The Biequivalence of Locally Cartesian Closed Categories and Martin-Löf Type Theories

Pierre Clairambault and Peter Dybjer

University of Bath and Chalmers University of Technology

Abstract. Seely's paper *Locally cartesian closed categories and type theory* contains a well-known result in categorical type theory: that the category of locally cartesian closed categories is equivalent to the category of Martin-Löf type theories with Π, Σ, and extensional identity types. However, Seely's proof relies on the problematic assumption that substitution in types can be interpreted by pullbacks. Here we prove a corrected version of Seely's theorem: that the Bénabou-Hofmann interpretation of Martin-Löf type theory in locally cartesian closed categories yields a biequivalence of 2-categories. To facilitate the technical development we employ categories with families as a substitute for syntactic Martin-Löf type theories. As a second result we prove that if we remove Π-types the resulting categories with families are biequivalent to left exact categories.

1 Introduction

It is "well-known" that locally cartesian closed categories (lcccs) are equivalent to Martin-Löf's intuitionistic type theory [9,10]. But how *known* is it really? Seely's original proof [13] contains a flaw, and the papers by Curien [3] and Hofmann [5] who address this flaw only show that Martin-Löf type theory can be interpreted in locally cartesian closed categories, but not that this interpretation is an equivalence of categories provided the type theory has Π, Σ, and extensional identity types. Here we complete the work and fully rectify Seely's result except that we do not prove an equivalence of categories but a *biequivalence* of 2-categories. In fact, a significant part of the endeavour has been to find an appropriate formulation of the result, and in particular to find a suitable notion analogous to Seely's "interpretation of Martin-Löf theories".

Categories with families and democracy. Seely turns a given Martin-Löf theory into a category where the objects are *closed* types and the morphisms from type A to type B are functions of type $A \to B$. Such categories are the objects of Seely's "category of Martin-Löf theories".

Instead of syntactic Martin-Löf theories we shall employ *categories with families (cwfs)* [4]. A cwf is a pair $(\mathbb{C}, T)$ where $\mathbb{C}$ is the category of contexts and explicit substitutions, and $T : \mathbb{C}^{op} \to \mathbf{Fam}$ is a functor, where $T(\Gamma)$ represents the family of sets of terms indexed by types in context Γ and $T(\gamma)$ performs the substitution of γ in types and terms. Cwf is an appropriate substitute for syntax for dependent types: its definition unfolds to a variable-free calculus of explicit substitutions [4], which is like Martin-Löf's [11,14] except that variables are encoded by projections. One advantage of this approach compared to Seely's is that we get a natural definition of morphism of cwfs, which preserves the structure of cwfs up to isomorphism. In contrast Seely's notion of "interpretation of Martin-Löf theories" is defined indirectly via the construction of an lccc associated with a Martin-Löf theory, and basically amounts to a functor preserving structure between the corresponding lcccs, rather than directly as something which preserves all the "structure" of Martin-Löf theories.

To prove our biequivalences we require that our cwfs are *democratic*. This means that each context is *represented* by a type. Our results require us to build local cartesian closed structure in the category of contexts. To this end we use available constructions on types and terms, and by democracy such constructions can be moved back and forth between types and contexts. Since Seely works with closed types only he has no need for democracy.

The coherence problem. Seely interprets type substitution in Martin-Löf theories as pullbacks in lcccs. However, this is problematic, since type substitution is already defined by induction on the structure of types, and thus fixed by the interpretation of the other constructs of type theory. It is not clear that the pullbacks can be chosen to coincide with this interpretation.

In the paper *Substitution up to isomorphism* [3] Curien describes the fundamental nature of this problem. He sets out

> ... to solve a difficulty arising from a mismatch between syntax and semantics: in locally cartesian closed categories, substitution is modelled by pullbacks (more generally pseudo-functors), that is, only up to isomorphism, unless split fibrational hypotheses are imposed. ... but not all semantics do satisfy them, and in particular not the general description of the interpretation in an arbitrary locally cartesian closed category. In the general case, we have to show that the isomorphisms between types arising from substitution are *coherent* in a sense familiar to category theorists.

To solve the problem Curien introduces a calculus with explicit substitutions for Martin-Löf type theory, with special terms witnessing applications of the type equality rule. In this calculus type equality can be interpreted as isomorphism in lcccs. The remaining coherence problem is to show that Curien's calculus is equivalent to the usual formulation of Martin-Löf type theory, and Curien proves this result by cut-elimination.

Somewhat later, Hofmann [5] gave an alternative solution based on a technique which had been used by Bénabou [1] for constructing a *split* fibration from an arbitrary fibration. In this way Hofmann constructed a model of Martin-Löf type theory with Π-types, Σ-types, and (extensional) identity types from a locally cartesian closed category. Hofmann used categories with attributes (cwa) in the sense of Cartmell [2] as his notion of model. In fact, cwas and cwfs are closely related: the notion of cwf arises by reformulating the axioms of cwas to make the connection with the usual syntax of dependent type theory more transparent. Both cwas and cwfs are split notions of model of Martin-Löf type theory, hence the relevance of Bénabou's construction.

However, Seely wanted to prove an equivalence of categories. Hofmann conjectured [5]:

> We have now constructed a cwa over $\mathcal{C}$ which can be shown to be equivalent to $\mathcal{C}$ in some suitable 2-categorical sense.

Here we spell out and prove this result, and thus fully rectify Seely's theorem. It should be apparent from what follows that this is not a trivial exercise. In our setting the result is a biequivalence analogous to Bénabou's (much simpler) result: that the 2-category of fibrations (with non-strict morphisms) is biequivalent to the 2-category of split fibrations (with non-strict morphisms).

While carrying out the proof we noticed that if we remove Π-types the resulting 2-category of cwfs is biequivalent to the 2-category of left exact (or finitely complete) categories. We present this result in parallel with the main result.

Plan of the paper. An equivalence of categories consists of a pair of functors which are inverses up to natural isomorphism. Biequivalence is the appropriate notion of equivalence for bicategories [8]. Instead of functors we have *pseudofunctors* which only preserve identity and composition up to isomorphism. Instead of natural isomorphisms we have *pseudonatural transformations* which are inverses up to *invertible modification*.

A 2-category is a strict bicategory, and the remainder of the paper consists of constructing two biequivalences of 2-categories. In Section 2 we introduce cwfs and show how to turn a cwf into an indexed category. In Section 3 we define the 2-categories $\mathbf{CwF}_{\mathrm{dem}}^{\mathrm{I_{ext}}\Sigma}$ of democratic cwfs which support extensional identity types and Σ-types and $\mathbf{CwF}_{\mathrm{dem}}^{\mathrm{I_{ext}}\Sigma\Pi}$ which also support Π-types. We also define the notions of pseudo cwf-morphism and pseudo cwf-transformation. In Section 4 we define the 2-categories $\mathbf{FL}$ of left exact categories and $\mathbf{LCC}$ of locally cartesian closed categories. We show that there are forgetful 2-functors $U : \mathbf{CwF}_{\mathrm{dem}}^{\mathrm{I_{ext}}\Sigma} \to \mathbf{FL}$ and $U : \mathbf{CwF}_{\mathrm{dem}}^{\mathrm{I_{ext}}\Sigma\Pi} \to \mathbf{LCC}$. In section 5 we construct the pseudofunctors $H : \mathbf{FL} \to \mathbf{CwF}_{\mathrm{dem}}^{\mathrm{I_{ext}}\Sigma}$ and $H : \mathbf{LCC} \to \mathbf{CwF}_{\mathrm{dem}}^{\mathrm{I_{ext}}\Sigma\Pi}$ based on the Bénabou-Hofmann construction. In section 6 we prove that H and U give rise to the biequivalences of $\mathbf{FL}$ and $\mathbf{CwF}_{\mathrm{dem}}^{\mathrm{I_{ext}}\Sigma}$ and of $\mathbf{LCC}$ and $\mathbf{CwF}_{\mathrm{dem}}^{\mathrm{I_{ext}}\Sigma\Pi}$.

An appendix containing the full proof of the biequivalences can be found at `http://www.cse.chalmers.se/~peterd/papers/categorytypetheory.html/`.

Acknowledgement. We are grateful to the anonymous reviewers for several useful remarks which have helped us improve the paper. We would also like to acknowledge the support of the (UK) EPSRC grant RC-CM1025 for the first author and of the (Swedish) Vetenskapsrådet grant "Types for Proofs and Programs" for the second author.

2 Categories with Families

2.1 Definition

Definition 1. *Let* **Fam** *be the category of families of sets defined as follows. An object is a pair* (A, B) *where* A *is a set and* $B(x)$ *is a family of sets indexed by* $x \in A$. *A morphism with source* (A, B) *and target* (A', B') *is a pair consisting of a function* $f : A \to A'$ *and a family of functions* $g(x) : B(x) \to B'(f(x))$ *indexed by* $x \in A$.

Note that **Fam** is equivalent to the arrow category $\mathbf{Set}^{\to}$.

Definition 2. *A* ***category with families (cwf)*** *consists of the following data:*

- *A base category* $\mathbb{C}$. *Its objects represent* contexts *and its morphisms represent* substitutions. *The identity map is denoted by* $\mathrm{id} : \Gamma \to \Gamma$ *and the composition of maps* $\gamma : \Delta \to \Gamma$ *and* $\delta : \Xi \to \Delta : \Xi \to \Gamma$ *is denoted by* $\gamma \circ \delta$ *or more briefly by* $\gamma\delta : \Xi \to \Gamma$.
- *A functor* $T : \mathbb{C}^{op} \to \mathbf{Fam}$. $T(\Gamma)$ *is a pair, where the first component represents the set* $\mathrm{Type}(\Gamma)$ *of* types *in context* Γ, *and the second component represents the type-indexed family* $(\Gamma \vdash A)_{A \in \mathrm{Type}(\Gamma)}$ *of sets of* terms *in context* Γ. *We write* $a : \Gamma \vdash A$ *for a term* $a \in \Gamma \vdash A$. *Moreover, if* γ *is a morphism in* $\mathbb{C}$, *then* $T(\gamma)$ *is a pair consisting of the* type substitution *function* $A \mapsto A[\gamma]$ *and the type-indexed family of* term substitution *functions* $a \mapsto a[\gamma]$.
- *A* terminal object $[]$ *of* $\mathbb{C}$ *which represents the* empty context *and a terminal map* $\langle\rangle : \Delta \to []$ *which represents the* empty substitution.
- *A* context comprehension *which to an object* Γ *in* $\mathbb{C}$ *and a type* $A \in \mathrm{Type}(\Gamma)$ *associates an object* $\Gamma \cdot A$ *of* $\mathbb{C}$, *a morphism* $\mathrm{p}_A : \Gamma \cdot A \to \Gamma$ *of* $\mathbb{C}$ *and a term* $\mathrm{q} \in \Gamma \cdot A \vdash A[\mathrm{p}]$ *such the following universal property holds: for each object* Δ *in* $\mathbb{C}$, *morphism* $\gamma : \Delta \to \Gamma$, *and term* $a \in \Delta \vdash A[\gamma]$, *there is a unique morphism* $\theta = \langle \gamma, a \rangle : \Delta \to \Gamma \cdot A$, *such that* $\mathrm{p}_A \circ \theta = \gamma$ *and* $\mathrm{q}[\theta] = a$. *(We remark that a related notion of comprehension for hyperdoctrines was introduced by Lawvere [7].)*

The definition of cwf can be presented as a system of axioms and inference rules for a variable-free generalized algebraic formulation of the most basic rules of dependent type theory [4]. The correspondence with standard syntax is explained by Hofmann [6] and the equivalence is proved in detail by Mimram [12]. The easiest way to understand this correspondence might be as a translation between the standard lambda calculus based syntax of dependent type theory and the language of cwf-combinators. In one direction the key idea is to translate a variable (de Bruijn number) to a projection of the form $\mathrm{q}[\mathrm{p}^n]$. In the converse direction, recall that the cwf-combinators yield a calculus of explicit substitutions whereas substitution is a meta-operation in usual lambda calculus. When we translate cwf-combinators to lambda terms, we execute the explicit substitutions, using the equations for substitution in types and terms as rewrite rules. The equivalence proof is similar to the proof of the equivalence of cartesian closed categories and the simply typed lambda calculus.

We shall now define what it means that a cwf supports extra structure corresponding to the rules for the various type formers of Martin-Löf type theory.

Definition 3. *A cwf* supports (extensional) identity types *provided the following conditions hold:*

Form. *If* $A \in \mathrm{Type}(\Gamma)$ *and* $a, a' : \Gamma \vdash A$*, there is* $\mathrm{I}_A(a, a') \in \mathrm{Type}(\Gamma)$*;*
Intro. *If* $a : \Gamma \vdash A$*, there is* $\mathrm{r}_{A,a} : \Gamma \vdash \mathrm{I}_A(a, a)$*;*
Elim. *If* $c : \Gamma \vdash \mathrm{I}_A(a, a')$ *then* $a = a'$ *and* $c = \mathrm{r}_{A,a}$*.*

Moreover, we have stability under substitution: if $\delta : \Delta \to \Gamma$ *then*

$$\begin{aligned} \mathrm{I}_A(a, a')[\delta] &= \mathrm{I}_{A[\delta]}(a[\delta], a'[\delta]) \\ \mathrm{r}_{A,a}[\delta] &= \mathrm{r}_{A[\delta],a[\delta]} \end{aligned}$$

Definition 4. *A cwf* supports Σ-types *iff the following conditions hold:*

Form. *If* $A \in \mathrm{Type}(\Gamma)$ *and* $B \in \mathrm{Type}(\Gamma \cdot A)$*, there is* $\Sigma(A, B) \in \mathrm{Type}(\Gamma)$*,*
Intro. *If* $a : \Gamma \vdash A$ *and* $b : \Gamma \vdash B[\langle \mathrm{id}, a\rangle]$*, there is* $\mathrm{pair}(a, b) : \Gamma \vdash \Sigma(A, B)$*,*
Elim. *If* $a : \Gamma \vdash \Sigma(A, B)$*, there are* $\pi_1(a) : \Gamma \vdash A$ *and* $\pi_2(a) : \Gamma \vdash B[\langle \mathrm{id}, \pi_1(a)\rangle]$ *such that*

$$\begin{aligned} \pi_1(\mathrm{pair}(a, b)) &= a \\ \pi_2(\mathrm{pair}(a, b)) &= b \\ \mathrm{pair}(\pi_1(c), \pi_2(c)) &= c \end{aligned}$$

Moreover, we have stability under substitution:

$$\begin{aligned} \Sigma(A, B)[\delta] &= \Sigma(A[\delta], B[\langle \delta \circ \mathrm{p}, \mathrm{q}\rangle]) \\ \mathrm{pair}(a, b)[\delta] &= \mathrm{pair}(a[\delta], b[\delta]) \\ \pi_1(c)[\delta] &= \pi_1(c[\delta]) \\ \pi_2(c)[\delta] &= \pi_2(c[\delta]) \end{aligned}$$

Note that in a cwf which supports extensional identity types and Σ-types surjective pairing, $\mathrm{pair}(\pi_1(c), \pi_2(c)) = c$, follows from the other conditions [10].

Definition 5. *A cwf* supports Π-types *iff the following conditions hold:*

Form. *If* $A \in \mathrm{Type}(\Gamma)$ *and* $B \in \mathrm{Type}(\Gamma \cdot A)$*, there is* $\Pi(A, B) \in \mathrm{Type}(\Gamma)$*.*
Intro. *If* $b : \Gamma \cdot A \vdash B$*, there is* $\lambda(b) : \Gamma \vdash \Pi(A, B)$*.*
Elim. *If* $c : \Gamma \vdash \Pi(A, B)$ *and* $a : \Gamma \vdash A$ *then there is a term* $\mathrm{ap}(c, a) : \Gamma \vdash B[\langle \mathrm{id}, a\rangle]$ *such that*

$$\begin{aligned} \mathrm{ap}(\lambda(b), a) &= b[\langle \mathrm{id}, a\rangle] : \Gamma \vdash B[\langle \mathrm{id}, a\rangle] \\ c &= \lambda(\mathrm{ap}(c[\mathrm{p}], \mathrm{q})) : \Gamma \vdash \Pi(A, B) \end{aligned}$$

Moreover, we have stability under substitution:

$$\begin{aligned}\Pi(A,B)[\gamma] &= \Pi(A[\gamma], B[\langle \gamma \circ \mathrm{p}, \mathrm{q}\rangle])\\ \lambda(b)[\gamma] &= \lambda(b[\langle \gamma \circ \mathrm{p}, \mathrm{q}\rangle])\\ \mathrm{ap}(c,a)[\gamma] &= \mathrm{ap}(c[\gamma], a[\gamma])\end{aligned}$$

Definition 6. *A cwf* $(\mathbb{C}, T)$ *is* democratic *iff for each object* Γ *of* $\mathbb{C}$ *there is* $\overline{\Gamma} \in \mathrm{Type}([])$ *and an isomorphism* $\Gamma \cong_{\gamma_\Gamma} [] \cdot \overline{\Gamma}$. *Each substitution* $\delta : \Delta \to \Gamma$ *can then be represented by the term* $\overline{\delta} = q[\gamma_\Gamma \delta \gamma_\Delta^{-1}] : [] \cdot \overline{\Delta} \vdash \overline{\Gamma}[p]$.

Democracy does not correspond to a rule of Martin-Löf type theory. However, a cwf generated inductively by the standard rules of Martin-Löf type theory with a one element type N_1 and Σ-types is democratic, since we can associate N_1 to the empty context and the closed type $\Sigma x_1 : A_1. \cdots .\Sigma x_n : A_n$ to a context $x_1 : A_1, \ldots, x_n : A_n$ by induction on n.

2.2 The Indexed Category of Types in Context

We shall now define the indexed category associated with a cwf. This will play a crucial role and in particular introduce the notion of *isomorphism* of types.

Proposition 7 (The Context-Indexed Category of Types). *If* $(\mathbb{C}, T)$ *is a cwf, then we can define a functor* $\boldsymbol{T} : \mathbb{C}^{op} \to \mathbf{Cat}$ *as follows:*

- *The objects of* $\boldsymbol{T}(\Gamma)$ *are types in* $\mathrm{Type}(\Gamma)$. *If* $A, B \in \mathrm{Type}(\Gamma)$, *then a morphism in* $\boldsymbol{T}(\Gamma)(A,B)$ *is a morphism* $\delta : \Gamma \cdot A \to \Gamma \cdot B$ *in* $\mathbb{C}$ *such that* $\mathrm{p}\delta = \mathrm{p}$.
- *If* $\gamma : \Delta \to \Gamma$ *in* $\mathbb{C}$, *then* $\boldsymbol{T}(\gamma) : \mathrm{Type}(\Gamma) \to \mathrm{Type}(\Delta)$ *maps an object* $A \in \mathrm{Type}(\Gamma)$ *to* $A[\gamma]$ *and a morphism* $\delta : \Gamma \cdot A \to \Gamma \cdot B$ *to* $\langle \mathrm{p}, \mathrm{q}[\delta \langle \gamma \circ \mathrm{p}, \mathrm{q}\rangle]\rangle : \Delta \cdot A[\gamma] \to \Delta \cdot B[\gamma]$.

We write $A \cong_\theta B$ if $\theta : A \to B$ is an isomorphism in $\boldsymbol{T}(\Gamma)$. If $a : \Gamma \vdash A$, we write $\{\theta\}(a) = q[\theta\langle id, a\rangle] : \Gamma \vdash B$ for the *coercion* of a to type B and $a =_\theta b$ if $a = \{\theta\}(b)$. Moreover, we get an alternative formulation of democracy.

Proposition 8. $(\mathbb{C}, T)$ *is democratic iff the functor from* $\boldsymbol{T}([])$ *to* $\mathbb{C}$, *which maps a closed type* A *to the context* $[] \cdot A$, *is an equivalence of categories.*

Seely's category **ML** of Martin-Löf theories [13] is essentially the category of categories $\boldsymbol{T}([])$ of closed types.

Fibres, slices and lcccs. Seely's interpretation of type theory in lcccs relies on the idea that a type $A \in \mathrm{Type}(\Gamma)$ can be interpreted as its *display map*, that is, a morphism with codomain Γ. For instance, the type $\texttt{list}(n)$ of lists of length $n : \texttt{nat}$ would be mapped to the function $l : \texttt{list} \to \texttt{nat}$ which to each list associates its length. Hence, types and terms in context Γ are interpreted in the *slice category* $\mathbb{C}/\Gamma$, since terms are interpreted as global sections. Syntactic types are connected with types-as-display-maps by the following result, an analogue of which was one of the cornerstones of Seely's paper.

Proposition 9. *If* $(\mathbb{C}, T)$ *is democratic and supports extensional identity and* Σ*-types, then* $\boldsymbol{T}(\Gamma)$ *and* $\mathbb{C}/\Gamma$ *are equivalent categories for all* Γ*.*

Proof. To each object (type) A in $\boldsymbol{T}(\Gamma)$ we associate the object p_A in $\mathbb{C}/\Gamma$. A morphism from A to B in $\boldsymbol{T}(\Gamma)$ is by definition a morphism from p_A to p_B in $\mathbb{C}/\Gamma$.

Conversely, to each object (morphism) $\delta : \Delta \to \Gamma$ of $\mathbb{C}/\Gamma$ we associate a type in $\mathrm{Type}(\Gamma)$. This is the inverse image $x : \Gamma \vdash \mathrm{Inv}(\delta)(x)$ which is defined type-theoretically by

$$\mathrm{Inv}(\delta)(x) = \Sigma y : \overline{\Delta}.\mathrm{I}_{\overline{\Gamma}}(\overline{x}, \overline{\delta}(y))$$

written in ordinary notation. In cwf combinator notation it becomes

$$\mathrm{Inv}(\delta) = \Sigma(\overline{\Delta}[\langle\rangle], \mathrm{I}_{\overline{\Gamma}[\langle\rangle]}(\mathrm{q}[\gamma_\Gamma \mathrm{p}], \overline{\delta}[\langle\langle\rangle, \mathrm{q}\rangle]) \in \mathrm{Type}(\Gamma)$$

These associations yield an equivalence of categories since $\mathrm{p}_{\mathrm{Inv}(\delta)}$ and δ are isomorphic in $\mathbb{C}/\Gamma$.

It is easy to see that $\boldsymbol{T}(\Gamma)$ has binary products if the cwf supports Σ-types and exponentials if it supports Π-types. Simply define $A \times B = \Sigma(A, B[\mathrm{p}])$ and $B^A = \Pi(A, B[\mathrm{p}])$. Hence by Proposition 9 it follows that $\mathbb{C}/\Gamma$ has products and $\mathbb{C}$ has finite limits in any democratic cwf which supports extensional identity types and Σ-types. If it supports Π-types too, then $\mathbb{C}/\Gamma$ is cartesian closed and $\mathbb{C}$ is locally cartesian closed.

3 The 2-Category of Categories with Families

3.1 Pseudo Cwf-Morphisms

A notion of *strict cwf-morphism* between cwfs $(\mathbb{C}, T)$ and $(\mathbb{C}', T')$ was defined by Dybjer [4]. It is a pair (F, σ), where $F : \mathbb{C} \to \mathbb{C}'$ is a functor and $\sigma : T \xrightarrow{\bullet} T'F$ is a natural transformation of family-valued functors, such that terminal objects and context comprehension are preserved on the nose. Here we need a weak version where the terminal object, context comprehension, and substitution of types and terms of a cwf are only preserved up to isomorphism. The pseudo-natural transformations needed to prove our biequivalences will be families of cwf-morphisms which do not preserve cwf-structure on the nose.

The definition of pseudo cwf-morphism will be analogous to that of *strict* cwf-morphism, but cwf-structure will only be preserved up to coherent isomorphism.

Definition 10. *A* ***pseudo cwf-morphism*** *from* $(\mathbb{C}, T)$ *to* $(\mathbb{C}', T')$ *is a pair* (F, σ) *where:*

- $F : \mathbb{C} \to \mathbb{C}'$ *is a functor,*
- *For each context* Γ *in* $\mathbb{C}$*,* σ_Γ *is a* **Fam**-*morphism from* $T\Gamma$ *to* $T'F\Gamma$*. We will write* $\sigma_\Gamma(A) : \mathrm{Type}'(F\Gamma)$ *for the type component and* $\sigma_\Gamma^A(a) : F\Gamma \vdash \sigma_\Gamma(A)$ *for the term component of this morphism.*

The following preservation properties must be satisfied:

- *Substitution is preserved: For each context* $\delta : \Delta \to \Gamma$ *in* $\mathbb{C}$ *and* $A \in \mathrm{Type}(\Gamma)$, *there is an isomorphism of types* $\theta_{A,\delta} : \sigma_\Gamma(A)[F\delta] \to \sigma_\Delta(A[\delta])$ *such that substitution on terms is also preserved, that is,* $\sigma_\Delta^{A[\gamma]}(a[\gamma]) =_{\theta_{A,\gamma}} \sigma_\Gamma^A(a)[F\gamma]$.
- *The terminal object is preserved:* $F[]$ *is terminal.*
- *Context comprehension is preserved:* $F(\Gamma{\cdot}A)$ *with the projections* $F(\mathrm{p}_A)$ *and* $\{\theta^{-1}_{A,\mathrm{p}}\}(\sigma^{A[p]}_{\Gamma{\cdot}A}(\mathrm{q}_A))$ *is a context comprehension of* $F\Gamma$ *and* $\sigma_\Gamma(A)$. *Note that the universal property on context comprehensions provides a unique isomorphism* $\rho_{\Gamma,A} : F(\Gamma\cdot A) \to F\Gamma\cdot\sigma_\Gamma(A)$ *which preserves projections.*

These data must satisfy naturality and coherence laws which amount to the fact that if we extend σ_Γ *to a functor* $\boldsymbol{\sigma}_\Gamma : \boldsymbol{T}(\Gamma) \to \boldsymbol{T'}F(\Gamma)$, *then* $\boldsymbol{\sigma}$ *is a pseudo natural transformation from* $\boldsymbol{T}$ *to* $\boldsymbol{T'}F$. *This functor is defined by* $\boldsymbol{\sigma}_\Gamma(A) = \sigma_\Gamma(A)$ *on an object* A *and* $\boldsymbol{\sigma}_\Gamma(f) = \rho_{\Gamma,B}F(f)\rho^{-1}_{\Gamma,A}$ *on a morphism* $f : A \to B$.

A consequence of this definition is that all cwf structure is preserved.

Proposition 11. *Let* (F,σ) *be a pseudo cwf-morphism from* $(\mathbb{C},T)$ *to* $(\mathbb{C}',T')$.

(1) Then substitution extension is preserved: for all $\delta : \Delta \to \Gamma$ *in* $\mathbb{C}$ *and* $a : \Delta \vdash A[\delta]$, *we have* $F(\langle\delta,a\rangle) = \rho^{-1}_{\Gamma,A}\langle F\delta, \{\theta^{-1}_{A,\delta}\}(\sigma_\Delta^{A[\delta]}(a))\rangle$.
(2) Redundancy terms/sections: for all $a \in \Gamma \vdash A$, $\sigma^A_\Gamma(a) = \mathrm{q}[\rho_{\Gamma,A}F(\langle\mathrm{id},a\rangle)]$.

If $(F,\sigma) : (\mathbb{C}_0,T_0) \to (\mathbb{C}_1,T_1)$ and $(G,\tau) : (\mathbb{C}_1,T_1) \to (\mathbb{C}_2,T_2)$ are two pseudo cwf-morphisms, we define their composition $(G,\tau)(F,\sigma)$ as $(GF,\tau\sigma)$ where:

$$(\tau\sigma)_\Gamma(A) = \tau_{F\Gamma}(\sigma_\Gamma(A))$$
$$(\tau\sigma)^A_\Gamma(a) = \tau^{\sigma_\Gamma(A)}_{F\Gamma}(\sigma^A_\Gamma(a))$$

The families θ^{GF} and ρ^{GF} are obtained from θ^F, θ^G and ρ^F and ρ^G in the obvious way. The fact that these data satisfy the necessary coherence and naturality conditions basically amounts to the stability of pseudonatural transformation under composition. There is of course an identity pseudo cwf-morphism whose components are all identities, which is obviously neutral for composition. So, there is a category of cwfs and pseudo cwf-morphisms.

Since the isomorphism $(\Gamma{\cdot}A){\cdot}B \cong \Gamma{\cdot}\Sigma(A,B)$ holds in an arbitrary cwf which supports Σ-types, it follows that pseudo cwf-morphisms automatically preserve Σ-types, since they preserve context comprehension. However, if cwfs support other structure, we need to define what it means that cwf-morphisms preserve this extra structure up to isomorphism.

Definition 12. *Let* (F,σ) *be a pseudo cwf-morphism between cwfs* $(\mathbb{C},T)$ *and* $(\mathbb{C}',T')$ *which support identity types,* Π*-types, and democracy, respectively.*

- (F,σ) preserves identity types *provided* $\sigma_\Gamma(\mathrm{I}_A(a,a')) \cong \mathrm{I}_{\sigma_\Gamma(A)}(\sigma^A_\Gamma(a), \sigma^A_\Gamma(a))$;
- (F,σ) preserves Π-types *provided* $\sigma_\Gamma(\Pi(A,B)) \cong \Pi(\sigma_\Gamma(A), \sigma_{\Gamma{\cdot}A}(B)[\rho^{-1}_{\Gamma,A}])$;

– (F, σ) preserves democracy *provided* $\sigma_{[]}(\overline{\Gamma}) \cong_{d_\Gamma} \overline{F\Gamma}[\langle\rangle]$*, and the following diagram commutes:*

$$\begin{array}{ccccc}
F\Gamma & & \xrightarrow{\;F\gamma_\Gamma\;} & & F([]\cdot\overline{\Gamma}) \\
\downarrow{\scriptstyle \gamma_{F\gamma}} & & & & \downarrow{\scriptstyle \rho_{[],\overline{\Gamma}}} \\
[]\cdot\overline{F\Gamma} & \xleftarrow{\;\langle\langle\rangle,\mathrm{q}\rangle\;} & F[]\cdot\overline{F\Gamma}[\langle\rangle] & \xleftarrow{\;d_\Gamma\;} & F[]\cdot\sigma_{[]}(\overline{\Gamma})
\end{array}$$

These preservation properties are all stable under composition and thus yield several different 2-categories of structure-preserving pseudo cwf-morphisms.

3.2 Pseudo Cwf-Transformations

Definition 13 (Pseudo cwf-transformation). *Let (F, σ) and (G, τ) be two cwf-morphisms from $(\mathbb{C}, T)$ to $(\mathbb{C}', T')$. A* pseudo cwf-transformation *from (F, σ) to (G, τ) is a pair (ϕ, ψ) where $\phi : F \stackrel{\bullet}{\to} G$ is a natural transformation, and for each Γ in $\mathbb{C}$ and $A \in \mathrm{Type}(\Gamma)$, a morphism $\psi_{\Gamma,A} : \sigma_\Gamma(A) \to \tau_\Gamma(A)[\phi_\Gamma]$ in $\boldsymbol{T'}(F\Gamma)$, natural in A and such that the following diagram commutes:*

$$\begin{array}{ccc}
\sigma_\Gamma(A)[F\delta] & \xrightarrow{\;\boldsymbol{T'}(F\delta)(\psi_{\Gamma,A})\;} & \tau_\Gamma(A)[\phi_\Gamma F(\delta)] \\
\downarrow{\scriptstyle \theta_{A,\delta}} & & \downarrow{\scriptstyle \boldsymbol{T'}(\phi_\Delta)(\theta'_{A,\delta})} \\
\sigma_\Delta(A[\delta]) & \xrightarrow[\;\psi_{\Delta,A[\delta]}\;]{} & \tau_\Delta(A[\delta])[\phi_\Delta]
\end{array}$$

where θ and θ' are the isomorphisms witnessing preservation of substitution in types in the definition of pseudo cwf-morphism.

Pseudo cwf-transformations can be composed both vertically (denoted by $(\phi', \psi')(\phi, \psi)$) and horizontally (denoted by $(\phi', \psi') \star (\phi, \psi)$), and these compositions are associative and satisfy the interchange law. Note that just as coherence and naturality laws for pseudo cwf-morphisms ensure that they give rise to pseudonatural transformations (hence morphisms of indexed categories) $\boldsymbol{\sigma}$ to $\boldsymbol{\tau}$, this definition exactly amounts to the fact that pseudo cwf-transformations between (F, σ) and (F, τ) correspond to modifications from $\boldsymbol{\sigma}$ to $\boldsymbol{\tau}$.

3.3 2-Categories of Cwfs with Extra Structure

Definition 14. *Let $\mathbf{CwF}^{\mathrm{I_{ext}}\Sigma}_{\mathrm{dem}}$ be the 2-category of small democratic categories with families which support extensional identity types and Σ-types. The 1-cells are cwf-morphisms preserving democracy and extensional identity types (and Σ-types automatically) and the 2-cells are pseudo cwf-transformations.*

Moreover, let $\mathbf{CwF}^{\mathrm{I_{ext}}\Sigma\Pi}_{\mathrm{dem}}$ be the sub-2-category of $\mathbf{CwF}^{\mathrm{I_{ext}}\Sigma}_{\mathrm{dem}}$ where also Π-types are supported and preserved.

4 Forgetting Types and Terms

Definition 15. *Let* **FL** *be the 2-category of small categories with finite limits (left exact categories). The* 1*-cells are functors preserving finite limits (up to isomorphism) and the* 2*-cells are natural transformations.*

Let **LCC** *be the 2-category of small locally cartesian closed categories. The* 1*-cells are functors preserving local cartesian closed structure (up to isomorphism), and the* 2*-cells are natural transformations.*

FL is a sub(2-)category of the 2-category of categories: we do not provide a choice of finite limits. Similarly, **LCC** is a sub(2-)category of **FL**. The first component of our biequivalences will be *forgetful* 2-functors.

Proposition 16. *The forgetful* 2*-functors*

$$U : \mathbf{CwF}_{\mathrm{dem}}^{\mathrm{I_{ext}}\Sigma} \to \mathbf{FL}$$
$$U : \mathbf{CwF}_{\mathrm{dem}}^{\mathrm{I_{ext}}\Sigma\Pi} \to \mathbf{LCC}$$

defined as follows on 0-, 1-, and 2-cells

$$U(\mathbb{C}, T) = \mathbb{C}$$
$$U(F, \sigma) = F$$
$$U(\phi, \psi) = \phi$$

are well-defined.

Proof. By definition U is a 2-functor from **CwF** to **Cat**, it remains to prove that it sends a cwf in $\mathbf{CwF}_{\mathrm{dem}}^{\mathrm{I_{ext}}\Sigma}$ to **FL** and a cwf in $\mathbf{CwF}_{\mathrm{dem}}^{\mathrm{I_{ext}}\Sigma\Pi}$ to **LCC**, along with the corresponding properties for 1-cells and 2-cells.

For 0-cells we already proved as corollaries of Proposition 9 that if $(\mathbb{C}, T)$ supports Σ-types, identity types and democracy, then $\mathbb{C}$ has finite limits; and if $(\mathbb{C}, T)$ also supports Π-types, then $\mathbb{C}$ is an lccc.

For 1-cells we need to prove that if (F, σ) preserves identity types and democracy, then F preserves finite limits; and if (F, σ) also preserves Π-types then F preserves local exponentiation. Since finite limits and local exponentiation in $\mathbb{C}$ and $\mathbb{C}'$ can be defined by the inverse image construction, these two statements boil down to the fact that if (F, σ) preserves identity types and democracy then inverse images are preserved. Indeed we have an isomorphism $F(\Gamma \cdot \mathrm{Inv}(\delta)) \cong F\Gamma \cdot \mathrm{Inv}(F\delta)$. This can be proved by long but mostly direct calculations involving all components and coherence laws of pseudo cwf-morphisms.

There is nothing to prove for 2-cells.

5 Rebuilding Types and Terms

Now, we turn to the reverse construction. We use the Bénabou-Hofmann construction to build a cwf from any finitely complete category, then generalize this operation to functors and natural transformations, and show that this gives rise to a pseudofunctor.

Proposition 17. *There are pseudofunctors*

$$H : \mathbf{FL} \to \mathbf{CwF}^{\mathrm{I_{ext}}\Sigma}_{\mathrm{dem}}$$
$$H : \mathbf{LCC} \to \mathbf{CwF}^{\mathrm{I_{ext}}\Sigma\Pi}_{\mathrm{dem}}$$

defined by

$$H\mathbb{C} = (\mathbb{C}, T_{\mathbb{C}})$$
$$HF = (F, \sigma_F)$$
$$H\phi = (\phi, \psi_\phi)$$

on 0-cells, 1-cells, and 2-cells, respectively, and where $T_{\mathbb{C}}, \sigma_F$*, and* ψ_ϕ *are defined in the following three subsections.*

Proof. The remainder of this Section contains the proof. We will in turn show the action on 0-cells, 1-cells, 2-cells, and then prove pseudofunctoriality of H.

5.1 Action on 0-Cells

As explained before, it is usual (going back to Cartmell [2]) to represent a type-in-context $A \in \mathrm{Type}(\Gamma)$ in a category as a *display map* [15], that is, as an object p_A in $\mathbb{C}/\Gamma$. A term $\Gamma \vdash A$ is then represented as a section of the display map for A, that is, a morphism a such that $\mathrm{p}_A \circ a = \mathrm{id}_\Gamma$. Substitution in types is then represented by pullback. This is essentially the technique used by Seely for interpreting Martin-Löf type theory in lcccs. However, as we already mentioned, it leads to a coherence problem.

To solve this problem Hofmann [5] used a construction due to Bénabou [1], which from any fibration builds an equivalent *split* fibration. Hofmann used it to build a category with attributes (cwa) [2] from a locally cartesian closed category. He then showed that this cwa supports Π, Σ, and extensional identity types. This technique essentially amounts to associating to a type A, not only a display map, but a whole family of display maps, one for each substitution instance $A[\delta]$. In other words, we choose a pullback square for every possible substitution and this choice is split, hence solving the coherence problem. As we shall explain below this family takes the form of a functor, and we refer to it as a *functorial family*.

Here we reformulate Hofmann's construction using cwfs. See Dybjer [4] for the correspondence between cwfs and cwas.

Lemma 18. *Let* $\mathbb{C}$ *be a category with terminal object. Then we can build a democratic cwf* $(\mathbb{C}, T_{\mathbb{C}})$ *which supports* Σ*-types. If* $\mathbb{C}$ *has finite limits, then* $(\mathbb{C}, T_{\mathbb{C}})$ *also supports extensional identity types. If* $\mathbb{C}$ *is locally cartesian closed, then* $(\mathbb{C}, T_{\mathbb{C}})$ *also supports* Π*-types.*

Proof. We only show the definition of types and terms in $T_{\mathbb{C}}(\Gamma)$. This construction is essentially the same as Hofmann's [5].

A *type* in $\mathrm{Type}_{\mathbb{C}}(\Gamma)$ is a *functorial family*, that is, a functor $\overrightarrow{A} : \mathbb{C}/\Gamma \to \mathbb{C}^{\to}$ such that $\mathrm{cod} \circ \overrightarrow{A} = \mathrm{dom}$ and if $\begin{array}{ccc} \Omega & \xrightarrow{\alpha} & \Delta \\ {\scriptstyle \delta\alpha} \searrow & \Gamma & \swarrow {\scriptstyle \delta} \end{array}$ is a morphism in $\mathbb{C}/\Gamma$, then $\overrightarrow{A}(\alpha)$ is a pullback square:

$$\begin{array}{ccc} \cdot & \xrightarrow{\overrightarrow{A}(\delta,\alpha)} & \cdot \\ {\scriptstyle \overrightarrow{A}(\delta\alpha)} \downarrow & & \downarrow {\scriptstyle \overrightarrow{A}(\delta)} \\ \Omega & \xrightarrow[\alpha]{} & \Delta \end{array}$$

Following Hofmann, we denote the upper arrow of the square by $\overrightarrow{A}(\delta, \alpha)$.

A *term* $a : \Gamma \vdash \overrightarrow{A}$ is a section of $\overrightarrow{A}(\mathrm{id}_\Gamma)$, that is, a morphism $a : \Gamma \to \Gamma \cdot \overrightarrow{A}$ such that $\overrightarrow{A}(\mathrm{id}_\Gamma)a = \mathrm{id}_\Gamma$, where we have defined context extension by $\Gamma \cdot \overrightarrow{A} = \mathrm{dom}(\overrightarrow{A}(\mathrm{id}_\Gamma))$. Interpreting types as functorial families makes it easy to define substitution in types. Substitution in terms is obtained by exploiting the universal property of pullback squares, yielding a functor $T_{\mathbb{C}} : \mathbb{C}^{op} \to \mathbf{Fam}$.

Note that $(\mathbb{C}, T_{\mathbb{C}})$ is a *democratic* cwf since to any context Γ we can associate a functorial family $\widehat{\langle\rangle} : \mathbb{C}/[] \to C^{\to}$, where $\langle\rangle : \Gamma \to []$ is the terminal projection. The isomorphism $\gamma_\Gamma : \Gamma \to [] \cdot \widehat{\langle\rangle}$ is just id_Γ.

5.2 Action on 1-Cells

Suppose that $\mathbb{C}$ and $\mathbb{C}'$ have finite limits and that $F : \mathbb{C} \to \mathbb{C}'$ preserves them. As described in the previous section, $\mathbb{C}$ and $\mathbb{C}'$ give rise to cwfs $(\mathbb{C}, T_{\mathbb{C}})$ and $(\mathbb{C}', T_{\mathbb{C}'})$. In order to extend F to a pseudo cwf-morphism, we need to define, for each object Γ in $\mathbb{C}$, a **Fam**-morphism $(\sigma_F)_\Gamma : T_{\mathbb{C}}(\Gamma) \to T_{\mathbb{C}'}F(\Gamma)$. Unfortunately, unless F is full, it does not seem possible to embed faithfully a functorial family $\overrightarrow{A} : \mathbb{C}/\Gamma \to \mathbb{C}^{\to}$ into a functorial family over $F\Gamma$ in $\mathbb{C}'$. However, there is such an embedding for display maps (just apply F) from which we will freely regenerate a functorial family from the obtained display map.

The "hat" construction. As remarked by Hofmann, any morphism $f : \Delta \to \Gamma$ in a category $\mathbb{C}$ with a (not necessarily split) choice of finite limits generates a functorial family $\widehat{f} : \mathbb{C}/\Gamma \to \mathbb{C}^{\to}$. If $\delta : \Delta \to \Gamma$ then $\widehat{f}(\delta) = \delta^*(f)$, where $\delta^*(f)$ is obtained by taking the pullback of f along δ (δ^* is known as the *pullback functor*):

$$\begin{array}{ccc} \cdot & \longrightarrow & \cdot \\ {\scriptstyle \delta^*(f)} \downarrow & & \downarrow {\scriptstyle f} \\ \Delta & \xrightarrow[\delta]{} & \Gamma \end{array}$$

Note that we can always choose pullbacks such that $\widehat{f}(\mathrm{id}_\Gamma) = \mathrm{id}_\Gamma^*(f) = f$. If $\begin{array}{ccc} \Omega & \xrightarrow{\alpha} & \Delta \\ {\scriptstyle \delta\alpha} \searrow & \Gamma & \swarrow {\scriptstyle \delta} \end{array}$ is a morphism in $\mathbb{C}/\Gamma$, we define $\widehat{f}(\alpha)$ as the left square in the following diagram:

$$\begin{array}{ccccc} \cdot & \xrightarrow{\widehat{f}(\delta,\alpha)} & \cdot & \longrightarrow & \cdot \\ {\scriptstyle \widehat{f}(\delta\alpha)}\downarrow & & \downarrow{\scriptstyle \widehat{f}(\delta)} & & \downarrow{\scriptstyle f} \\ \Delta' & \xrightarrow[\alpha]{} & \Delta & \xrightarrow[\delta]{} & \Gamma \end{array}$$

This is a pullback, since both the outer square and the right square are pullbacks.

Translation of types. The hat construction can be used to extend F to types:

$$\sigma_F(\overrightarrow{A}) = \widehat{F(\overrightarrow{A}(\mathrm{id}))}$$

Note that $F(\Gamma\cdot\overrightarrow{A}) = F(\mathrm{dom}(\overrightarrow{A}(\mathrm{id}))) = \mathrm{dom}(F(\overrightarrow{A}(\mathrm{id}))) = \mathrm{dom}(\sigma_\Gamma(\overrightarrow{A})(\mathrm{id})) = F\Gamma\cdot\sigma_\Gamma(\overrightarrow{A})$, so context comprehension is preserved on the nose. However, substitution on types is *not* preserved on the nose. Hence we have to define a coherent family of isomorphisms $\theta_{\overrightarrow{A},\delta}$.

Completion of cwf-morphisms. Fortunately, whenever F preserves finite limits there is a canonical way to generate all the remaining data.

Lemma 19 (Generation of isomorphisms). *Let* $(\mathbb{C}, T)$ *and* $(\mathbb{C}', T')$ *be two cwfs,* $F : \mathbb{C} \to \mathbb{C}'$ *a functor preserving finite limits,* $\sigma_\Gamma : \mathrm{Type}(\Gamma) \to \mathrm{Type}'(F\Gamma)$ *a family of functions, and* $\rho_{\Gamma,A} : F(\Gamma\cdot A) \to F\Gamma\cdot\sigma_\Gamma(A)$ *a family of isomorphisms such that* $\mathrm{p}\rho_{\Gamma,A} = F\mathrm{p}$. *Then there exists an unique choice of functions* σ_Γ^A *on terms and of isomorphisms* $\theta_{A,\delta}$ *such that* (F, σ) *is a pseudo cwf-morphism.*

Proof. By item *(2)* of Proposition 11, the unique way to extend σ to terms is to set $\sigma_\Gamma^A(a) = \mathrm{q}[\rho_{\Gamma,A}F(\langle\mathrm{id}, a\rangle)]$. To generate θ, we use the two squares below:

$$\begin{array}{ccc} F\Delta\cdot\sigma_\Gamma(A)[F\delta] & \xrightarrow{\langle (F\delta)\mathrm{pq}\rangle} & F\Gamma\cdot\sigma_\Gamma(A) \\ {\scriptstyle \mathrm{p}}\downarrow & & \downarrow{\scriptstyle \mathrm{p}} \\ F\Delta & \xrightarrow[F\delta]{} & F\Gamma \end{array} \qquad \begin{array}{ccc} F\Delta\cdot\sigma_\Delta(A[\delta]) & \xrightarrow{\rho_{\Gamma,A}F(\langle\delta\mathrm{p},\mathrm{q}\rangle)\rho^{-1}_{\Delta,A[\delta]}} & F\Gamma\cdot\sigma_\Gamma(A) \\ {\scriptstyle \mathrm{p}}\downarrow & & \downarrow{\scriptstyle \mathrm{p}} \\ F\Delta & \xrightarrow[F\delta]{} & F\Gamma \end{array}$$

The first square is a substitution pullback. The second is a pullback because F preserves finite limits and $\rho_{\Gamma,A}$ and $\rho_{\Delta,A[\delta]}$ are isomorphisms. The isomorphism $\theta_{A,\delta}$ is defined as the unique mediating morphism from the first to the second. It follows from the universal property of pullbacks that the family θ satisfies the necessary naturality and coherence conditions. There is no other choice for $\theta_{A,\delta}$, because if (F, σ) is a pseudo cwf-morphism with families of isomorphisms θ and ρ, then $\rho_{\Gamma,A}F(\langle\delta\mathrm{p}, \mathrm{q}\rangle)\rho^{-1}_{\Delta,A[\delta]}\theta_{A,\delta} = \langle(F\delta)\mathrm{p}, \mathrm{q}\rangle$. Hence if F preserves finite limits, $\theta_{A,\delta}$ must coincide with the mediating morphism.

Preservation of additional structure. As a pseudo cwf-morphism, (F, σ_F) automatically preserves Σ-types. Since the democratic structure of $(\mathbb{C}, T_{\mathbb{C}})$ and $(\mathbb{C}', T_{\mathbb{C}'})$ is trivial it is clear that it is preserved by (F, σ_F). To prove that it also preserves type constructors, we use the following proposition.

Proposition 20. *Let (F, σ) be a pseudo cwf-morphism between $(\mathbb{C}, T)$ and $(\mathbb{C}', T')$ supporting Σ-types and democracy. Then:*

- *If $(\mathbb{C}, T)$ and $(\mathbb{C}', T')$ both support identity types, then (F, σ) preserves identity types provided F preserves finite limits.*
- *If $(\mathbb{C}, T)$ and $(\mathbb{C}', T')$ both support Π-types, then (F, σ) preserves Π-types provided F preserves local exponentiation.*

Proof. For the first part it remains to prove that if F preserves finite limits, then (F, σ) preserves identity types. Since $a, a' \in \Gamma \vdash A$, $\mathrm{p}_{\mathrm{I}_A(a,a')} : \Gamma \cdot \mathrm{I}_A(a, a') \to \Gamma$ is an equalizer of $\langle \mathrm{id}, a \rangle$ and $\langle \mathrm{id}, a' \rangle$ and F preserves equalizers, it follows that $F(\mathrm{p}_{\mathrm{I}_A(a,a')})$ is an equalizer of $\langle \mathrm{id}, \sigma_\Gamma^A(a) \rangle$ and $\langle \mathrm{id}, \sigma_\Gamma^A(a') \rangle$, and by uniqueness of equalizers it is isomorphic to $\mathrm{I}_{\sigma_\Gamma(A)}(\sigma_\Gamma^A(a), \sigma_\Gamma^A(a'))$.

The proof of preservation of Π-types exploits in a similar way the uniqueness (up to iso) of "Π-objects" of $A \in \mathrm{Type}(\Gamma)$ and $B \in \mathrm{Type}(\Gamma \cdot A)$.

5.3 Action on 2-Cells

Similarly to the case of 1-cells, under some conditions a natural transformation $\phi : F \xrightarrow{\bullet} G$ where (F, σ) and (G, τ) are pseudo cwf-morphisms can be completed to a pseudo cwf-transformation (ϕ, ψ_ϕ), as stated below.

Lemma 21 (Completion of pseudo cwf-transformations). *Suppose (F, σ) and (G, τ) are pseudo cwf-morphisms from $(\mathbb{C}, T)$ to $(\mathbb{C}', T)$ such that F and G preserve finite limits and $\phi : F \xrightarrow{\bullet} G$ is a natural transformation, then there exists a family of morphisms $(\psi_\phi)_{\Gamma, A} : \sigma_\Gamma(A) \to \tau_\Gamma(A)[\phi_\Gamma]$ such that (ϕ, ψ_ϕ) is a pseudo cwf-transformation from (F, σ) to (G, τ).*

Proof. We set $\psi_{\Gamma,A} = \langle \mathrm{p}, \mathrm{q}[\rho'_{\Gamma,A} \phi_{\Gamma \cdot A} \rho_{\Gamma,A}^{-1}] \rangle : F\Gamma \cdot \sigma_\Gamma A \to F\Gamma \cdot \tau_\Gamma(A)[\phi_\Gamma]$. To check the coherence law, we apply the universal property of a well-chosen pullback square (exploiting the fact that G preserves finite limits).

This completion operation on 2-cells commutes with units and both notions of composition, as will be crucial to prove pseudofunctoriality of H:

Lemma 22. *If $\phi : F \xrightarrow{\bullet} G$ and $\phi' : G \xrightarrow{\bullet} H$, then*

$$(\phi', \psi_{\phi'})(\phi, \psi_\phi) = (\phi'\phi, \psi_{\phi'\phi})$$
$$(\phi, \psi_\phi) \star 1 = (\phi \star 1, \psi_{\phi \star 1})$$
$$1 \star (\phi, \psi_\phi) = (1 \star \phi, \psi_{1 \star \phi})$$
$$(\phi', \psi_{\phi'}) \star (\phi, \psi_\phi) = (\phi' \star \phi, \psi_{\phi' \star \phi})$$

whenever these expressions typecheck.

Proof. Direct calculations.

5.4 Pseudofunctoriality of *H*

Note that H is *not* a functor, because for any $F : \mathbb{C} \to \mathbb{D}$ with finite limits and functorial family $\overrightarrow{A}$ over Γ (in $\mathbb{C}$), $\sigma_\Gamma(\overrightarrow{A})$ forgets all information on $\overrightarrow{A}$ except its display map $\overrightarrow{A}(\mathrm{id})$, and later extends $F(\overrightarrow{A}(\mathrm{id}))$ to an independent functorial family. However if $F : \mathbb{C} \to \mathbb{D}$ and $G : \mathbb{D} \to \mathbb{E}$ preserve finite limits, the two pseudo cwf-morphisms $(G, \sigma^G) \circ (F, \sigma^F) = (GF, \sigma^G \sigma^F)$ and (GF, σ^{GF}) are related by the pseudo cwf-transformation $(1_{GF}, \psi_{1_{GF}})$, which is obviously an isomorphism. The coherence laws only involve vertical and horizontal compositions of units and pseudo cwf-transformations obtained by completion, hence they are easy consequences of Lemma 22.

6 The Biequivalences

Theorem 23. *We have the following biequivalences of* 2*-categories.*

$$\mathbf{FL} \underset{U}{\overset{H}{\rightleftarrows}} \mathbf{CwF}^{\mathrm{I_{ext}}\,\Sigma}_{\mathrm{dem}} \qquad\qquad \mathbf{LCC} \underset{U}{\overset{H}{\rightleftarrows}} \mathbf{CwF}^{\mathrm{I_{ext}}\,\Sigma\Pi}_{\mathrm{dem}}$$

Proof. Since $UH = \mathrm{Id}$ (the identity 2-functor) it suffices to construct pseudonatural transformations of pseudofunctors:

$$\mathrm{Id} \underset{\epsilon}{\overset{\eta}{\rightleftarrows}} HU$$

which are inverse up to invertible modifications. Since $HU(\mathbb{C}, T) = (\mathbb{C}, T^{\mathbb{C}})$, these pseudonatural transformations are families of equivalences of cwfs:

$$(\mathbb{C}, T) \underset{\epsilon_{(\mathbb{C},T)}}{\overset{\eta_{(\mathbb{C},T)}}{\rightleftarrows}} (\mathbb{C}, T^{\mathbb{C}})$$

which satisfy the required conditions for pseudonatural transformations.

Construction of $\eta_{(\mathbb{C},T)}$. Using Lemma 19, we just need to define a base functor, which will be $\mathrm{Id}_{\mathbb{C}}$, and a family σ^η_Γ which translates types (in the sense of T) to functorial families. This is easy, since types in the cwf $(\mathbb{C}, T)$ come equipped with a chosen behaviour under substitution. Given $A \in \mathrm{Type}(\Gamma)$, we define:

$$\sigma^\eta_\Gamma(A)(\delta) = \mathrm{p}_{A[\delta]}$$
$$\sigma^\eta_\Gamma(A)(\delta, \gamma) = \langle \gamma \mathrm{p}, \mathrm{q} \rangle$$

For each pseudo cwf-morphism (F, σ), the pseudonaturality square relates two pseudo cwf-morphisms whose base functor is F. Hence, the necessary invertible pseudo cwf-transformation is obtained using Lemma 21 from the identity natural transformation on F. The coherence conditions are straightforward consequences of Lemma 22.

Construction of $\epsilon_{(\mathbb{C},T)}$. As for η, the base functor for $\epsilon_{(\mathbb{C},T)}$ is $\mathrm{Id}_{\mathbb{C}}$. Using Lemma 19 again we need, for each context Γ, a function $\sigma^{\epsilon}_{\Gamma}$ which given a functorial family $\overrightarrow{A}$ over Γ will build a syntactic type $\sigma^{\epsilon}_{\Gamma}(\overrightarrow{A}) \in \mathrm{Type}(\Gamma)$. In other terms, we need to find a syntactic representative of an arbitrary display map, that is, an arbitrary morphism in $\mathbb{C}$. We use the inverse image:

$$\sigma^{\epsilon}_{\Gamma}(\overrightarrow{A}) = \mathrm{Inv}(\overrightarrow{A}(\mathrm{id})) \in \mathrm{Type}(\Gamma)$$

The family ϵ is pseudonatural for the same reason as η above.

Invertible modifications. For each cwf $(\mathbb{C}, T)$, we need to define invertible pseudo cwf-transformations $m_{(\mathbb{C},T)} : (\epsilon\eta)_{(\mathbb{C},T)} \rightarrow \mathrm{id}_{(\mathbb{C},T)}$ and $m'_{(\mathbb{C},T)} : (\eta\epsilon)_{(\mathbb{C},T)} \rightarrow \mathrm{id}_{(\mathbb{C},T)}$. As pseudo cwf-transformations between pseudo cwf-morphisms with the same base functor, their first component will be the identity natural transformation, and the second will be generated by Lemma 21. The coherence law for modifications is a consequence of Lemma 22.

References

1. Jean Bénabou. Fibred categories and the foundation of naive category theory. *Journal of Symbolic Logic*, 50:10–37, 1985.
2. John Cartmell. Generalized algebraic theories and contextual categories. *Annals of Pure and Applied Logic*, 32:209–243, 1986.
3. Pierre-Louis Curien. Substitution up to isomorphism. *Fundamenta Informaticae*, 19(1,2):51–86, 1993.
4. 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.
5. 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.
6. Martin Hofmann. Syntax and semantics of dependent types. In Andrew Pitts and Peter Dybjer, editors, *Semantics and Logics of Computation*. Cambridge University Press, 1996.
7. 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.
8. Tom Leinster. Basic bicategories. `arXiv:math/9810017v1`, 1999.
9. Per Martin-Löf. Constructive mathematics and computer programming. In *Logic, Methodology and Philosophy of Science, VI, 1979*, pages 153–175. North-Holland, 1982.
10. Per Martin-Löf. *Intuitionistic Type Theory*. Bibliopolis, 1984.
11. Per Martin-Löf. Substitution calculus. Notes from a lecture given in Göteborg, November 1992.
12. Samuel Mimram. Decidability of equality in categories with families. Report, Magistère d'Informatique et Modelisation, École Normale Superieure de Lyon, `http://www.pps.jussieu.fr/~smimram/`, 2004.
13. Robert Seely. Locally cartesian closed categories and type theory. *Math. Proc. Cambridge Philos. Soc.*, 95(1):33–48, 1984.

14. 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. Licentiate Thesis.
15. Paul Taylor. *Practical Foundations of Mathematics*. Cambridge University Press, 1999.