\bullet} : t\{y_i \mapsto v_i\}\{x_j \mapsto u_j\} \Rightarrow t'\{y_i \mapsto v_i'\{x_j \mapsto u_j'\}\} : C}$$

Figure 7. Naturality rules on structural rewrites

$$\frac{x_1 : A_1, \dots, x_n : A_n \vdash t : B \qquad (\Delta \vdash u_i : A_i)_{i=1,\dots,n}}{\Delta \vdash t\{x_i \mapsto \varrho^{(i)}_{u_\bullet}\} \bullet \mathsf{assoc}_{t,x_\bullet,u_\bullet} \bullet \iota_t\{x_i \mapsto u_i\} \equiv \mathrm{id}_{t\{x_i \mapsto u_i\}} : t\{x_i \mapsto u_i\} \Rightarrow t\{x_i \mapsto u_i\} : B}$$

$$\frac{\begin{array}{cc} z_1 : C_1, \dots, z_l : C_l \vdash t : D & (x_1 : A_1, \dots, x_m : A_m \vdash v_i : B_i)_{i=1,\dots,n} \\ (y_1 : B_1, \dots, y_n : B_n \vdash w_j : C_k)_{k=1,\dots,l} & (\Delta \vdash u_j : A_j)_{j=1,\dots m} \end{array}}{\begin{array}{r} \Delta \vdash t\{z_k \mapsto \mathsf{assoc}_{w_k,v_\bullet,u_\bullet}\} \bullet \mathsf{assoc}_{t,w_\bullet\{y_j \mapsto v_j\},u_\bullet} \bullet \mathsf{assoc}_{t,w_\bullet,v_\bullet}\{x_j \mapsto u_j\} \equiv \mathsf{assoc}_{t,w_\bullet,v_\bullet\{x_j \mapsto u_i\}} \bullet \mathsf{assoc}_{t\{z_k \mapsto w_k\},v_\bullet,u_\bullet} \\ : t\{z_k \mapsto w_k\}\{y_i \mapsto v_i\}\{x_j \mapsto u_j\} \Rightarrow t\{z_k \mapsto w_k\{y_i \mapsto v_i\{x_j \mapsto u_j\}\}\} : D \end{array}}$$

Figure 8. Biclone laws

Equational theory for structural rewrites in $\Lambda^{\mathrm{b}}_{\mathrm{ps}}$, $\Lambda^{\times}_{\mathrm{ps}}$ and $\Lambda^{\times,\to}_{\mathrm{ps}}$. For $\Lambda^{\mathrm{b}}_{\mathrm{ps}}$ the rules are restricted to unary contexts.

2) We write $\langle f_i \rangle_{i=1,\dots,n}$ or simply $\langle f_\bullet \rangle$ for the n-ary tupling $\langle f_1, \dots, f_n \rangle$.

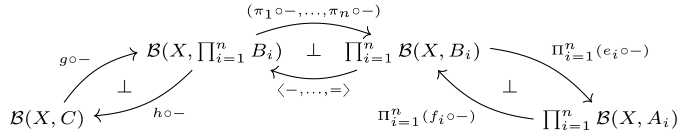
There are different ways of specifying the adjoint equivalence (1) (see *e.g.* [41, Chapter IV]). One option is to specify an invertible unit and counit subject to naturality and triangle laws. This matches the η-expansion and β-reduction rules of STLC (*c.f.* [24], [26], [27], [29]), but in the pseudo or lax settings requires a cumbersome proliferation of introduction rules. Instead, we characterise the counit $\varpi = (\varpi^{(1)}, \dots, \varpi^{(n)})$ as a universal arrow. Thus, for any finite family of 1-cells $(t_i : X \to A_i)_{i=1,\dots,n}$ we require a 1-cell $\langle t_1, \dots, t_n \rangle : X \to \prod_n(A_1, \dots, A_n)$ and a family of 2-cells $(\varpi^{(k)}_{t_1,\dots,t_n} : \pi_k \circ \langle t_\bullet \rangle \Rightarrow t_k)_{k=1,\dots,n}$, universal in the sense that for any family of 2-cells $(\alpha_i : \pi_i \circ u \Rightarrow t_i : \Gamma \to A_i)_{i=1,\dots,n}$ there exists a unique 2-cell $\mathsf{p}^\dagger(\alpha_1, \dots, \alpha_n) : u \Rightarrow \langle t_\bullet \rangle : \Gamma \to \prod_{i=1}^n A_i$ such that

$$\varpi^{(k)}_{t_1,\dots,t_n} \bullet \big(\pi_k \circ \mathsf{p}^\dagger(\alpha_1, \dots, \alpha_n)\big) = \alpha_k : \pi_k \circ u \Rightarrow t_k$$

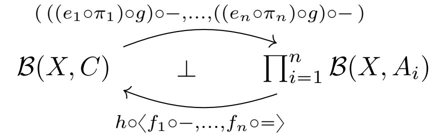
for $k = 1, \dots, n$.

Example VI.5. In any bicategory, unary-product structure may be chosen to be canonically given as follows: $\Pi_1(A) = A$, $\pi_1^A = \mathrm{Id}_A$, $\langle t \rangle = t$ and $\varpi_t = \mathfrak{l}_t : \mathrm{Id} \circ t \Rightarrow t$.

Remark VI.6. As it is well-known, product structure is unique up to equivalence. Given adjoint equivalences $(g : C \leftrightarrows \Pi_{i=1}^n B_i : h)$ and $(e_i : B_i \leftrightarrows A_i : f_i)_{i=1,\dots,n}$ in a bicategory $\mathcal{B}$, the composite



yields an adjoint equivalence



presenting C as the product of $A_1,\ldots,A_n$.

Definition VI.7. An *fp-pseudofunctor* $(F,k^{\times})$ between fp-bicategories $(\mathcal{B},\Pi_n(-))$ and $(\mathcal{C},\Pi_n(-))$ is a pseudofunctor $F:\mathcal{B}\to\mathcal{C}$ equipped with adjoint equivalences

$$\big(F\big(\textstyle\prod_{i=1}^n A_i\big),\prod_{i=1}^n F(A_i),\langle F\pi_1,\ldots,F\pi_n\rangle,k^{\times}_{A_1,\ldots,A_n}\big)$$

for every $n\in\mathbb{N}$ and $A_1,\ldots,A_n\in\mathcal{B}$.

We call $(F,k^{\times})$ *strict* if F is strict and satisfies

$$\begin{aligned}
F\big(\Pi_n(A_1,\ldots,A_n)\big) &= \Pi_n(FA_1,\ldots,FA_n)\\
F(\pi_i^{A_1,\ldots,A_n}) &= \pi_i^{FA_1,\ldots,FA_n}\\
F\langle t_1,\ldots,t_n\rangle &= \langle Ft_1,\ldots,Ft_n\rangle\\
F\varpi^{(i)}_{t_1,\ldots,t_n} &= \varpi^{(i)}_{Ft_1,\ldots,Ft_n}\\
k^{\times}_{A_1,\ldots,A_n} &= \mathrm{Id}_{\Pi_n(FA_1,\ldots,FA_n)}
\end{aligned}$$

and the adjoint equivalences are canonically induced by the 2-cells $\mathsf{p}^{\dagger}(\mathbf{r}_{\pi_1},\ldots,\mathbf{r}_{\pi_n}):\mathrm{Id}\overset{\cong}{\Rightarrow}\langle\pi_1,\ldots,\pi_n\rangle$.

Thus, a strict fp-pseudofunctor strictly preserves both (global) biuniversal arrows and (local) universal arrows.

VII. A type theory for fp-bicategories

We extend the basic language $\Lambda^{\mathrm{b}}_{\mathrm{ps}}$ to a type theory $\Lambda^{\times}_{\mathrm{ps}}$ with finite products. The addition of products enables us to use arbitrary contexts, defined as finite lists of variable-and-type pairs in which variable names must not be repeated. The underlying signature is therefore a 2-multigraph and the bicategorical structure of $\Lambda^{\mathrm{b}}_{\mathrm{ps}}$ is extended to a biclone.

The additional structure required for products is synthesised directly from the cases in Definition VI.1. These additional rules are collected in Figures 9–11. As for $\Lambda^{\mathrm{b}}_{\mathrm{ps}}$, we work up to α-equivalence of terms and rewrites. The well-formedness properties of $\Lambda^{\mathrm{b}}_{\mathrm{ps}}$ extend to $\Lambda^{\times}_{\mathrm{ps}}$.

For every $n\in\mathbb{N}$ and types $A_1,\ldots,A_n$ we introduce a product type $\prod_n(A_1,\ldots,A_n)$; the case $n=0$ yields the unit type $\mathbf{1}$. We fix a set of base types S and let the set of types $\mathrm{T}_0(S)$ be generated by the grammar

$$\begin{aligned}
A_1,\ldots,A_n \quad ::= &\quad X\in S\\
\mid &\quad \textstyle\prod_n(A_1,\ldots,A_n) \qquad (n\in\mathbb{N})
\end{aligned}$$

On top of this, we fix a 2-multigraph $\mathcal{G}$ with $\mathcal{G}_0=\mathrm{T}_0(S)$.

We generally abuse notation by adopting the conventions of Notation VI.4(1). For the biuniversal arrows we introduce distinguished constants $\pi_k(p):A_k$ $(k=1,\ldots,n)$ for every context $\big(p:\prod_n(A_1,\ldots,A_n)\big)$. As for adjunction (1), we postulate an operator $\mathsf{tup}(-,\ldots,=)$, a family of rewrites

$$\varpi_{t_1,\ldots,t_n}=\big(\varpi^{(k)}_{t_1,\ldots,t_n}:\pi_k\{\mathsf{tup}(t_1,\ldots,t_n)\}\Rightarrow t_k\big)_{k=1,\ldots,n}$$

(recall Notation V.1(3)) for every family of derivable terms $(\Gamma\vdash t_i:A_i)_{i=1,\ldots,n}$, and provide a natural bijective correspondence between rewrites as follows:

$$\frac{\alpha_i:\pi_i\{u\}\Rightarrow t_i\quad(i=1,\ldots,n)}{\mathsf{p}^{\dagger}(\alpha_1,\ldots,\alpha_n):u\Rightarrow\mathsf{tup}(t_1,\ldots,t_n)}\tag{2}$$

The rules of Figure 11 achieve this by making ϖ universal: this is precisely the content of Lemma VII.2 below. These rules define *lax* products. To obtain the adjoint equivalence (1) one introduces explicit inverses for the unit and counit. In this extended abstract these are omitted.

Remark VII.1. The product structure arises from two *nested* universal transposition structures. We conjecture that a calculus for fp-*tri*categories (resp. fp-∞-categories) would have *three* (resp. a countably infinite tower of) such correspondences.

A. The product structure of $\Lambda^{\times}_{\mathrm{ps}}$

The product structure derives from the following universal property of ϖ.

Lemma VII.2. *If the judgements* $(\Gamma\vdash\alpha_i:\pi_i\{u\}\Rightarrow t_i:A_i)_{i=1,\ldots,n}$ *are derivable in* $\Lambda^{\times}_{\mathrm{ps}}(\mathcal{G})$ *then* $\mathsf{p}^{\dagger}(\alpha_1,\ldots,\alpha_n):u\Rightarrow\mathsf{tup}(t_1,\ldots,t_n)$ *is the unique rewrite* γ *(modulo* $\alpha\equiv$*-equivalence) such that the following equality is derivable for* $k=1,\ldots,n$*:*

$$\Gamma\vdash\varpi^{(k)}_{t_1,\ldots,t_n}\bullet\pi_k\{\gamma\}\equiv\alpha_k:\pi_i\{u\}\Rightarrow t_k:A_k$$

From this lemma it follows that the $\mathsf{tup}(-,\ldots,=)$ operator extends to a functor on rewrites, and that one may define the unit of adjunction (1) by applying the universal property to the identity.

Definition VII.3.

1) For every family of derivable rewrites $(\Gamma\vdash\tau_i:t_i\Rightarrow t'_i:A_i)_{i=1,\ldots,n}$ define $\mathsf{tup}(\tau_1,\ldots,\tau_n):\mathsf{tup}(t_1,\ldots,t_n)\Rightarrow\mathsf{tup}(t'_1,\ldots,t'_n)$ to be the rewrite $\mathsf{p}^{\dagger}(\tau_1\bullet\varpi^{(1)}_{t_1,\ldots,t_n},\ldots,\tau_n\bullet\varpi^{(n)}_{t_1,\ldots,t_n})$ in context Γ.
2) For every derivable term $\Gamma\vdash t:\prod_n(A_1,\ldots,A_n)$ define the unit $\varsigma_t:t\Rightarrow\mathsf{tup}(\pi_1\{t\},\ldots,\pi_n\{t\})$ to be the rewrite $\mathsf{p}^{\dagger}(\mathrm{id}_{\pi_1\{t\}},\ldots,\mathrm{id}_{\pi_n\{t\}})$ in context Γ.

Likewise, one obtains naturality and the triangle laws relating the unit ς and counit $\varpi=(\varpi^{(1)},\ldots,\varpi^{(n)})$, thereby recovering a presentation of products in the style of [24], [25], [29]. These admissible rules are collected in Figure 12.

B. The syntactic model for $\Lambda^{\times}_{\mathrm{ps}}$

We construct the syntactic model for $\Lambda^{\times}_{\mathrm{ps}}$ and prove the freeness universal property establishing that it is the internal language of fp-bicategories. We begin with a model in which we allow arbitrary variables and consider all contexts. We then

$$\frac{\Gamma \vdash t_1 : A_1 \quad \ldots \quad \Gamma \vdash t_n : A_n}{\Gamma \vdash \mathsf{tup}(t_1, \ldots, t_n) : \prod_n(A_1, \ldots, A_n)}\; n\text{-pair} \qquad \frac{}{p : \prod_n(A_1, \ldots, A_n) \vdash \pi_k(p) : A_k}\; k\text{-proj } (1 \leqslant k \leqslant n)$$

Figure 9. Terms for product structure

$$\frac{\Gamma \vdash t_1 : A_1 \quad \ldots \quad \Gamma \vdash t_n : A_n}{\Gamma \vdash \varpi^{(k)}_{t_1,\ldots,t_n} : \pi_k\{\mathsf{tup}(t_1, \ldots, t_n)\} \Rightarrow t_k : A_k}\; \varpi^{(k)}\text{-intro } (1 \leqslant k \leqslant n)$$

$$\frac{\Gamma \vdash u : \prod_n(A_1, \ldots, A_n) \qquad \Gamma \vdash \alpha_1 : \pi_1\{u\} \Rightarrow t_1 : A_1 \quad \ldots \quad \Gamma \vdash \alpha_n : \pi_n\{u\} \Rightarrow t_n : A_n}{\Gamma \vdash \mathsf{p}^{\dagger}(\alpha_1, \ldots, \alpha_n) : u \Rightarrow \mathsf{tup}(t_1, \ldots, t_n) : \prod_n(A_1, \ldots, A_n)}\; \mathsf{p}^{\dagger}\text{-intro}$$

Figure 10. Rewrites for product structure

$$\frac{\Gamma \vdash \alpha_1 : \pi_1\{u\} \Rightarrow t_1 : A_1 \quad \ldots \quad \Gamma \vdash \alpha_n : \pi_n\{u\} \Rightarrow t_n : A_n}{\Gamma \vdash \alpha_k \equiv \varpi^{(k)}_{t_1,\ldots,t_n} \bullet \pi_k\{\mathsf{p}^{\dagger}(\alpha_1, \ldots, \alpha_n)\} : \pi_k\{u\} \Rightarrow t_k : A_k}\; \text{U1 } (1 \leqslant k \leqslant n)$$

$$\frac{\Gamma \vdash \gamma : u \Rightarrow \mathsf{tup}(t_1, \ldots, t_n) : \prod_n(A_1, \ldots, A_n)}{\Gamma \vdash \gamma \equiv \mathsf{p}^{\dagger}(\varpi^{(1)}_{t_1,\ldots,t_n} \bullet \pi_1\{\gamma\}, \ldots, \varpi^{(n)}_{t_1,\ldots,t_n} \bullet \pi_n\{\gamma\}) : u \Rightarrow \mathsf{tup}(t_1, \ldots, t_n) : \prod_n(A_1, \ldots, A_n)}\; \text{U2}$$

$$\frac{\big(\Gamma \vdash \alpha_i \equiv \alpha'_i : \pi_i\{u\} \Rightarrow t_i : A_i\big)_{i=1,\ldots,n}}{\Gamma \vdash \mathsf{p}^{\dagger}(\alpha_1, \ldots, \alpha_n) \equiv \mathsf{p}^{\dagger}(\alpha'_1, \ldots, \alpha'_n) : u \Rightarrow \mathsf{tup}(t_1, \ldots, t_n) : \prod_n(A_1, \ldots, A_n)}\; \text{cong}$$

Figure 11. Universal property of ϖ

Rules for $\Lambda^{\times}_{\mathrm{ps}}(\mathcal{G})$.

restrict to unary contexts and a single named variable (*c.f.* Constructions V.2 and V.4) in order to obtain a strict universal property (*c.f.* Theorem V.5).

Construction VII.4. *Define a bicategory $\mathcal{T}^{\times}_{\mathrm{ps}}(\mathcal{G})$ as follows. The objects are contexts $\Gamma, \Delta, \ldots$. The 1-cells $\Gamma \to (y_j : B_j)_{j=1,\ldots,m}$ are m-tuples of α-equivalence classes of terms $(\Gamma \vdash t_j : B_j)_{j=1,\ldots,m}$ derivable in $\Lambda^{\times}_{\mathrm{ps}}(\mathcal{G})$; the 2-cells are m-tuples of $\alpha{\equiv}$-equivalence classes of rewrites $(\Gamma \vdash \tau : t_j \Rightarrow t'_j : B_j)_{j=1,\ldots,m}$.*

Vertical composition is given pointwise by the $\bullet$ operation, and horizontal composition by explicit substitution:

$$(t_1, \ldots, t_l), (u_1, \ldots, u_m) \mapsto (t_1\{x_i \mapsto u_i\}, \ldots, t_l\{x_i \mapsto u_i\})$$

$$(\tau_1, \ldots, \tau_l), (\sigma_1, \ldots, \sigma_m) \mapsto (\tau_1\{x_i \mapsto \sigma_i\}, \ldots, \tau_l\{x_i \mapsto \sigma_i\})$$

The identity on $\Delta = (y_j : B_j)_{j=1,\ldots,m}$ is given by the $\mathtt{var}$ *rule $(\Delta \vdash y_j : B_j)_{j=1,\ldots,m}$. The structural isomorphisms* **l**, **r** *and* **a** *are given pointwise by ϱ, ι^{-1} and* assoc*, respectively.*

The first step in showing that $\mathcal{T}^{\times}_{\mathrm{ps}}(\mathcal{G})$ is an fp-bicategory is constructing n-ary products of unary contexts. Much of the work required is contained in the admissible rules of Figure 12. For example, there is an adjoint equivalence $\mathcal{T}^{\times}_{\mathrm{ps}}(\mathcal{G})\big((x : X), (p : \prod_n(A_1, \ldots, A_n))\big) \simeq \prod_{i=1}^n \mathcal{T}^{\times}_{\mathrm{ps}}(\mathcal{G})\big((x : X), (x_i : A_i)\big)$ defining products of unary contexts in $\mathcal{T}^{\times}_{\mathrm{ps}}(\mathcal{G})$ with unit ς and counit $\varpi = (\varpi^{(1)}, \ldots, \varpi^{(n)})$.

Lemma VII.5. *For any 2-multigraph $\mathcal{G}$, the following holds in $\mathcal{T}^{\times}_{\mathrm{ps}}(\mathcal{G})$.*

1) *For any $n \in \mathbb{N}$ the n-ary product $\prod_{i=1}^n (x_i : A_i)$ exists and is given by $(p : \prod_{i=1}^n A_i)$.*
2) *For any context $\Gamma = (x_1 : A_1, \ldots, x_n : A_n)$ there exists an adjoint equivalence $\Gamma \leftrightarrows \big(p : \prod_n(A_1, \ldots, A_n)\big)$.*

Remark VII.6. The existence of the adjoint equivalence of Lemma VII.5(2) in $\mathcal{T}^{\times}_{\mathrm{ps}}(\mathcal{G})$ is equivalent to the existence of a *pseudo-universal* multimap $[A_1, \ldots, A_n] \to \prod_n(A_1, \ldots, A_n)$ in the biclone associated to $\Lambda^{\times}_{\mathrm{ps}}(\mathcal{G})$ (*c.f.* [50]).

We define products of arbitrary contexts using Remark VI.6 and the preceding lemma. Define the product of

$$(x_i^{(1)} : A_i^{(1)})_{i=1,\ldots,m_1} \times \cdots \times (x_i^{(n)} : A_i^{(n)})_{i=1,\ldots,m_n}$$

to be the product $(p_1 : \prod_{i=1}^{m_1} A_i^{(1)}) \times \cdots \times (p_n : \prod_{i=1}^{m_n} A_i^{(n)})$ of unary contexts.

Corollary VII.7. *For any 2-multigraph $\mathcal{G}$, the syntactic model $\mathcal{T}^{\times}_{\mathrm{ps}}(\mathcal{G})$ of $\Lambda^{\times}_{\mathrm{ps}}(\mathcal{G})$ is an fp-bicategory.*

$\mathcal{T}^{\times}_{\mathrm{ps}}(\mathcal{G})$ contains redundancy in two ways: as well as the equivalent objects given by bijectively renaming variables in contexts, every context $(x_i : A_i)_{i=1,\ldots,n}$ is equivalent to the context $(p : \prod_{i=1}^n A_i)$. To obtain a strict universal property we cut out such multiplicities (*c.f.* Construction V.4).

Construction VII.8. *Let $\mathcal{S}^{\times}_{\mathrm{ps}}(\mathcal{G})$ denote the bicategory with objects unary contexts $(x : A)$ for x a* fixed *variable and horizontal and vertical composition operations as in $\mathcal{T}^{\times}_{\mathrm{ps}}(\mathcal{G})$.*

The product structure of $\mathcal{T}^{\times}_{\mathrm{ps}}(\mathcal{G})$ restricts to $\mathcal{S}^{\times}_{\mathrm{ps}}(\mathcal{G})$ and the two bicategories are equivalent as fp-bicategories. To obtain

$$\frac{(\Gamma \vdash \mathrm{id}_{t_i} : t_i \Rightarrow t_i : A_i)_{i=1,\dots,n}}{\Gamma \vdash \mathsf{tup}(\mathrm{id}_{t_1}, \dots, \mathrm{id}_{t_n}) \equiv \mathrm{id}_{\mathsf{tup}(t_1,\dots,t_n)} : \mathsf{tup}(t_1, \dots, t_n) \Rightarrow \mathsf{tup}(t_1, \dots, t_n) : \prod_n(A_1, \dots, A_n)}$$

$$\frac{(\Gamma \vdash \tau'_i : t'_i \Rightarrow t''_i : A_i)_{i=1,\dots,n} \qquad (\Gamma \vdash \tau_i : t_i \Rightarrow t'_i : A_i)_{i=1,\dots,n}}{\Gamma \vdash \mathsf{tup}(\tau'_1, \dots, \tau'_n) \bullet \mathsf{tup}(\tau_1, \dots, \tau_n) \equiv \mathsf{tup}(\tau'_1 \bullet \tau_1, \dots, \tau'_n \bullet \tau_n) : \mathsf{tup}(t_1, \dots, t_n) \Rightarrow \mathsf{tup}(t''_1, \dots, t''_n) : \prod_n(A_1, \dots, A_n)}$$

$$\frac{\Gamma \vdash \sigma : u \Rightarrow u' : \prod_n(A_1, \dots, A_n)}{\Gamma \vdash \varsigma_{u'} \bullet \sigma \equiv \mathsf{tup}(\pi_1\{\sigma\}, \dots, \pi_n\{\sigma\}) \bullet \varsigma_u : u \Rightarrow \mathsf{tup}(\pi_1\{u'\}, \dots, \pi_n\{u'\}) : \prod_n(A_1, \dots, A_n)}\ \varsigma\text{-nat}$$

$$\frac{(\Gamma \vdash \tau_i : t_i \Rightarrow t'_i : A_i)_{i=1,\dots,n}}{\Gamma \vdash \varpi^{(k)}_{t'_1,\dots,t'_n} \bullet \pi_k\{\mathsf{tup}(\tau_1, \dots, \tau_n)\} \equiv \tau_k \bullet \varpi^{(k)}_{t_1,\dots,t_n} : \pi_k\{\mathsf{tup}(t_1, \dots, t_n)\} \Rightarrow t_k : A_k}\ \varpi^{(k)}\text{-nat } (1 \leqslant k \leqslant n)$$

$$\frac{\Gamma \vdash \mathsf{tup}(t_1, \dots, t_n) : \prod_n(A_1, \dots, A_n)}{\Gamma \vdash \mathsf{tup}(\varpi^{(1)}_{t_\bullet}, \dots, \varpi^{(n)}_{t_\bullet}) \bullet \varsigma_{\mathsf{tup}(t_\bullet)} \equiv \mathrm{id}_{\mathsf{tup}(t_\bullet)} : \mathsf{tup}(t_\bullet) \Rightarrow \mathsf{tup}(t_\bullet) : \prod_n(A_1, \dots, A_n)}\ \text{triangle-law-1}$$

$$\frac{\Gamma \vdash \pi_k\{u\} : A_k}{\Gamma \vdash \varpi^{(k)}_{t_1,\dots,t_n} \bullet \pi_k\{\varsigma_u\} \equiv \mathrm{id}_{\pi_k\{u\}} : \pi_k\{u\} \Rightarrow \pi_k\{u\} : A_k}\ \text{triangle-law-2 } (1 \leqslant k \leqslant n)$$

Figure 12. Admissible rules for $\Lambda^{\times}_{\mathrm{ps}}(\mathcal{G})$

$$
\begin{aligned}
h[\![X]\!] &:= h(X) \qquad \text{for } X \text{ a base type} \\
h[\![\textstyle\prod_n(B_1, \dots, B_n)]\!] &:= \textstyle\prod_{i=1}^n h[\![B_i]\!] \\
h[\![A \Rightarrow B]\!] &:= h[\![A]\!] \Rightarrow h[\![B]\!] \\
h[\![x_1 : A_1, \dots, x_n : A_n]\!] &:= \textstyle\prod_{i=1}^n h[\![A_i]\!] \qquad \text{on } n\text{-ary contexts} \\
h[\![x_1 : A_1, \dots, x_n : A_n \vdash x_k : A_i]\!] &:= \pi_k^{A_1,\dots,A_n} \\
h[\![x_1 : A_1, \dots, x_n : A_n \vdash c(x_1, \dots, x_n) : B]\!] &:= h(c) \qquad \text{for } c \in \mathcal{G}(A_\bullet; B) \\
h[\![\Delta \vdash t\{x_i^{A_i} \mapsto u_i\}_{i=1}^n : B]\!] &:= h[\![x_1 : A_1, \dots, x_n : A_n \vdash t : B]\!] \\
&\qquad \circ \langle h[\![\Delta \vdash u_i : A_i]\!] \rangle_{i=1,\dots,n} \\
h[\![\Gamma \vdash \mathsf{tup}(t_1, \dots, t_n) : \textstyle\prod_n(B_1, \dots, B_n)]\!] &:= \langle h[\![\Gamma \vdash t_1 : B_1,]\!], \dots, h[\![\Gamma \vdash t_n : B_n]\!] \rangle \\
h[\![p : \textstyle\prod_n(B_1, \dots, B_n) \vdash \pi_k(p) : B_k]\!] &:= \pi_k^{B_1,\dots,B_n} \\
h[\![f : A \Rightarrow B, a : A \vdash \mathsf{eval}(f, a) : B]\!] &:= \mathrm{eval}_{A,B} \\
h[\![\Gamma \vdash \lambda x.t : A \Rightarrow B]\!] &:= \lambda\big(h[\![\Gamma, x : A \vdash t : B]\!] \circ \simeq\big) \\
h[\![\Gamma \vdash \mathrm{id}_t : t \Rightarrow t : B]\!] &:= \mathrm{id}_{h[\![\Gamma \vdash t : B]\!]} \\
h[\![x_1 : A_1, \dots, x_n : A_n \vdash \sigma(x_\bullet) : c(x_\bullet) \Rightarrow c'(x_\bullet) : B]\!] &:= h(\sigma) \qquad \text{for } \sigma \in \mathcal{G}(A_\bullet, B)(c, c') \\
h[\![\Gamma \vdash \varpi^{(k)}_{t_1,\dots,t_n} : \pi_k\{\mathsf{tup}(t_1, \dots, t_n)\} \Rightarrow t_k : B_k]\!] &:= \varpi^{(k)}_{h[\![t_1]\!],\dots,h[\![t_n]\!]} \\
h[\![\Gamma \vdash \mathsf{p}^\dagger(\alpha_1, \dots, \alpha_n) : u \Rightarrow \mathsf{tup}(t_1, \dots, t_n) : \textstyle\prod_n(B_1, \dots, B_n)]\!] &:= \mathsf{p}^\dagger(h[\![\Gamma \vdash \alpha_i : \pi_i\{u\} \Rightarrow t_i : B_i]\!])_{i=1,\dots,n} \\
h[\![\Gamma, x : A \vdash \epsilon_t : \mathsf{eval}\{(\lambda x.t)\{\mathrm{inc}_x\}, x\} \Rightarrow t : B]\!] &:= \epsilon_{(h[\![\Gamma, x : A \vdash t : B]\!] \circ \simeq)} \\
h[\![\Gamma \vdash \mathsf{e}^\dagger(x.\alpha) : u \Rightarrow \lambda x.t : A \Rightarrow B]\!] &:= \mathsf{e}^\dagger(h[\![\Gamma, x : A \vdash \alpha : \mathsf{eval}\{u\{\mathrm{inc}_x\}, x\} \Rightarrow t : B]\!] \circ \simeq) \\
h[\![\Gamma \vdash \tau' \bullet \tau : t \Rightarrow t'' : B]\!] &:= h[\![\Gamma \vdash \tau' : t' \Rightarrow t'' : B]\!] \bullet h[\![\Gamma \vdash \tau : t \Rightarrow t' : B]\!] \\
h[\![\Delta \vdash \tau\{x_i^{A_i} \mapsto \sigma_i\}_{i=1}^n : t\{x_i^{A_i} \mapsto u_i\}_{i=1}^n \Rightarrow t'\{x_i^{A_i} \mapsto u'_i\}_{i=1}^n : B]\!] &:= h[\![x_1 : A_1, \dots, x_n : A_n \vdash \tau : t \Rightarrow t' : B]\!] \\
&\qquad \circ \langle h[\![\Delta \vdash \sigma_i : u_i \Rightarrow u'_i : A_i]\!] \rangle_{i=1,\dots,n}
\end{aligned}
$$

Figure 13. Bicategorical semantics

the freeness universal property for $\mathcal{T}^{\times}_{\mathrm{ps}}(\mathcal{G})$ with respect to 2-multigraphs $\mathcal{G}$, it suffices to show that $\mathcal{S}^{\times}_{\mathrm{ps}}(\mathcal{G})$ is the free fp-bicategory for 2-graphs $\mathcal{G}$. Universal extensions are induced by the semantics in Figure 13 restricted to $\mathcal{S}^{\times}_{\mathrm{ps}}(\mathcal{G})$.

Theorem VII.9. *Let $\mathcal{G}$ be a 2-graph with $\mathcal{G}_0 = \mathrm{T}_0(S)$ for a set of base types S. For every fp-bicategory $\mathcal{B}$ and 2-graph homomorphism $h : \mathcal{G} \to \mathcal{B}$ such that $h\big(\Pi_n(A_1, \ldots, A_n)\big) = \Pi_n(hA_1, \ldots, hA_n)$, there exists a unique strict fp-pseudofunctor $h^{\#} : \mathcal{S}^{\times}_{\mathrm{ps}}(\mathcal{G}) \to \mathcal{B}$ such that $h^{\#} \circ \iota = h$, where $\iota : \mathcal{G} \hookrightarrow \mathcal{S}^{\times}_{\mathrm{ps}}(\mathcal{G})$ denotes the inclusion.*

VIII. Cartesian closed bicategories

To give a cartesian closed structure on an fp-bicategory $\mathcal{B}$ is to specify a biadjunction $(-) \times A \dashv (A \Rightarrow -)$ for each object $A \in \mathcal{B}$. Cartesian closed bicategories are the objects of a tricategory $\mathbf{CCBicat}$.

Definition VIII.1. A *cartesian closed bicategory* or *CC-bicategory* is an fp-bicategory $\big(\mathcal{B}, \Pi_n(-)\big)$ equipped with the following data for every $A, B \in \mathcal{B}$:

1) an *exponential* object $(A \Rightarrow B)$,
2) an *evaluation* 1-cell $\mathrm{eval}_{A,B} : (A \Rightarrow B) \times A \to B$,
3) for every $X \in \mathcal{B}$ an adjoint equivalence

$$\mathcal{B}(X, A \Rightarrow B) \underset{\lambda}{\overset{\mathrm{eval}_{A,B} \circ (- \times A)}{\rightleftarrows}} \mathcal{B}(X \times A, B) \quad \bot \qquad (3)$$

Remark VIII.2. The uniqueness of exponentials up to equivalence manifests itself in the same way as for products. For instance, given an adjoint equivalence $e : E \simeq (A \Rightarrow B) : f$, the object E inherits an exponential structure by composition with e and f (*c.f.* Remark VI.6).

As for products, we choose to define the adjunction (3) by characterising its counit ϵ as a universal arrow (*c.f.* the discussion after Notation VI.4). Thus, for every 1-cell $t : X \times A \to B$ we require a 1-cell $\lambda t : X \to (A \Rightarrow B)$ and a 2-cell $\epsilon_t : \mathrm{eval}_{A,B} \circ (\lambda t \times A) \Rightarrow t$ that is universal in the sense that for any 2-cell $\alpha : \mathrm{eval}_{A,B} \circ (u \times A) \Rightarrow t$ there exists a unique 2-cell $\mathsf{e}^{\dagger}(\alpha) : u \Rightarrow \lambda t$ such that $\epsilon_t \bullet \big(\mathrm{eval}_{A,B} \circ (\mathsf{e}^{\dagger}(\alpha) \times A)\big) = \alpha$.

Definition VIII.3. A *cartesian closed pseudofunctor* or *CC-pseudofunctor* between CC-bicategories $\big(\mathcal{B}, \Pi_n(-), \Rightarrow\big)$ and $\big(\mathcal{C}, \Pi_n(-), \Rightarrow\big)$ is an fp-pseudofunctor $(F, k^{\times})$ equipped with equivalences

$$\big(F(A \Rightarrow B), (FA \Rightarrow FB), s_{A,B}, k^{\Rightarrow}_{A,B}\big)$$

for every $A, B \in \mathcal{B}$, where $s_{A,B} : F(A \Rightarrow B) \to (FA \Rightarrow FB)$ is the transpose of

$$F(\mathrm{eval}_{A,B}) \circ k^{\times}_{A \Rightarrow B, A} : F(A \Rightarrow B) \times FA \to FB$$

A CC-pseudofunctor $(F, k^{\times}, k^{\Rightarrow})$ is *strict* if $(F, k^{\times})$ is strict and satisfies

$$\begin{aligned} F(A \Rightarrow B) &= (FA \Rightarrow FB) \\ F(\mathrm{eval}_{FA,FB}) &= \mathrm{eval}_{FA,FB} \\ F(\lambda t) &= \lambda(Ft) \\ F(\epsilon_t) &= \epsilon_{Ft} \\ k^{\Rightarrow}_{A,B} &= \mathrm{Id}_{FA \Rightarrow FB} \end{aligned}$$

and the adjoint equivalences are canonically induced by the 2-cells $\mathsf{e}^{\dagger}(\kappa) : \mathrm{Id}_{FA \Rightarrow FB} \overset{\cong}{\Rightarrow} \lambda(\mathrm{eval}_{FA,FB})$ where κ is the following composite of canonical 2-cells

$$\mathrm{eval}_{FA,FB} \circ (\mathrm{Id} \times FA) \cong \mathrm{eval}_{FA,FB} \circ \mathrm{Id} \cong \mathrm{eval}_{FA,FB}$$

IX. A type theory for cartesian closed bicategories

We extend $\Lambda^{\times}_{\mathrm{ps}}$ to the full language $\Lambda^{\times,\to}_{\mathrm{ps}}$ by synthesising the additional rules from the cases of Definition VIII.1; these rules are Figures 14–16.

We extend the grammar for types with an arrow type former $(-) \Rightarrow (=)$. We henceforth fix a set of base types S, let $\mathrm{T}(S)$ denote the set of all types over S, and consider 2-multigraphs $\mathcal{G}$ with set of nodes $\mathcal{G}_0 = \mathrm{T}(S)$.

We postulate a constant $\mathsf{eval}(f, x)$ for every context $(f : A \Rightarrow B, x : A)$; the usual application operation becomes a derived rule:

$$\frac{\Gamma \vdash t : A \Rightarrow B \qquad \Gamma \vdash u : A}{\Gamma \vdash \mathsf{eval}\{t, u\} : B}$$

The adjunction (3) requires the functor $(-) \times A$ on the syntactic model. As for STLC, this corresponds to context extension and an associated notion of weakening. The explicit nature of substitution in our type theory gives rise to correspondingly explicit structural operations on contexts.

Definition IX.1. A *context renaming*

$$r : (x_i : A_i)_{i=1,\ldots,n} \to (y_j : B_j)_{j=1,\ldots,m}$$

is a map $r : \{x_1, \ldots, x_n\} \to \{y_1, \ldots, y_m\}$ such that $A_i = B_j$ whenever $r(x_i) = y_j$.

It is immediate that for every context renaming $r : \Gamma \to \Delta$ the following rules are admissible in $\Lambda^{\times,\to}_{\mathrm{ps}}(\mathcal{G})$:

$$\frac{\Gamma \vdash t : A \qquad r : \Gamma \to \Delta}{\Delta \vdash t\{x_i \mapsto r(x_i)\} : A}$$

$$\frac{\Gamma \vdash \tau : t \Rightarrow t' : A \qquad r : \Gamma \to \Delta}{\Delta \vdash \tau\{x_i \mapsto r(x_i)\} : t\{x_i \mapsto r(x_i)\} \Rightarrow t'\{x_i \mapsto r(x_i)\} : A}$$

Notation IX.2. For a context renaming r we write $t\{r\}$ and $\tau\{r\}$ for the terms and rewrites formed using the preceding admissible rules.

Weakening arises by taking the inclusion of contexts $\mathrm{inc}_x : \Gamma \hookrightarrow (\Gamma, x : A)$. For the counit of (3) we therefore postulate a rewrite

$$\epsilon_t : \mathsf{eval}\{(\lambda x.t)\{\mathrm{inc}_x\}, x\} \Rightarrow t$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \Rightarrow B}\text{ lam} \qquad \frac{}{f : A \Rightarrow B, a : A \vdash \mathsf{eval}(f, a) : B}\text{ eval}$$

Figure 14. Terms for cartesian closed structure

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma, x : A \vdash \epsilon_t : \mathsf{eval}\{(\lambda x.t)\{\mathrm{inc}_x\}, x\} \Rightarrow t : B}\ \epsilon\text{-intro} \qquad \frac{\Gamma, x : A \vdash t : B \quad \Gamma \vdash u : A \Rightarrow B \quad \Gamma, x : A \vdash \alpha : \mathsf{eval}\{u\{\mathrm{inc}_x\}, x\} \Rightarrow t : B}{\Gamma \vdash \mathsf{e}^\dagger(x.\alpha) : u \Rightarrow \lambda x.t : A \Rightarrow B}\ \mathsf{e}^\dagger\text{-intro}$$

Figure 15. Rewrites for cartesian closed structure

$$\frac{\Gamma, x : A \vdash \alpha : \mathsf{eval}\{u\{\mathrm{inc}_x\}, x\} \Rightarrow t : B}{\Gamma, x : A \vdash \alpha \equiv \epsilon_t \bullet \mathsf{eval}\{\mathsf{e}^\dagger(x.\alpha)\{\mathrm{inc}_x\}, x\} : \mathsf{eval}\{u\{\mathrm{inc}_x\}, x\} \Rightarrow t : B}\text{ U1}$$

$$\frac{\Gamma \vdash \gamma : u \Rightarrow \lambda x.t : A \Rightarrow B}{\Gamma \vdash \gamma \equiv \mathsf{e}^\dagger(x.\epsilon_t \bullet \mathsf{eval}\{\gamma\{\mathrm{inc}_x\}, x\}) : u \Rightarrow \lambda x.t : A \Rightarrow B}\text{ U2}$$

$$\frac{\Gamma, x : A \vdash \alpha \equiv \alpha' : \mathsf{eval}\{u\{\mathrm{inc}_x\}, x\} \Rightarrow t : B}{\Gamma \vdash \mathsf{e}^\dagger(x.\alpha) \equiv \mathsf{e}^\dagger(x.\alpha') : u \Rightarrow \lambda x.t : A \Rightarrow B}\text{ cong}$$

Figure 16. Universal property of ϵ

Rules for $\Lambda_{\mathrm{ps}}^{\times,\rightarrow}(\mathcal{G})$.

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.\mathrm{id}_t \equiv \mathrm{id}_{\lambda x.t} : \lambda x.t \Rightarrow \lambda x.t : A \Rightarrow B} \qquad \frac{\Gamma, x : A \vdash \tau' : t' \Rightarrow t'' : B \quad \Gamma, x : A \vdash \tau : t \Rightarrow t' : B}{\Gamma \vdash \lambda x.(\tau' \bullet \tau) \equiv (\lambda x.\tau') \bullet (\lambda x.\tau) : \lambda x.t \Rightarrow \lambda x.t'' : A \Rightarrow B}$$

$$\frac{\Gamma \vdash \sigma : u \Rightarrow u' : A \Rightarrow B}{\Gamma \vdash \eta_{u'} \bullet \sigma \equiv \lambda x.\mathsf{eval}\{\sigma\{\mathrm{inc}_x\}, x\} \bullet \eta_u : u \Rightarrow \lambda x.\mathsf{eval}\{u'\{\mathrm{inc}_x\}, x\} : A \Rightarrow B}\ \eta\text{-nat}$$

$$\frac{\Gamma, x : A \vdash \tau : t \Rightarrow t' : B}{\Gamma, x : A \vdash \tau \bullet \epsilon_t \equiv \epsilon_{t'} \bullet \mathsf{eval}\{(\lambda x.\tau)\{\mathrm{inc}_x\}, x\} : \mathsf{eval}\{(\lambda x.t)\{\mathrm{inc}_x\}, x\} \Rightarrow t' : B}\ \epsilon\text{-nat}$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash (\lambda x.\epsilon_t) \bullet \eta_t \equiv \mathrm{id}_{\lambda x.t} : \lambda x.t \Rightarrow \lambda x.t : A \Rightarrow B}\text{ triangle-law-1}$$

$$\frac{\Gamma \vdash u : A \Rightarrow B}{\Gamma, x : A \vdash \epsilon_{\mathsf{eval}\{u\{\mathrm{inc}_x\},x\}} \bullet \mathsf{eval}\{\eta_u\{\mathrm{inc}_x\}, x\} \equiv \mathrm{id}_{\mathsf{eval}\{u\{\mathrm{inc}_x\},x\}} : \mathsf{eval}\{u\{\mathrm{inc}_x\}, x\} \Rightarrow \mathsf{eval}\{u\{\mathrm{inc}_x\}, x\} : B}\text{ triangle-law-2}$$

Figure 17. Admissible rules for $\Lambda_{\mathrm{ps}}^{\times,\rightarrow}(\mathcal{G})$

for every term t typeable in a context extended by the variable x.

The encoding of the universal property of ϵ_t in the type theory yields a bijective correspondence of rewrites modulo $\alpha\equiv$-equivalence as follows (*c.f.* (2)):

$$\frac{(x : A) \quad \alpha : \mathsf{eval}\{u\{\mathrm{inc}_x\}, x\} \Rightarrow t : B}{\mathsf{e}^\dagger(x.\alpha) : u \Rightarrow \lambda x.t : A \Rightarrow B}$$

where we write $(x : A)$ to indicate that the variable x of type A is in the context (*c.f.* [30]). Our approach to achieving this matches that for products. For every rewrite $\alpha : \mathsf{eval}\{u\{\mathrm{inc}_x\}, x\} \Rightarrow t$ we introduce the rewrite $\mathsf{e}^\dagger(x.\alpha) : u \Rightarrow \lambda x.t$ and make it unique in inducing a factorisation of α through ϵ_t; the variable x is bound by this operation. The rules governing this construction are given in Figures 15–16; these provide *lax* exponentials (*c.f.* [25]). Finally, one requires explicit inverses for the unit and counit. In this extended abstract they are omitted.

Remark IX.3. As for products (*c.f.* Remark VII.1), we obtain a nesting of two universal transposition structures. In the same vein, we conjecture that a calculus for cartesian closed *tri*categories (cartesian closed ∞-categories) would have *three* (a countably infinite tower of) of such correspondences. Indeed, this should extend to general type structures arising from weak adjunctions.

We continue to work up to α-equivalence of terms and rewrites. The well-formedness properties of $\Lambda_{\mathrm{ps}}^{\mathrm{b}}$ and $\Lambda_{\mathrm{ps}}^{\times}$ lift to $\Lambda_{\mathrm{ps}}^{\times,\rightarrow}$.

A. Cartesian closed structure of $\Lambda_{\mathrm{ps}}^{\times,\to}$

The ϵ-introduction rule (Figure 15) only 'evaluates' lambda abstractions at variables. The general form of explicit β-reduction is derivable.

Definition IX.4. For derivable terms $\Gamma, x : A \vdash t : B$ and $\Gamma \vdash u : A$ define $\beta_{x.t,u} : \mathsf{eval}\{\lambda x.t, u\} \Rightarrow t\{\mathrm{id}_\Gamma, x \mapsto u\}$ to be the rewrite $\epsilon_t\{\mathrm{id}_\Gamma, x \mapsto u\} \bullet \tau$ in context Γ (recall Notation IX.2), where τ is a composite of structural isomorphisms.

As expected, the remaining exponential structure follows from the universal property of ϵ (*c.f.* Lemma VII.2).

Lemma IX.5. *For every rewrite* $(\Gamma, x : A \vdash \alpha : \mathsf{eval}\{u\{\mathrm{inc}_x\}, x\} \Rightarrow t : B)$ *in* $\Lambda_{\mathrm{ps}}^{\times,\to}(\mathcal{G})$*, the rewrite* $\mathsf{e}^\dagger(x.\alpha)$ *is the unique* γ *(modulo* $\alpha\equiv$*-equivalence) such that*

$$\Gamma, x : A \vdash \alpha \equiv \epsilon_t \bullet \mathsf{eval}\{\gamma\{\mathrm{inc}_x\}, x\} : \mathsf{eval}\{u\{\mathrm{inc}_x\}, x\} \Rightarrow t : B$$

It follows that the lambda-abstraction operator extends to a functor, and we obtain a unit for the adjunction (3).

Definition IX.6.

1) For any derivable rewrite $(\Gamma, x : A \vdash \tau : t \Rightarrow t' : B)$ define $\lambda x.\tau : \lambda x.t \Rightarrow \lambda x.t'$ to be the rewrite $\mathsf{e}^\dagger(x.\tau \bullet \epsilon_t)$ in context Γ.
2) For any derivable term $(\Gamma \vdash u : A \Rightarrow B)$ define the unit $\eta_u : u \Rightarrow \lambda x.\mathsf{eval}\{u\{\mathrm{inc}_x\}, x\}$ to be the rewrite $\mathsf{e}^\dagger(x.\mathrm{id}_{\mathsf{eval}\{u\{\mathrm{inc}_x\},x\}})$ in context Γ.

The unit η and counit ϵ satisfy naturality and triangle laws, collected in Figure 17. Thus we recover the unit-counit presentation of both products and exponentials (*c.f.* [24], [25], [29]).

B. The syntactic model for $\Lambda_{\mathrm{ps}}^{\times,\to}$

We finally turn to constructing the syntactic model for $\Lambda_{\mathrm{ps}}^{\times,\to}(\mathcal{G})$ and proving its freeness universal property. The construction of the syntactic model follows the pattern established by $\Lambda_{\mathrm{ps}}^{\mathrm{b}}$ and $\Lambda_{\mathrm{ps}}^{\times}$.

Construction IX.7. *Define a bicategory* $\mathcal{T}_{\mathrm{ps}}^{\times,\to}(\mathcal{G})$ *as follows. The objects are contexts* $\Gamma, \Delta, \ldots$*. The 1-cells* $\Gamma \to (y_j : B_j)_{j=1,\ldots,m}$ *are* m*-tuples of* α*-equivalence classes of terms* $(\Gamma \vdash t_j : B_j)_{j=1,\ldots,m}$ *derivable in* $\Lambda_{\mathrm{ps}}^{\times,\to}(\mathcal{G})$*; 2-cells are* m*-tuples of* $\alpha\equiv$*-equivalence classes of rewrites* $(\Gamma \vdash \tau : t_j \Rightarrow t'_j : B_j)_{j=1,\ldots,m}$*. Horizontal and vertical composition are as in Construction VII.4.*

$\mathcal{T}_{\mathrm{ps}}^{\times,\to}(\mathcal{G})$ inherits the product structure of $\mathcal{T}_{\mathrm{ps}}^{\times}(\mathcal{G})$. To construct cartesian closed structure it suffices to construct exponentials of unary contexts. Much of the work required is contained in the admissible rules of Figure 17.

Lemma IX.8. *For unary contexts* $(x : A)$*,* $(y : B)$*, the exponential* $(x : A) \Rightarrow (y : B)$ *in* $\mathcal{T}_{\mathrm{ps}}^{\times,\to}(\mathcal{G})$ *exists and is given by* $(f : A \Rightarrow B)$*.*

Using Remark VIII.2 and the preceding lemma, we define the exponential object $(\Gamma \Rightarrow \Delta)$ for $\Gamma := (x_i : A_i)_{i=1,\ldots,n}$ and $\Delta := (y_j : B_j)_{j=1,\ldots,m}$ using the equivalent unary contexts, as

$$\big(p : \textstyle\prod_n(A_1, \ldots, A_n)\big) \Rightarrow \big(q : \textstyle\prod_m(B_1, \ldots, B_m)\big).$$

Corollary IX.9. *The syntactic model* $\mathcal{T}_{\mathrm{ps}}^{\times,\to}(\mathcal{G})$ *is a CC-bicategory.*

From Sections V-A and VII-B one might expect that restricting to a sub-bicategory with unary contexts and a fixed variable name is sufficient to obtain the required freeness universal property. This suggests the following definition.

Construction IX.10. *Let* $\mathcal{S}_{\mathrm{ps}}^{\times,\to}(\mathcal{G})$ *denote the bicategory with objects unary contexts* $(x : A)$ *for* x *a* fixed *variable and horizontal and vertical composition operations as in* $\mathcal{T}_{\mathrm{ps}}^{\times,\to}(\mathcal{G})$*.*

The cartesian closed structure of $\mathcal{T}_{\mathrm{ps}}^{\times,\to}(\mathcal{G})$ restricts to $\mathcal{S}_{\mathrm{ps}}^{\times,\to}(\mathcal{G})$ and the two are equivalent as cartesian closed bicategories. However, one does not obtain a strict freeness universal property. We recover uniqueness by restricting to cartesian closed bicategories in which the product structure is strict.

Theorem IX.11. *Let* $\mathcal{G}$ *be a 2-graph with* $\mathcal{G}_0 = \mathrm{T}(S)$ *for a set of base types* S*. For every cartesian 2-category (*i.e. *2-category with 2-categorical products) with pseudo-exponentials* $\mathcal{C}$ *and every 2-graph homomorphism* $h : \mathcal{G} \to \mathcal{C}$ *such that* $h\big(\Pi_n(A_1, \ldots, A_n)\big) = \Pi_n(hA_1, \ldots, hA_n)$ *and* $h(A \Rightarrow B) = (hA \Rightarrow hB)$*, there exists a unique strict CC-pseudofunctor* $h^\# : \mathcal{S}_{\mathrm{ps}}^{\times,\to}(\mathcal{G}) \to \mathcal{C}$ *such that* $h^\# \circ \iota = h$*, for* $\iota : \mathcal{G} \hookrightarrow \mathcal{S}_{\mathrm{ps}}^{\times,\to}(\mathcal{G})$ *the inclusion.*

Thus, one obtains the same result as for Sections V-A and VII-B, albeit with a restricted freeness universal property. This, modulo the coherence result of Power [14] applied to fp-bicategories, yields that

$\Lambda_{\mathrm{ps}}^{\times,\to}$ is the internal language of
cartesian closed bicategories.

In fact, it is possible to adjust the definition of exponentials to obtain a type theory for which the induced syntactic models are equivalent as CC-bicategories to those above and satisfy a strict freeness universal property with respect to arbitrary CC-bicategories. This type theory is however no longer in the spirit of STLC, and so will be presented elsewhere.

Acknowledgements

We thank Dylan McDermott and Ian Orton for helpful discussions on type theories with weak substitution.

References

[1] J. Bénabou, "Introduction to bicategories," in *Reports of the Midwest Category Seminar*. Springer Berlin Heidelberg, 1967, pp. 1–77.

[2] R. Street, "Categorical structures," in *Handbook of Algebra*, M. Hazewinkel, Ed. Elsevier, 1995, vol. 1, ch. 15, pp. 529–577.

[3] G.L. Cattani, M. Fiore, and G. Winskel, "A theory of recursive domains with applications to concurrency," in *Proceedings of the 13th Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society, 1998, pp. 214–225.

[4] S. Castellan, P. Clairambault, S. Rideau, and G. Winskel, "Games and strategies as event structures," *Logical Methods in Computer Science*, vol. 13, no. 3, 2017. [Online]. Available: https://doi.org/10.23638/LMCS-13(3:35)2017

[5] M. G. Abbott, "Categories of containers," Ph.D. dissertation, University of Leicester, 2003.

[6] P.-E. Dagand and C. McBride, "A categorical treatment of ornaments," in *Proceedings of the 28th Annual ACM/IEEE Symposium on Logic in Computer Science*. IEEE Computer Society, 2013, pp. 530–539.

[7] M. Fiore, N. Gambino, M. Hyland, and G. Winskel, "The cartesian closed bicategory of generalised species of structures," *Journal of the London Mathematical Society*, vol. 77, no. 1, pp. 203–220, 2007.

[8] N. Gambino and J. Kock, "Polynomial functors and polynomial monads," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 154, no. 1, pp. 153–192, 2013.

[9] M. Fiore and A. Joyal, "Theory of para-toposes," talk at the *Category Theory 2015 Conf.*, Departamento de Matematica, Universidade de Aveiro, Aveiro (Portugal), 2015. [Online]. Available: http://sweet.ua.pt/dirk/ct2015/abstracts/fiore_m.pdf

[10] N. Gambino and A. Joyal, "On operads, bimodules and analytic functors," *Memoirs of the American Mathematical Society*, vol. 249, no. 1184, pp. 153–192, 2017.

[11] M. Fiore, N. Gambino, M. Hyland, and G. Winskel, "Relative pseudomonads, Kleisli bicategories, and substitution monoidal structures," *Selecta Mathematica New Series*, 2017.

[12] J. W. Gray, *Formal Category Theory: Adjointness for 2-Categories*, ser. Lecture Notes in Mathematics. Springer, 1974, vol. 391.

[13] S. Mac Lane and R. Paré, "Coherence for bicategories and indexed categories," *Journal of Pure and Applied Algebra*, vol. 37, pp. 59–80, 1985.

[14] A. J. Power, "Coherence for bicategories with finite bilimits I," in *Categories in Computer Science and Logic: Proceedings of the AMS-IMS-SIAM Joint Summer Research Conference Held June 14-20, 1987*, J. W. Gray and A. Scedrov, Eds. American Mathematical Society, 1989, vol. 92, pp. 341–349.

[15] P. T. Johnstone, *Sketches of an Elephant*. Oxford University Press, 2002.

[16] N. Gambino and M. Hyland, "Wellfounded trees and dependent polynomial functors," in *Types for Proofs and Programs*, S. Berardi, M. Coppo, and F. Damiani, Eds. Springer Berlin Heidelberg, 2004, pp. 210–225.

[17] J. Lambek, "Cartesian closed categories and typed lambda- calculi," in *Proceedings of the Thirteenth Spring School of the LITP on Combinators and Functional Programming Languages*. Springer-Verlag, 1986, pp. 136–175.

[18] M. Makkai and G. Reyes, *First Order Categorical Logic: Model-Theoretical Methods in the Theory of Topoi and Related Categories*. Springer, 1977.

[19] A. Church, "A formulation of the simple theory of types," *The Journal of Symbolic Logic*, vol. 5, no. 2, pp. 56–68, 1940.

[20] M. Fiore, "An algebraic combinatorial approach to opetopic structure," talk at the *Seminar on Higher Structures, Program on Higher Structures in Geometry and Physics*, Max Planck Institute for Mathematics, Bonn (Germany), 2016. [Online]. Available: https://www.mpim-bonn.mpg.de/node/6586

[21] N. Yamada and S. Abramsky, "Dynamic game semantics," *arXiv*, 2018. [Online]. Available: https://arxiv.org/abs/1601.04147

[22] D. E. Rydeheard and J. G. Stell, "Foundations of equational deduction: A categorical treatment of equational proofs and unification algorithms," in *Category Theory and Computer Science*, D. H. Pitt, A. Poigné, and D. E. Rydeheard, Eds. Springer Berlin Heidelberg, 1987, pp. 114–139.

[23] A. J. Power, "An abstract formulation for rewrite systems," in *Category Theory and Computer Science*, D. H. Pitt, D. E. Rydeheard, P. Dybjer, A. M. Pitts, and A. Poigné, Eds. Springer Berlin Heidelberg, 1989, pp. 300–312.

[24] R. A. G. Seely, "Modelling computations: A 2-categorical framework," in *Proceedings of the Second Annual IEEE Symp. on Logic in Computer Science*, D. Gries, Ed. IEEE Computer Society Press, June 1987, pp. 65–71.

[25] B. P. Hilken, "Towards a proof theory of rewriting: the simply typed 2λ-calculus," *Theoretical Computer Science*, vol. 170, no. 1, pp. 407–444, 1996.

[26] C. B. Jay and N. Ghani, "The virtues of eta-expansion," *Journal of Functional Programming*, vol. 5, no. 2, pp. 135–154, 1995.

[27] N. Ghani, "Adjoint rewriting," Ph.D. dissertation, University of Edinburgh, 1995.

[28] N. Tabareau, "Aspect oriented programming: A language for 2-categories," in *Proceedings of the 10th International Workshop on Foundations of Aspect-oriented Languages*, ser. FOAL '11. ACM, 2011, pp. 13–17.

[29] T. Hirschowitz, "Cartesian closed 2-categories and permutation equivalence in higher-order rewriting," *Logical Methods in Computer Science*, vol. 9, pp. 1–22, 07 2013.

[30] P. Martin-Löf, *Intuitionistic Type Theory*. Bibliopolis, 1984.

[31] D. R. Licata and R. Harper, "2-dimensional directed type theory," *Electronic Notes in Theoretical Computer Science*, vol. 276, pp. 263–289, 2011, twenty-seventh Conference on the Mathematical Foundations of Programming Semantics (MFPS XXVII).

[32] P.-L. Curien, "Substitution up to isomorphism," *Fundam. Inf.*, vol. 19, no. 1-2, pp. 51–85, Sep. 1993.

[33] The Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study: https://homotopytypetheory.org/book, 2013.

[34] M. Abadi, L. Cardelli, P.-L. Curien, and J.-J. Levy, "Explicit substitutions," in *Proceedings of the 17th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, ser. POPL '90. ACM, 1990, pp. 31–46.

[35] E. Ritter and V. de Paiva, "On explicit substitutions and names (extended abstract)," in *Automata, Languages and Programming*, P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, Eds. Springer Berlin Heidelberg, 1997, pp. 248–258.

[36] B. Plotkin, *Universal Algebra, Algebraic Logic, and Databases*. Springer, 1994.

[37] R. Gordon, A. Power, and R. Street, *Coherence for tricategories*. Memoirs of the American Mathematical Society, 1995.

[38] N. Gurski, *Coherence in Three-Dimensional Category Theory*. Cambridge University Press, 2013.

[39] Agda contributors, "The Agda proof assistant," https://wiki.portal.chalmers.se/agda/pmwiki.php.

[40] A. J. Power, "2-categories," *BRICS Notes Series*, 1998.

[41] S. Mac Lane, *Categories for the Working Mathematician*, 2nd ed., ser. Graduate Texts in Mathematics. Springer-Verlag New York, 1998, vol. 5.

[42] T. Fiore, *Pseudo Limits, Biadjoints, and Pseudo Algebras: Categorical Foundations of Conformal Field Theory*, ser. Memoirs of the American Mathematical Society. AMS, 2006.

[43] F. W. Lawvere, "Adjoints in and among bicategories," in *Logic & Algebra*, ser. Lecture Notes in Pure and Applied Algebra, vol. 180, 1996, pp. 181–189.

[44] S. Lack and R. Street, "Skew monoidales, skew warpings and quantum categories," *Theory and Applications of Categories*, 2012.

[45] R. Street, "Fibrations in bicategories," *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, vol. 21, no. 2, pp. 111–160, 1980.

[46] S. Staton, "An algebraic presentation of predicate logic," in *Foundations of Software Science and Computation Structures*, F. Pfenning, Ed. Springer Berlin Heidelberg, 2013, pp. 401–417.

[47] M. Fiore, "On the concrete representation of discrete enriched abstract clones," *Tbilisi Mathematical Journal*, vol. 10, no. 3, pp. 297–328, 2017.

[48] K. Szlachányi, "Skew-monoidal categories and bialgebroids," *Advances in Mathematics*, vol. 231, no. 3, pp. 1694–1730, 2012.

[49] T. Leinster, *Higher operads, higher categories*, ser. London Mathematical Society Lecture Note Series. Cambridge University Press, 2004, no. 298.

[50] C. Hermida, "Representable multicategories," *Advances in Mathematics*, vol. 151, no. 2, pp. 164–225, 2000.

[51] R. L. Crole, *Categories for Types*. Cambridge University Press, 1994.

[52] A. Carboni and R. Walters, "Cartesian bicategories I," *Journal of Pure and Applied Algebra*, vol. 49, no. 1, pp. 11–32, 1987.

[53] A. Carboni, G. Kelly, R. Walters, and R. Wood, "Cartesian bicategories II," *Theory and Applications of Categories*, vol. 19, no. 6, pp. 93–124, 2008.