Categories with Families: Unityped, Simply Typed, and Dependently Typed

Simon Castellan, Pierre Clairambault, and Peter Dybjer

Abstract We show how the categorical logic of untyped, simply typed and dependently typed lambda calculus can be structured around the notion of category with family (cwf). To this end we introduce subcategories of simply typed cwfs (scwfs), where types do not depend on variables, and unityped cwfs (ucwfs), where there is only one type. We prove several equivalence and biequivalence theorems between cwf-based notions and basic notions of categorical logic, such as cartesian operads, Lawvere theories, categories with finite products and limits, cartesian closed categories, and locally cartesian closed categories. Some of these theorems depend on the restrictions of contextuality (in the sense of Cartmell) or democracy (used by Clairambault and Dybjer for their biequivalence theorems). Some theorems are equivalences between notions with strict preservation of chosen structure. Others are biequivalences between notions where properties are only preserved up to isomorphism. In addition to this we discuss various constructions of initial ucwfs, scwfs, and cwfs with extra structure.

Simon Castellan
Imperial College London, London SW7 2AZ, England
e-mail: simon.castellan@phis.me

Pierre Clairambault
CNRS, ENS de Lyon, 46 alle d'Italie, 69364 Lyon, France
e-mail: pierre.clairambault@ens-lyon.fr

Peter Dybjer
Chalmers University of Technology, SE-412 96 Gteborg, Sweden.
e-mail: peterd@chalmers.se

1 Introduction

An important part of categorical logic is to establish correspondences between languages (from logic) and categorical models. For example, in their book "Introduction to higher order categorical logic" [24] Lambek and Scott prove equivalences between typed lambda calculi and cartesian closed categories, between untyped lambda calculi and C-monoids, and between intuitionistic type theories and toposes. Lambek and Scott's intuitionistic type theories are intuitionistic versions of Church's simple theory of types, which should not be confused with Martin-Löf's intuitionistic type theories. Interestingly, in the preface of their book [24, p viii] Lambek and Scott express a desire to include a result concerning the latter too:

> We also claim that intuitionistic type theories and toposes are closely related, in as much as there is a pair of adjoint functors between their respective categories. This is worked out out in Part II. The relationship between Martin-Löf type theories and locally cartesian closed categories was established too recently (by Robert Seely) to be treated here.

Seely's seminal paper [30] claims to prove that a category of Martin-Löf type theories is equivalent to a category of locally cartesian closed categories (lcccs). However, his result relies on an interpretation of substitution as pullback, and these are only defined up to isomorphism. It is not clear how to choose pullbacks in such a way that the strict laws for substitution are satisfied. This coherence problem was identified and solved by Curien [14] and Hofmann [20] who provided alternative methods for interpreting Martin-Löf type theory in lcccs (see also [15]). By using Hofmann's interpretation Clairambault and Dybjer [11, 12] could then show that there is an actual biequivalence of 2-categories.

In this paper we ask ourselves what it would take to add the missing chapter on Martin-Löf type theory and its correspondence with lcccs to the book by Lambek and Scott.

First we would need to add some material to Part 0 in the book on "Introduction to category theory", including introductions to lcccs, 2-categories, bicategories, pseudofunctors, and biequivalences. But more profoundly, our biequivalence theorem differs from Lambek and Scott's (and Seely's) equivalence theorems in important respects, since we replace Seely's category of Martin-Löf theories by a 2-category of categories with families (cwfs), with extra structure for type formers $\mathrm{I}_{\mathrm{ext}}, \Sigma, \Pi$, and pseudo cwf-morphisms which preserve the structure up to isomorphism. Thus cwfs with extra structure replace Martin-Löf theories on the "syntactic" side of the biequivalence.

This style of presenting the correspondence between "syntax" and "semantics" for Martin-Löf's dependent type theory applies equally well to the simply typed lambda calculus and the untyped lambda calculus, provided we consider subcategories of simply typed cwfs (scwfs), where types do not depend on variables, and of untyped cwfs (ucwfs), where there is only one type. As for full cwfs we need to provide extra structure for modelling λ-abstraction and application in the untyped λ-calculus and for modelling type formers in the simply typed λ-calculus.

This suggests that we ought to rewrite Lambek and Scott's Part I "Cartesian closed categories and λ-calculus" in a way which harmonizes with our presentation of the biequivalence between Martin-Löf type theory and lcccs.

Categories wifh families (cwfs) model the most basic rules of dependent type theory, those which deal with substitution, context formation, assumption, and general reasoning about equality. A key feature of cwfs is that the definition can be unfolded to yield a generalized algebraic theory in Cartmell's sense [8]. As such it suggests a language of cwf-combinators which can be used for the construction of initial cwfs (with extra structure for modelling type formers).

We prove several correspondence theorems between "syntax" in the guise of a number of cwf-based notions and "semantics" in the guise of some basic notions from category theory. Some of our theorems require "contextuality", a notion introduced by Cartmell [8] for his contextual categories. Others require "democracy", a notion introduced by Clairambault and Dybjer for their biequivalence theorems. Moreover, our equivalence theorems require strict preservation of chosen cwf-structure, while our biequivalence theorems only require preservation of cwf-structure up to isomorphism. In this way we can relate a number of notions from categorical logic such as cartesian operads, Lawvere's algebraic theories, Obtułowicz' algebraic theories of type λ-$\beta\eta$ [29], categories with finite products and limits, cccs, and lcccs, to the corresponding cwf-based notions. In addition to this we discuss different constructions of initial ucwfs, scwfs and cwfs (with extra structure) with and without explicit substitutions.

The purpose of our work is not so much to prove new results, but to suggest a new way to organize basic correspondence theorems in categorical logic, where the ucwf-scwf-cwf-sequence provides a smooth progression of the categorical model theory of untyped, simply typed, and dependently typed λ-calculi. We will also highlight some of the subtleties which arise when relating syntactic and semantic notions. Another important feature is that the correspondences between logical theories and categorical notions are now split into two phases: (i) equivalences and biequivalences between cwf-based notions and basic categorical notions, and (ii) the constructions of initial cwf-based notions. This yields an "abstract syntax" perspective of formal systems, where specific formalisms for untyped, simply typed and dependently typed λ-calculi are instances of the respective isomorphism classes of initial cwf-based notions. This is particularly important for dependent types and Martin-Löf type theory, since different authors make different choices in the exact formulation of the syntax and inference rules. Being initial in the appropriate category of cwfs is a suitable correctness criterion for these formulations.

Plan of the paper

In Section 2 we will introduce cwfs and explain their connection to Martin-Löf type theory. We will also define contextual and democratic cwfs. In Section 3 we consider unityped cwfs and show that contextual ucwfs are equivalent both to cartesian operads and to Lawvere theories. We will then add extra structure to model the untyped $\lambda\beta\eta$-calculus, and show how initial ucwfs can be built both as calculi of explicit and implicit substitutions. In Section 4 we consider simply typed cwfs. We first show the equivalence between contextual scwfs with finite product types and categories with finite products as structure. Then we show the biequivalence between democratic scwfs and categories with finite products as property. Moreover, we add function types to our contextual scwfs with finite product types and show their equivalence to cartesian closed categories as structure. (This is our analogoue of the equivalence between simply typed λ-calculi and cartesian closed categories in Lambek and Scott.) In Section 5 we discuss alternative definitions of full dependently typed cwfs. We also show an explicit construction of a free cwf. Moreover, we present two biequivalences: one is between categories with finite limits and democratic cwfs with extensional identity types, Σ-types; the other is between lcccs and democratic cwfs with extensional identity types, Σ-types and Π-types. Finally, we outline the construction of a free lccc, and how we can use our biequivalence theorem to prove that equality in this lccc is undecidable. In this section we only give an overview of these theorems and refer the reader to the journal articles [12, 10] for detailed proofs.

2 Dependent type theory and categories with families

We recall the general structure of judgments and inference rules of Martin-Löf type theory and explain its connection to the definition of cwfs.

2.1 Martin-Löf type theory

We shall here consider Martin-Löf type theory with extensional identity types in the style of [26, 27].

2.1.1 Judgments

In Martin-Löf [26] a new formulation of intuitionistic type theory as a formal system with four forms of judgment was introduced:

$$\Gamma \vdash A \ \texttt{type}$$
$$\Gamma \vdash A = A'$$
$$\Gamma \vdash a : A$$
$$\Gamma \vdash a = a' : A$$

These respectively state that A is a well-formed type; that A and A' are equal types; that a is a valid term of type A; and that a and a' are equal terms of type A. These four forms of judgments are hypothetical, that is, relative to a context $\Gamma = x_1 : A_1, \ldots, x_n : A_n$ which assigns types to free variables.

Martin-Löf [28] gave an alternative version of the theory in the form of a *substitution calculus* (see also [32]) with four new forms of judgments:

$$\Gamma \ \texttt{context}$$
$$\Gamma = \Gamma'$$
$$\Delta \to \gamma : \Gamma$$
$$\Delta \to \gamma = \gamma' : \Gamma$$

One aim was to make the rules for context formation explicit. Another was to formulate a calculus where substitution is a first-class citizen (a term constructor), and not just an operation defined by induction on the types and terms of the theory. The judgment $\Delta \to \gamma : \Gamma$ expresses that γ is a substitution (an assignment) of terms for free variables $x_1 = a_1, \ldots, x_n = a_n$, where $a_1 : A_1, \ldots, a_n : A_n$ are terms in the context Δ. Substitutions can be applied to terms and types, *e.g.* if $\Delta \to \gamma : \Gamma$ and $\Gamma \vdash A \ \texttt{type}$ and $\Gamma \vdash a : A$, then $\Delta \vdash A[\gamma]$ and $\Delta \vdash a[\gamma] : A[\gamma]$. But rather than defining $a[\gamma]$ and $A[\gamma]$ (which simultaneously substitute terms for free variables) by induction, they are instead explicit term constructors, and the effect of replacing a variable x_i by a term a_i is expressed *a posteriori* via judgmental equality.

2.1.2 Inference rules

The inference rules of intuitionistic type theory can be separated into two kinds. The first kind are the general rules, the most basic rules for reasoning with dependent types. They deal with *substitution*, *context formation*, *assumption*, and *general equality reasoning*. They form the backbone of the dependently typed structure, and bear no information yet on term and type formers. The second kind consists of the rules for type formers, such as Π, Σ, and extensional identity types. These rules are divided into *formation*, *introduction*, *elimination*, and *equality rules* (also called *computation rules*).

Categories with families capture models of the first kind of rules: the backbone of Martin-Löf type theory, independently of type and term constructors.

2.2 Categories with families

We first give the definition and then explain the connection to type theory.

2.2.1 Definition

The definition uses the category Fam of *families of sets*. Its objects are families $(U_x)_{x\in X}$. A morphism with source $(U_x)_{x\in X}$ and target $(V_y)_{y\in Y}$ is a pair consisting of a reindexing function $f : X \to Y$, and a family $(g_x)_{x\in X}$ where for each $x \in X$, $g_x : U_x \to V_{f(x)}$ is a function.

Definition 1 A category with families (cwf) consists of the following:

- A category $\mathcal{C}$ with a terminal object 1.
  Notation and terminology. We use Γ, Δ, *etc*, to range over objects of $\mathcal{C}$, and refer to those as "contexts". Likewise, we use δ, γ, *etc*, to range over morphisms, and refer to those as "substitutions". We refer to 1 as the *empty context.* We write $\langle\rangle_\Gamma \in \mathcal{C}(\Gamma, 1)$ for the terminal map, representing the empty substitution.
- A Fam-valued presheaf, *i.e.* a functor $T : \mathcal{C}^{\mathrm{op}} \to \mathrm{Fam}$.
  Notation and terminology. If $T(\Gamma) = (U_x)_{x\in X}$, we write $X = \mathrm{Ty}(\Gamma)$ and refer to its elements as *types over* Γ – we use A, B, C to range over such "types". For $A \in X = \mathrm{Ty}(\Gamma)$, we write $U_A = \mathrm{Tm}(\Gamma, A)$ and refer to its elements as *terms of type* A *in context* Γ. Finally, for $\gamma : \Delta \to \Gamma$, the functorial action yields
  $$T(\gamma) : (\mathrm{Tm}(\Gamma, A))_{A\in\mathrm{Ty}(\Gamma)} \to (\mathrm{Tm}(\Delta, B))_{B\in\mathrm{Ty}(\Delta)}$$
  consisting of a pair of a reindexing function $\_[\gamma] : \mathrm{Ty}(\Gamma) \to \mathrm{Ty}(\Delta)$ referred to as *substitution on types*, and for each $A \in \mathrm{Ty}(\Gamma)$ a function $\_[\gamma] : \mathrm{Tm}(\Gamma, A) \to \mathrm{Tm}(\Delta, A[\gamma])$ referred to as *substitution on terms.*
- A *context comprehension operation* which to a given context $\Gamma \in \mathcal{C}_0$ and type $A \in \mathrm{Ty}(\Gamma)$ assigns a context $\Gamma \cdot A$ and two projections
  $$\mathrm{p}_{\Gamma,A} : \Gamma \cdot A \to \Gamma \qquad \mathrm{q}_{\Gamma,A} \in \mathrm{Tm}(\Gamma \cdot A, A[\mathrm{p}_{\Gamma,A}])$$
  satisfying the following universal property: for all $\gamma : \Delta \to \Gamma$, for all $a \in \mathrm{Tm}(\Delta, A[\gamma])$, there is a unique $\langle\gamma, a\rangle : \Gamma \to \Delta \cdot A[\gamma]$ such that
  $$\mathrm{p}_{\Gamma,A} \circ \langle\gamma, a\rangle = \gamma \qquad \mathrm{q}_{\Gamma,A}[\langle\gamma, a\rangle] = a\,.$$
  We say that $(\Gamma \cdot A, \mathrm{p}_{\Gamma,A}, \mathrm{q}_{\Gamma,A})$ is a *context comprehension* of Γ and A.

Observe the similarity between the universal properties of context comprehension and cartesian products – the former is a skewed dependently typed version of the latter. It is also closely related to Lawvere comprehension [25].

This definition is the standard, historical definition of cwfs [16]. As notations and terminology suggest, it is closely connected to the syntax of type theory, and particularly to Martin-Löf's *substitution calculus*.

Remark 1 The structure from Definition 1 exactly matches that of Martin-Löf's *substitution calculus* mentioned before. The correspondence follows:

- $\Gamma \in \mathcal{C}_0$ models the judgment Γ `context` and $\Gamma = \Gamma' \in \mathcal{C}_0$ models $\Gamma = \Gamma'$.
- $\gamma \in \mathcal{C}(\Delta, \Gamma)$ models the judgment $\Delta \to \gamma : \Gamma$ and $\gamma = \gamma' \in \mathcal{C}(\Delta, \Gamma)$ models $\Delta \to \gamma = \gamma' : \Gamma$.
- $A \in \mathrm{Ty}(\Gamma)$ models the judgment $\Gamma \vdash A$ `type` and $A = A' \in \mathrm{Ty}(\Gamma)$ models $\Gamma \vdash A = A'$.
- $a \in \mathrm{Tm}(\Gamma, A)$ models the judgment $\Gamma \vdash a : A$ and $a = a' \in \mathrm{Tm}(\Gamma, A)$ models $\Gamma \vdash a = a' : A$.

The connection with Martin-Löf's substitution calculus contributes to the appeal of cwfs: they give rise to categorical combinators for dependent types in the same way as cccs give rise to categorical combinators for the simply typed λ-calculus [13]. However, Definition 1 has sometimes been criticized for being *too* close to the syntax, or for relying on less standard mathematical objects such as Fam-valued presheaves. We will see in Section 5.1 alternative formulations of cwfs highlighting other aspects of the structure.

Remark 2 A key feature of the notion of cwf is that it can be presented as a generalized algebraic theory in the sense of Cartmell [8].

- The generalized algebraic theory of categories introduces the sorts $\mathcal{C}_0$ and $\mathcal{C}(\Delta, \Gamma)$, the operations $\delta \circ \gamma$ and id_Γ, and associativity and identity laws.
- The Fam-valued presheaf adds the sorts $\mathrm{Ty}(\Gamma)$ and $\mathrm{Tm}(\Gamma, A)$, the operations $A[\gamma]$ and $a[\gamma]$, and associativity and identity laws for both.
- The terminal object adds the operation 1 and $\langle\rangle$, and its uniqueness law.
- Context comprehension adds the operations $\Gamma \cdot A, \mathrm{p}_{\Gamma,A}, \mathrm{q}_{\Gamma,A}$, and $\langle \gamma, a \rangle$, and the projection and surjective pairing laws.

See [16] for a complete presentation of this generalized algebraic theory.

A cwf is thus a structure $(\mathcal{C}, T, 1, \langle\rangle, \cdot, \mathrm{p}, \mathrm{q}, \langle -, - \rangle)$, subject to some equations. However, we often refer to a cwf by the first two components $(\mathcal{C}, T)$ or even the first component $\mathcal{C}$. We also freely drop arguments to operations in order to simplify notation, and for example, write p_A or p for the official $\mathrm{p}_{\Gamma,A}$, etc. We also remark that we sometimes write $\gamma : \Delta \to \Gamma$ for $\gamma \in \mathcal{C}(\Delta, \Gamma)$.

As already mentioned, cwfs only organize the core of dependent type theory, the basic structure and operations on contexts, types, terms, and substitutions. We will see later how cwfs naturally generalize well-known notions in categorical logic to dependent types. We will also see how they may be enriched with type and term formers, in order to capture Martin-Löf type theory with Σ-types, Π-types and identity types. Finally, we will see that the syntax of Martin-Löf type theory may be defined as the *initial cwf*, in a sense to be made precise later on.

2.2.2 Structure of contexts

The definition of cwfs just contains two operations on contexts: the terminal object representing the empty context and an operation mapping a context Γ and a type $A \in \mathrm{Ty}(\Gamma)$ to a new context $\Gamma \cdot A$. It is however not required that all contexts are obtained in this way. In contrast to this, Cartmell [8] added such a constraint on the structure of context for his contextual categories. We shall use the following formulation, which is equivalent to Cartmell's:

Definition 2 (Contextuality) A cwf is *contextual* iff there is a length function

$$l : \mathcal{C}_0 \to \mathbb{N}$$

such that $l(\Gamma) = 0$ iff $\Gamma = 1$, and $l(\Gamma) = n + 1$ iff there are unique $\Delta \in \mathcal{C}_0$ and $A \in \mathrm{Ty}(\Delta)$ such that $\Gamma = \Delta \cdot A$, and $l(\Delta) = n$.

Although this requirement will be used in some of our equivalence theorems, it is not part of our definition of cwf. The reason is that unlike the other parts of the definition of cwfs, it does not correspond to an inference rule of dependent type theory, and it is not expressed in the language of generalized algebraic theories. However, the free cwf is contextual.

Without going as far as requiring all contexts to be defined inductively, we sometimes wish to overcome the intrinsic distinction between contexts and types by asking that up to isomorphism, every context is represented by a type [11, 12].

Definition 3 (Democracy) A cwf is *democratic* provided each context Γ is represented by a type $\overline{\Gamma}$ in the sense that there is an isomorphism

$$\gamma_\Gamma : \Gamma \cong 1.\overline{\Gamma}$$

Democracy does not imply contextuality. However, in the presence of unit types and Σ-types, the converse holds: any context $1 \cdot A_1 \cdot \ldots \cdot A_n$ may be represented by the iterated Σ-type $\Sigma(A_1, \Sigma(A_2, \ldots, \Sigma(A_{n-1}, A_n) \ldots)) \in \mathrm{Ty}(1)$. Like contextuality, democracy does not correspond to an inference rule of dependent type theory. However, unlike contextuality, democracy can be expressed in the language of generalized algebraic theories.

2.2.3 Strict morphisms of cwfs

We will now introduce a notion of morphisms between cwfs.

Definition 4 A *(strict) cwf-morphism from cwf* $(\mathcal{C}, T_\mathcal{C})$ *to* $(\mathcal{D}, T_\mathcal{D})$ is a pair (F, σ) where $F : \mathcal{C} \to \mathcal{D}$ is a functor preserving 1 on the nose, and

$$\sigma : T_\mathcal{C} \Rightarrow T_\mathcal{D}$$

is a natural transformation between Fam-valued presheaves, preserving context comprehension on the nose.

For each $\Gamma \in \mathcal{C}_0$, σ gives $\sigma_\Gamma : \mathrm{Ty}_\mathcal{C}(\Gamma) \to \mathrm{Ty}_\mathcal{D}(F\Gamma)$, and for $A \in \mathrm{Ty}(\Gamma)$

$$\sigma_\Gamma^A : \mathrm{Tm}_\mathcal{C}(\Gamma, A) \to \mathrm{Tm}_\mathcal{D}(F\Gamma, \sigma_\Gamma A)\,.$$

It is convenient to alleviate notations and write all the components of a cwf-morphism as F; for instance writing $F(A)$ for $\sigma_\Gamma(A)$ and $F(a)$ for σ_Γ^A. Naturality of σ amounts to preservation of substitution, *i.e.*, for all $\gamma : \Delta \to \Gamma$ in $\mathcal{C}$, we have

$$F(A[\gamma]) = (FA)[F\gamma] \qquad\qquad F(a[\gamma]) = (Fa)[\gamma]\,.$$

Finally, preservation of context comprehension on the nose amounts to the fact that $F(\Gamma \cdot A) = F\Gamma \cdot FA$, with $F(\mathrm{p}_{\Gamma,A}) = \mathrm{p}_{F\Gamma,FA}$ and $F(\mathrm{q}_{\Gamma,A}) = \mathrm{q}_{F\Gamma,FA}$.
Small cwfs and strict cwfs-morphisms form a category, written **Cwf**.

3 Unityped cwfs

As we explained in the introduction, a key feature of the notion of cwf is that it can be presented as a generalized algebraic theory. As a consequence it can be seen both as a notion of model and as an (idealized) language for dependent type theory. It is therefore a suitable intermediary between traditional formal systems for dependent type theory and categorical notions of model. This paper is based on the observation that restricted classes of cwfs can play a similar role for untyped and simply typed systems. In this section we will look at cwfs with only one type and claim that they provide a similar role for untyped systems as cwfs do for dependently typed systems.

3.1 Plain ucwfs

The definition of cwfs with only one typed can be simplified as follows.

Definition 5 A unityped category with families (ucwf) consists of the following:

- A category $\mathcal{C}$ with a terminal object, written 0.
  Notation and terminlogy. We use n, m, *etc* to range over objects of $\mathcal{C}$, and refer to those as “contexts”. Likewise, we use δ, γ, *etc* to range over morphisms, and refer to those as “substitutions”. We refer to 0 as the *empty context.* We write $\langle\rangle_n \in \mathcal{C}(n, 0)$ for the terminal morphism, representing the empty substitution.

- A presheaf $\mathrm{Tm} : \mathcal{C}^{\mathrm{op}} \to \mathrm{Set}$.
  Notation and terminlogy. We refer to the elements of $\mathrm{Tm}(n)$ as the *terms of arity* n – we use a, b, etc to range over terms. Finally, for $\gamma : n \to m$, the functorial action of Tm yields a *substitution operation*

  $$\mathrm{Tm}(\gamma) = \_[\gamma] : \mathrm{Tm}(m) \to \mathrm{Tm}(n)\,.$$

- A *context comprehension operation* which to a given context $n \in \mathcal{C}_0$ assigns a context $s(n) \in \mathcal{C}_0$ along with two *projections*

  $$\mathrm{p}_n : s(n) \to n \qquad\qquad \mathrm{q}_n \in \mathrm{Tm}(s(n))$$

  satisfying the following universal property: for all $\gamma : m \to n$, for all $a \in \mathrm{Tm}(m)$, there is a unique $\langle \gamma, a\rangle : m \to s(n)$ such that

  $$\mathrm{p}_n \circ \langle \gamma, a\rangle = \gamma \qquad\qquad \mathrm{q}_n[\langle \gamma, a\rangle] = a\,.$$

Remark 3 The context comprehension operation for ucwfs amounts to the data of a representation for the presheaf

$$\mathcal{C}(-, n) \times \mathrm{Tm}(-) : \mathcal{C}^{\mathrm{op}} \to \mathrm{Set}$$

for all $n \in \mathcal{C}_0$.

If $n \in \mathbb{N}$ is a natural number, we write $\underline{n} = s^n(0)$ for the context obtained by n applications of the context comprehension operation. As the notation suggests, we think of objects of $\mathcal{C}$ as natural numbers, the *arities*, although nothing forces all objects to be natural numbers, *i.e.* of the form $\underline{n}$. We think of “terms” in $\mathrm{Tm}(\underline{n})$ as *boxes* with n inputs and one output. “Substitutions” $\gamma : \underline{m} \to \underline{n}$ are then tuples

$$\gamma = \langle\langle \ldots \langle\langle\rangle, a_1\rangle, \ldots\rangle, a_n\rangle : \underline{m} \to \underline{n}$$

where, for $a_1, \ldots, a_n \in \mathrm{Tm}(\underline{m})$. For convenience we write $\gamma = \langle a_1, \ldots, a_n\rangle$. For $a \in \mathrm{Tm}(\underline{m})$, performing the substitution $a[\gamma]$ amounts to connecting the box a_i to the i-th input of the box a, see Figure 1.

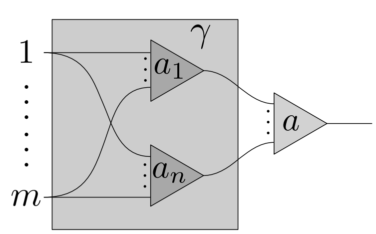


Fig. 1 Substitution as plugging boxes

Figure 1 reminds us about other categorical notions that aim to capture algebraic theories, such as Lawvere theories. We will come back later to this similarity. However, Figure 1 is misleading in one respect: it suggests that the boxes $a_1, \ldots, a_n$ have free variables in $\{1, \ldots, m\}$. In reality, in the context of ucwfs, these free variables are not first-class citizens but are obtained indirectly through sequences of *projections*. More precisely, the term

$$\pi_i^m = \mathrm{q}_{i-1}[\mathrm{p}_i] \cdots [\mathrm{p}_{m-1}] \in \mathrm{Tm}(m)$$

will serve in place of the free variable i in the context of size m. With the notations introduced, we then have the expected equation

$$\pi_i^m[\langle a_1, \ldots, a_m \rangle] = a_i$$

These notations suggest a correspondence to *cartesian operads* [33], and we will come back to this connection. Before that, writing **Ucwf** for the category of small ucwfs and strict cwf-morphisms, we note:

Proposition 1 *The category* **Ucwf** *has an initial object.*

Construction 1. From the definition of ucwf it is immediately clear that an initial ucwf $\mathcal{T}_{\mathrm{ucwf}}$ can be generated as a higher inductive type: we simultaneously define the three families $\mathcal{C}_0, \mathcal{C}(n, m)$, and $\mathrm{Tm}(n)$ where the ucwf-operations become point introduction rules and the equations become path introduction rules. (In other words, we define $\mathcal{T}_{\mathrm{ucwf}}$ by generators and relations.)

Construction 2. Alternatively, we can construct it as the presheaf of variables over the category of renamings:

- The category $\mathcal{N}$ of *renamings*, with objects $\mathbb{N}$ and $\mathcal{N}(n, m) = n^m$;
- *Presheaf* $\mathrm{Tm} : \mathcal{N}^{\mathrm{op}} \to \mathrm{Set}$ given by $\mathrm{Tm}(n) = n$ and $i[(a_1, \ldots, a_m)] = a_i$;
- *Context comprehension* as $s(n) = n + 1$, $\mathrm{p}_n = (1, \ldots, n)$, and $\mathrm{q}_n = n + 1$.

This construction and its isomorphism with $\mathcal{T}_{\mathrm{ucwf}}$ have been formalized in Agda by Brilakis [7].

Interestingly, this is the same as the free cartesian operad. This initial ucwf has the interesting property that it is *contextual*. As we will see, this requirement is necessary for the connection with cartesian operads.

3.2 Contextual ucwfs

3.2.1 Cartesian operads

Let us now consider the special case of a *contextual ucwf* $\mathcal{C}$, where the length function induces a bijection $\mathcal{C}_0 \cong \mathbb{N}$. It follows from the laws of cwfs that

$$\mathcal{C}(\underline{m}, \underline{n}) \cong \mathrm{Tm}(\underline{m})^n$$

From right to left we use the n-ary tupling introduced above, while from left-to-right we apply projections. Thus contextual ucwfs correspond to *cartesian operads*.

Definition 6 A *cartesian operad* consists of the following:

- a family $\mathrm{Tm}(n)$, where $n \in \mathbb{N}$, of "n-ary operations",
- an operation which maps $a \in \mathrm{Tm}(m)$ and $\gamma \in \mathrm{Tm}(n)^m$ to $a[\gamma] \in \mathrm{Tm}(n)$ of "operad composition" which satisfies identity and associativity laws,
- projections $\pi_i^n \in \mathrm{Tm}(n)$, such that $\pi_i^n[\langle a_1, \ldots, a_n \rangle] = a_i$.

Focusing on contextual ucwfs allows us to extract the mechanism for terms and substitutions that is at play in full cwfs, but without considering types.

3.2.2 Lawvere theories

Contextual ucwfs are also equivalent to *Lawvere theories*. The following definition is from [23].

Definition 7 A *Lawvere theory* consists of a small category $\mathcal{C}$ with (necessarily strictly associative) finite products and a strict finite-product preserving identity-on-objects functor $L : \mathcal{N} \to \mathcal{C}$. Here, $\mathcal{N}$ refers to the category of *renamings* introduced in the proof of Proposition 1, with terminal object 0 and finite products defined by $+$.

A map of Lawvere theories from $(L, \mathcal{C})$ to $(L', \mathcal{C}')$ is a (necessarily strict) finite-product preserving functor from $\mathcal{C}$ to $\mathcal{C}'$ that commutes with the functors L and L'. Lawvere theories and their maps form a category **Law**.

Note that $L : \mathcal{N} \to \mathcal{C}$ differs from the perhaps more familiar looking $\mathcal{N}_0^{\mathbf{op}} \to \mathcal{C}$, where $\mathcal{N}_0$, a skeleton of the category of finite sets and functions, has an initial object 0 and finite coproducts given by the sum of integers.

Theorem 1 *The full subcategory* $\mathbf{Ucwf}_{\mathrm{ctx}}$ *of* **Ucwf***, having as object* contextual *ucwfs, is equivalent to* **Law***.*

Proof Let $(\mathcal{C}, T_\mathcal{C})$ be a contextual ucwf. We already noted that $\mathcal{C}_0 \cong \mathbb{N}$. Moreover, we observe that there is a *unique* contextual ucwf $(\mathcal{D}, T_\mathcal{D})$, isomorphic to $(\mathcal{C}, T_\mathcal{C})$ and such that $\mathcal{D}_0 = \mathbb{N}$ with 0 terminal and $s(n) = n + 1$ for all $n \in \mathcal{D}_0$. Hence we assume from now on that contextual ucwfs have natural numbers as objects. Now, by Proposition 1, $\mathcal{N}$ is the base category of the *initial ucwf*. Hence, if $(\mathcal{C}, T_\mathcal{C})$ is a contextual ucwf, there is a unique functor

$$L : \mathcal{N} \to \mathcal{C},$$

which is the first component of a cwf-morphism. In particular, L preserves the terminal object and context comprehension on the nose, from which it immediately follows that it is identity-on-objects and strictly finite-product preserving.

If (F, σ) is a cwf-functor between contextual ucwfs (assuming *w.l.o.g.* that these ucwfs have $\mathbb{N}$ as objects), it follows that F is a morphism between the corresponding Lawvere theories. Conversely, for any morphism F between the corresponding Lawvere theories, σ can be uniquely recovered from F and projection. This yields a full and faithful functor from $\mathbf{Ucwf}_{\mathrm{ctx}}$ to **Law**.

Finally this functor is surjective on objects: from $(L, \mathcal{C})$ a Lawvere theory, there is a ucwf with category $\mathcal{C}$; terms $\mathrm{Tm}(n) = \mathcal{C}(n, 1)$, and context comprehension $s(n) = n + 1$. The universal property follows from that of the finite product. From all that, it follows that $\mathbf{Ucwf}_{\mathrm{ctx}}$ and **Law** are equivalent. □

Remark 4 In a recent paper Fiore and Voevodsky [18] prove a closely related result about C-systems, a variant of Cartmell's contextual categories. They prove that their category of Lawvere theories is isomorphic to the subcategory of C-systems whose length functions (in the definition of contextuality) are bijections.

A proof of the equivalence of Lawvere theories and cartesian operads is given by Trimble [33].

3.3 $\boldsymbol{\lambda\beta\eta}$-ucwfs

Ucwfs capture the combinatorics of terms and substitution in a similar way as cartesian operads and Lawvere theories. This primitive structure may then be enriched with operations and equations for capturing specific theories, such as the pure $\lambda\beta\eta$-calculus.

Definition 8 A $\lambda\beta\eta$-ucwf is a ucwf $(\mathcal{C}, \mathrm{Tm})$ with two more operations:

$$\begin{aligned} \lambda_n &: \mathrm{Tm}(s(n)) \to \mathrm{Tm}(n) \\ \mathrm{ap}_n &: \mathrm{Tm}(n) \times \mathrm{Tm}(n) \to \mathrm{Tm}(n) \end{aligned}$$

for all $n \in \mathcal{C}_0$, and four more equations:

$$\begin{aligned} \lambda_n(b)[\gamma] &= \lambda_m(b[\langle \gamma \circ \mathrm{p}_m, \mathrm{q}_m \rangle]) \\ \mathrm{ap}_n(c, a)[\gamma] &= \mathrm{ap}_m(c[\gamma], a[\gamma]) \\ \mathrm{ap}_n(\lambda_n(b), a) &= b[\langle \mathrm{id}_n, a \rangle] \qquad (\beta) \\ \lambda_n(\mathrm{ap}_{s(n)}(c[\mathrm{p}_n], \mathrm{q}_n)) &= c \qquad (\eta) \end{aligned}$$

for $\gamma : m \to n$, $b \in \mathrm{Tm}(s(n))$, and $c, a \in \mathrm{Tm}(n)$. A *λβ-ucwf* has the same operations, but it not subject to the (η) equation.

The definition above is natural and close to the syntax; and as we will see later on, it is the direct simplification of the notions of arrow and *Π*-

types in the simply-typed and the dependently typed case discussed later on. However, in the unityped case, this definition can be simplified dramatically.

Proposition 2 *Let* $(\mathcal{C}, \mathrm{Tm})$ *be a ucwf. Then,* $\lambda\beta\eta$*-structures on* $\mathcal{C}$ *are equivalently defined as natural isomorphisms between presheaves*

$$\mathrm{Tm}(\mathrm{s}(-)) \overset{\lambda}{\cong} \mathrm{Tm}(-)$$

where the functorial action of s *is defined as* $\mathrm{s}(\gamma) = \langle \gamma \circ \mathrm{p}_m, \mathrm{q}_m \rangle$*. More precisely,* (1) *for any* $\lambda\beta\eta$*-structure* λ *is such a natural isomorphism, and* (2) *given such a natural isomorphism, there is a unique* $\lambda\beta\eta$*-structure giving rise to it.*

Proof For *(1)*, given a $\lambda\beta\eta$-structure on $(\mathcal{C}, \mathrm{Tm})$, we first observe that λ_n is natural by the substitution law. For $c \in \mathrm{Tm}(n)$ we set $\lambda_n^{-1}(c) = \mathrm{ap}_{\mathrm{s}(n)}(c[\mathrm{p}_n], \mathrm{q}_n) \in \mathrm{Tm}(\mathrm{s}(n))$ – using β, η and the substitution law for λ, λ_n and λ_n^{-1} are inverse.

For *(2)*, given a natural iso λ_n, we set $\mathrm{ap}_n(c, a) = \lambda_n^{-1}(c)[\langle \mathrm{id}_n, a \rangle]$. The β-rule follows from the fact that $\lambda_n^{-1} \circ \lambda_n$ is the identity. The η-rule follows from the naturality of λ^{-1} plus the fact that $\lambda_n \circ \lambda_n^{-1}$ is the identity. The substitution law for λ is by naturality of λ, and the substitution law for ap is by naturality of λ^{-1}. Finally, uniqueness of the $\lambda\beta\eta$-structure (*i.e.* of ap_n) relies on the substitution rule for ap. □

3.3.1 Some related models of the untyped λ-calculus

There are many notions of model of λ-calculus, see for example Barendregt [5]. We will only briefly discuss the ones given by Obtułowicz [29], Aczel [2], and Lambek and Scott [24].

Obtułowicz's *algebraic theories of type* λ–$\beta\eta$ are Lawvere theories similar to the Lawvere theories corresponding to $\lambda\beta\eta$-ucwfs, but use an evaluation morphism ε as a primitive instead of ap. These operations are interdefinable, via $\varepsilon = \mathrm{ap}(\pi_1^2, \pi_2^2) \in \mathrm{Tm}(2)$ and $\mathrm{ap}(c, a) = \varepsilon[\langle c, a \rangle] \in \mathrm{Tm}(n)$ for $c, a \in \mathrm{Tm}(n)$.

As a basis for his notion of Frege structure, Aczel introduces a notion of *lambda structure*. This in turn is based on the notion of an *explicitly closed family*, which is a cartesian operad where $\mathrm{Tm}(n) \subseteq \mathrm{Tm}(0)^n \to \mathrm{Tm}(0)$, so that $a[\gamma]$ is function composition, and projections are the projections in the metalanguage. It is thus a cartesian operad which is well-pointed in the sense that $a, a' \in \mathrm{Tm}(n)$ and $a[\gamma] = a'[\gamma]$ for all $\gamma \in \mathcal{C}(0, n)$ implies $a = a'$. To model the $\lambda\beta$-calculus he added two operations

$$\begin{aligned} \lambda_0 &: \mathrm{Tm}(1) \to \mathrm{Tm}(0) \\ \mathrm{ap}_0 &: \mathrm{Tm}(0) \times \mathrm{Tm}(0) \to \mathrm{Tm}(0) \end{aligned}$$

The resulting notion of lambda structure is equivalent to well-pointed $\lambda\beta$-ucwfs. Since terms are *functions*, there is a unique way to define the operations λ_n and ap_n for $n > 0$ so that they satisfy the substitution laws of $\lambda\beta$-ucwfs:

$$\lambda_n(b)(\gamma) = \lambda_0(b(\gamma \circ \mathrm{p}_0, \mathrm{q}_0))$$
$$\mathrm{ap}_n(c, a)(\gamma) = \mathrm{ap}_0(c(\gamma), a(\gamma))$$

for $\gamma \in \mathcal{C}(0, n)$. The general substitution rules follow from this definition.

Lambek and Scott propose *C-monoids* as their notion of model of the untyped λ-calculus. These are monoids with extra structure coming from combinators of cartesian closed categories. C-monoids capture the equational behaviour of *closed* rather than open terms. Like in $\lambda\beta\eta$-ucwfs and cartesian closed categories, variables are dealt with indirectly as projections. But in $\lambda\beta\eta$-ucwfs, variable addressing is *external*, that is, handled by the ucwfs structure. There are no *term* constructors for pairs and projections – in particular closed terms, *i.e.* terms in $\mathrm{Tm}(0)$, do not form a C-monoid as they support no pairing and projection operations. In contrast, C-monoids handle variable addressing through pairs and projections at the *term level*.

We expect a strong relationship between C-monoids and $\lambda\beta\eta$-ucwfs with term-level pairs and projections. The proof should follow [24], encoding open terms and substitution within C-monoids via *functional completeness*.

3.3.2 Initial $\boldsymbol{\lambda\beta\eta}$-ucwfs

To conclude the discussion on $\lambda\beta\eta$-ucwf, we include a construction of the untyped λ-calculus as the *initial* such structure. For that, let us say that a strict cwf-morphism F between $\lambda\beta\eta$-ucwfs $(\mathcal{C}, \mathrm{Tm}_\mathcal{C})$ and $(\mathcal{D}, \mathrm{Tm}_\mathcal{D})$ is a *strict $\lambda\beta\eta$-ucwf-morphism* iff the action of F on terms preserves all the term constructors on the nose. Let us write $\mathbf{Ucwf}^{\lambda\beta\eta}$ for the category of small $\lambda\beta\eta$-ucwfs and strict $\lambda\beta\eta$-ucwf-morphisms. Then, we have:

Proposition 3 *The category* $\mathbf{Ucwf}^{\lambda\beta\eta}$ *has an initial object.*

Construction 1. The most direct construction is as a higher inductive type where each operation of the generalized algebraic theory of $\lambda\beta\eta$-ucwf is turned into a point constructor, and there is one path constructor for each equation. This construction can be viewed as a well-scoped variable free version of the $\lambda\sigma$-calculus of Abadi, Cardelli, Curien, and Lévy [1].

Construction 2. Another initial ucwf can be constructed from the (well-scoped) $\lambda\beta\eta$-calculus. It is a higher inductive type with the point constructors

$$\pi_i^n : \mathrm{Tm}(n)$$
$$\lambda_n : \mathrm{Tm}(s(n)) \to \mathrm{Tm}(n)$$
$$\mathrm{ap}_n : \mathrm{Tm}(n) \times \mathrm{Tm}(n) \to \mathrm{Tm}(n)$$

and with path constructors for the following equations:

$$\mathrm{ap}_n(\lambda_n(b), a) \sim_{\beta\eta} b[\langle \mathrm{id}_n, a\rangle]$$
$$\lambda_n(\mathrm{ap}_{s(n)}(c[\mathrm{p}_n], \mathrm{q}_n)) \sim_{\beta\eta} c$$

where we define substitution by induction on $\mathrm{Tm}(n)$:

$$\pi_i^n[\langle a_1, ..., a_n\rangle] = a_i$$
$$\lambda_n(b)[\gamma] = \lambda_{s(n)}(b[\langle \gamma \circ \mathrm{p}_m, \mathrm{q}_m\rangle])$$
$$\mathrm{ap}_n(c, a)[\gamma] = \mathrm{ap}_m(c[\gamma], a[\gamma])$$

for $\gamma \in \mathcal{C}(m, n)$, and then define the equational theory of $\beta\eta$-conversion:

$$\mathrm{ap}_n(\lambda_n(b), a) \sim_{\beta\eta} b[\langle \mathrm{id}_n, a\rangle]$$
$$\lambda_n(\mathrm{ap}_{s(n)}(c[\mathrm{p}_n], \mathrm{q}_n)) \sim_{\beta\eta} c$$

for $b \in \mathrm{Tm}(s(n))$ and $c \in \mathrm{Tm}(n)$. Note that this construction is an extension of the construction of $\mathcal{N}$ as an initial plain ucwf.

The two constructions, and the fact that they give rise to initial $\lambda\beta\eta$-ucwfs, have been formalised in Agda by Brilakis [7].

4 Simply-typed cwfs

En route to full cwfs, we now add types yielding *simply-typed cwfs (scwfs)*. We will then study the relationship with cartesian and cartesian closed categories.

4.1 Plain scwfs

A *simply-typed cwf (scwf)* is a cwf where the presheaf of types $\mathrm{Ty} : \mathcal{C}^{\mathrm{op}} \to \mathrm{Set}$ is constant, *i.e.* forms a set Ty not depending on the context and invariant under substitution. We can thus simplify the definition as follows:

Definition 9 An scwf consists of the following:

- A category $\mathcal{C}$ with a terminal object 1.
- A set Ty.
- A family of presheaves $\mathrm{Tm}_A : \mathcal{C}^{\mathrm{op}} \to \mathrm{Set}$ for $A \in \mathrm{Ty}$. (We also write $\mathrm{Tm}(\Gamma, A)$ for $\mathrm{Tm}_A(\Gamma)$.)
- A *context comprehension* operation which to $\Gamma \in \mathcal{C}_0$ and $A \in \mathrm{Ty}$ assigns a context $\Gamma \cdot A$ and two projections

$$\mathrm{p}_{\Gamma,A} : \Gamma \cdot A \to \Gamma \qquad\qquad \mathrm{q}_{\Gamma,A} \in \mathrm{Tm}(\Gamma \cdot A, A)$$

satisfying the following universal property: for all $\gamma : \Delta \to \Gamma$, for all $a \in \mathrm{Tm}(\Delta, A)$, there is a unique $\langle \gamma, a \rangle : \Gamma \to \Delta \cdot A$ such that

$$\mathrm{p}_{\Gamma,A} \circ \langle \gamma, a \rangle = \gamma \qquad \mathrm{q}_{\Gamma,A}[\langle \gamma, a \rangle] = a$$

We say that $(\Gamma \cdot A, \mathrm{p}_{\Gamma,A}, \mathrm{q}_{\Gamma,A})$ is a *context comprehension* of Γ and A.

Remark 5 Context comprehension for scwfs amounts to the data of a representation for the presheaf

$$\mathcal{C}(-, \Gamma) \times \mathrm{Tm}_A(-) : \mathcal{C}^{\mathrm{op}} \to \mathrm{Set}$$

for all $\Gamma \in \mathcal{C}_0$ and $A \in \mathrm{Ty}$.

An scwf is a particular kind of cwf, and a *(strict) scwf-morphism* is simply a (strict) cwf-morphism between scwfs. Small scwfs and strict cwf-morphisms form a category **Scwf**.

Scwf has an initial object, but it is not very interesting since its set of types is empty and its base category is restricted to the terminal object. Therefore we fix as a set $\mathcal{B}$ of *basic types* and consider the category $\mathcal{B}$-**Scwf** where objects are small scwfs $(\mathcal{C}, \mathrm{Ty}_{\mathcal{C}}, \mathrm{Tm}_{\mathcal{C}})$ together with an interpretation function $[\![-]\!]_{\mathcal{C}} : \mathcal{B} \to \mathrm{Ty}$. Morphisms are strict scwfs-morphisms that commute with the interpretation.

Proposition 4 *For all sets $\mathcal{B}$ the category $\mathcal{B}$-**Scwf** has an initial object.*

Proof The initial $\mathcal{B}$-scwf, also called the *free scwf over a set of types $\mathcal{B}$*, can be constructed in much the same way as we constructed initial ucwfs, either directly as a higher inductive type formalizing scwfs combinators and their equations, or as a typed version of the category $\mathcal{N}$ of Proposition 1. For the second construction we let $\mathrm{Ty} = \mathcal{B}$. Then we define

$$\mathcal{C}_0 = \mathrm{List}(\mathrm{Ty})$$
$$\mathcal{C}(\Gamma, [A_1, \ldots, A_n]) = \mathrm{Tm}(\Gamma, A_1) \times \cdots \times \mathrm{Tm}(\Gamma, A_n)$$

where $\mathrm{Tm}(\Gamma, A)$ is the set $\{\mathrm{var}_n(i) \mid A_i = A\}$ of *variables*, for $\Gamma = [A_1, \ldots, A_n]$. Moreover, we define the *projections* as

$$\mathrm{p}_{\Gamma,A} = (\mathrm{var}_{n+1}(1), \ldots, \mathrm{var}_{n+1}(n))$$
$$\mathrm{q}_{\Gamma,A} = \mathrm{var}_{n+1}(n+1)\,,$$

and substitution $\mathrm{var}_n(i)[\langle a_1, \ldots, a_n \rangle] = a_i$. □

Just as for ucwfs, the free scwf over a set $\mathcal{B}$ is contextual. In analogy with Section 3.2 we could relate scwfs to coloured cartesian operads (multi-categories) and multi-sorted Lawvere theories, but we omit the unsurprising details. Instead, we discuss their relationship with *cartesian categories.*

4.2 Finite products as structure

Cartesian categories (categories with finite products) are categories with a terminal object and binary cartesian products. Straightforward as it seems, this definition hides some subtleties which are put to the forefront when one considers the associated notions of *morphism* between cartesian categories. Namely, does the mere *existence* of a product for any two objects suffice to obtain a cartesian category, or are the finite products part of the *data* of a cartesian category? Texts in category theory adopt one view or the other, not always explicitely. In Lambek and Scott's book, the latter view is explicitely adopted; and morphisms between cartesian categories must preserve this explicit data *on the nose*.

Definition 10 A *cartesian category (with structure)* consists of the following:

- A category $\mathcal{C}$ with a terminal object 1,
- A *product operation* which to any $A, B \in \mathcal{C}_0$ assigns an object $A \times B \in \mathcal{C}_0$ and two projections

$$\mathrm{fst}_{A,B} : A \times B \to A \qquad \mathrm{snd}_{A,B} : A \times B \to B$$

satisfying the following universal property: for all $a : X \to A$, for all $b : X \to B$, there is a unique $\langle a, b \rangle : X \to A \times B$ such that

$$\mathrm{fst}_{A,B} \circ \langle a, b \rangle = a \qquad \mathrm{snd}_{A,B} \circ \langle a, b \rangle = b\,.$$

A *strict cartesian functor* from cartesian category $\mathcal{C}$ (leaving implicit the other components) to $\mathcal{D}$ is a functor $F : \mathcal{C} \to \mathcal{D}$ preserving the structure *on the nose*, *i.e.* for all $A, B \in \mathcal{C}_0$, $F(A \times_{\mathcal{C}} B) = F(A) \times_{\mathcal{D}} F(B)$, and

$$F(\mathrm{fst}^{\mathcal{C}}_{A,B}) = \mathrm{fst}^{\mathcal{D}}_{F(A),F(B)} \qquad F(\mathrm{snd}^{\mathcal{C}}_{A,B}) = \mathrm{snd}^{\mathcal{D}}_{F(A),F(B)}\,.$$

We write **CCs** for the category of small cartesian categories (with structure) and strict cartesian functors, preserving the structure on the nose.

We will later consider another notion of cartesian category, where we are content with the *mere existence* of a cartesian product for any two objects, but the only data part of the structure is the basic category $\mathcal{C}$ – there are no *chosen products*. This, of course, constrains the maps: the notion of strict cartesian morphisms as above would make no sense without chosen structure.

We observe in passing that from the preservation of projections and the universal property, it follows directly that strict cartesian functors also preserve tuples on the nose, in the sense that $F(\langle a, b \rangle) = \langle F(a), F(b) \rangle$.

4.2.1 Finite product types

We now wish to compare cartesian categories with structure, as above, with scwfs. They are very similar, the main difference being that scwfs distinguish *contexts* and *types*; whereas cartesian categories do not have this distinction. In particular, when constructing an scwf from a cartesian category we must recover the *types* as the objects of $\mathcal{C}$. It follows that the resulting scwfs supports a finite product operation on types. Looking for an equivalence, we define what it means for an scwf to support finite product types and introduce a binary product type, and a unit type $\mathbb{N}_1$.

Definition 11 An $\mathbb{N}_1$-*structure* on an scwf $\mathcal{C}$ consists of a type $\mathbb{N}_1 \in \mathrm{Ty}$, and for each Γ a term $0_1 \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, \mathbb{N}_1)$ such that for all $c \in \mathrm{Tm}(\Gamma, \mathbb{N}_1)$, $0_1 = c$.

Definition 12 A $\times$-*structure* on an scwf $\mathcal{C}$ consists of, for each $\Gamma \in \mathcal{C}_0$ and $A, B \in \mathrm{Ty}$, a type $A \times B \in \mathrm{Ty}$ and term formers

$$\begin{aligned}
\mathrm{fst}_{\Gamma,A,B}(-) &: \mathrm{Tm}(\Gamma, A \times B) \to \mathrm{Tm}(\Gamma, A) \\
\mathrm{snd}_{\Gamma,A,B}(-) &: \mathrm{Tm}(\Gamma, A \times B) \to \mathrm{Tm}(\Gamma, B) \\
\langle -, - \rangle &: \mathrm{Tm}(\Gamma, A) \times \mathrm{Tm}(\Gamma, B) \to \mathrm{Tm}(\Gamma, A \times B)
\end{aligned}$$

such that, for appropriate γ, a, b, c

$$\begin{aligned}
\mathrm{fst}(\langle a, b \rangle) &= a \\
\mathrm{snd}(\langle a, b \rangle) &= b \\
\langle \mathrm{fst}(c), \mathrm{snd}(c) \rangle &= c \\
\langle a, b \rangle[\gamma] &= \langle a[\gamma], b[\gamma] \rangle \,.
\end{aligned}$$

We thus have a type formation rule for $A \times B$ and term formation rules for projections and pairs. The first three equations are the usual rules for product types with surjective pairing, while the last one states stability under substitution. By an *scwf with finite product types* we mean an *scwf with an* $\mathbb{N}_1$-*structure and a* $\times$-*structure.*

Remark 6 Having a $\times$-structure on scwf $\mathcal{C}$ amounts to requiring that there is a binary type former $\times$ and a natural isomorphism of preheaves

$$\mathrm{Tm}_{\mathcal{C}}(-, A) \times \mathrm{Tm}_{\mathcal{C}}(-, B) \cong \mathrm{Tm}_{\mathcal{C}}(-, A \times B) \,.$$

Likewise, an $\mathbb{N}_1$-structure corresponds to a type $\mathbb{N}_1$ and a natural isomorphism between $\mathrm{Tm}_{\mathcal{C}}(-, \mathbb{N}_1)$ and the constant singleton presheaf.

The product type structure on scwfs should be *preserved* by morphisms.

Definition 13 If $\mathcal{C}, \mathcal{D}$ have the adequate structure, a strict cwf-morphism $F : \mathcal{C} \to \mathcal{D}$ *(strictly) preserves* $\mathbb{N}_1$-*structure* if $F({\mathbb{N}_1}^{\mathcal{C}}) = {\mathbb{N}_1}^{\mathcal{D}}$, and $F(0_1^{\mathcal{C}}) = 0_1^{\mathcal{D}}$.

Similarly, it *(strictly) preserves $\times$-structure* if $F(A \times^{\mathcal{C}} B) = F(A) \times^{\mathcal{D}} F(B)$, $F(\mathrm{fst}^{\mathcal{C}}_{A,B}(c)) = \mathrm{fst}^{\mathcal{D}}_{FA,FB}(F(c))$ and $F(\mathrm{snd}^{\mathcal{C}}_{A,B}(c)) = \mathrm{snd}^{\mathcal{D}}_{FA,FB}(F(c))$.

We write $\mathbf{Scwf}^{\mathbb{N}_1,\times}$ for the category where the objects are small scwfs with product types, and morphisms are strict structure-preserving cwf-morphisms.

Given an scwf with product types we construct a cartesian category as follows. First, we define the *category of types and terms in context*.

Definition 14 Let $\mathcal{C}$ be an scwf with products. We define the category of *types and terms in context* $\Gamma \in \mathcal{C}_0$ as having: *(1)* as objects, Ty; *(2)* as morphisms from $A \in \mathrm{Ty}$ to $B \in \mathrm{Ty}$, the terms $b \in \mathrm{Tm}(\Gamma \cdot A, B)$, composed with, for $b \in \mathrm{Tm}(\Gamma \cdot A, B)$ and $c \in \mathrm{Tm}(\Gamma \cdot B, C)$,

$$c \circ b = c[\langle \mathrm{p}_{\Gamma,A}, b \rangle]$$

with identity $\mathrm{id}_{\Gamma,A} = \mathrm{q}_{\Gamma,A} \in \mathrm{Tm}(\Gamma \cdot A, A)$. We write $\mathrm{Ty}(\Gamma)$ for this category.

It follows that $\mathrm{Ty}(\Gamma)$ is a cartesian category with structure for all $\Gamma \in \mathcal{C}_0$.

Lemma 1 *For any Γ, we let $\mathbb{N}_1$ be the chosen terminal object in $\mathrm{Ty}(\Gamma)$. For every $A, B \in \mathrm{Ty}$, we let their product be $A \times B$ and the projections be*

$$\mathrm{fst}(\mathrm{q}_{\Gamma,A\times B}) \in \mathrm{Ty}(\Gamma)(A \times B, A) \qquad \mathrm{snd}(\mathrm{q}_{\Gamma,A\times B}) \in \mathrm{Ty}(\Gamma)(A \times B, B)$$

We omit the (straightforward) proof. This entails that given an scwf with products $\mathcal{C}$, there is one canonical cartesian category with structure, the cartesian category of closed terms $\mathrm{Ty}_{\mathcal{C}}(1_{\mathcal{C}})$.

Proposition 5 *There is a functor*

$$\mathbf{C} : \mathbf{Scwf}^{\mathbb{N}_1,\times} \to \mathbf{CCs}$$

which to any scwf with products $\mathcal{C}$ associates $\mathrm{Ty}_{\mathcal{C}}(1_{\mathcal{C}})$, and which to any structure-preserving cwf-morphism $F : \mathcal{C} \to \mathcal{D}$ associates $\mathbf{C}(F) : \mathrm{Ty}_{\mathcal{C}}(1_{\mathcal{C}}) \to \mathrm{Ty}_{\mathcal{D}}(1_{\mathcal{D}})$ given by the action of F on types and terms.

4.2.2 From cartesian categories to scwfs

Since $\mathbf{C}$ forgets the structure of contexts and only remembers closed types, we expect a functor $\mathbf{L}$ in the opposite direction to somehow reconstruct contexts. There are two natural candidates for this. The first is to reverse the effect of $\mathbf{C}$ by reconstructing the context formally, in an operation analogous to the construction of the cartesian category of *polynomials* in Lambek and Scott. We will detail this below. In this way we do not directly get an equivalence, because if $\mathcal{C}$ is an arbitrary scwf, the contexts of $\mathbf{LC}(\mathcal{C})$ are *generated inductively* from types. However, we get an equivalence for *contextual* scwfs.

The other option is to let the category of contexts of $\mathbf{L}(\mathcal{C})$ be $\mathcal{C}$, reflecting the dual role of objects in cartesian categories as both contexts and types. Context comprehension is defined via finite products. As simple as it looks, this construction does *not* yield an equivalence even when restricting scwfs. (It would if one restricted to democratic scwfs such that for each $\Gamma \in \mathcal{C}_0$, we have $\Gamma = 1 \cdot \overline{\Gamma}$, but this is not a natural hypothesis as it is not satisfied by the syntax). It is, however, behind the *biequivalence* between scwfs and cartesian categories as property which we shall discuss in the following subsection.

Definition 15 If $\mathcal{C}$ is a cartesian category, we define an scwf $\mathbf{L}(\mathcal{C})$ analogously to the free scwf in Proposition 4. The set of *types* is $\mathrm{Ty} = \mathcal{C}_0$. We define

$$\mathbf{L}(\mathcal{C})_0 = \mathrm{List}(\mathcal{C}_0)$$
$$\mathbf{L}(\mathcal{C})(\Gamma, [A_1, \dots, A_n]) = \mathcal{C}(\Pi\Gamma, A_1) \times \cdots \times \mathcal{C}(\Pi\Gamma, A_n)$$

where $\Pi[B_1, \dots, B_m] = (\dots(B_1 \times B_2) \cdots \times B_m)$ and $\Pi[] = 1$. The *terms* are

$$\mathrm{Tm}(\Gamma, A) = \mathcal{C}(\Pi\Gamma, A).$$

Substitution is defined as

$$a[(\gamma_1, \dots, \gamma_n)] = a \circ \langle\langle \dots \langle\gamma_1, \gamma_2\rangle, \dots, \rangle, \gamma_n\rangle \in \mathrm{Tm}(\Delta, A)$$

for $(\gamma_1, \dots, \gamma_n) : \Delta \to [A_1, \dots, A_n]$ and $a \in \mathrm{Tm}(\Gamma, A)$ and *composition* in $\mathbf{L}(\mathcal{C})$ by $(a_1, \dots, a_n) \circ \gamma = (a_1[\gamma], \dots, a_n[\gamma])$.

For $\Gamma = [A_1, \dots, A_n]$ and $1 \leq i \leq n$, we write $\mathrm{var}_n(i) \in \mathrm{Tm}(\Gamma, A_i)$ for the corresponding variable, obtained as the n-ary projection. For $\Gamma = [A_1, \dots, A_n] \in \mathbf{L}(\mathcal{C})_0$ and $A \in \mathrm{Ty}$ we let $\Gamma \cdot A = [A_1, \dots, A_n, A]$. The *projections* are defined by

$$\mathrm{p}_{\Gamma,A} = (\mathrm{var}_{n+1}(1), \dots, \mathrm{var}_{n+1}(n))$$
$$\mathrm{q}_{\Gamma,A} = \mathrm{var}_{n+1}(n+1)\,.$$

We thus get an scwf with finite product types: the $\mathbb{N}_1$-structure is the terminal object of $\mathcal{C}$ and $A \times B$ is given by the cartesian product of $\mathcal{C}$. If $c \in \mathrm{Tm}(\Gamma, A \times B)$, the projections

$$\mathrm{fst}_{A,B}(c) = \mathrm{fst}_{A,B} \circ c \qquad\qquad \mathrm{snd}_{A,B}(c) = \mathrm{snd}_{A,B} \circ c$$

are immediate. (Note the overloading of fst and snd.)

We observe that our construction yields that $\mathbf{C}(\mathbf{L}(\mathcal{C})) = \mathcal{C}$ for each cartesian category with structure $\mathcal{C}$. This construction can be lifted to a functor

$$\mathbf{L} : \mathbf{CCs} \to \mathbf{Scwf}^{\mathbb{N}_1,\times}$$

where given $\mathbf{L}(F) : \mathbf{L}(\mathcal{C}) \to \mathbf{L}(\mathcal{D})$ is obtained by letting $F : \mathcal{C} \to \mathcal{D}$ act component-wise. It is thus clear that all structure is preserved. Finally, $\mathbf{C}$ and

$\mathbf{L}$ do *not* yet form an equivalence: indeed, $\mathbf{L}(\mathbf{C}(\mathcal{C}))$ is always *contextual*, as its contexts are inductively generated; whereas $\mathcal{C}$ might not be. For instance, any context of $\mathcal{C}$ which is not described as an iterated context extension of types is not accounted for in $\mathbf{L}(\mathbf{C}(\mathcal{C}))$. However, we have:

Theorem 2 *The functors* $\mathbf{C}$ *and* $\mathbf{L}$ *form an equivalence of categories:*

$$\mathbf{CCs} \underset{\mathbf{C}}{\overset{\mathbf{L}}{\rightleftarrows}} \mathbf{Scwf}^{\mathbb{N}_1,\times}_{\mathrm{ctx}}$$

Proof It only remains to observe that for any contextual scwf $\mathcal{C}$ with finite products we have the isomorphism $\mathcal{C} \cong \mathbf{L}(\mathbf{C}(\mathcal{C}))$, where this isomorphism sends a context $1 \cdot A_1 \cdot \ldots \cdot A_n$ to $[A_1, \ldots, A_n]$. □

Theorem 2 is an analogue (without functions) to Theorem 11.3 in Lambek and Scott [24], where $\mathbf{Scwf}^{\mathbb{N}_1,\times}_{\mathrm{ctx}}$ plays the role of the category of typed λ-calculi.

4.3 Finite products as property

4.3.1 Cartesian categories as property

We now define a notion of cartesian category where finite products are defined as a property of a category.

Definition 16 Let $\mathcal{C}$ be a category. It is *cartesian (as a property)* if *there exists* a terminal object in $\mathcal{C}$, and if for any two objects $A, B \in \mathcal{C}_0$, *there exists* a cartesian product of A and B, *i.e.* a triple (P, π, π') with $P \in \mathcal{C}_0$, $\pi : P \to A$, $\pi' : P \to B$ satisfying the usual universal property.

By "there exists", we mean *mere existence*. The choice of terminal objects and the assignment of (P, π, π') from A and B are not part of the structure. A cartesian category is just a particular kind of category with no additional data. Likewise, cartesian functors may be defined as:

Definition 17 A functor $F : \mathcal{C} \to \mathcal{D}$ is *cartesian* if the image of a terminal object is terminal, and the image of a product (P, π, π') is a product $(FP, F\pi, F\pi')$. We let $\mathbf{CCp}^2$ be the 2-category of small cartesian categories (with property) as objects, cartesian functors as 1-cells, and natural transformations as 2-cells.

Thus cartesian functors are just certain functors with no additional data. We regard $\mathbf{CCp}^2$ as a 2-category because we will prove a *biequivalence* rather than an equivalence. Indeed, scwfs have chosen structure while cartesian categories (with property) do not. Going from an scwf to a cartesian category and back the structure is forgotten and then *chosen* again, and we cannot recover the original scwf up to isomorphism, only up to equivalence.

4.3.2 Pseudo scwf-morphisms

Previously, we defined a notion of strict scwf-morphism, but this does not match the notion of cartesian functor as property. To address this mismatch we need a more relaxed notion of *pseudo cwf-morphisms* where structure is only preserved *up to isomorphism.*

Definition 18 A *pseudo scwf-morphism* from scwf $\mathcal{C}$ to scwf $\mathcal{D}$ consists of a functor $F : \mathcal{C} \to \mathcal{D}$, a function $F^{\mathrm{Ty}} : \mathrm{Ty}^{\mathcal{C}} \to \mathrm{Ty}^{\mathcal{D}}$, and a family

$$F_A^{\mathrm{Tm}} : \mathrm{Tm}_A^{\mathcal{C}}(-) \Rightarrow \mathrm{Tm}_{F^{\mathrm{Ty}}A}^{\mathcal{D}}(-)$$

of natural transformations. We write $F_{\Gamma,A}^{\mathrm{Tm}}$ for the component of F_A^{Tm} on Γ.

These data are subject to the conditions that *(1)* $F1$ is terminal in $\mathcal{D}$, and *(2)* for all $\Gamma \in \mathcal{C}_0$, $A \in \mathrm{Ty}^{\mathcal{C}}(\Gamma)$, the triple $(F(\Gamma \cdot A), F(\mathrm{p}_{\Gamma,A}), F_{\Gamma,A}^{\mathrm{Tm}}(\mathrm{q}_{\Gamma,A}))$ is a context comprehension of $F\Gamma$ and $F^{\mathrm{Ty}}(A)$ in $\mathcal{D}$.

For strict cwf-morphisms, the triple $(F(\Gamma \cdot A), F(\mathrm{p}_{\Gamma,A}), F_{\Gamma,A}^{\mathrm{Tm}}(\mathrm{q}_{\Gamma,A}))$ must coincide with the context comprehension of $F\Gamma$ and $F^{\mathrm{Ty}}(A)$ chosen by the scwf structure of $\mathcal{D}$. Here, we drop that assumption.

For related reasons, the equivalence will work slightly differently than in Section 4.2. There the equivalence followed the slogan "the cartesian category corresponding to a scwf is its category of closed terms". However, in Theorem 2 products in a cartesian category $\mathcal{C}$ are used in a central way in the definition of terms in $\mathbf{L}(\mathcal{C})$. These can be chosen once and for all, but the preservation of products up to isomorphism leads to an unwieldy definition of the functorial action of $\mathbf{L}$. Instead, we adopt a simpler approach under the slogan "the cartesian category corresponding to an scwf is its base category". With the aim of getting a biequivalence rather than an equivalence, the requirement of *contextuality* will be replaced by the weaker *democracy.* This additional structure must then also be preserved.

Definition 19 A *democratic* pseudo scwf-morphism between democratic scwfs $\mathcal{C}$ and $\mathcal{D}$ additionally has, for each $\Gamma \in \mathcal{C}_0$, as isomorphism $\mathrm{d}_\Gamma : F^{\mathrm{Ty}}(\overline{\Gamma}) \cong \overline{F\Gamma}$ in the category $\mathrm{Ty}^{\mathcal{C}}(1)$, subject to a coherence diagram expressing that $F(\gamma_\Gamma) = \gamma_{F\Gamma}$ modulo some transports (see [12], p.19).

A democratic scwf always has finite product types defined by $A \times B = \overline{1 \cdot A \cdot B}$. So our biequivalence holds without adding these as extra structure.

Definition 20 We write $\mathbf{Scwf}^2_{\mathrm{dem}}$ for the 2-category having democratic small scwfs as objects, democratic pseudo scwf-morphisms as 1-cells, and natural transormations between the underlying functors as 2-cells.

One may wonder why the definition of 2-cells does not have a type component. Such a component could be added, but then the biequivalence requires a coherence law which makes it redundant (see [10], p.7 and the expanded discussion in Appendix B).

4.3.3 The biequivalence

We now provide the components of the biequivalence. First we observe that there is a forgetful 2-functor

$$\mathbf{C}^2 : \mathbf{Scwf}^2_{\mathrm{dem}} \to \mathbf{CCp}^2$$

which to a democratic scwf associates its base category, to a democratic pseudo cwf-morphism associates its base functor, and leaves 2-cells unchanged. That this is well-defined relies on the facts that *(1)* if $\mathcal{C}$ is a democratic scwf, then, the base category $\mathcal{C}$ is cartesian (with property), because products in $\mathcal{C}$ may be defined as $\Gamma \times \Delta = \Gamma \cdot \overline{\Delta}$; *(2)* if F is a democratic pseudo cwf-morphism, then the fact that the base functor preserves products follows from preservation of context comprehension and democracy.

We now construct a 2-functor in the other direction.

Proposition 6 *There is a* 2*-functor* $\mathbf{L}^2 : \mathbf{CCp}^2 \to \mathbf{Scwf}^2_{\mathrm{dem}}$.

Proof In every cartesian category (with property) $\mathcal{C}$ we choose a terminal object 1 and a product $(A \times B, \mathrm{fst}_{A,B}, \mathrm{snd}_{A,B})$ for every $A, B \in \mathcal{C}_0$.

Now, we can turn any cartesian category $\mathcal{C}$ into an scwf in the following way. Its category of contexts is $\mathcal{C}$. Its set of types is $\mathrm{Ty} = \mathcal{C}_0$. If $\Gamma \in \mathcal{C}_0$ and $A \in \mathrm{Ty}$, the terms are $\mathrm{Tm}_{\mathcal{C}}(\Gamma, A) = \mathcal{C}(\Gamma, A)$. The context comprehension is simply given by finite products. Democracy is the isomorphism $\Gamma \cong 1 \times \Gamma$. Likewise, a cartesian functor $F : \mathcal{C} \to \mathcal{D}$ is extended to types and terms in the obvious way, and yields a democratic pseudo cwf-morphism. □

Theorem 3 *The* 2*-functors* $\mathbf{C}^2$ *and* $\mathbf{L}^2$ *form a biequivalence of* 2*-categories:*

$$\mathbf{CCp}^2 \underset{\mathbf{C}^2}{\overset{\mathbf{L}^2}{\rightleftarrows}} \mathbf{Scwf}^2_{\mathrm{dem}}$$

Proof We have $\mathbf{C}^2\mathbf{L}^2 = \mathrm{I}_{\mathbf{CCp}}$. To obtain a biequivalence we must construct pseudonatural transformations of 2-functors (or pseudofunctors)

$$\mathrm{I}_{\mathbf{Scwf}^2_{\mathrm{dem}}} \underset{\epsilon}{\overset{\eta}{\rightleftarrows}} \mathbf{L}^2\mathbf{C}^2$$

which are inverses up to invertible modifications. Concretely, we construct, for each democratic scwf $\mathcal{C}$, pseudo cwf-morphisms $\eta_{\mathcal{C}} : \mathcal{C} \to \mathbf{L}^2\mathbf{C}^2\mathcal{C}$ and

$\epsilon_{\mathcal{C}} : \mathbf{L}^2\mathbf{C}^2\mathcal{C} \to \mathcal{C}$, both with the identity functor as base component. Besides we have $\eta_{\mathcal{C}}^{\mathrm{Ty}}(A) = 1.A$ and $\epsilon_{\mathcal{C}}^{\mathrm{Ty}}(\Gamma) = \overline{\Gamma}$ on types, and

$$\eta_{\mathcal{C}}^{\mathrm{Tm}}{}_{\Gamma,A}(a) = \langle\langle\rangle, a\rangle \qquad \epsilon_{\mathcal{C}}^{\mathrm{Tm}}{}_{\Delta,\Gamma}(\gamma) = \mathrm{q}_{1,\overline{\Gamma}}[\gamma_\Gamma \circ \Gamma]$$

on terms. These two pseudo cwf-morphisms are pseudonatural in $\mathcal{C}$, and form an equivalence in $\mathbf{Scwf}^2_{\mathrm{dem}}$: $\eta_{\mathcal{C}} \circ \epsilon_{\mathcal{C}}$ and $\mathrm{id}_{\mathbf{C}^2\mathbf{L}^2\mathcal{C}}$ (resp. $\epsilon_{\mathcal{C}} \circ \eta_{\mathcal{C}}$ and $\mathrm{id}_{\mathcal{C}}$) are related by invertible 2-cells with the identity as component. Finally, by unfolding definitions it follows that the invertible 2-cells satisfy the coherence condition for modifications between pseudonatural transformations. □

Together, Theorems 2 and 3 give two different points of view on the correspondence between scwfs and cartesian categories. It is interesting that we need to use a biequivalence not just for Martin-Löf's type theory and locally cartesian closed categories as in [12], but already for cartesian categories if one is to omit chosen structure. For cartesian categories we can build both an equivalence (with structure) and a biequivalence (with property), whereas it seems that only the latter is possible for finitely complete categories and locally cartesian closed categories.

4.4 Adding function types

Before going on to the dependently typed case, we mention how to add function types to the previous equivalences and biequivalences. First, we recall:

Definition 21 A *cartesian closed category (with structure)* is a cartesian category (with structure) and an operation which to $A, B \in \mathcal{C}_0$ assigns an object $A \Rightarrow B \in \mathcal{C}_0$ with an *evaluation*

$$\varepsilon_{A,B} : (A \Rightarrow B) \times A \to B$$

such that for all $f : C \times A \to B$, $\exists! h : C \to A \Rightarrow B$ s.t. $\varepsilon_{A,B} \circ (h \times A) = f$.

We write **CCCs** for the category having small cccs (with structure) as objects, and as morphisms, functors $F : \mathcal{C} \to \mathcal{D}$ preserving the structure on the nose, *i.e.* $F(A \Rightarrow^{\mathcal{C}} B) = FA \Rightarrow^{\mathcal{D}} FB$, and $F(\varepsilon^{\mathcal{C}}_{A,B}) = \varepsilon^{\mathcal{D}}_{FA,FB}$.

Likewise, we may add function types to scwfs in the following way.

Definition 22 A $\Rightarrow$*-structure* on an scwf $\mathcal{C}$ consists of, for each $\Gamma \in \mathcal{C}_0$ and $A, B \in \mathrm{Ty}$, a type $A \Rightarrow B$ along with term formers

$$\lambda_{\Gamma,A,B} : \mathrm{Tm}(\Gamma \cdot A, B) \to \mathrm{Tm}(\Gamma, A \Rightarrow B)$$
$$\mathrm{ap}_{\Gamma,A,B} : \mathrm{Tm}(\Gamma, A \Rightarrow B) \times \mathrm{Tm}(\Gamma, A) \to \mathrm{Tm}(\Gamma, B)$$

s.t., for $a \in \mathrm{Tm}(\Gamma, A), b \in \mathrm{Tm}(\Gamma \cdot A, B), c \in \mathrm{Tm}(\Gamma, A \Rightarrow B), \gamma \in \mathcal{C}(\Delta, \Gamma)$:

$$\lambda_{\Gamma,A,B}(b)[\gamma] = \lambda_{\Delta,A,B}(b[\langle \gamma \circ \mathrm{p}_{\Delta,A}, \mathrm{q}_{\Delta,A}\rangle])$$
$$\mathrm{ap}_{\Gamma,A,B}(c,a)[\gamma] = \mathrm{ap}_{\Delta,A,B}(c[\gamma], a[\gamma])$$
$$\mathrm{ap}_{\Gamma,A,B}(\lambda_{\Gamma,A,B}(b), a) = b[\langle \mathrm{id}_\Gamma, a\rangle]$$
$$\lambda_{\Gamma,A,B}(\mathrm{ap}_{\Gamma\cdot A,A,B}(c[\mathrm{p}_{\Gamma,A}], \mathrm{q}_{\Gamma,A})) = c\,.$$

If $\mathcal{C}, \mathcal{D}$ have $\Rightarrow$-structure, a (strict) cwf-morphism $F : \mathcal{C} \to \mathcal{D}$ *preserves* it if $F(A \Rightarrow^{\mathcal{C}} B) = FA \Rightarrow^{\mathcal{D}} FB$, and $F(\mathrm{ap}^{\mathcal{C}}_{\Gamma,A,B}(c,a)) = \mathrm{ap}^{\mathcal{D}}_{F\Gamma,FA,FB}(F(c), F(a))$.

Let us write $\mathbf{Scwf}^{\mathbb{N}_1,\times,\Rightarrow}_{\mathrm{ctx}}$ for the category having as objects small contextual scwfs with an $\mathbb{N}_1$-structure, a $\times$-structure and a $\Rightarrow$-structure, and as morphisms the strict cwf-morphisms preserving this structure on the nose.

Theorem 4 *The categories* **CCCs** *and* $\mathbf{Scwf}^{\mathbb{N}_1,\times,\Rightarrow}_{\mathrm{ctx}}$ *are equivalent.*

Proof Straightforward extension of Theorem 2, which boils down to the definition of evaluation from application and vice versa. □

As discussed after Theorem 2, we regard this as our version of one of the main results of the Lambek and Scott book, namely Theorem 3.11, the equivalence between simply-typed λ-calculi and cartesian closed categories, where $\mathbf{Scwf}^{\mathbb{N}_1,\times,\Rightarrow}_{\mathrm{ctx}}$ plays the role of the category of typed λ-calculi. And indeed, the simply-typed λ-calculus arises as the free scwf with $\Rightarrow$-structure over a set of types. If $\mathcal{B}$ is a set of basic types, we consider $\mathcal{B}$-$\mathbf{Scwf}^{\Rightarrow}_{\mathrm{ctx}}$ to be the category with objects scwfs with $\Rightarrow$-structure together with an interpretation function $[\![-]\!]_{\mathcal{C}} : \mathcal{B} \to \mathrm{Ty}$; and as morphisms the strict scwf-morphisms that preserve $\Rightarrow$-structure and commute with the interpretation. Then we have:

Proposition 7 *For all sets $\mathcal{B}$, the category $\mathcal{B}$-*$\mathbf{Scwf}^{\Rightarrow}$ *has an initial object.*

Construction 1. Just as for ucwfs and scwfs, we can immediately turn the definition of scwf with $\times$-structure into a definition of a free one as a higher inductive type. The resulting theory is a well-scoped variable-free version of the typed $\lambda\sigma$-calculus with base types in $\mathcal{B}$.

Construction 2. Alternatively, we can construct an scwf with $\Rightarrow$-structure free over $\mathcal{B}$ from a well-scoped version of the simply-typed $\lambda\beta\eta$-calculus. The construction follows that of Proposition 4, except that the terms are now inductively defined with the following three constructors:

$$\pi^n_i : \mathrm{Tm}([A_1, \ldots, A_i, \ldots, A_n], A_i)$$
$$\lambda_{\Gamma,A,B} : \mathrm{Tm}(\Gamma \cdot A, B) \to \mathrm{Tm}(\Gamma, A \Rightarrow B)$$
$$\mathrm{ap}_{\Gamma,A,B} : \mathrm{Tm}(\Gamma, A \Rightarrow B) \times \mathrm{Tm}(\Gamma, A) \to \mathrm{Tm}(\Gamma, B)\,.$$

The definition of substitution is then extended with

$$\lambda_{\Gamma,A,B}(b)[\gamma] = \lambda_{\Delta,A,B}(b[\langle \gamma \circ \mathrm{p}_{\Delta,A}, \mathrm{q}_{\Delta,A}\rangle])$$
$$\mathrm{ap}_{\Gamma,A,B}(c,a)[\gamma] = \mathrm{ap}_{\Delta,A,B}(c[\gamma], a[\gamma])$$

for $\gamma \in \mathcal{C}(\Delta, \Gamma)$. Finally we define the equational theory of $\beta\eta$-conversion:

$$\mathrm{ap}_{\Gamma,A,B}(\lambda_{\Gamma,A,B}(b), a) \sim b[\langle \mathrm{id}_\Gamma, a\rangle]$$
$$\lambda_{\Gamma,A,B}(\mathrm{ap}_{\Gamma\cdot A,A,B}(c[\mathrm{p}_{\Gamma,A}], \mathrm{q}_{\Gamma,A})) \sim c$$

The two constructions and the proof of their equivalence have been formalised in Agda by Brilakis [7].

We can of course construct free objects in $\mathbf{Scwf}^{\mathbb{N}_1,\times,\Rightarrow}$ and other categories of scwfs with extra type structure in a similar way. Likewise, we can show a *biequivalence* between the 2-category of cccs *as property* rather than structure, and the extension of the 2-category $\mathbf{Scwf}^2_{\mathrm{dem}}$ where scwfs additionally have a $\Rightarrow$-structure, which is preserved up to isomorphism. We omit the details since they are very similar to those in the proof of Theorem 3.

5 Dependently typed categories with families

We now return to full dependently typed cwfs. After discussing alternative definitions, we show an explicit construction of a free cwf. We then add the type formers I, Σ, and Π and give an overview of the biequivalence theorems in Clairambault and Dybjer [11, 12]. Finally we outline the construction of a bifree lccc and the proof of undecidability of equality [9, 10] in this lccc.

5.1 Plain cwfs

5.1.1 Alternative definitions

As already mentioned in Section 2.2.1, we shall discuss alternative definitions of cwfs.

Context comprehension via representable presheaves. First, observe that the family valued presheaf $T : \mathcal{C}^{\mathrm{op}} \to \mathrm{Fam}$ may equivalently be given by two Set-valued presheaves

$$\mathrm{Ty} : \mathcal{C}^{\mathrm{op}} \to \mathrm{Set} \qquad\qquad \mathrm{Tm} : (\int^{\mathcal{C}} \mathrm{Ty})^{\mathrm{op}} \to \mathrm{Set}$$

where $\int^{\mathcal{C}} \mathrm{Ty}$ is the category of elements of the first presheaf. Context comprehension for full cwfs is then equivalent to requiring that the presheaves

$$\Delta \mapsto \sum_{\gamma \in \mathcal{C}(\Delta,\Gamma)} \mathrm{Tm}(\Delta, A[\gamma])$$
$$\delta \mapsto (\gamma, a) \mapsto (\gamma \circ \delta, a[\delta])$$

are representable for all $\Gamma \in \mathcal{C}_0$ and $A \in \mathrm{Ty}(\Gamma)$.

Natural models. More radically, Awodey [4] and Fiore [17] propose to replace the Fam-valued presheaf $T : \mathcal{C}^{\mathrm{op}} \to \mathrm{Fam}$ by two set valued presheaves

$$\mathrm{Ty} : \mathcal{C}^{\mathrm{op}} \to \mathrm{Set} \qquad\qquad \mathrm{Tm} : \mathcal{C}^{\mathrm{op}} \to \mathrm{Set}$$

and a natural transformation $\mathsf{typeof} : \mathrm{Tm} \Rightarrow \mathrm{Ty}$. One can then define context comprehension in terms of representable natural transformations. Let $\mathrm{Y} : \mathcal{C} \to (\mathcal{C}^{\mathrm{op}} \to \mathrm{Set})$ be the Yoneda embedding. A natural transformation $\sigma : G \Rightarrow F$ between presheaves on $\mathcal{C}$ is representable in the sense of Grothendieck, if for all $C \in \mathcal{C}_0$ and $c \in F(C)$, there are $D \in \mathcal{C}_0, p \in \mathcal{C}(D, C)$, and $d \in G(D)$, such that the following diagram in the category of presheaves is a pullback:

$$\begin{array}{ccc} \mathrm{Y}(D) & \overset{d}{\Longrightarrow} & G \\ {\scriptstyle \mathrm{Y}(p)}\Big\Downarrow & \lrcorner & \Big\Downarrow{\scriptstyle \sigma} \\ \mathrm{Y}(C) & \underset{c}{\Longrightarrow} & F \end{array}$$

where c and d in the diagram are shorthand for the corresponding respective natural transformations $F(-)(c)$ and $G(-)(d)$ from the Yoneda lemma.

Hence, $\mathsf{typeof} : \mathrm{Tm} \Rightarrow \mathrm{Ty}$ is representable provided for all $\Gamma \in \mathcal{C}_0$ and $A \in \mathrm{Ty}(\Gamma)$, there is $\Gamma \cdot A \in \mathcal{C}_0$, $\mathrm{p}_{\Gamma,A} \in \mathcal{C}(\Gamma \cdot A, \Gamma)$, and $\mathrm{q}_{\Gamma,A} \in \mathrm{Tm}(\Gamma \cdot A)$, such that the following diagram is a pullback:

$$\begin{array}{ccc} \mathcal{C}(-, \Gamma \cdot A) & \overset{\mathrm{q}_{\Gamma,A}[-]}{\Longrightarrow} & \mathrm{Tm} \\ {\scriptstyle \mathrm{p}_{\Gamma,A} \circ -}\Big\Downarrow & \lrcorner & \Big\Downarrow{\scriptstyle \mathsf{typeof}} \\ \mathcal{C}(-, \Gamma) & \underset{A[-]}{\Longrightarrow} & \mathrm{Ty} \end{array}$$

We emphasize that the function that maps Γ and A to the triple $\Gamma \cdot A, \mathrm{p}_{\Gamma,A}, \mathrm{q}_{\Gamma,A}$ is part of the structure – the natural transformation $\mathsf{typeof} : \mathrm{Tm} \Rightarrow \mathrm{Ty}$ is *represented*.

Awodey [4] has studied this notion under the name *natural models*. We refer to his paper for further development of the theory. Note that this approach suggests an essentially algebraic view of cwfs rather than a generalized algebraic one. Here we have only one sort $\mathrm{Tm}(\Gamma)$ containing *all* terms a of some type $A = \mathsf{typeof}(a) \in \mathrm{Ty}(\Gamma)$. In other words terms are "fibred" over types.

Categories with attributes. There is a certain redundancy in the definition of categories with families, since we can show (using context comprehension) that *terms* are in one-to-one correspondence with certain morphisms of the base category:

$$\mathrm{Tm}(\Gamma, A) \cong \{\gamma \in \mathcal{C}(\Gamma, \Gamma \cdot A) \mid \mathrm{p}_{\Gamma,A} \circ \gamma = \mathrm{id}_\Gamma\}$$

In other words, terms in $\mathrm{Tm}(\Gamma, A)$ correspond to *sections* of $\mathrm{p}_{\Gamma,A}$, the *display map* for the type A.

We can thus remove the term part of cwfs and get the closely related notion of a *category with attributes* (cwa) [19]. This consists of a category $\mathcal{C}$ with a terminal object, a presheaf $\mathrm{Ty} : \mathcal{C}^{\mathrm{op}} \to \mathrm{Set}$ for types and sustitutions, and an operation which given $\Gamma \in \mathcal{C}_0$ and $A \in \mathrm{Ty}(\Gamma)$ associates a context $\Gamma \cdot A$ and a "display map" $\mathrm{p}_{\Gamma,A} : \Gamma \cdot A \to \Gamma$; and for each $\gamma : \Delta \to \Gamma$, a chosen *pullback square*:

$$\begin{array}{ccc}
\Delta \cdot A[\gamma] & \longrightarrow & \Gamma \cdot A \\
{\scriptstyle \mathrm{p}_{\Delta.A[\gamma]}}\big\downarrow & \lrcorner & \big\downarrow{\scriptstyle \mathrm{p}_{\Gamma,A}} \\
\Delta & \xrightarrow[\gamma]{} & \Gamma
\end{array}$$

This follows the idea of "substitutions as pullbacks" familiar in categorical logic. This choice of pullbacks is finally required to be *split*, in the sense that the association of substitutions to pullback squares is functorial.

It is fairly easy to prove that categories with attributes are equivalent to categories with families [22]. This proof and several other proofs relating models of dependent type theory are formalized in the UniMath system by Ahrens, Lumsdaine, and Voevodsky [3].

Cwfs were originally introduced [16] because of the desire to represent the most basic rules of dependent types as a generalized algebraic theory, closely related to Martin-Löf's substitution calculus. This was achieved by making the family of terms into an explicit part of the definition, and formalize the sets of types and terms and their substitution operations as a family valued presheaf. In this way the pullback property of type substitution could be removed from the definition since it can be derived from the other part of the structure.

5.1.2 A free cwf

As in the previous sections, we next show how to build a free cwf. Recall that in the simply-typed case, we built a free scwf over a set $\mathcal{B}$ of basic types. We could do the same here, however for simplicity, and because it suffices to prove the undecidability theorem, we will assume that there is only one basic type o. The rules used for the construction of this plain free cwf (except the rule for the base type) correspond to the general rules for dependent type theory. This construction can then be extend to the case where we add rules for specific type formers.

The construction of initial ucwfs and free scwfs are rather immediate from the definitions: simply turn the definitions into simultaneous inductive definitions of their terms and substitutions, and then define equality of terms and substitutions by another simultaneous inductive definition. Unfortunately, the construction of free full cwfs is no longer as direct. What complicates the matter is the type-equality rule, which means that typability of terms may depend on proofs of equality of types. Thus we have to define equality of contexts, context morphisms, types, and terms simultaneously with their elements. Apart from this the recipe is similar: take the definition of the generalised algebraic theory of cwfs and turn it into a mutual inductive definition where all equality reasoning is made explicit.

To build the free cwf we first define raw contexts, raw substitutions, raw types, and raw terms.

$$\begin{aligned} \Gamma \in \texttt{Ctx} &::= 1 \mid \Gamma \cdot A \\ \gamma \in \texttt{Sub} &::= \gamma \circ \gamma \mid \mathrm{id}_\Gamma \mid \langle\rangle_\Gamma \mid \mathrm{p}_A \mid \langle \gamma, a \rangle_A \\ A \in \texttt{Ty} &::= o \mid A[\gamma] \\ a \in \texttt{Tm} &::= a[\gamma] \mid \mathrm{q}_A \end{aligned}$$

We then need to define the well-formed contexts and types, and the well-typed substitutions and terms. In Martin-Löf's substitution calculus these are defined by a system of inference rules for all the eight forms of judgments. Here we choose a more economical way, by only defining well-formed *equal* contexts and types, and the well-typed *equal* substitutions and terms. Thus we define four families of partial equivalence relations (pers), corresponding to the four forms of equality judgments, by a mutual inductive definition:

$$\Gamma = \Gamma' \vdash \qquad \Gamma \vdash A = A' \qquad \Delta \vdash \gamma = \gamma' : \Gamma \qquad \Gamma \vdash a = a' : A$$

where $\Gamma, \Gamma' \in \texttt{Ctx}, \gamma, \gamma' \in \texttt{Sub}, A, A' \in \texttt{Ty}$, and $a, a' \in \texttt{Tm}$. The basic judgment forms can then be defined as the reflexive instances of the pers:

- $\Gamma \vdash$ abbreviates $\Gamma = \Gamma \vdash$,
- $\Gamma \vdash A$ abbreviates $\Gamma \vdash A = A$,
- $\Delta \vdash \gamma : \Gamma$ abbreviates $\Delta \vdash \gamma = \gamma : \Gamma$,
- $\Gamma \vdash a : A$ abbreviates $\Gamma \vdash a = a : A$.

The four families of partial equivalence relations (pers) are given by a simultaneous inductive definition with the following introduction rules:

Per-rules

$$\frac{\Gamma = \Gamma' \vdash \qquad \Gamma' = \Gamma'' \vdash}{\Gamma = \Gamma'' \vdash} \qquad \frac{\Gamma = \Gamma' \vdash}{\Gamma' = \Gamma \vdash}$$

$$\frac{\Delta \vdash \gamma = \gamma' : \Gamma \qquad \Delta \vdash \gamma' = \gamma'' : \Gamma}{\Delta \vdash \gamma = \gamma'' : \Gamma} \qquad \frac{\Delta \vdash \gamma = \gamma' : \Gamma}{\Delta \vdash \gamma' = \gamma : \Gamma}$$

$$\frac{\Gamma \vdash A = A' \qquad \Gamma \vdash A' = A''}{\Gamma \vdash A = A''} \qquad \frac{\Gamma \vdash A = A'}{\Gamma \vdash A' = A}$$

$$\frac{\Gamma \vdash a = a' : A \qquad \Gamma \vdash a' = a'' : A}{\Gamma \vdash a = a'' : A} \qquad \frac{\Gamma \vdash a = a' : A}{\Gamma \vdash a' = a : A}$$

Preservation rules for judgments

$$\frac{\Gamma = \Gamma' \vdash \qquad \Delta = \Delta' \vdash \qquad \Gamma \vdash \gamma = \gamma' : \Delta}{\Gamma' \vdash \gamma = \gamma' : \Delta'} \qquad \frac{\Gamma = \Gamma' \vdash \qquad \Gamma \vdash A = A'}{\Gamma' \vdash A = A'}$$

$$\frac{\Gamma = \Gamma' \vdash \qquad \Gamma \vdash A = A' \qquad \Gamma \vdash a = a' : A}{\Gamma' \vdash a = a' : A'}$$

Congruence rules for operators and the base type

$$\frac{}{1 = 1 \vdash} \qquad \frac{\Gamma = \Gamma' \vdash \qquad \Gamma \vdash A = A'}{\Gamma \cdot A = \Gamma' \cdot A' \vdash} \qquad \frac{}{1 \vdash o = o}$$

$$\frac{\Gamma \vdash A = A' \qquad \Delta \vdash \gamma = \gamma' : \Gamma}{\Delta \vdash A[\gamma] = A'[\gamma']} \qquad \frac{\Gamma = \Gamma' \vdash}{\Gamma \vdash \mathrm{id}_\Gamma = \mathrm{id}_{\Gamma'} : \Gamma}$$

$$\frac{\Gamma = \Gamma' \vdash}{\Gamma \vdash \langle\rangle_\Gamma = \langle\rangle_{\Gamma'} : 1} \qquad \frac{\Gamma \vdash \delta = \delta' : \Delta \qquad \Delta \vdash \gamma = \gamma' : \Theta}{\Gamma \vdash \gamma \circ \delta = \gamma' \circ \delta' : \Theta}$$

$$\frac{\Gamma \vdash A = A'}{\Gamma \cdot A \vdash \mathrm{p}_A = \mathrm{p}_{A'} : \Gamma}$$

$$\frac{\Gamma \vdash A = A' \qquad \Delta \vdash \gamma = \gamma' : \Gamma \qquad \Delta \vdash a = a' : A[\gamma]}{\Delta \vdash \langle \gamma, a \rangle_A = \langle \gamma', a' \rangle_{A'} : \Gamma \cdot A}$$

$$\frac{\Gamma \vdash a = a' : A \qquad \Delta \vdash \gamma = \gamma' : \Gamma}{\Delta \vdash a[\gamma] = a'[\gamma'] : A[\gamma]} \qquad \frac{\Gamma \vdash A = A'}{\Gamma \cdot A \vdash \mathrm{q}_A = \mathrm{q}_{A'} : A[\mathrm{p}_A]}$$

Conversion rules

$$\frac{\Delta \vdash \theta : \Theta \qquad \Gamma \vdash \delta : \Delta \qquad \Xi \vdash \gamma : \Gamma}{\Xi \vdash (\theta \circ \delta) \circ \gamma = \theta \circ (\delta \circ \gamma) : \Theta} \qquad \frac{\Gamma \vdash \gamma : \Delta}{\Gamma \vdash \gamma = \mathrm{id}_\Delta \circ \gamma : \Delta}$$

$$\frac{\Gamma \vdash \gamma : \Delta}{\Gamma \vdash \gamma = \gamma \circ \mathrm{id}_\Gamma : \Delta} \qquad \frac{\Gamma \vdash A \qquad \Delta \vdash \gamma : \Gamma \qquad \Theta \vdash \delta : \Delta}{\Theta \vdash A[\gamma \circ \delta] = (A[\gamma])[\delta]}$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A[\mathrm{id}_\Gamma] = A} \qquad \frac{\Gamma \vdash a : A \qquad \Delta \vdash \gamma : \Gamma \qquad \Theta \vdash \delta : \Delta}{\Theta \vdash a[\gamma \circ \delta] = (a[\gamma])[\delta] : (A[\gamma])[\delta]}$$

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash a[\mathrm{id}_\Gamma] = a : A} \qquad \frac{\Gamma \vdash \gamma : 1}{\Gamma \vdash \gamma = \langle\rangle_\Gamma : 1}$$

$$\frac{\Gamma \vdash A \qquad \Delta \vdash \gamma : \Gamma \qquad \Delta \vdash a : A[\gamma]}{\Delta \vdash \mathrm{p}_A \circ \langle \gamma, a \rangle_A = \gamma : \Gamma}$$

$$\frac{\Gamma \vdash A \qquad \Delta \vdash \gamma : \Gamma \qquad \Delta \vdash a : A[\gamma]}{\Delta \vdash \mathrm{q}_A[\langle \gamma, a \rangle_A] = a : A[\gamma]} \qquad \frac{\Delta \vdash \gamma : \Gamma \cdot A}{\Delta \vdash \gamma = \langle \mathrm{p}_A \circ \gamma, \mathrm{q}_A[\gamma] \rangle_A : \Gamma \cdot A}$$

These rules (except the base type rule) correspond to the *general rules* for intuitionistic type theory, that is, the rules which are given before the rules for the type formers, see the discussion in Section 2.1.2.

Theorem 5 *The cwf* $\mathcal{T}$*, defined with the following data:*

- $\mathcal{T}_0 = \{\Gamma \mid \Gamma \vdash\}/{=^c}$, *where* $\Gamma =^c \Gamma'$ *if* $\Gamma = \Gamma' \vdash$ *is derivable.*
- $\mathcal{T}([\Gamma],[\Delta]) = \{\gamma \mid \Gamma \vdash \gamma : \Delta\}/{=^\Gamma_\Delta}$ *where* $\gamma =^\Gamma_\Delta \gamma'$ *iff* $\Gamma \vdash \gamma = \gamma' : \Delta$ *is derivable. Note that this makes sense since it only depends on the equivalence classes* $[\Gamma]$ *and* $[\Delta]$ *of* Γ *and* Δ *(morphisms and morphism equality are preserved by object equality).*
- $\mathrm{Ty}_\mathcal{T}([\Gamma]) = \{A \mid \Gamma \vdash A\}/ =^\Gamma$ *where* $A =^\Gamma B$ *if* $\Gamma \vdash A = B$.
- $\mathrm{Tm}_\mathcal{T}([\Gamma],[A]) = \{a \mid \Gamma \vdash a : A\}/ =^\Gamma_A$ *where* $a =^\Gamma_A a'$ *if* $\Gamma \vdash a = a' : A$.

is the free cwf on one base type.

By *free cwf on one base type* we mean that it is initial in the category $\mathbf{Cwf}^o$ having as objects small cwfs with a chosen type $o \in \mathrm{Ty}(1)$, and as morphisms the strict cwf-morphisms which preserve the chosen base type. We refer to Castellan, Clairambault, and Dybjer [9, 10] for the proof of freeness of $\mathcal{T}$.

In the papers mentioned above, we show that the free cwf is also *bifree* in the fully dependent version $\mathbf{Cwf}^2_{\mathrm{dem}}$of the 2-category $\mathbf{Scwf}^2_{\mathrm{dem}}$ of Section 4.3.3 (with a base type). Before we consider additional structure, let us define the morphisms of this 2-category, which will play an important role later on.

Definition 23 A *pseudo-cwf morphism* from a cwf $\mathcal{C}$ to a cwf $\mathcal{D}$ is a pair (F, σ) where $F : \mathcal{C} \to \mathcal{D}$ is a functor and for each $\Gamma \in \mathcal{C}_0$, σ_Γ is a **Fam**-morphism from $T\Gamma$ to $T'F\Gamma$ preserving the structure up to isomorphism. In particular there are isomorphisms (again writing F for all components):

$$\begin{array}{rrcll} \theta_{A,\gamma} : & FA[F\gamma] & \cong_{F\Gamma} & F(A[\gamma]) & \qquad (\text{for } \gamma : \Gamma \to \Delta) \\ !_F : & 1 & \cong & F1 & \\ \rho_{\Gamma,A} : & F(\Gamma \cdot A) & \cong & F\Gamma \cdot FA & \end{array}$$

where $\cong_{F\Gamma}$ means that it is an isomorphism in the category $\mathrm{Ty}^{\mathcal{D}}(F\Gamma)$ of types over $F\Gamma$ in $\mathcal{D}$. These data must satisfy some coherence diagrams (see [12], Definition 3.1 for details).

We can now prove the following.

Theorem 6 *The cwf $\mathcal{T}$ is* bifree over one base type, i.e. *it is bi-initial in the* 2*-category* $\mathbf{Cwf}_o^2$ *having as objects small cwfs with one base type, as* 1*-cells pseudo cwf-morphisms preserving the base type up to iso, and as* 2*-cells natural transformations between the base functors.*

Proof We refer the reader to [10] for details and proofs. We remark that $\mathcal{T}$ is democratic and also bi-initial in $\mathbf{Cwf}^2_{o,\mathrm{dem}}$. □

Previously we defined initial ucwfs and scwfs (with extra structure) in two ways: with and without explicit substitutions. The above construction of the free cwf gives rise to a calculus with explicit substitutions. There is an analogous construction of a free cwf where substitution is instead defined as a meta-operation, which is close to the standard formulation of the general rules for dependent type theory. We do not have space to spell out this construction with implicit substitutions here, but refer to Streicher [31].

5.2 Extensional identity types, Σ-types, and finite limits

5.2.1 Extensional identity types and Σ-types

We now add extensional identity types and Σ-types to cwfs.

Definition 24 An $\mathrm{I}_{\mathrm{ext}}$-structure on cwf $\mathcal{C}$ comprises, for each $\Gamma \in \mathcal{C}_0$, $A \in \mathrm{Ty}(\Gamma)$ and $a, a' \in \mathrm{Tm}(\Gamma, A)$, a type $\mathrm{I}_A(a, a')$; and a term

$$\mathrm{r}_{A,a} \in \mathrm{Tm}(\Gamma, \mathrm{I}_A(a, a))$$

such that if $c \in \mathrm{Tm}(\Gamma, \mathrm{I}_A(a, a'))$ then $a = a'$ and $c = \mathrm{r}_{A,a}$, and such that

$$\mathrm{I}_A(a, a')[\gamma] = \mathrm{I}_{A[\gamma]}(a[\gamma], a'[\gamma])$$

for any $\gamma \in \mathcal{C}(\Delta, \Gamma)$.

This captures *extensional*, rather than *intensional*, identity types. The difference is that the two equations $a = a'$ and $c = \mathrm{r}_{A,a}$ whenever $\mathrm{Tm}(\Gamma, \mathrm{I}_A(a, a'))$ is inhabited are only valid for extensional identity types. It follows that the reflexivity term is preserved by substitution as well: $\mathrm{r}_{A,a}[\gamma]$ is forced to coincide with $\mathrm{r}_{A[\gamma],a[\gamma]}$ as the two both inhabit $\mathrm{I}_{A[\gamma]}(a[\gamma], a[\gamma])$.

As in the previous sections, strict cwf-functors between cwfs equipped with an extensional identity type structure are said to *preserve it strictly* if the action of morphisms maps the components of the source to the components of the target, on the nose. In the remainder of this paper, a more important role will be played by morphisms preserving this structure up to isomorphism. If $F : \mathcal{C} \to \mathcal{D}$ is a pseudo cwf-morphism where $\mathcal{C}$ and $\mathcal{D}$ are equipped with an $\mathrm{I}_{\mathrm{ext}}$-structure, we say that it *preserves* it provided there is, for any $A \in \mathrm{Ty}^{\mathcal{C}}(\Gamma)$, $a, a' \in \mathrm{Tm}^{\mathcal{C}}(\Gamma, A)$, an isomorphism

$$F(\mathrm{I}_A(a, a')) \cong \mathrm{I}_{FA}(Fa, Fa')$$

in $\mathrm{Ty}^{\mathcal{D}}(\Gamma)$.

Definition 25 The definition of an $\mathbb{N}_1$-structure for a cwf is the same as for an scwf, except that $\mathbb{N}_1$ now is also required to be stable under substitution, *i.e.*, a natural transformation of type presheaves

$$1 \Rightarrow \mathrm{Ty}(-)$$

and, as before, a natural isomorphism between preheaves

$$1 \cong \mathrm{Tm}_{\mathcal{C}}(-, \mathbb{N}_1)$$

where again 1 is the constant singleton presheaf.

If a cwf is democratic it has an $\mathbb{N}_1$-structure, since $\mathbb{N}_1$ may be defined as $\overline{1} \in \mathrm{Ty}(1)$.

Definition 26 A *Σ-structure* on a cwf $\mathcal{C}$ consists of, for each $\Gamma \in \mathcal{C}_0$, $A \in \mathrm{Ty}(\Gamma)$, $B \in \mathrm{Ty}(\Gamma \cdot A)$, a type $\Sigma(A, B) \in \mathrm{Ty}(\Gamma)$, and term formers

$$\begin{aligned}
\mathrm{fst}_{\Gamma,A,B}(-) &: \mathrm{Tm}(\Gamma, \Sigma(A, B)) \to \mathrm{Tm}(\Gamma, A)\\
\mathrm{snd}_{\Gamma,A,B}(-) &: \Pi_{c \in \mathrm{Tm}(\Gamma, \Sigma(A,B))} \mathrm{Tm}(\Gamma, B[\langle \mathrm{id}, \mathrm{fst}(c)\rangle])\\
\langle -, - \rangle &: (\Sigma_{a \in \mathrm{Tm}(\Gamma, A)} \mathrm{Tm}(\Gamma, B[\langle \mathrm{id}, a\rangle])) \to \mathrm{Tm}(\Gamma, \Sigma(A, B))
\end{aligned}$$

subject to the same equations as in Definition 12, plus the additional

$$\Sigma(A, B)[\gamma] = \Sigma(A[\gamma], B[\langle \gamma \circ \mathrm{p}, \mathrm{q}\rangle])\,.$$

A Σ-type structure thus gives rise to a natural transformation Σ of type presheaves

$$\sum_{A \in \mathrm{Ty}_{\mathcal{C}}(\Gamma)} \mathrm{Ty}_{\mathcal{C}}(\Gamma \cdot A) \Rightarrow \mathrm{Ty}_{\mathcal{C}}(\Gamma)$$

and isomorphisms

$$\sum_{a \in \mathrm{Tm}_{\mathcal{C}}(\Gamma,A)} \mathrm{Tm}_{\mathcal{C}}(\Gamma, B[\langle \mathrm{id}, a\rangle]) \cong \mathrm{Tm}_{\mathcal{C}}(\Gamma, \Sigma_{\Gamma}(A,B))$$

which are stable under substitution (see Definition 12). Note the difference between this and the characterization of a $\times$-structure as natural isomorphisms between presheaves. Since $A \in \mathrm{Ty}(\Gamma)$ and $B \in \mathrm{Ty}(\Gamma \cdot A)$ are dependent types the two sides of the isomorphism are no longer presheaves.

It follows by induction on the length of a context that a *contextual* cwf with an $\mathbb{N}_1$-structure and a Σ-structure is democratic.

As usual, a strict cwf-morphism between cwfs with Σ-structure *preserves* it if it maps the structure in the source cwf to the structure in the target cwf on the nose. For *pseudo*-morphisms, it turns out that there is nothing to add. In any cwf $\mathcal{C}$ with a Σ-structure, for any $\Gamma \in \mathcal{C}_0$, $A \in \mathrm{Ty}(\Gamma)$ and $B \in \mathrm{Ty}(\Gamma \cdot A)$, we have the isomorphism $\Gamma \cdot A \cdot B \cong \Gamma \cdot \Sigma(A,B)$. Since pseudo cwf-morphisms are already known to preserve context extension, it follows that they automatically preserve Σ-structures, in the following sense.

Proposition 8 *A pseudo cwf-morphism F from $\mathcal{C}$ to $\mathcal{D}$, where both cwfs have a Σ-structure, also preserves it in the sense that there is an isomorphism:*

$$s_{A,B} : F(\Sigma(A,B)) \cong \Sigma(FA, FB[\rho^{-1}_{\Gamma,A}])$$

such that projections and pairs are preserved, modulo some transports (notably following $s_{A,B}$, see Proposition 3.5 in [12] for details).

Let us write $\mathbf{Cwf}^{2,\mathrm{I}_{\mathrm{ext}},\Sigma}_{\mathrm{dem}}$ for the 2-category having as objects small democratic cwfs with an $\mathrm{I}_{\mathrm{ext}}$-structure and a Σ-structure; as morphisms the pseudo cwf-morphisms preserving these structures up to isomorphism, and as 2-cells natural transformations between the base functors.

5.2.2 The biequivalence with finitely complete categories

In Theorem 3, we proved a biequivalence between democratic scwfs and cartesian categories. We now give the dependently typed version: a biequivalence between democratic cwfs with $\mathrm{I}_{\mathrm{ext}}$- and Σ-structures; and *finitely complete* (also called *left exact*) categories.

Definition 27 A category $\mathcal{C}$ is *finitely complete* if it has all finite limits. A functor $F : \mathcal{C} \to \mathcal{D}$ between finitely complete categories is *left exact* if it preserves finite limits: the image of a limiting cone is a limiting cone.

We write **FL** for the 2-category with small finitely complete categories as objects, left exact functors as 1-cells, and natural transformations as 2-cells.

In the light of Section 4, it is natural to insist that we consider finitely complete categories to be categories with *property*, rather than with additional structure. How could we make a corresponding notion of finitely complete categories with structure? One could ask for a cartesian category with structure additionally equipped with a choice of equalizers; one could ask for a choice of pullbacks (and a terminal object); or one could directly ask for a choice of a limit of any finite diagram. One could then consider a category of these, where structure is preserved on the nose. However, the strict equivalence of Theorem 4 does *not* to extend to this situation: whereas in the simply-typed case we may prove an equivalence with structure or a biequivalence with property, it seems that the only possibility to relate the two in the present case is the biequivalence. We will comment again on that later.

Let us now give some information about the main ingredients of the biequivalence. The first observation is that if $\mathcal{C}$ is a cwf with $\mathrm{I}_{\mathrm{ext}}$ and Σ-structures, then for each $\Gamma \in \mathcal{C}_0$ there is an equivalence between the category of types over a context $\mathrm{Ty}(\Gamma)$, and the *slice category* $\mathcal{C}/\Gamma$. Indeed, Each type $A \in \mathrm{Ty}(\Gamma)$ yields a *display map* $\mathrm{p}_{\Gamma,A} : \Gamma \cdot A \to \Gamma$ regarded as an object in $\mathcal{C}/\Gamma$. In the other direction, any $\gamma : \Delta \to \Gamma$ is isomorphic (in $\mathcal{C}/\Gamma$) to

$$\mathrm{p}_{\Gamma,\mathsf{Inv}(\gamma)} : \Gamma \cdot \mathsf{Inv}(\gamma) \to \Gamma\,,$$

a projection corresponding to a type $\mathsf{Inv}(\gamma) \in \mathrm{Ty}(\Gamma)$, the "inverse image", defined as (a cwf formalization of) $x : \Gamma \vdash \Sigma_{y:\Delta}(\gamma(x) = y)$ type. Via this equivalence of categories it follows that for each $\Gamma \in \mathcal{C}_0$ the slice category $\mathcal{C}/\Gamma$ has products, that is, $\mathcal{C}$ has pullbacks. Since it has a terminal object, it has all finite limits. Likewise, if $F : \mathcal{C} \to \mathcal{D}$ is a 1-cell in $\mathbf{Cwf}^{2,\mathrm{I}_{\mathrm{ext}},\Sigma}_{\mathrm{dem}}$ then it preserves pullbacks in $\mathcal{C}$ – in fact, we have an equivalence:

Lemma 2 *Let $\mathcal{C}$ and $\mathcal{D}$ be democratic cwfs with $\mathrm{I}_{\mathrm{ext}}$- and Σ-structures and $F : \mathcal{C} \to \mathcal{D}$ be a pseudo cwf-morphism preserving democracy. Then, F preserves the $\mathrm{I}_{\mathrm{ext}}$-structure if and only if F preserves pullbacks.*

Proof The harder direction is *only if*, which boils down to the preservation of the inverse image. This can be proved from intricate calculations on cwf combinators. Details appear in [12], Lemma 4.3 and Proposition 4.4. □

We can then show that there is a *forgetful 2-functor*:

$$\mathbf{C} : \mathbf{Cwf}^{2,\mathrm{I}_{\mathrm{ext}},\Sigma}_{\mathrm{dem}} \to \mathbf{FL}\,.$$

The other direction is much more complicated. The equivalence of categories $\mathrm{Ty}(\Gamma) \simeq \mathcal{C}/\Gamma$ together with Seely's approach to interpreting type theory in locally cartesian closed categories [30] suggest, from a finitely complete category $\mathcal{C}$, to redefine the *types* over Γ as the objects of the slice category $\mathcal{C}/\Gamma$. However, there is a problem: how to define *substitution*, *i.e.*

$$-[\gamma] : (\mathcal{C}/\Gamma)_0 \to (\mathcal{C}/\Delta)_0$$

for $\gamma : \Delta \to \Gamma$? In categorical logic, substitution is usually defined *by pullback*. However, there is a problem: for an arbitrary choice of pullbacks, there is no reason why this assignment should be functorial. Consider the following two pullback squares:

$$\begin{array}{ccccc} \cdot & \longrightarrow & \cdot & \longrightarrow & \Gamma \cdot A \\ \downarrow \lrcorner & & \downarrow \lrcorner & & \downarrow \mathrm{p}_{\Gamma,A} \\ \Omega & \xrightarrow{\delta} & \Delta & \xrightarrow{\gamma} & \Gamma \end{array} \qquad\qquad \begin{array}{ccc} \cdot & \longrightarrow & \Gamma \cdot A \\ \downarrow \lrcorner & & \downarrow \mathrm{p}_{\Gamma,A} \\ \Omega & \xrightarrow{\gamma\circ\delta} & \Gamma \end{array}$$

There is no reason why the left hand side diagram, which is a composition of chosen pullbacks, and the right hand side diagram, which is a chosen pullback, should coincide – although they are always isomorphic. In other words the codomain fibration is not *split*, whereas the fibration implicit in a cwf is always split. This is a fundamental issue: Seely's proposed interpretation [30] sends types that are provably equal in the syntax to morphisms in $\mathcal{C}$ that are only known to be isomorphic. This *coherence problem* may be solved in two ways: Curien [14] proposes to change the syntax by weakening equality to isomorphism *in the syntax*, enriching it with explicit coercions between isomorphic types, and showing the extended syntax equivalent to the original via a difficult coherence theorem.

In [20], Hofmann proposes instead to solve the problem by exploiting a construction of Bénabou [6], which associates to each fibration an equivalent split fibration. This construction can be extended to dependent types: given a category $\mathcal{C}$ with finite limits, we build a cwf whose types are no longer just objects of $\mathcal{C}/\Gamma$, but objects of $\mathcal{C}/\Gamma$ with a pre-chosen substitution pullback, for every possible substitution – such that this choice is split. For details, the reader is referred to [12], Section 5. As we show there, Hofmann's construction yields a *pseudofunctor*:

$$\mathbf{L} : \mathbf{FL} \to \mathbf{Cwf}_{\mathrm{dem}}^{2,\mathrm{I}_{\mathrm{ext}},\Sigma}.$$

This is *not* a functor: when sending $F : \mathcal{C} \to \mathcal{D}$ to $\mathbf{Cwf}_{\mathrm{dem}}^{2,\mathrm{I}_{\mathrm{ext}},\Sigma}$, one must extend F to types, *i.e.* display maps p together with a fixed choice of substitution pullbacks. But the choice of substitution pullbacks for p in $\mathcal{C}$ does not suffice to completely determine a choice of substitution pullbacks for Fp in $\mathcal{D}$, hence those must be *chosen*; causing $\mathbf{L}$ to fail functoriality on the nose.

For this reason, it seems unlikely that switching to finite limits categories with structure would yield an equivalence of categories with strict maps, unless one considers categories with finite limits with a split choice of pullbacks.

Theorem 7 *There is a biequivalence of* 2*-categories* $\mathbf{FL}^2 \simeq \mathbf{Cwf}_{\mathrm{dem}}^{2,\mathrm{I}_{\mathrm{ext}},\Sigma}$.

Proof Once the mediating pseudofunctors are constructed, the proof is fairly close to that of Theorem 3. The reader is referred to [12], Section 6 for details. □

5.3 Π-types and locally cartesian closed categories

We now add Π-types and extend the results of Section 5.2. We shall show that Theorem 7 yields a biequivalence with locally cartesian closed categories.

5.3.1 Π-types

Definition 28 A Π-type structure on a cwf $\mathcal{C}$ consists of, for each $\Gamma \in \mathcal{C}_0$, $A \in \mathrm{Ty}(\Gamma)$, $B \in \mathrm{Ty}(\Gamma \cdot A)$, a type $\Pi(A, B) \in \mathrm{Ty}(\Gamma)$, and term formers

$$\begin{aligned} \lambda_{\Gamma,A,B} &: \mathrm{Tm}(\Gamma \cdot A, B) \to \mathrm{Tm}(\Gamma, \Pi(A, B)) \\ \mathrm{ap}_{\Gamma,A,B} &: \mathrm{Tm}(\Gamma, \Pi(A, B)) \to \Pi_{a \in \mathrm{Tm}(\Gamma,A)} \mathrm{Tm}(\Gamma, B[\langle \mathrm{id}, a \rangle] \end{aligned}$$

subject to the equations of Definition 22, plus the additional

$$\Pi(A, B)[\gamma] = \Pi(A[\gamma], B[\langle \gamma \circ \mathrm{p}, \mathrm{q} \rangle])\,.$$

We note that a Π-type structure gives rise to a natural transformation Π of type presheaves:

$$\sum_{A \in \mathrm{Ty}_{\mathcal{C}}(\Gamma)} \mathrm{Ty}_{\mathcal{C}}(\Gamma.A) \Rightarrow \mathrm{Ty}_{\mathcal{C}}(\Gamma)$$

and isomorphisms

$$\mathrm{Tm}_{\mathcal{C}}(\Gamma.A, B) \cong \mathrm{Tm}_{\mathcal{C}}(\Gamma, \Pi_\Gamma(A, B))$$

which are stable under substitution (see Definition 22). Note the difference between this and the characterization of an $\Rightarrow$-structure as natural isomorphisms between presheaves. Since $A \in \mathrm{Ty}(\Gamma)$ and $B \in \mathrm{Ty}(\Gamma \cdot A)$ are dependent types $\mathrm{Tm}_{\mathcal{C}}(\Gamma.A, B)$ and $\mathrm{Tm}_{\mathcal{C}}(\Gamma, \Pi_\Gamma(A, B))$ are no longer families of presheaves.

Strict cwf-morphisms between cwfs with Π-structure *preserve* it if they map all components of the Π-structure in the source cwf to the same component in the target cwf, on the nose. For *pseudo-morphisms*, we define:

Definition 29 A pseudo cwf-morphism F from $\mathcal{C}$ to $\mathcal{D}$, where both cwfs have a Π-structure, also *preserves* it provided for all $\Gamma \in \mathcal{C}_0$, $A \in \mathrm{Ty}(\Gamma)$ and $B \in \mathrm{Ty}(\Gamma \cdot A)$ there is an isomorphism in $\mathrm{Ty}^{\mathcal{D}}(F\Gamma)$

$$i_{A,B} : F(\Pi^{C}(A, B)) \cong \Pi^{\mathcal{D}}(FA, FB[\rho^{-1}_{\Gamma,A}])$$

such that application is preserved, modulo some transports (notably by $i_{A,B}$, see Definition 9 in [12] for details).

It is sufficient to require preservation of application, preservation of abstraction then follows. This follows the situation in cartesian closed categories with structure where evaluation is part of the structure, whereas abstraction is defined uniquely by the universal property – and although functors are only required to preserve evaluation, it follows that they preserve abstraction too.

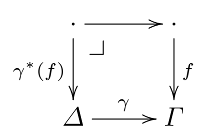
Let us write $\mathbf{Cwf}^{2,\mathrm{I}_{\mathrm{ext}},\Sigma,\Pi}_{\mathrm{dem}}$ for the 2-category where the objects are small democratic cwfs with $\mathrm{I}_{\mathrm{ext}}$-structure, Σ-structure and Π-structure; the morphisms are pseudo cwf-morphisms preserving this structure (up to isomorphism), and the 2-cells are natural transformations between the base functors.

5.3.2 The biequivalence with locally cartesian closed categories

Let us first recall locally cartesian closed categories.

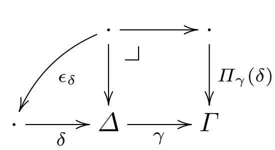
Definition 30 A category $\mathcal{C}$ is *locally cartesian closed (lccc)* if it has a terminal object and if for all $\Gamma \in \mathcal{C}_0$, the slice category $\mathcal{C}/\Gamma$ is cartesian closed.

Again, this definition is in terms of property rather than structure. This is one of the two usual definitions of locally cartesian closed categories. Equivalently, one could ask $\mathcal{C}$ to have finite limits, and require that for all $\gamma : \Delta \to \Gamma$, the *pullback functor* $\gamma^* : \mathcal{C}/\Gamma \to \mathcal{C}/\Delta$, which associates to any $f : \cdot \to \Delta$ the left hand side morphism of the pullback diagram

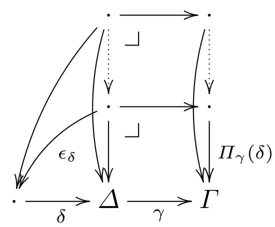


obtained via finite limits, has a right adjoint $\Pi_\delta : \mathcal{C}/\Delta \to \mathcal{C}/\Gamma$. It is this right adjoint that was proposed by Seely for the interpretation of Π-types.

As usual, this right adjoint to the pullback functor may be equivalently described as the data of cofree objects. Instantiated here, a right adjoint to $\gamma^* : \mathcal{C}/\Gamma \to \mathcal{C}/\Delta$ consists of, for all $\delta : \cdot \to \Delta$ an object of $\mathcal{C}/\Delta$, an object $\Pi_\gamma(\delta) : \cdot \to \Gamma$ in $\mathcal{C}/\Gamma$ together with a co-unit $\epsilon_\delta : \gamma^*(\Pi_\gamma(\delta)) \to \delta$ (a morphism in $\mathcal{C}/\Delta$) satisfying the universal property of co-free objects. These data, along with the universal property, amount to a *dependent product diagram*:



which is universal among any such diagram over δ and γ, as described below.



In other words, a locally cartesian closed category may be defined as a category $\mathcal{C}$ with finite limits such that, additionally, for every $\cdot \stackrel{\delta}{\to} \Delta \stackrel{\gamma}{\to} \Gamma$ in $\mathcal{C}$ there is a dependent product diagram as above. A *locally cartesian closed functor* may then be defined as a left exact functor (the image of a terminal object is terminal and the image of a pullback diagram is a pullback diagram) such that the image of a dependent product diagram is a dependent product diagram. There is a 2-category $\mathbf{LCC}^2$ having small lcccs as objects, locally cartesian closed functors as 1-cells, and natural transformations as 2-cells.

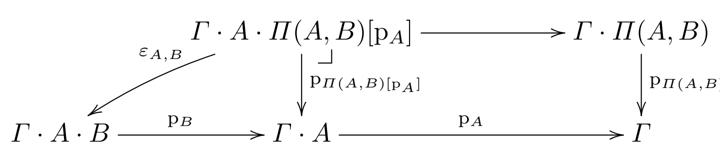
Now we can build our biequivalence. First, we define a *forgetful* 2*-functor*

$$\mathbf{C} : \mathbf{Cwf}^{2,\mathrm{I}_{\mathrm{ext}},\Sigma,\Pi}_{\mathrm{dem}} \to \mathbf{LCC}^2 \,,$$

which omits all components and only keeps the base category, as in Section 5.2.2. We must prove that if $\mathcal{C}$ is the base category of a democratic cwf with $\mathrm{I}_{\mathrm{ext}}$-structure, Σ-structure and Π-structure, then $\mathcal{C}$ is locally cartesian closed. This is straightforward: using Π-types it is easy to show that for each Γ, $\mathrm{Ty}(\Gamma)$ is cartesian closed, but $\mathrm{Ty}(\Gamma)$ is equivalent to $\mathcal{C}/\Gamma$ as shown in Section 5.2.2. Alternatively one may construct dependent products: from $\cdot \stackrel{\delta}{\to} \Delta \stackrel{\gamma}{\to} \Gamma$ one may construct an isomorphic (in the obvious sense) sequence of projections via inverse image:

$$\Gamma \cdot \mathsf{Inv}(\gamma) \cdot \mathsf{Inv}(\delta) \stackrel{\mathrm{p}}{\to} \Gamma \cdot \mathsf{Inv}(\gamma) \stackrel{\mathrm{p}}{\to} \Gamma \,.$$

For any such sequence of projections $\Gamma \cdot A \cdot B \to \Gamma \cdot A \to \Gamma$ there is a dependent product diagram, called the *chosen dependent product diagram*:



where $\varepsilon_{A,B} = \langle \mathrm{p}, \mathrm{ap}(\mathrm{q}, \mathrm{q}[\mathrm{p}]) \rangle$. Combined with inverse image, this shows that any $\cdot \stackrel{\delta}{\to} \Delta \stackrel{\gamma}{\to} \Gamma$ has a dependent product diagram. Dependent product diagrams also permit a nice characterisation of the preservation of Π-structures:

Lemma 3 *Let $F : \mathcal{C} \to \mathcal{D}$ be a pseudo cwf-morphism between cwfs with Π-structure. Then F preserves the Π-structure if and only if the image of any chosen dependent product diagrams is a dependent product diagram.*

Proof Proved through intricate calculations – see [12], Proposition 4.8, for details. □

This completes the definition of the forgetful 2-functor $\mathbf{C} : \mathbf{Cwf}^{2,\mathrm{I}_{\mathrm{ext}},\Sigma,\Pi}_{\mathrm{dem}} \to \mathbf{LCC}^2$. In the other direction, the construction is the same as for Theorem 7; the fact that the pseudofunctor $\mathbf{L}$ yields pseudo cwf-morphisms preserving Π-structures follows from Lemma 3. We conclude:

Theorem 8 *There is a biequivalence of* 2*-categories* $\mathbf{LCC}^2 \simeq \mathbf{Cwf}^{2,\mathrm{I}_{\mathrm{ext}},\Sigma,\Pi}_{\mathrm{dem}}$.

5.3.3 Undecidability in the bifree locally cartesian closed category

The construction of a free cwf in Theorem 5 can easily be extended when we add $\mathrm{I}_{\mathrm{ext}}, \mathbb{N}_1, \Sigma$ and Π-types. This is done by adding the per-rules corresponding to formation, introduction, elimination, and equality rules for $\mathrm{I}_{\mathrm{ext}}, \mathbb{N}_1, \Sigma$ and Π. We do not have room for explicitly displaying those rules, but they can be found in Castellan, Clairambault, and Dybjer [10]. We can thus construct a free cwf $\mathcal{T}^{\mathrm{I}_{\mathrm{ext}},\mathbb{N}_1,\Sigma,\Pi,o}$ with $\mathrm{I}_{\mathrm{ext}}, \mathbb{N}_1, \Sigma$ and Π-type structures. It is both a *free* cwf (with the extra structure) on one base type and with respect to strict morphisms, and a *bifree* cwf (with the extra structure) on one base type and with respect to pseudo morphisms.

Theorem 9 *The cwf* $\mathcal{T}^{\mathrm{I}_{\mathrm{ext}},\mathbb{N}_1,\Sigma,\Pi,o}$ *is* initial *in the category* $\mathbf{Cwf}^{\mathrm{I}_{\mathrm{ext}},\mathbb{N}_1,\Sigma,\Pi,o}$*, as well as* bi-initial *in the* 2*-category* $\mathbf{Cwf}^{2,\mathrm{I}_{\mathrm{ext}},\mathbb{N}_1,\Sigma,\Pi,o}$*. Moreover, it is democratic and bi-initial in* $\mathbf{Cwf}^{2,\mathrm{I}_{\mathrm{ext}},\Sigma,\Pi,o}_{\mathrm{dem}}$.

Proof Details appear in [10]. □

An object I is *bi-initial* in a 2-category iff for any A there is an arrow $I \to A$ and for any two arrows $f, g : I \to A$ there exists a unique 2-cell $\theta : f \Rightarrow g$. It follows that θ is invertible, and that bi-initial objects are equivalent. In the statement above, o denotes a base type. The objects of both $\mathbf{Cwf}^{\mathrm{I}_{\mathrm{ext}},\mathbb{N}_1,\Sigma,\Pi,o}$ and $\mathbf{Cwf}^{2,\mathrm{I}_{\mathrm{ext}},\mathbb{N}_1,\Sigma,\Pi,o}_{\mathrm{dem}}$ have a distinguished type $o \in \mathrm{Ty}(1)$, preserved on the nose for $\mathbf{Cwf}^{\mathrm{I}_{\mathrm{ext}},\mathbb{N}_1,\Sigma,\Pi,o}$ and up to isomorphism for $\mathbf{Cwf}^{2,\mathrm{I}_{\mathrm{ext}},\mathbb{N}_1,\Sigma,\Pi,o}_{\mathrm{dem}}$.

The presence of the base type o does not affect the biequivalence in Theorem 8, which extends to a biequivalence $\mathbf{Cwf}^{2,\Sigma,\Pi,o}_{\mathrm{dem}} \simeq \mathbf{LCC}^{2,o}$ where the latter has a distinguished object $o \in \mathcal{C}_0$ preserved by functors up to isomorphism. As bi-initial objects are transported to bi-initial objects via a biequivalence, it follows from Theorem 9 that the base category of the cwf $\mathcal{T}^{\mathrm{I}_{\mathrm{ext}},\mathbb{N}_1,\Sigma,\Pi,o}$ is bi-initial in $\mathbf{LCC}^{2,o}$:

Theorem 10 *The base category of* $\mathcal{T}^{\mathrm{I}_{\mathrm{ext}},\mathbb{N}_1,\Sigma,\Pi,o}$ *is the bifree lccc on one object.*

Having constructed a bifree lccc, it is natural to consider its *word problem*: given two (syntactic) substitutions $\gamma, \gamma' : \Delta \to \Gamma$ in $\mathcal{T}^{\mathrm{I}_{\mathrm{ext}},\mathbb{N}_1,\Sigma,\Pi,o}$, is it decidable whether they are equal, or equivalently whether they have the same interpretation in any locally cartesian closed category? As γ, γ' are syntactic constructs in extensional type theory, one expects undecidability. However, prior undecidability proofs for extensional type theory rely on structure not available in our case. The folklore argument uses a universe and Hofmann's undecidability proof [21] uses a type of natural numbers.

In [10], we generalize the folklore result to hold with only one base type, *i.e.* in $\mathcal{T}^{\mathrm{I_{ext}}, \mathbb{N}_1, \Sigma, \Pi, o}$, but without natural numbers or a universe. This relies on an encoding of combinatory logic in Martin-Löf type theory with I-types, Π-types, and a base type o; as a context Γ_{CL} containing:

$$\begin{aligned} k &: o, \\ s &: o, \\ \cdot &: o \Rightarrow o \Rightarrow o, \\ ax_k &: \Pi xy : o.\ \mathrm{I}(o,\ k \cdot x \cdot y,\ x), \\ ax_s &: \Pi xyz : o.\ \mathrm{I}(o,\ s \cdot x \cdot y \cdot z,\ x \cdot z \cdot (y \cdot z)) \end{aligned}$$

where the left-associative binary infix symbol "$\cdot$" stands for application. While the above uses the syntax of type theory, it is easy to set up the same context just using the cwf combinators available in $\mathcal{T}^{\mathrm{I_{ext}}, \mathbb{N}_1, \Sigma, \Pi, o}$, hence reducing the decision of equality between terms M, N in combinatory logic to deciding

$$\Gamma_{\mathsf{CL}} \vdash M \overset{?}{=} N : o$$

e.g. an equality of two terms $M, N \in \mathrm{Tm}(\Gamma_{\mathsf{CL}}, o)$. We conclude:

Theorem 11 *Equality is undecidable in the bifree locally cartesian closed category on one base type.*

Acknowledgements Peter Dybjer is grateful for the generous support of the Centre for Advanced Study at the Norwegian Academy of Science and Letters in Oslo, where he was a fellow in the Homotopy Type Theory and Univalent Foundations project while working on this paper.

References

1. Martín Abadi, Luca Cardelli, Pierre-Louis Curien, and Jean-Jacques Lévy. Explicit substitutions. In *Conference Record of the Seventeenth Annual ACM Symposium on Principles of Programming Languages, San Francisco, California, USA, January 1990*, pages 31–46, 1990.
2. Peter Aczel. *Frege Structures and the Notions of Proposition, Truth, and Set*, pages 31–59. North-Holland, 1980.
3. Benedikt Ahrens, Peter LeFanu Lumsdaine, and Vladimir Voevodsky. Categorical structures for type theory in univalent foundations. *Logical Methods in Computer Science*, 14(3), 2018.
4. Steve Awodey. Natural models of homotopy type theory. *Mathematical Structures in Computer Science*, 28(2):241–286, 2018.

5. Henk P. Barendregt. *The Lambda Calculus*. North-Holland, 1984. Revised edition.
6. Jean Bénabou. Fibred categories and the foundation of naive category theory. *Journal of Symbolic Logic*, 50:10–37, 1985.
7. Konstantinos Brilakis. On Initial Categories with Families – Formalization of Unityped and Simply Typed CwFs in Agda. Master's thesis, Chalmers University of Technology, 2018.
8. John Cartmell. Generalized algebraic theories and contextual categories. *Annals of Pure and Applied Logic*, 32:209–243, 1986.
9. Simon Castellan, Pierre Clairambault, and Peter Dybjer. Undecidability of equality in the free locally cartesian closed category. In *13th International Conference on Typed Lambda Calculi and Applications, TLCA 2015, July 1-3, 2015, Warsaw, Poland*, pages 138–152, 2015.
10. Simon Castellan, Pierre Clairambault, and Peter Dybjer. Undecidability of equality in the free locally cartesian closed category (extended version). *Logical Methods in Computer Science*, 13(4), 2017.
11. Pierre Clairambault and Peter Dybjer. The biequivalence of locally cartesian closed categories and Martin-Löf type theories. In *Typed Lambda Calculi and Applications - 10th International Conference, TLCA 2011, Novi Sad, Serbia, June 1-3, 2011. Proceedings*, pages 91–106, 2011.
12. Pierre Clairambault and Peter Dybjer. The biequivalence of locally cartesian closed categories and Martin-Löf type theories. *Mathematical Structures in Computer Science*, 24(6), 2014.
13. Pierre-Louis Curien. Categorical combinators. *Information and Control*, 69(1-3):188–254, 1986.
14. Pierre-Louis Curien. Substitution up to isomorphism. *Fundamenta Informaticae*, 19(1,2):51–86, 1993.
15. Pierre-Louis Curien, Richard Garner, and Martin Hofmann. Revisiting the categorical interpretation of dependent type theory. *Theoretical Computer Science*, 546:99–119, 2014.
16. Peter Dybjer. Internal type theory. In *TYPES '95, Types for Proofs and Programs*, number 1158 in Lecture Notes in Computer Science, pages 120–134. Springer, 1996.
17. Marcelo Fiore. Discrete generalised polynomial functors. Slides for a talk given at the 39th International Colloquium on Automata, Languages and Programming (ICALP), 2012.
18. Marcelo Fiore and Vladimir Voevodsky. Lawvere theories and C-systems. 2017. Accepted for publication in Proceedings of the American Mathematical Society.
19. Martin Hofmann. Interpretation of type theory in locally cartesian closed categories. In *Proceedings of CSL*. Springer LNCS, 1994.
20. Martin Hofmann. On the interpretation of type theory in locally cartesian closed categories. In Leszek Pacholski and Jerzy Tiuryn, editors, *CSL*, volume 933 of *Lecture Notes in Computer Science*. Springer, 1994.
21. Martin Hofmann. *Extensional concepts in intensional type theory*. PhD thesis, University of Edinburgh, 1995.
22. Martin Hofmann. Syntax and semantics of dependent types. In Andrew Pitts and Peter Dybjer, editors, *Semantics and Logics of Computation*. Cambridge University Press, 1996.
23. Martin Hyland and John Power. The category theoretic understanding of universal algebra: Lawvere theories and monads. *Electr. Notes Theor. Comput. Sci.*, 172:437–458, 2007.
24. J. Lambek and P. J. Scott. *Introduction to higher order categorical logic*. Number 7 in Cambridge studies in advanced mathematics. Cambridge University Press, 1986.
25. F. William Lawvere. Equality in hyperdoctrines and comprehension schema as an adjoint functor. In A. Heller, editor, *Applications of Categorical Algebra, Proceedings of Symposia in Pure Mathematics*. AMS, 1970.

26. Per Martin-Löf. Constructive mathematics and computer programming. In *Logic, Methodology and Philosophy of Science, VI, 1979*, pages 153–175. North-Holland, 1982.
27. Per Martin-Löf. *Intuitionistic Type Theory*. Bibliopolis, 1984.
28. Per Martin-Löf. Substitution calculus. Notes from a lecture given in Göteborg, November 1992.
29. Adam Obtułowicz. Functorial semantics of the type free lambda-betaeta calculus. In *Foundations of Computation Theory*, pages 302–307, 1977.
30. R. A. G. Seely. Locally cartesian closed categories and type theory. *Proceedings of the Cambridge Philosophical Society*, 95:33–48, 1984.
31. Thomas Streicher. *Semantics of Type Theory*. Birkhäuser, 1991.
32. Alvaro Tasistro. Formulation of Martin-Löf's theory of types with explicit substitutions. Technical report, Department of Computer Sciences, Chalmers University of Technology and University of Göteborg, 1993.
33. Todd Trimble. Towards a doctrine of operads. Article in nlab `https://ncatlab.org/toddtrimble/published/Towards+a+doctrine+of+operads#cartesian_operads_are_equivalent_to_lawvere_theories`, 2013 (accessed 8 March 2019).