Undecidability of Equality in the Free Locally Cartesian Closed Category (Extended Version)

Simon Castellan, Pierre Clairambault, Peter Dybjer

UNDECIDABILITY OF EQUALITY IN THE FREE LOCALLY CARTESIAN CLOSED CATEGORY (EXTENDED VERSION)

SIMON CASTELLAN, PIERRE CLAIRAMBAULT, AND PETER DYBJER

Univ Lyon, CNRS, ENS de Lyon, UCB Lyon 1, LIP
e-mail address: {simon.castellan,pierre.clairambault}@ens-lyon.fr

Chalmers University of Technology
e-mail address: peterd@chalmers.se

Abstract. We show that a version of Martin-Löf type theory with an extensional identity type former I, a unit type N_1, Σ-types, Π-types, and a base type is a free category with families (supporting these type formers) both in a 1- and a 2-categorical sense. It follows that the underlying category of contexts is a free locally cartesian closed category in a 2-categorical sense because of a previously proved biequivalence. We show that equality in this category is undecidable by reducing it to the undecidability of convertibility in combinatory logic. Essentially the same construction also shows a slightly strengthened form of the result that equality in extensional Martin-Löf type theory with one universe is undecidable.

1. Introduction

In previous work [5, 6] we showed the biequivalence of locally cartesian closed categories (lcccs) and the I, Σ, Π-fragment of extensional Martin-Löf type theory. More precisely, we showed the biequivalence of the following two 2-categories.

- The first has as *objects* lcccs, as *arrows* functors which preserve the lccc-structure (up to isomorphism), and as *2-cells* natural transformations.
- The second has as *objects* categories with families (cwfs) [8, 11] which support extensional identity types (I-types), Σ-types, Π-types, and are *democratic*, as *arrows* pseudo cwf-morphisms (preserving structure up to isomorphism), and as *2-cells* pseudo cwf-transformations. A cwf is democratic iff there is an equivalence between its category of contexts and its category of closed types.

This result is a corrected version of a result by Seely [15] concerning the equivalence of the category of lcccs and the category of Martin-Löf type theories. Seely's paper did not address the coherence problem caused by the interpretation of substitution as pullbacks [7].

1998 ACM Subject Classification: F.3.2, F.4.1.
Key words and phrases: Extensional Type Theory, Undecidability, Locally Cartesian Closed Categories.
This paper is an extended version of [4].

LOGICAL METHODS IN COMPUTER SCIENCE
DOI:10.2168/LMCS-???

As Hofmann showed [9], this coherence problem can be solved by extending a construction of Bénabou [2]. Our biequivalence is based on this construction.

Cwfs are models of the most basic rules of dependent type theory; those dealing with substitution, assumption, and context formation, the rules which come before any rules for specific type formers. The distinguishing feature of cwfs, compared to other categorical notions of model of dependent types, is that they are formulated in a way which makes the connection with the ordinary syntactic formulation of dependent type theory transparent. They can be defined purely equationally [8] as a generalised algebraic theory (gat) [3], where each sort symbol corresponds to a judgment form, and each operator symbol corresponds to an inference rule in a variable free formulation of Martin-Löf's explicit substitution calculus for dependent type theory [13, 17].

Cwfs provide a basic theory of dependently typed n-place functions. We remark that *non-dependent* cwfs, in which there is a *fixed* set of *types*, are closely related to (cartesian) multicategories, where the *terms* of the cwf correspond to *multiarrows*. A difference is however that a multiarrow always comes with a finite list of input objects, whereas the cwf-axioms do not force the input context of a term to be a list.

Cwfs are not only models of dependent type theory, but also suggest an answer to the question what dependent type theory is as a mathematical object. Perhaps surprisingly, this is a non-trivial question, and Voevodsky has remarked that "a type system is not a mathematical notion". There are numerous variations of Martin-Löf type theory in the literature, even of the formulation of the most basic rules for dependent types. There are systems with explicit and implicit substitutions, and there are variations in assumption, context formation, and substitution rules. There are formulations with de Bruijn indices and with ordinary named variables, etc. In fact, there are so many rules that most papers do not try to provide a complete list; and if you do try to list all of them how can you be sure that you have not forgotten any? Nevertheless, there is a tacit assumption that most variations are equivalent and that a complete list of rules could be given if needed. However, from a mathematical point of view this is neither clear nor elegant.

To remedy this situation we suggest to define Martin-Löf type theory (and other dependent type theories) abstractly as the initial cwf (with extra structure). The category of cwfs and morphisms which preserve cwf-structure on the nose was defined by Dybjer [8]. We suggest that the correctness of a definition or an implementation of dependent type theory means that it gives rise to an initial object in this category of cwfs (with extra structure). Here we shall construct the initial object in this category explicitly in the simplest possible way following closely the definition of the generalised algebraic theory of cwfs. Note however that the notion of a generalised algebraic theory is itself based on dependent type theory, that is, on cwf-structure. So just defining the initial cwf as the generalised algebraic theory of cwfs would be circular.

Instead we construct the initial cwf explicitly by giving grammar and inference rules which follow closely the operators of the gat of cwfs. However, we must also make equality reasoning explicit. To decrease the number of rules, we present a "per-style" system rather than an ordinary one. We will mutually define four partial equivalence relations (pers): for the judgments of context equality $\Gamma = \Gamma'$, substitution equality $\Delta \vdash \gamma = \gamma' : \Gamma$, type equality $\Gamma \vdash A = A'$, and term equality $\Gamma \vdash a = a' : A$. The ordinary judgments will be defined as the reflexive instances. For example, $\Gamma \vdash a : A$ will be defined as $\Gamma \vdash a = a : A$. There are altogether 32 inference rules for the pure theory of cwfs: the first 8 rules express that we define four families of pers; the second 3 rules that judgments preserve equality of contexts and types; the next 10 rules express the typing and congruence of the 10 cwf-operations; and the final 11 rules are the conversion rules for these operations. In addition to the pure theory of cwfs, we have 1 rule for the base type.

Our only optimisation is the elimination of some redundant arguments of operators. For example, the composition operator in the gat of cwfs has five arguments: three objects and two arrows. However, the three object arguments can be recovered from the arrows, and can hence be omitted.

The goal of the present paper is to prove the undecidability of equality in the free lccc. To this end we extend our formal system for cwfs with rules for extensional I-types, $\mathrm{N}_1, \Sigma, \Pi$, and a base type. (Note that we have added the unit type N_1 to the type formers needed for the proof of biequivalence with lcccs. This is because we need to construct a democratic cwf, where there is a bijection between types and contexts (see above). Therefore we need the type N_1 which corresponds to the empty context.) There are 5 rules for I-types, 3 rules for N_1, 11 rules for Σ, and 8 rules for Π. We want to show that this yields a free lccc on one object, by appealing to our biequivalence theorem. However, in order to use our biequivalence it does not suffice to show that we get a free cwf in the 1-category of cwfs and strict cwf-morphisms: we must show that it is also free ("bifree") in the 2-category of cwfs and pseudo cwf-morphisms. Indeed, biequivalences do not preserve initial objects in general: uniquess of a morphism $\mathbf{0} \to A$ out of an initial object is lost. The proof of bifreeness is technically more involved because of the complexity of the notion of pseudo cwf-morphism.

Once we have constructed the free lccc (as a cwf-formulation of Martin-Löf type theory with extensional I-types, $\mathrm{N}_1, \Sigma, \Pi$, and one base type) we will be able to prove undecidability. It is well-known that extensional Martin-Löf type theory with one universe (folklore) or with natural numbers [10] has undecidable equality, and we only need to show that a similar construction can be made without a universe and without natural numbers, provided we have a base type. We do this by encoding untyped combinatory logic as a context, and use the undecidability of equality in this theory.

Related work. Palmgren and Vickers [14] show how to construct free models of essentially algebraic theories in general. We could use this result to build a free cwf, but this only shows freeness in the 1-categorical sense. We also think that the explicit construction of the free (and bifree) cwf is interesting in its own right.

Plan. In Section 2 we prove a few undecidability theorems, including the undecidability of equality in Martin-Löf type theory with extensional I-types, Π-types, and one base type. In Section 3 we construct a free cwf on one base type. We show that it is free and bifree. In Section 4 we construct a free and bifree cwf with extensional identity types, $\mathrm{N}_1, \Sigma, \Pi$, and one base type. Since this cwf is democratic we can use the biequivalence result to conclude that this yields a free lccc in a 2-categorical sense.

2. Undecidability in Martin-Löf type theory

Like any other single-sorted first order equational theory, combinatory logic can be encoded as a context in Martin-Löf type theory with I-types, Π-types, and a base type o. The context Γ_{CL} for combinatory logic is the following:

$$k \;:\; o,$$

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

The left-associative binary infix symbol "$\cdot$" stands for application. Note that $k, s, \cdot, ax_k, ax_s$ are all variables.

Theorem 2.1. *Type-inhabitation in Martin-Löf type theory with (intensional or extensional) identity-types,* Π*-types and a base type is undecidable.*

This follows from the undecidability of convertibility in combinatory logic, since the type

$$\Gamma_{\mathrm{CL}} \vdash \mathrm{I}(o,\ M,\ M')$$

is inhabited iff the closed combinatory terms M and M' are convertible. Clearly, if the combinatory terms are convertible, it can be formalised in this fragment of type theory. For the other direction we build a model of the context Γ_{CL} where o is interpreted as the set of combinatory terms modulo convertibility.

Theorem 2.2. *Judgmental equality in Martin-Löf type theory with extensional identity-types,* Π*-types and a base type is undecidable.*

With extensional identity types [12] the above identity type is inhabited iff the corresponding equality judgment is valid:

$$\Gamma_{\mathrm{CL}} \vdash M = M' : o$$

This theorem also holds if we add N_1 and Σ-types to the theory. The remainder of the paper will show that the category of contexts of the resulting fragment of Martin-Löf type theory is bifree in the 2-category of lcccs (Theorem 4.25). Our main result follows:

Theorem 2.3. *Equality of arrows in the bifree lccc on one object is undecidable.*

We remark that the following folklore theorem can be proved in the same way as Theorem 2.2. (We are not aware of any published proof of this theorem, but see Hofmann [10] for a proof which instead uses the natural number type.)

Theorem 2.4. *Judgmental equality in Martin-Löf type theory with extensional identity-types,* Π*-types and a universe* U *is undecidable.*

If we have a universe we can instead work in the context

$$\begin{array}{rcl} X & : & \mathrm{U}, \\ k & : & X, \\ s & : & X, \\ \cdot & : & X \to X \to X, \\ ax_k & : & \Pi xy : X.\ \mathrm{I}(X,\ k \cdot x \cdot y,\ x), \\ ax_s & : & \Pi xyz : X.\ \mathrm{I}(X,\ s \cdot x \cdot y \cdot z,\ x \cdot z \cdot (y \cdot z)) \end{array}$$

and prove undecidability for this theory (without a base type) in the same way as above.

Note that we don't need any closure properties at all for U – only the ability to quantify over small types. Hence we prove a slightly stronger theorem than the folklore theorem which assumes that U is closed under function types and uses the context

$$\begin{aligned} X &: \mathrm{U}, \\ x &: \mathrm{I}(\mathrm{U}, X, X \to X) \end{aligned}$$

so that X is a model of the untyped lambda calculus.

3. A free category with families

In this section we define a free cwf syntactically, as a *term model* consisting of derivable well-formed contexts, substitutions, types and terms modulo derivable equality. To this end we give syntax and inference rules for a cwf-calculus, that is, a variable-free explicit substitution calculus for dependent type theory.

We first prove that this calculus yields a free cwf in the category where morphisms preserve cwf-structure on the nose. The free cwf on one object is a rather degenerate structure, since there are no non-trivial dependent types. However, we have nevertheless chosen to present this part of the construction separately. Cwfs model the common core of dependent type theory, including all generalised algebraic theories, pure type systems [1], and fragments of Martin-Löf type theory. The construction of a free pure cwf is thus the common basis for constructing free and initial cwfs with appropriate extra structure for modelling specific dependent type theories.

In Section 3.1 we start by recalling the definition of cwfs, the associated morphisms – both those preserving structure in the strict sense and up to isomorphism – and some related definitions and notations. In Section 3.2, we introduce our syntax and inference rules. In Section 3.3, we show that these inference rules give rise to a *free* cwf, in the category of cwfs and strict cwf-morphisms. Finally, in Section 3.4 we prove that our free cwf is also bifree in the 2-category of cwf-morphisms preserving structure up to isomorphism.

3.1. The 2-category of categories with families.

The 2-category of cwfs and pseudo-morphisms which preserve cwf-structure up to isomorphism was defined in [5, 6]. Here we only give an outline.

Notations. We write **Fam** for the category of families of sets: objects are families of sets $(X_i)_{i\in I}$ and maps from $(X_i)_{i\in I}$ to $(Y_j)_{j\in J}$ are pairs $(f : I \to J, (f_i : X_i \to Y_{f(i)})_{i\in I})$. In a category with families, contexts and substitutions form the objects and arrows of a category $\mathcal{C}$. The set of objects will be written $\mathrm{Ctx}_{\mathcal{C}}$ and the set of morphisms from Δ to Γ will be written $\mathrm{Sub}_{\mathcal{C}}(\Delta, \Gamma)$. Types and terms over a context Γ form a family $(\mathrm{Tm}_{\mathcal{C}}(\Gamma, A))_{A\in \mathrm{Ty}_{\mathcal{C}}\Gamma}$, and substitution gives rise to a functorial action on such a family. Thus we have a functor

$$T : \mathcal{C}^{\mathrm{op}} \to \mathbf{Fam}$$

The action of T on objects is $T\Gamma = (\mathrm{Tm}_{\mathcal{C}}(\Gamma, A))_{A\in \mathrm{Ty}_{\mathcal{C}}\Gamma}$, and its action on a type A is written $A[\_]$: if $\gamma \in \mathrm{Sub}_{\mathcal{C}}(\Gamma, \Delta)$ and $A \in \mathrm{Ty}_{\mathcal{C}}(\Delta)$, then $A[\gamma] \in \mathrm{Ty}_{\mathcal{C}}(\Gamma)$. Similarly, if $a \in \mathrm{Tm}_{\mathcal{C}}(\Delta, A)$, we write $a[\gamma] \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, A[\gamma])$ for the functorial action of T on a.

Definition 3.1 (Category with families)**.** A cwf is given by a category $\mathcal{C}$ and a functor $T : \mathcal{C}^{\mathrm{op}} \to \mathbf{Fam}$ together with the following chosen structure:

- (*Empty context*) $\mathcal{C}$ has a terminal object 1.

- (*Context comprehension*) For each $\Delta \in \mathrm{Ctx}_{\mathcal{C}}$ and $A \in \mathrm{Ty}_{\mathcal{C}}(\Delta)$ there is the extended context $\Delta.A \in \mathrm{Ctx}_{\mathcal{C}}$ with a substitution $\mathtt{p}_A \in \mathrm{Sub}_{\mathcal{C}}(\Delta.A, \Delta)$ and a term $\mathtt{q}_A \in \mathrm{Tm}_{\mathcal{C}}(\Delta.A, A[\mathtt{p}_A])$, such that for every pair $\gamma \in \mathrm{Sub}_{\mathcal{C}}(\Gamma, \Delta)$ and $a \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, A[\gamma])$ there exists a unique
$$\langle \gamma, a \rangle \in \mathrm{Sub}_{\mathcal{C}}(\Gamma, \Delta.A)$$
such that $\mathtt{p}_A \circ \langle \gamma, a \rangle = \gamma$ and $\mathtt{q}_A[\langle \gamma, a \rangle] = a$.

When talking about cwfs, we will often refer only to the base category $\mathcal{C}$ and keep the rest of the structure implicit.

Note that with the notation $\mathrm{Ty}_{\mathcal{C}}$ and $\mathrm{Tm}_{\mathcal{C}}$ there is no need to explicitly mention the functor T when working with categories with families, and we will often keep it implicit. Given a substitution $\gamma : \Delta \to \Gamma$, and $A \in \mathrm{Ty}_{\mathcal{C}}(\Gamma)$, we write $\gamma \uparrow A$ or γ^+ (when A can be inferred from the context) for the lifting of γ to A: $\langle \gamma \circ \mathtt{p}, \mathtt{q} \rangle : \Delta.A[\gamma] \to \Gamma.A$.

The indexed category. In [5, 6] it is shown that any cwf $\mathcal{C}$ induces a functor $\mathbf{T} : \mathcal{C}^{\mathrm{op}} \to \mathbf{Cat}$ assigning to each context Γ the category whose objects are types in $\mathrm{Ty}_{\mathcal{C}}(\Gamma)$ and morphisms from A to B are substitutions $\varphi : \Gamma.A \to \Gamma.B$ such that $\mathtt{p} \circ \varphi = \mathtt{p}$ – those are in bijection with terms of type $\Gamma.A \vdash B[\mathtt{p}]$. The functorial action of $\mathbf{T}$ is given by
$$\mathbf{T}(\gamma)(\varphi) = \langle \mathtt{p}, \mathtt{q}[\varphi \circ (\gamma \uparrow A)] \rangle : \Delta.A[\gamma] \to \Delta.B[\gamma]$$
for $\gamma : \Delta \to \Gamma$.

Any morphism φ in $\mathbf{T}\Gamma$ from a type A to a type B induces a function $\{\varphi\} : \mathrm{Tm}_{\mathcal{C}}(\Gamma, A) \to \mathrm{Tm}_{\mathcal{C}}(\Gamma, B)$ which is defined by
$$\{\varphi\}(a) = \mathtt{q}[\varphi \circ \langle \mathtt{id}, a \rangle]$$

We will use this construction when transporting terms through *isomorphism of types* $\theta : A \cong_\Gamma B$, that is, isomorphisms in $\mathbf{T}\Gamma$. We note the following:

Lemma 3.2. *For any $\gamma : \Delta \to \Gamma$, $\varphi : \Gamma.A \to \Gamma.B$ in $\mathbf{T}\Gamma$, and $a \in \mathrm{Tm}_{\mathcal{C}}(\Delta, A[\gamma])$,*
$$\{\mathbf{T}(\gamma)(\varphi)\}(a) = \mathtt{q}[\varphi \circ \langle \gamma, a \rangle]$$

Proof. Immediate from the definition. □

Definition 3.3 (Pseudo cwf-morphisms)**.** A pseudo-cwf morphism from a cwf $\mathcal{C}$ to a cwf $\mathcal{C}'$ is a pair (F, σ) where $F : \mathcal{C} \to \mathcal{C}'$ is a functor and for each $\Gamma \in \mathrm{Ctx}_{\mathcal{C}}$, σ_Γ is a **Fam**-morphism from $T\Gamma$ to $T'F\Gamma$ preserving the structure up to isomorphism. In particular there are isomorphisms
$$\begin{array}{rccll} \rho_{\Gamma,A} : & F(\Gamma.A) & \cong & F\Gamma.FA & \\ \theta_{A,\gamma} : & FA[F\gamma] & \cong_{F\Gamma} & F(A[\gamma]) & \qquad (\text{for } \gamma : \Gamma \to \Delta) \\ !_F : & 1 & \cong & F1 & \end{array}$$
satisfying some coherence diagrams, see Appendix A for the complete definition.

Since σ_Γ is a **Fam**-morphism from $(\mathrm{Tm}_{\mathcal{C}}(\Gamma, A))_{A \in \mathrm{Ty}_{\mathcal{C}}(\Gamma)}$ to $(\mathrm{Tm}_{\mathcal{C}'}(F\Gamma, B))_{B \in \mathrm{Ty}_{\mathcal{C}'}(F\Gamma)}$ it has an action both on types and on terms. We write FA for the image of A by the function $\mathrm{Ty}_{\mathcal{C}}(\Gamma) \to \mathrm{Ty}_{\mathcal{C}'}(F\Gamma)$ induced by σ_Γ, and Fa for the image of $a \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, A)$ through the function $\mathrm{Tm}_{\mathcal{C}}(\Gamma, A) \to \mathrm{Tm}_{\mathcal{C}'}(F\Gamma, FA)$ induced by σ_Γ. As for cwfs, we will often refer to a pseudo cwf-morphism (F, σ) just by F, keeping σ implicit. This goes in line with the notations introduced above, which do not mention σ.

A pseudo cwf-morphism is strict whenever $\theta_{A,\gamma}$ and $\rho_{\Gamma,A}$ are both identities and $F1 = 1$. Cwfs and strict cwf-morphisms form a category $\mathbf{CwF}_s$.

Definition 3.4 (Pseudo cwf-transformation)**.** A pseudo cwf-transformation between pseudo cwf-morphisms F and G is a pair (φ, ψ) where $\varphi : F \Rightarrow G$ is a natural transformation, and for each $\Gamma \in \mathrm{Ctx}_{\mathcal{C}}$ and $A \in \mathrm{Ty}_{\mathcal{C}}(\Gamma)$, $\psi_{\Gamma,A}$ is a type isomorphism $FA \cong_{F\Gamma} GA[\varphi_\Gamma]$ satisfying:

$$\varphi_{\Gamma.A} = F(\Gamma.A) \xrightarrow{\rho^F_{\Gamma,A}} F\Gamma.FA \xrightarrow{\psi_{\Gamma,A}} F\Gamma.GA[\varphi_\Gamma] \xrightarrow{\varphi^+_\Gamma} G\Gamma.GA \xrightarrow{{\rho^G_{\Gamma,A}}^{-1}} G(\Gamma.A),$$

This means in particular that ψ is uniquely determined from φ. However, it matches our inductive proof later on to have both φ and ψ explicitely in the definition, with this coherence diagram. This definition corrects the one given in [6]; see Appendix B for a discussion on that. We will write **CwF** for the resulting 2-category.

3.2. **Syntax and inference rules for the free category with families.**

3.2.1. *Raw terms.* In this section we define the syntax and inference rules for a minimal dependent type theory with one base type o. This theory is closely related to the generalised algebraic theory of cwfs [8], but here we define it as a usual logical system with a grammar and a collection of inference rules. The grammar has four syntactic categories: contexts `Ctx`, substitutions `Sub`, types `Ty` and terms `Tm`.

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

These terms have as few annotations as possible, only what is necessary to recover the domain and codomain of a substitution, the context of a type, and the type of a term:

$$\begin{array}{rlrl}
\mathrm{dom}(\gamma \circ \gamma') = \mathrm{dom}(\gamma') & & \mathrm{cod}(\gamma \circ \gamma') & = \mathrm{cod}(\gamma) \\
\mathrm{dom}(\mathtt{id}_\Gamma) = \Gamma & & \mathrm{cod}(\mathtt{id}_\Gamma) & = \Gamma \\
\mathrm{dom}(\langle\rangle_\Gamma) = \Gamma & & \mathrm{cod}(\langle\rangle_\Gamma) & = 1 \\
\mathrm{dom}(\mathtt{p}_A) = \text{ctx-of}(A).A & & \mathrm{cod}(\mathtt{p}_A) & = \text{ctx-of}(A) \\
\mathrm{dom}(\langle \gamma, a \rangle_A) = \mathrm{dom}(\gamma) & & \mathrm{cod}(\langle \gamma, a \rangle_A) & = \mathrm{cod}(\gamma).A \\
\\
\text{ctx-of}(o) = 1 & & \text{type-of}(a[\gamma]) & = (\text{type-of}(a))[\gamma] \\
\text{ctx-of}(A[\gamma]) = \mathrm{dom}(\gamma) & & \text{type-of}(\mathtt{q}_A) & = A[\mathtt{p}_A]
\end{array}$$

These functions will be used to define the interpretation.

3.2.2. *Inference rules.* We simultaneously inductively define four families of partial equivalence relations (pers) for the four forms of equality judgments:

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

In the inference rules which generate these pers we will use the following abbreviations for the basic judgment forms: $\Gamma \vdash$ abbreviates $\Gamma = \Gamma \vdash$, $\Gamma \vdash A$ abbreviates $\Gamma \vdash A = A$, $\Delta \vdash \gamma : \Gamma$ abbreviates $\Delta \vdash \gamma = \gamma : \Gamma$, and $\Gamma \vdash a : A$ abbreviates $\Gamma \vdash a = a : A$. The inference rules are divided into four kinds: *per-rules*, which axiomatise symmetry and transitivity of

Per-rules for the four forms of judgments

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

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

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

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

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

$$\frac{\Gamma \vdash A = A' \qquad \Delta \vdash \gamma = \gamma' : \Gamma}{\Delta \vdash A[\gamma] = A'[\gamma']} \qquad \frac{\Gamma = \Gamma' \vdash}{\Gamma \vdash \mathtt{id}_\Gamma = \mathtt{id}_{\Gamma'} : \Gamma} \qquad \frac{\Gamma = \Gamma' \vdash}{\Gamma \vdash \langle\rangle_\Gamma = \langle\rangle_{\Gamma'} : 1}$$

$$\frac{\Gamma \vdash \delta = \delta' : \Delta \qquad \Delta \vdash \gamma = \gamma' : \Theta}{\Gamma \vdash \gamma \circ \delta = \gamma' \circ \delta' : \Theta} \qquad \frac{\Gamma \vdash A = A'}{\Gamma.A \vdash \mathtt{p}_A = \mathtt{p}_{A'} : \Gamma}$$

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

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

equality; *preservation rules*, which express that equality preserves judgments; *congruence rules* for operators with respect to equality, and *conversion rules*.

Note that our syntax is annotated in order to ensure that a raw term has a unique (up to judgmental equality) type given by the function type-of, and that a type has a unique (up to judgemental equality) context given by the function ctx-of. Similarly, dom and cod return the unique domain and codomain of a substitution.

Lemma 3.5. *We have the following:*

- *If* $\Gamma \vdash A$ *is derivable, then* $\Gamma = \text{ctx-of}(A) \vdash$ *is also derivable.*
- *If* $\Gamma \vdash a : A$ *is derivable, then* $\Gamma = \text{ctx-of}(A) \vdash$ *and* $\Gamma \vdash A = \text{type-of}(a)$ *are derivable.*
- *If* $\Delta \vdash \gamma : \Gamma$ *is derivable, then* $\Delta = \text{dom}(\gamma) \vdash$ *and* $\Gamma = \text{cod}(\gamma) \vdash$ *are derivable.*

3.2.3. *The syntactic cwf* $\mathcal{T}$. We can now define a term model as the syntactic cwf obtained by the well-formed contexts, substitutions, types, and terms, modulo judgmental equality.

Conversion rules

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

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

$$\frac{\Gamma \vdash a : A \qquad \Delta \vdash \gamma : \Gamma \qquad \Theta \vdash \delta : \Delta}{\Theta \vdash a[\gamma \circ \delta] = (a[\gamma])[\delta] : (A[\gamma])[\delta]} \qquad \frac{\Gamma \vdash a : A}{\Gamma \vdash a[\mathtt{id}_\Gamma] = a : A} \qquad \frac{\Gamma \vdash \gamma : 1}{\Gamma \vdash \gamma = \langle\rangle_\Gamma : 1}$$

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

$$\frac{\Delta \vdash \gamma : \Gamma.A}{\Delta \vdash \gamma = \langle \mathtt{p}_A \circ \gamma, \mathtt{q}_A[\gamma] \rangle_A : \Gamma.A}$$

We use brackets for equivalence classes in this definition. (Note that brackets are also used for substitution in types and terms. However, this should not cause confusion since we will soon drop the equivalence class brackets.)

Definition 3.6. The term model $\mathcal{T}$ is given by:

- $\mathrm{Ctx}_\mathcal{T} = \{\Gamma \mid \Gamma \vdash\}/{=^c}$, where $\Gamma =^c \Gamma'$ if $\Gamma = \Gamma' \vdash$ is derivable.
- $\mathrm{Sub}_\mathcal{T}([\Gamma],[\Delta]) = \{\gamma \mid \Gamma \vdash \gamma : \Delta\}/{=^\Gamma_\Delta}$ 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 class of Γ (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 $\Gamma \vdash a = a' : A$.

The cwf-operations on $\mathcal{T}$ can now be defined in a straightforward way. For example, if $\Delta \vdash \theta : \Theta$, $\Gamma \vdash \delta : \Delta$, we define $[\theta] \circ_\mathcal{T} [\delta] = [\theta \circ \delta]$, which is well-defined since composition preserves equality.

3.3. **Freeness of $\mathcal{T}$.** We shall show that $\mathcal{T}$ is a free cwf on one base type, in the sense that for an arbitrary cwf $\mathcal{C}$ and type $o_\mathcal{C} \in \mathrm{Ty}_\mathcal{C}(1_\mathcal{C})$, there exists a unique strict cwf morphism $\mathcal{T} \to \mathcal{C}$ which maps $[o]$ to $o_\mathcal{C}$. Such a morphism can be defined by first defining a partial function for each sort of raw terms (where `Ctx` denotes the set of raw contexts, `Sub` the set of raw substitutions, and so on defined by the grammar of Section 3.2.1), cf Streicher [16].

$$\begin{aligned}
[\![-]\!] &: \mathtt{Ctx} \rightharpoonup \mathrm{Ctx}_\mathcal{C} \\
[\![-]\!] &: (\gamma \in \mathtt{Sub}) \rightharpoonup \mathrm{Sub}_\mathcal{C}(\mathrm{dom}(\gamma), \mathrm{cod}(\gamma)) \\
[\![-]\!] &: (A \in \mathtt{Ty}) \rightharpoonup \mathrm{Ty}_\mathcal{C}(\text{ctx-of}(A)) \\
[\![-]\!] &: (t \in \mathtt{Tm}) \rightharpoonup \mathrm{Tm}_\mathcal{C}(\text{ctx-of}(\text{type-of}(t)), \text{type-of}(t))
\end{aligned}$$

We use the notation $(x \in A) \rightharpoonup B(x)$ for the *partial* dependent function space, that is, the set of partial functions f which map $x \in A$ to $f(x) \in B(x)$ whenever $f(x)$ is defined.

Note that we use the same notation for all four interpretation functions. These partial interpretation functions are defined by mutual induction on the structure of raw terms:

$$
\begin{array}{rclrclrcl}
[\![1]\!] & = & 1_{\mathcal{C}} & [\![\gamma' \circ \gamma]\!] & = & [\![\gamma']\!] \circ_{\mathcal{C}} [\![\gamma]\!] & [\![\langle\rangle_\Gamma]\!] & = & (\langle\rangle_{\mathcal{C}})_{[\![\Gamma]\!]} \\
[\![\Gamma.A]\!] & = & [\![\Gamma]\!].{}_{\mathcal{C}}[\![A]\!] & [\![\mathtt{id}_\Gamma]\!] & = & (\mathrm{id}_{\mathcal{C}})_{[\![\Gamma]\!]} & [\![a[\gamma]]\!] & = & [\![a]\!][[\![\gamma]\!]]_{\mathcal{C}} \\
[\![o]\!] & = & o_{\mathcal{C}} & [\![\langle\gamma, a\rangle_A]\!] & = & \langle[\![\gamma]\!], [\![a]\!]\rangle_{\mathcal{C}} & [\![\mathtt{q}_A]\!] & = & (\mathtt{q}_{\mathcal{C}})_{[\![A]\!]} \\
[\![A[\gamma]]\!] & = & [\![A]\!][[\![\gamma]\!]]_{\mathcal{C}} & [\![\mathtt{p}_A]\!] & = & (\mathtt{p}_{\mathcal{C}})_{[\![A]\!]} & & &
\end{array}
$$

Partiality arises because, for instance, $[\![\gamma']\!] \circ_{\mathcal{C}} [\![\gamma]\!]$ is only defined when $[\![\gamma']\!]$ and $[\![\gamma]\!]$ are defined and $\mathrm{dom}([\![\gamma']\!]) = \mathrm{cod}([\![\gamma]\!])$. However, we can prove by induction on the inference rules that the interpretation of equal well-formed contexts, equal well-typed substitutions, equal well-formed types, and equal well-typed terms are always defined and equal:

Lemma 3.7.

- *If $\Gamma = \Gamma' \vdash$, then both $[\![\Gamma]\!]$ and $[\![\Gamma']\!]$ are defined in $\mathrm{Ctx}_{\mathcal{C}}$, and equal.*
- *If $\Delta \vdash \gamma = \gamma' : \Gamma$, then $[\![\gamma]\!] = [\![\gamma']\!] \in \mathrm{Sub}_{\mathcal{C}}([\![\Delta]\!], [\![\Gamma]\!])$ are defined and equal.*
- *If $\Gamma \vdash A = A'$, then $[\![A]\!] = [\![A']\!] \in \mathrm{Ty}_{\mathcal{C}}([\![\Gamma]\!])$ are defined and equal.*
- *If $\Gamma \vdash a = a' : A$, then $[\![a]\!] = [\![a']\!] \in \mathrm{Tm}_{\mathcal{C}}([\![\Gamma]\!], [\![A]\!])$ are defined and equal.*

It follows in particular that if we have $\Gamma \vdash$ (which abbreviates $\Gamma = \Gamma \vdash$), then $[\![\Gamma]\!]$ is defined – and likewise for the other reflexive judgements. Hence, we can define total interpretation functions on the term model by restricting the partial interpretation function to the well-formed contexts, etc, and then lift it to the quotient:

$$
\begin{array}{rcl}
\overline{[\![-]\!]} & : & \mathrm{Ctx}_{\mathcal{T}} \ \to \ \mathrm{Ctx}_{\mathcal{C}} \\
\overline{[\![-]\!]}_{[\Gamma],[\Delta]} & : & \mathrm{Sub}_{\mathcal{T}}([\Gamma], [\Delta]) \ \to \ \mathrm{Sub}_{\mathcal{C}}(\overline{[\![[\Gamma]]\!]}, \overline{[\![[\Delta]]\!]}) \\
\overline{[\![-]\!]}_{[\Gamma]} & : & \mathrm{Ty}_{\mathcal{T}}([\Gamma]) \ \to \ \mathrm{Ty}_{\mathcal{C}}(\overline{[\![[\Gamma]]\!]}) \\
\overline{[\![-]\!]}_{[\Gamma],[A]} & : & \mathrm{Tm}_{\mathcal{T}}([\Gamma], [A]) \ \to \ \mathrm{Tm}_{\mathcal{C}}(\overline{[\![[\Gamma]]\!]}, \overline{[\![[A]]\!]}_{[\Gamma]})
\end{array}
$$

by

$$
\begin{array}{rcl}
\overline{[\![[\Gamma]]\!]} & = & [\![\Gamma]\!] \\
\overline{[\![[\gamma]]\!]}_{[\Gamma],[\Delta]} & = & [\![\gamma]\!] \\
\overline{[\![[A]]\!]}_{[\Gamma]} & = & [\![A]\!] \\
\overline{[\![[a]]\!]}_{[\Gamma],[A]} & = & [\![a]\!]
\end{array}
$$

which is well-defined by Lemma 3.5.

This defines a strict cwf morphism $\mathcal{T} \to \mathcal{C}$ which maps $[o]$ to $o_{\mathcal{C}}$. In order to prove that it is unique, we assume that $F : \mathcal{T} \to \mathcal{C}$ is another strict cwf morphism, and prove by induction on the inference rules (the pers) that if $\Gamma = \Gamma' \vdash$ then $F[\Gamma] = \overline{[\![[\Gamma]]\!]}$, etc. For example, $1 = 1 \vdash$ and we prove $F[1] = 1_{\mathcal{C}} = \overline{[\![[1]]\!]}$ by preservation of the terminal object. The other cases are similarly straightforward, since strict cwf-morphisms preserve the structure on the nose.

This concludes the proof of our theorem:

Theorem 3.8. *$\mathcal{T}$ is a free cwf on one object, that is, for every other cwf $\mathcal{C}$ and $o_{\mathcal{C}} \in \mathrm{Ty}_{\mathcal{C}}(1_{\mathcal{C}})$ there is a unique strict cwf morphism $\mathcal{T} \to \mathcal{C}$ which maps $[o]$ to $o_{\mathcal{C}}$.*

It is in fact *the* free cwf on one object up to isomorphism, since any two free cwfs are related by a unique isomorphism.

From now on we will uniformly drop the equivalence class brackets and for example write Γ for $[\Gamma]$. There should be no risk of confusion, but we remark that proofs by induction on syntax and inference rules are on representatives rather than equivalence classes.

3.4. **Bifreeness of $\mathcal{T}$.** We eventually wish to add the type formers $\mathrm{N}_1, \Sigma, \Pi$ and I, and construct the free cwf which supports these type formers. However, as we explained in the introduction, this freeness property will not transport to lcccs. Indeed, our correspondence between cwfs (with support for these type formers) and lcccs is a *biequivalence* [6] rather than an equivalence, and freeness is not preserved by biequivalence (the dimensions of the two notions mismatch – likewise, isomorphism is not preserved by biequivalence, but *equivalence* is). Moreover, so far we proved that $\mathcal{T}$ is free in the category of cwfs and *strict* cwf-morphisms which preserve cwf-structure on the nose. In lcccs, finite limits and local exponents are usually not treated as extra *structure*, but as *properties* of categories. Thus functors can only preserve these properties up to *isomorphism*, since it would not even make sense to say that these properties are preserved on the nose. As a consequence, in our biequivalence result we moved to *pseudo* cwf-morphisms (Definition 3.3) that only preserve structure up to coherent isomorphism. The cwf $\mathcal{T}$ is not free in the category of cwfs with pseudo cwf-morphisms – in fact, there is no free cwf in this category. However, we can move to a 2-categorical setting and show that $\mathcal{T}$ is *bifree*.

We recall that an object I is bi-initial in a 2-category iff for any object A there exists 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.

Definition 3.9. A cwf $\mathcal{C}$ is *bifree* on one base type iff it is bi-initial in the 2-category $\mathbf{CwF}^o$:

- *Objects*: pairs $(\mathcal{C}, o_\mathcal{C})$ where $\mathcal{C}$ is a cwf and $o_\mathcal{C} \in \mathrm{Ty}_\mathcal{C}(1_\mathcal{C})$.
- *Morphisms between* $(\mathcal{C}, o_\mathcal{C})$ *and* $(\mathcal{D}, o_\mathcal{D})$: pairs (F, α_F) of pseudo cwf-morphisms $F : \mathcal{C} \to \mathcal{D}$ and isomorphisms $\alpha_F : F(o_\mathcal{C})[!_F] \cong o_\mathcal{D}$ in the category of closed types $\mathbf{T}(1_\mathcal{D})$ (recall that $!_F : 1 \to F1$).
- *2-cells between the morphisms* $(F, \alpha_F), (G, \alpha_G) : (\mathcal{C}, o_\mathcal{C}) \to (\mathcal{D}, o_\mathcal{D})$: pseudo cwf-transformations (φ, ψ) from F to G satisfying $\psi_{o_\mathcal{C}} = \alpha_G^{-1} \circ \alpha_F : F(o_\mathcal{C})[!_F] \cong_{1_\mathcal{D}} G(o_\mathcal{C})[!_G]$.

The rest of the section is dedicated to the proof of the following:

Theorem 3.10. *$\mathcal{T}$ is a bifree cwf on one base type.*

We have shown that for every cwf $\mathcal{C}$, and $o_\mathcal{C} \in \mathrm{Ty}_\mathcal{C}(1_\mathcal{C})$, the interpretation $\overline{[\![-]\!]}$ is a strict cwf-morphism mapping o to $o_\mathcal{C}$. Hence it is a morphism in $\mathbf{CwF}^o$. It remains to show that for any other morphism $F : \mathcal{T} \to \mathcal{C}$ in $\mathbf{CwF}^o$, there is a unique 2-cell (pseudo cwf-transformation) $(\varphi, \psi) : \overline{[\![-]\!]} \to F$, which is an isomorphism. This asymmetric version of bi-initiality is equivalent to that given above.

3.4.1. *Existence of* (φ, ψ). We construct (φ, ψ) by induction on the inference rules and simultaneously prove their naturality properties:

- If $\Gamma = \Gamma' \vdash$, then there exists *an isomorphism* $\varphi_\Gamma = \varphi_{\Gamma'} : \overline{[\![\Gamma]\!]} \cong F\Gamma$.
- If $\Gamma \vdash A = A'$, then there exists *an isomorphism* $\psi_A = \psi_{A'} : [\![A]\!] \cong_{\overline{[\![\Gamma]\!]}} FA[\varphi_\Gamma]$.
- If $\Gamma \vdash \gamma = \gamma' : \Delta$, then $F\gamma \circ \varphi_\Gamma = \varphi_\Delta \circ \overline{[\![\gamma]\!]}$.
- If $\Gamma \vdash a = a' : A$, then $Fa[\varphi_\Gamma] = \{\psi_A\}(\overline{[\![a]\!]})$.

It follows that (φ, ψ) is a pseudo cwf-transformation. We show some crucial cases:

Empty context. F preserves terminal objects and we let $\varphi_1 = !_F : \overline{[\![1]\!]} = 1_{\mathcal{C}} \cong F1$.

Context extension. By induction, we have $\psi_A : \overline{[\![A]\!]} \cong FA[\varphi_\Gamma]$ in $\mathbf{T}(\Gamma)$. We define $\varphi_{\Gamma.A}$ as the following composition of isomorphisms:

$$\overline{[\![\Gamma.A]\!]} = \overline{[\![\Gamma]\!]}.\overline{[\![A]\!]} \xrightarrow{\psi_A} \overline{[\![\Gamma]\!]}.FA[\varphi_\Gamma] \xrightarrow{\langle \varphi_\Gamma \circ \mathtt{p}, \mathtt{q}\rangle} F\Gamma.FA \xrightarrow{\rho^{-1}_{\Gamma,A}} F(\Gamma.A)$$

We remark that this case of the induction concerns the rule that not only expresses the well-formedness of context extension, but more generally, that context extension preserves equality. So officially, we need to prove that $\varphi_\Gamma = \varphi_{\Gamma'} : \overline{[\![\Gamma]\!]} \cong F\Gamma$ and $\psi_A = \psi_{A'} : [\![A]\!] \cong_{\overline{[\![\Gamma]\!]}} FA[\varphi_\Gamma]$ entail $\varphi_{\Gamma.A} = \varphi_{\Gamma'.A'}$ which follows immediately. We also remark that we have dropped the official index A in $\mathtt{p}_A$ and $\mathtt{q}_A$ in the above definition. Both remarks apply in other cases too.

Base type. By definition, F is equipped with $\alpha_F : \overline{[\![o]\!]} \cong F(o)[!_F]$. We define $\psi_o = \alpha_F : \overline{[\![o]\!]} \cong F(o)[!_F]$ in $\mathrm{Ty}_{\mathcal{C}}(1)$.

Type substitution. Let $\Gamma \vdash \gamma : \Delta$ and $\Delta \vdash A$. The induction hypotheses are $\varphi_\Delta \circ \overline{[\![\gamma]\!]} = (F\gamma) \circ \varphi_\Gamma$ and $\psi_A : \overline{[\![A]\!]} \cong_{\overline{[\![\Delta]\!]}} FA[\varphi_\Delta]$. Since $\mathbf{T}$ is a contravariant functor, $\mathbf{T}\overline{[\![\gamma]\!]}$ is a functor from $\mathbf{T}\overline{[\![\Delta]\!]}$ to $\mathbf{T}\overline{[\![\Gamma]\!]}$ thus,

$$\mathbf{T}(\overline{[\![\gamma]\!]})(\psi_A) : \overline{[\![A[\gamma]]\!]} \cong_{\overline{[\![\Gamma]\!]}} FA[\varphi_\Delta \circ \overline{[\![\gamma]\!]}] = FA[F\gamma][\varphi_\Gamma]$$

by induction hypothesis on γ. So we define

$$\psi_{A[\gamma]} = \mathbf{T}(\varphi_\Gamma)(\theta_{A,\gamma}) \circ \mathbf{T}(\overline{[\![\gamma]\!]})(\psi_A) : \overline{[\![A[\gamma]]\!]} \cong_{\overline{[\![\Gamma]\!]}} (F(A[\gamma]))[\varphi_\Gamma]$$

Projection. We have $\Gamma.A \vdash \mathtt{p}_A : \Gamma$ and need to check that $F\mathtt{p}_A \circ \varphi_{\Gamma.A} = \varphi_\Gamma \circ \mathtt{p}$. This is a simple calculation:

$$\begin{aligned}
F\mathtt{p}_A \circ \varphi_{\Gamma.A} &= F\mathtt{p} \circ \rho^{-1}_{\Gamma,A} \circ \langle \varphi_\Gamma \circ \mathtt{p}, \mathtt{q}\rangle \circ \psi_A && \text{(definition of } \varphi_{\Gamma.A}\text{)}\\
&= \mathtt{p} \circ \langle \varphi_\Gamma \circ \mathtt{p}, \mathtt{q}\rangle \circ \psi_A && \text{(property of } \rho_{\Gamma,A}\text{)}\\
&= \varphi_\Gamma \circ \mathtt{p} \circ \psi_A = \varphi_\Gamma \circ \mathtt{p} && \text{(because } \psi_A \text{ is a map in } \mathbf{T}\overline{[\![\Gamma]\!]}\text{)}
\end{aligned}$$

Extension. Assume we have $\Gamma \vdash \gamma : \Delta$ and $\Gamma \vdash t : A[\gamma]$ so that $\langle \gamma, t\rangle_A$ is a morphism from Γ to $\Delta.A$. Using Proposition A.2, we get that $F\langle \gamma, t\rangle_A \circ \varphi_\Gamma = \rho^{-1}_{\Delta,A} \circ \langle F\gamma, \{\theta^{-1}_{A,\gamma}\}(Ft)\rangle \circ \varphi_\Gamma$. After calculation, we get

$$
\begin{array}{rcl}
F\langle\gamma,t\rangle_A\circ\varphi_\Gamma & = & \rho^{-1}_{\Delta,A}\circ\langle F\gamma,\{\theta^{-1}_{A,\gamma}\}(Ft)\rangle\circ\varphi_\Gamma \\
& = & \rho^{-1}_{\Delta,A}\circ\langle F\gamma\circ\varphi_\Gamma,\{\theta^{-1}_{A,\gamma}\}(Ft)[\varphi_\Gamma]\rangle \\
& & \text{(Lemma A.4)} \\
& = & \rho^{-1}_{\Delta,A}\circ\langle F\gamma\circ\varphi_\Gamma,\{\mathbf{T}(\varphi_\Gamma)(\theta^{-1}_{A,\gamma})\}(Ft[\varphi_\Gamma])\rangle \\
& & \text{(I.H. on } \gamma \text{ and } t) \\
& = & \rho^{-1}_{\Delta,A}\circ\langle\varphi_\Delta\circ\overline{[\![\gamma]\!]},\{\mathbf{T}(\varphi_\Gamma)(\theta^{-1}_{A,\gamma})\}(\{\psi_{A[\gamma]}\}(\overline{[\![t]\!]}))\rangle \\
& & \text{(definition of } \psi_{A[\gamma]}) \\
& = & \rho^{-1}_{\Delta,A}\circ\langle\varphi_\Delta\circ\overline{[\![\gamma]\!]},\{\mathbf{T}(\overline{[\![\gamma]\!]})(\psi_A)\}(\overline{[\![t]\!]})\rangle \\
& & \text{(definition of } \mathbf{T}) \\
& = & \rho^{-1}_{\Delta,A}\circ\langle\varphi_\Delta\circ\overline{[\![\gamma]\!]},\mathtt{q}[\psi_A\circ\langle\overline{[\![\gamma]\!]},\overline{[\![t]\!]}\rangle]\rangle \\
& = & \rho^{-1}_{\Delta,A}\circ\langle\varphi_\Delta\circ\mathtt{p},\mathtt{q}\rangle\circ\psi_A\circ\overline{[\![\langle\gamma,t\rangle_A]\!]} \\
& = & \rho^{-1}_{\Delta,A}\circ\varphi^{+}_{\Delta}\circ\psi_A\circ\overline{[\![\langle\gamma,t\rangle_A]\!]} \\
& = & \varphi_{\Delta.A}\circ\overline{[\![\langle\gamma,t\rangle_A]\!]}
\end{array}
$$

Term substitution. Assume we have $\Gamma\vdash\gamma:\Delta$ and $\Delta\vdash t:A$. Unfolding the definition of $\psi_{A[\gamma]}$, we get:

$$
\begin{array}{rcl}
\{\psi_{A[\gamma]}\}(\overline{[\![t[\gamma]]\!]}) & = & \{\mathbf{T}(\varphi_\Gamma)(\theta_{A,\gamma})\}(\{\mathbf{T}(\overline{[\![\gamma]\!]})(\psi_A)\}(\overline{[\![t[\gamma]]\!]})) \\
& = & \{\mathbf{T}(\varphi_\Gamma)(\theta_{A,\gamma})\}(\{\mathbf{T}(\overline{[\![\gamma]\!]})(\psi_A)\}(\overline{[\![t]\!]}[\overline{[\![\gamma]\!]}])) \\
& & \text{(Lemma A.4)} \\
& = & \{\mathbf{T}(\varphi_\Gamma)(\theta_{A,\gamma})\}(\{\psi_A\}(\overline{[\![t]\!]})[\overline{[\![\gamma]\!]}]) \\
& & \text{(I.H. on } t) \\
& = & \{\mathbf{T}(\varphi_\Gamma)(\theta_{A,\gamma})\}((Ft)[\varphi_\Delta\circ\overline{[\![\gamma]\!]}]) \\
& & \text{(I.H. on } \gamma) \\
& = & \{\mathbf{T}(\varphi_\Gamma)(\theta_{A,\gamma})\}((Ft)[F\gamma\circ\varphi_\Gamma]) \\
& & \text{(Lemma A.4)} \\
& = & \{\theta_{A,\gamma}\}(Ft[F\gamma])[\varphi_\Gamma] \\
& & \text{(Definition of pseudo cwf-morphisms)} \\
& = & F(t[\gamma])[\varphi_\Gamma]
\end{array}
$$

Variable. Assume we have $\Gamma.A\vdash\mathtt{q}_A:A[\mathtt{p}]$. Unfolding the definition of $\psi_{A[\mathtt{p}]}$ yields, after some simplifications:

$$
\begin{aligned}
\{\psi_{A[\mathtt{p}]}\}(\mathtt{q}_A) &= \mathtt{q}[\theta_{A,\mathtt{p}}\circ\langle\varphi_\Gamma,\mathtt{q}[\psi_A\circ\langle\mathtt{p},\mathtt{q}\rangle]\rangle] \\
&\qquad (\langle\mathtt{p},\mathtt{q}\rangle=\mathtt{id}) \\
&= \mathtt{q}[\theta_{A,\mathtt{p}}\circ\langle\varphi_{\Gamma.A},\mathtt{q}[\psi_A]\rangle]
\end{aligned}
$$

We need to prove that this is equal to:

$$
\begin{aligned}
F\mathtt{q}[\varphi_{\Gamma.A}] &= \{\theta_{A,\mathtt{p}}\}\left(\mathtt{q}\left[\rho_{\Gamma,A}\right]\right)\left[\varphi_{\Gamma.A}\right] \\
&\qquad \text{(definition of pseudo cwf-morphism)} \\
&= \mathtt{q}[\theta_{A,\mathtt{p}}\circ\langle\mathtt{id},\mathtt{q}[\rho_{\Gamma,A}]\rangle\circ\varphi_{\Gamma.A}] \\
&= \mathtt{q}[\theta_{A,\mathtt{p}}\circ\langle\varphi_{\Gamma.A},\mathtt{q}[\rho_{\Gamma,A}\circ\varphi_{\Gamma.A}]\rangle] \\
&\qquad \text{(definition of } \varphi_{\Gamma.A}) \\
&= \mathtt{q}[\theta_{A,\mathtt{p}}\circ\langle\varphi_{\Gamma.A},\mathtt{q}[\langle\varphi_\Gamma\circ\mathtt{p},\mathtt{q}\rangle\circ\psi_A\rangle]
\end{aligned}
$$

$$= \mathtt{q}[\theta_{A,\mathtt{p}} \circ \langle \varphi_{\Gamma.A}, \mathtt{q}[\psi_A]\rangle]$$

Thus the equality holds.

Functoriality of substitution. Assume we have $\Gamma \vdash \gamma : \Delta$ and $\Delta \vdash \delta : \Theta$. We want to show the equality $\psi_{A[\delta][\gamma]} = \psi_{A[\delta\circ\gamma]}$ and $\psi_{A[\mathtt{id}]} = \psi_A$ for $\Theta \vdash A$. The second equation is easy: by functoriality of $\mathbf{T}$, $\mathbf{T}(\overline{[\![\mathtt{id}]\!]})(\psi_A) = \psi_A$ and properties of F, $\theta_{A,\mathtt{id}} = \mathtt{id}$.

For the other equation, unfolding the definitions gives:

$$\begin{aligned}
\psi_{A[\delta][\gamma]} &= \mathbf{T}(\overline{[\![\gamma]\!]})(\mathbf{T}(\overline{[\![\delta]\!]})(\psi_A) \circ \mathbf{T}(\varphi_\Delta)(\theta_{A,\delta})) \circ \mathbf{T}(\varphi_\Gamma)(\theta_{A,\gamma}) \\
&= \mathbf{T}(\overline{[\![\gamma]\!]})\left(\mathbf{T}(\overline{[\![\delta]\!]})(\psi_A)\right) \circ \mathbf{T}(\overline{[\![\gamma]\!]})\left(\mathbf{T}(\varphi_\Delta)(\theta_{A,\delta})\right) \circ \mathbf{T}(\varphi_\Gamma)(\theta_{A,\gamma}) \\
&\qquad \text{(functoriality of } \mathbf{T} \text{ and induction hypothesis on } \gamma) \\
&= \mathbf{T}(\overline{[\![\delta]\!]} \circ \overline{[\![\gamma]\!]})(\psi_A) \circ \mathbf{T}(\varphi_\Gamma)(\mathbf{T}(F\gamma)(\theta_{A,\delta})) \circ \mathbf{T}(\varphi_\Gamma)(\theta_{A,\gamma}) \\
&\qquad \text{(functoriality of } \mathbf{T}(\varphi_\Gamma)) \\
&= \mathbf{T}(\overline{[\![\delta]\!]} \circ \overline{[\![\gamma]\!]})(\psi_A) \circ \mathbf{T}(\varphi_\Gamma)(\mathbf{T}(F\gamma)(\theta_{A,\delta}) \circ (\theta_{A,\gamma})) \\
&\qquad \text{(coherence for } \theta) \\
&= \mathbf{T}(\overline{[\![\delta]\!]} \circ \overline{[\![\gamma]\!]})(\psi_A) \circ \mathbf{T}(\varphi_\Gamma)(\theta_{A,\delta\circ\gamma}) \\
&= \psi_{A[\delta\circ\gamma]}
\end{aligned}$$

Other cases arising from conversion rules and per-rules are straightfoward.

3.4.2. *Uniqueness of* (φ, ψ). Let $(\varphi', \psi') : \overline{[\![\cdot]\!]} \to F$ be another pseudo cwf-transformation in $\mathbf{CwF}^o$. We prove the following by induction:

- If $\Gamma = \Gamma' \vdash$, then $\varphi_\Gamma = \varphi'_\Gamma$
- If $\Gamma \vdash A = A'$, then $\psi_A = \psi'_A$

Empty context. There is a unique morphism between the terminal objects $\overline{[\![1]\!]}$ and $F1$, so $\varphi_1 = \varphi'_1$.

Context extension. Assume by induction $\varphi_\Gamma = \varphi'_\Gamma$ and $\psi_A = \psi'_A$. By the coherence law of pseudo cwf-transformations, we have $\varphi'_{\Gamma.A} = \rho^{-1}_{\Gamma,A} \circ {\varphi'_\Gamma}^+ \circ \psi'_A$ from which the equality $\varphi_{\Gamma.A} = \varphi'_{\Gamma.A}$ follows.

Type substitution. Assume we have $\Delta \vdash A$ and $\Gamma \vdash \gamma : \Delta$, and consider $\psi_{A[\gamma]}$ and $\psi'_{A[\gamma]}$. By Lemma B.2, we have:

$$\psi'_{A[\gamma]} = \mathbf{T}(\varphi'_\Gamma)(\theta_{A,\gamma}) \circ \mathbf{T}(\overline{[\![\gamma]\!]})(\psi'_A)$$

and likewise for $\psi_{A[\gamma]}$. Since we know by induction hypothesis that $\varphi_\Gamma = \varphi'_\Gamma$ and $\psi_A = \psi'_A$, it follows that $\psi_{A[\gamma]} = \psi'_{A[\gamma]}$.

Base type. The definition of 2-cells in $\mathbf{CwF}^o$ entails $\psi'_o = \alpha_F^{-1} : \overline{[\![o]\!]} \to F([\![o]\!])$.

This concludes the proof that $\mathcal{T}$ is a bifree cwf on one object. In the next section, we will prove that this result still holds in the presence of type constructors.

4. A free lccc

This section will basically follow the plan of Section 3. We will first recall what it means for categories with families to support the extra structure for I, N_1, Π and Σ-types. Then we will extend our cwf-calculus with these type constructors. Finally, we will also extend our proofs of freeness and bifreeness. In particular, bifreeness will be transported by our biequivalence [6]. It follows that the underlying category of contexts of the syntactic cwf with extra structure is a bifree lccc.

4.1. Cwfs which support $\mathrm{I}, \mathrm{N}_1, \Sigma, \Pi$.

We recall here from [8, 6] what it means for a cwf to support type constructors and prove a few properties of the corresponding combinators.

Definition 4.1. A cwf $\mathcal{C}$ supports extensional identity types iff it is equipped with the following extra structure:

- *Formation.* If $A \in \mathrm{Ty}_{\mathcal{C}}(\Gamma)$ and $a, a' \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, A)$, then there is $\mathrm{I}(A, a, a') \in \mathrm{Ty}_{\mathcal{C}}(\Gamma)$.
- *Introduction.* If $a \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, A)$, then there is $\mathrm{r}(a) \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, \mathrm{I}(A, a, a))$.
- *Elimination.* If $c \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, \mathrm{I}(A, a, a'))$, then $a = a'$ and $c = \mathrm{r}(a)$.

such that the following laws with respect to substitution are satisfied, for any $\gamma : \Delta \to \Gamma$:

$$\begin{array}{rcl} \mathrm{I}(A, a, a')[\gamma] & = & \mathrm{I}(A[\gamma], a[\gamma], a'[\gamma]) \\ \mathrm{r}(a)[\gamma] & = & \mathrm{r}(a[\gamma]) \end{array}$$

Definition 4.2. A cwf $\mathcal{C}$ supports Σ-types iff it is equipped with the following extra structure:

- *Formation.* If $A \in \mathrm{Ty}_{\mathcal{C}}(\Gamma)$ and $B \in \mathrm{Ty}_{\mathcal{C}}(\Gamma.A)$, there is $\Sigma(A, B) \in \mathrm{Ty}_{\mathcal{C}}(\Gamma)$,
- *Introduction.* If $a \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, A)$ and $b \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, B[\langle \mathtt{id}, a \rangle])$, there is $\mathrm{pair}(a, b) \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, \Sigma(A, B))$,
- *Elimination.* If $c \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, \Sigma(A, B))$, there are $\mathrm{fst}(c) \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, A)$ and $\mathrm{snd}(c) \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, B[\langle \mathtt{id}, \mathrm{fst}(c) \rangle])$ such that

$$\begin{array}{rcl} \mathrm{fst}(\mathrm{pair}(a, b)) & = & a \\ \mathrm{snd}(\mathrm{pair}(a, b)) & = & b \\ \mathrm{pair}(\mathrm{fst}(c), \mathrm{snd}(c)) & = & c \end{array}$$

and we also have stability under substitution. If $\gamma : \Delta \to \Gamma$ then

$$\begin{array}{rcl} \Sigma(A, B)[\gamma] & = & \Sigma(A[\gamma], B[\langle \gamma \circ \mathtt{p}, \mathtt{q} \rangle]) \\ \mathrm{pair}(a, b)[\gamma] & = & \mathrm{pair}(a[\gamma], b[\gamma]) \\ \mathrm{fst}(c)[\gamma] & = & \mathrm{fst}(c[\gamma]) \\ \mathrm{snd}(c)[\gamma] & = & \mathrm{snd}(c[\gamma]) \end{array}$$

Before going on to the definition of cwfs supporting Π-types, it is useful to recall a few lemmas about Σ-types on cwfs. First we recall from [6]:

Lemma 4.3. *For any $A \in \mathrm{Ty}_{\mathcal{C}}(\Gamma)$ and $B \in \mathrm{Ty}_{\mathcal{C}}(\Gamma.A)$, there is an isomorphism:*

$$\chi_{A,B} : \Gamma.A.B \to \Gamma.\Sigma(A, B)$$

such that $\mathtt{p} \circ \chi_{A,B} = \mathtt{p} \circ \mathtt{p}$.

Proof. The isomorphism is defined by the following inverse substitutions:

$$\begin{array}{rcl}\langle \mathtt{p} \circ \mathtt{p}, \mathrm{pair}(\mathtt{q}[\mathtt{p}], \mathtt{q})\rangle & : & \Gamma.A.B \to \Gamma.\Sigma(A,B) \\ \langle\langle \mathtt{p}, \mathrm{fst}(\mathtt{q})\rangle, \mathrm{snd}(\mathtt{q})\rangle & : & \Gamma.\Sigma(A,B) \to \Gamma.A.B\end{array}$$

An easy calculation shows that they are mutual inverses. ☐

The type constructor Σ can also be extended to act on morphisms in the adequate fibres, in a functorial way. This is formalized in the following lemma.

Lemma 4.4. *Let $A, A' \in \mathrm{Ty}_{\mathcal{C}}(\Gamma), B \in \mathrm{Ty}_{\mathcal{C}}(\Gamma.A)$, and $B' \in \mathrm{Ty}_{\mathcal{C}}(\Gamma.A')$. Moreover, consider morphisms $f_A : A \to A'$ in $\mathbf{T}(\Gamma)$ (*i.e.* $f_A : \Gamma.A \to \Gamma.A'$ such that $\mathtt{p} \circ f_A = \mathtt{p}$), and $f_B : B \to B'[f_A]$ in $\mathbf{T}(\Gamma.A)$ (*i.e.* $f_B : \Gamma.A.B \to \Gamma.A.B'[f_A]$ such that $\mathtt{p} \circ f_B = \mathtt{p}$).*

Then, defining:

$$\Sigma(f_A, f_B) : \langle \mathtt{p}, \mathrm{pair}(\mathtt{q}[f_A \circ \langle \mathtt{p}, \mathrm{fst}(\mathtt{q})\rangle], \mathtt{q}[f_B \circ \langle\langle \mathtt{p}, \mathrm{fst}(\mathtt{q})\rangle, \mathrm{snd}(\mathtt{q})\rangle])\rangle$$

we have $\Sigma(f_A, f_B) : \Sigma(A,B) \to \Sigma(A', B')$ in $\mathbf{T}(\Gamma)$. Moreover, it is functorial in the following sense. For f_A, f_B as above and $g_A : A' \to A'', g_B : B' \to B''[g_A]$, then:

$$\Sigma(g_A, g_B) \circ \Sigma(f_A, f_B) = \Sigma(g_A \circ f_A, \mathbf{T}(f_A)(g_B) \circ f_B)$$

Proof. Direct verification. ☐

This strengthens Lemma B.1 of [6], which states that the type constructor Σ preserves isomorphisms of types. We will also use the following lemma, which states compatibility of the functorial action of Σ with that of substitution.

Lemma 4.5. *Let f_A, f_B be as in the lemma above. Then, for any $\gamma : \Delta \to \Gamma$, we have:*

$$\mathbf{T}(\gamma)(\Sigma(f_A, f_B)) = \Sigma(\mathbf{T}(\gamma)(f_A), \mathbf{T}(\gamma \uparrow A)(f_B))$$

Both are morphisms from $\Sigma(A[\gamma], B[\gamma \uparrow A])$ to $\Sigma(A'[\gamma], B'[\gamma \uparrow A])$ in $\mathbf{T}(\Delta)$.

Proof. Direct calculation. ☐

Now, we go on to define what it means for a cwf to support Π-types.

Definition 4.6. A cwf $\mathcal{C}$ supports Π-types iff it is equipped with the following extra structure:

- *Formation.* If $A \in \mathrm{Ty}_{\mathcal{C}}(\Gamma)$ and $B \in \mathrm{Ty}_{\mathcal{C}}(\Gamma.A)$, there is $\Pi(A,B) \in \mathrm{Ty}_{\mathcal{C}}(\Gamma)$.
- *Introduction.* If $b \in \mathrm{Tm}_{\mathcal{C}}(\Gamma.A, B)$, there is $\lambda(b) \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, \Pi(A,B))$.
- *Elimination.* If $c \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, \Pi(A,B))$ and $a \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, A)$ then there is a term $\mathrm{app}(c,a) \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, B[\langle \mathtt{id}, a\rangle])$ such that

$$\begin{array}{rcl}\mathrm{app}(\lambda(b), a) & = & b[\langle \mathtt{id}, a\rangle] \\ \lambda(\mathrm{app}(c[\mathtt{p}], \mathtt{q})) & = & c\end{array}$$

and we also have stability under substitution. If $\gamma : \Delta \to \Gamma$ then

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

Just like for Σ-types, Π-types can be given a functorial action on the fibres.

Lemma 4.7. *Let $A, A' \in \mathrm{Ty}_\mathcal{C}(\Gamma), B \in \mathrm{Ty}_\mathcal{C}(\Gamma.A)$, and $B' \in \mathrm{Ty}_\mathcal{C}(\Gamma.A')$. Moreover, consider morphisms $f_A : A' \to A$ in $\mathbf{T}(\Gamma)$ and $f_B : B[f_A] \to B'$ in $\mathbf{T}(\Gamma.A')$.*

Then, defining:

$$\Pi(f_A, f_B) = \langle \mathtt{p}, \lambda(\mathtt{q}[f_B \circ \langle\langle \mathtt{p} \circ \mathtt{p}, \mathtt{q}\rangle, \mathrm{app}(\mathtt{q}[\mathtt{p}], \mathtt{q}[f_A \circ \langle \mathtt{p} \circ \mathtt{p}, \mathtt{q}\rangle])\rangle])\rangle$$

we have $\Pi(f_A, f_B) : \Pi(A, B) \to \Pi(A', B')$ in $\mathbf{T}(\Gamma)$. Morever, the action of Π is functorial, in the sense that for f_A, f_B as above and $g_A : A'' \to A'$, $g_B : B'[g_A] \to B''$, we have:

$$\Pi(g_A, g_B) \circ \Pi(f_A, f_B) = \Pi(f_A \circ g_A, g_B \circ \mathbf{T}(g_A)(f_B))$$

Proof. Tedious calculations on cwf-combinators. □

Just as for Σ-types, the functorial action of Π commutes with the functorial action of substitution.

Lemma 4.8. *Let f_A, f_B as in the lemma above, and $\gamma : \Delta \to \Gamma$. Then, we have:*

$$\mathbf{T}(\gamma)(\Pi(f_A, f_B)) = \Pi(\mathbf{T}(\gamma)(f_A), \mathbf{T}(\gamma \uparrow A')(f_B))$$

where both terms are morphisms from $\Pi(A[\gamma], B[\gamma \uparrow A])$ to $\Pi(A'[\gamma], B'[\gamma \uparrow A'])$ in $\mathbf{T}(\Delta)$.

Proof. Direct calculation. □

Definition 4.9. A cwf $\mathcal{C}$ supports N_1 iff it is equipped with the following extra structure:

- *Formation.* There is $\mathrm{N}_1 \in \mathrm{Ty}_\mathcal{C}(1)$.
- *Introduction.* There is $0_1 \in \mathrm{Tm}_\mathcal{C}(1, \mathrm{N}_1)$.
- *Elimination.* For any $c \in \mathrm{Tm}_\mathcal{C}(1, \mathrm{N}_1)$, $c = 0_1$.

We will be interested in cwfs that support N_1. However, both the cwfs that come from syntax (including $\mathcal{T}$) and the cwfs in correspondence with lcccs through our biequivalence satisfy a stronger property: they are *democratic*.

Definition 4.10 (Democratic cwfs)**.** A cwf $\mathcal{C}$ is democratic when for each context Γ there is a type $\overline{\Gamma} \in \mathrm{Ty}_\mathcal{C}(1)$ with an isomorphism $\gamma_\Gamma : \Gamma \cong 1.\overline{\Gamma}$.

Lemma 4.11. *Let $\mathcal{C}$ be a democratic cwf. Then, it supports N_1.*

Proof. We simply define $\mathrm{N}_1 = \overline{1}$. This type automatically has an inhabitant $0_1 = \mathtt{q}[\gamma_1] \in \mathrm{Tm}_\mathcal{C}(1, \overline{1})$; its uniqueness is an easy consequence of the fact that 1 is terminal. □

As a consequence we do not need to mention support for N_1 for democratic cwfs . We will show in Lemma 4.17 that in the presence of Σ-types and N_1, the syntactically generated cwf is democratic.

For each of these type constructors, it is easy to define what it means for strict cwf-morphisms to preserve them. We simply ask that everything – both type constructors and the associated combinators – is preserved on the nose. For instance, we ask that

$$F(\Gamma.\Sigma(A, B)) = F\Gamma.\Sigma(FA, FB)$$

and $F(\mathrm{pair}(a, b)) = \mathrm{pair}(Fa, Fb)$, *etc.*.

However, as emphasized before, for the correspondence with lcccs one needs notions of cwf-morphisms that only preserve structure *up to isomorphism.*

4.2. **Pseudo cwf-morphisms preserving structure up to isomorphism.** We now recall the definitions of preservation of structure up to isomorphism for pseudo cwf-morphisms from [6]. Note first that for cwfs which support Σ-types, pseudo cwf-morphisms automatically preserve Σ-types.

Proposition 4.12. *A pseudo cwf-morphism F from $\mathcal{C}$ to $\mathcal{C}'$, where both cwfs support Σ-types, also preserves them 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 are preserved up to isomorphism. For any term $c \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, \Sigma(A,B))$, or terms $a \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, A)$ and $b \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, B[\langle \mathtt{id}, a\rangle])$.

$$\begin{array}{rcl} F(\mathrm{fst}(c)) & = & \mathrm{fst}(\{s_{A,B}\}(Fc)) \\ F(\mathrm{snd}(c)) & = & \{\theta_{B,\langle \mathtt{id}, \mathrm{fst}(c)\rangle}\}(\mathrm{snd}(\{s_{A,B}\}(Fc))) \\ F(\mathrm{pair}(a,b)) & = & \{s^{-1}_{A,B}\}(\mathrm{pair}(Fa, \{\theta^{-1}_{B,\langle \mathtt{id}, a\rangle}\}(Fb))) \end{array}$$

Proof. Proposition 3.5 in [6]. $\square$

On the other hand, neither the preservation of the other type constructors nor the preservation of democracy is automatic. We recall the following definitions from [6].

Definition 4.13. Let $\mathcal{C}, \mathcal{C}'$ be cwfs supporting identity types and $F : \mathcal{C} \to \mathcal{C}'$ be a pseudo cwf-morphism. Then, F preserves identity types provided there is an isomorphism:

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

in $\mathbf{T}'(\Gamma)$.

Likewise, we have for democracy:

Definition 4.14. Let $\mathcal{C}, \mathcal{C}'$ be democratic cwfs, and $F : \mathcal{C} \to \mathcal{C}'$ be pseudo cwf-morphisms. Then, F preserves democracy provided there is an isomorphism

$$d_\Gamma : F(\overline{\Gamma}) \cong \overline{F\Gamma}[\langle\rangle]$$

in $\mathbf{T}'(1)$ such that the following diagram commutes:

$$\begin{tikzcd} F\Gamma \arrow[rr, "F\gamma_\Gamma"] \arrow[d, "\gamma_{F\Gamma}"'] & & F(1.\overline{\Gamma}) \arrow[d, "\rho_{1,\overline{\Gamma}}"] \\ 1.\overline{F\Gamma} & F1.\overline{F\Gamma}[\langle\rangle] \arrow[l, "\langle\langle\rangle, \mathtt{q}\rangle"'] & F1.F(\overline{\Gamma}) \arrow[l, "d_\Gamma"'] \end{tikzcd}$$

We saw before that democratic cwfs automatically support N_1 – likewise, pseudo cwf-morphisms that preserve democracy automatically preserve N_1 in the obvious sense.

Finally, we define preservation of Π-types.

Definition 4.15. Let $\mathcal{C}$ and $\mathcal{C}'$ be cwfs supporting Π-types, and F a pseudo cwf-morphism. Then F preserves Π-types iff for each types $A \in \mathrm{Ty}_{\mathcal{C}}(\Gamma)$ and $B \in \mathrm{Ty}_{\mathcal{C}}(\Gamma.A)$ there is an isomorphism in $\mathbf{T}'(\Gamma)$:

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

such that for any substitution $\gamma : \Delta \to \Gamma$, for any terms $c \in \mathrm{Tm}_{\mathcal{C}}(\Delta, \Pi(A,B)[\gamma])$ and $a \in \mathrm{Tm}_{\mathcal{C}}(\Gamma, A[\gamma])$, we have:

$$F(\mathrm{app}(c,a)) = \{\theta_{B,\langle\gamma,a\rangle}\}(\mathrm{app}(\{\mathbf{T}'(F\gamma)(i_{A,B}) \circ \theta^{-1}_{\Pi(A,B),\gamma}\}(Fc), \{\theta^{-1}_{A,\gamma}\}(Fa)))$$

The definition of preservation of Π-types for pseudo cwf-morphisms only require them to preserve application. In fact, as remarked in [6], it is sufficient to ensure that abstraction is preserved as well.

Lemma 4.16. *If $F : \mathcal{C} \to \mathcal{C}'$ is a pseudo cwf-morphism preserving* Π*-types, then it preserves the abstraction combinator, in the sense that for any $b \in \mathrm{Tm}_{\mathcal{C}}(\Gamma.A, B)$,*

$$F(\lambda(b)) = \{i^{-1}_{A,B}\}(\lambda((Fb)[\rho^{-1}_{\Gamma,A}]))$$

Proof. Immediate consequence of Lemma A.6. □

We now go on to extend our syntactic cwf $\mathcal{T}$ with all the extra structure mentioned above, before proving that it is bifree.

4.3. **The syntactic cwf with extensional** I**,** N_1**,** Σ**, and** Π**.** We extend the grammar and the set of inference rules with rules for $\mathrm{I}, \mathrm{N}_1, \Sigma$, and Π-types:

$$\begin{array}{rcl} A & ::= & \cdots \mid \mathrm{I}(A, a, a) \mid \mathrm{N}_1 \mid \Sigma(A, A) \mid \Pi(A, A) \\ a & ::= & \cdots \mid \mathrm{r}(a) \mid 0_1 \mid \mathrm{fst}(A, a) \mid \mathrm{snd}(A, A, a) \mid \mathrm{pair}(A, A, a, a) \mid \mathrm{app}(A, A, a, a) \mid \lambda(A, a) \end{array}$$

For each type we define its context:

$$\begin{aligned} \text{ctx-of}(\mathrm{I}(A, a, a')) &= \text{ctx-of}(A) \\ \text{ctx-of}(\mathrm{N}_1) &= 1 \\ \text{ctx-of}(\Sigma(A, B)) &= \text{ctx-of}(A) \\ \text{ctx-of}(\Pi(A, B)) &= \text{ctx-of}(A) \end{aligned}$$

For each term we define its type:

$$\begin{aligned} \text{type-of}(0_1) &= \mathrm{N}_1 \\ \text{type-of}(\mathrm{fst}(A, c)) &= A \\ \text{type-of}(\mathrm{snd}(A, B, c) &= B\,[\langle \mathtt{id}_{\text{ctx-of}(A)}, \mathrm{fst}(A, c)\rangle_A] \\ \text{type-of}(\mathrm{pair}(A, B, a, b)) &= \Sigma(A, B) \\ \text{type-of}(\mathrm{r}(a)) &= \mathrm{I}(\text{type-of}(a), a, a) \\ \text{type-of}(\lambda(A, c)) &= \Pi(A, \text{type-of}(c)) \\ \text{type-of}(\mathrm{app}(A, B, c, a)) &= B\,[\langle \mathtt{id}_{\text{ctx-of}(A)}, a\rangle_A] \end{aligned}$$

There is still some redundancy in the type annotations: one could omit the annotation A in $\mathrm{I}(A, a, a')$ and only have $\mathrm{I}(a, a')$. Its context can then be recovered as ctx-of(type-of(a)) instead of ctx-of(A). However, this optimization makes the termination of the mutually recursive functions ctx-of and type-of less immediately evident as it is no longer a simple structural induction[1]. We therefore opted for the present slightly redundant version.

[1]We are grateful to one of the reviewers for this observation.

Rules for Σ-types

$$\frac{\Gamma \vdash A = A' \qquad \Gamma.A \vdash B = B'}{\Gamma \vdash \Sigma(A,B) = \Sigma(A',B')} \qquad \frac{\Gamma \vdash A = A' \qquad \Gamma \vdash c = c' : \Sigma(A,B)}{\Gamma \vdash \mathrm{fst}(A,c) = \mathrm{fst}(A',c') : A}$$

$$\frac{\Gamma \vdash A = A' \qquad \Gamma.A \vdash B = B' \qquad \Gamma \vdash c = c' : \Sigma(A,B)}{\Gamma \vdash \mathrm{snd}(A,B,c) = \mathrm{snd}(A',B',c') : B\,[\langle \mathtt{id}_\Gamma, \mathrm{fst}(A,c)\rangle_A]}$$

$$\frac{\Gamma \vdash A = A' \qquad \Gamma.A \vdash B = B' \qquad \Gamma \vdash a = a' : A' \qquad \Gamma \vdash b = b' : B\,[\langle \mathtt{id}_\Gamma, \mathrm{fst}(A,c)\rangle_A]}{\Gamma \vdash \mathrm{pair}(A,B,a,b) = \mathrm{pair}(A',B',a',b') : \Sigma(A,B)}$$

$$\frac{\Gamma \vdash A \qquad \Gamma.A \vdash B \qquad \Gamma \vdash a : A \qquad \Gamma \vdash b : B\,[\langle \mathtt{id}_\Gamma, \mathrm{fst}(A,c)\rangle_A]}{\Gamma : \mathrm{fst}(A,\mathrm{pair}(A,B,a,b)) = a : A}$$

$$\frac{\Gamma \vdash A \qquad \Gamma.A \vdash B \qquad \Gamma \vdash a : A \qquad \Gamma \vdash b : B\,[\langle \mathtt{id}_\Gamma, \mathrm{fst}(A,c)\rangle_A]}{\Gamma \vdash \mathrm{snd}(A,B,\mathrm{pair}(A,B,a,b)) = b : B\,[\langle \mathtt{id}_\Gamma, \mathrm{fst}(A,c)\rangle_A]}$$

$$\frac{\Gamma \vdash c : \Sigma(A,B)}{\Gamma \vdash c = \mathrm{pair}(A,B,\mathrm{fst}(A,c),\mathrm{snd}(A,B,c)) : \Sigma(A,B)}$$

$$\frac{\Gamma \vdash A \qquad \Gamma.A \vdash B \qquad \Delta \vdash \gamma : \Gamma}{\Delta \vdash \Sigma(A,B)[\gamma] = \Sigma(A[\gamma], B[\gamma^+])} \qquad \frac{\Gamma \vdash A \qquad \Gamma \vdash c : \Sigma(A,B) \qquad \Delta \vdash \gamma : \Gamma}{\Delta \vdash \mathrm{fst}(A,c)[\gamma] = \mathrm{fst}(A[\gamma], c[\gamma]) : A}$$

$$\frac{\Gamma \vdash A \qquad \Gamma.A \vdash B \qquad \Gamma \vdash c : \Sigma(A,B) \qquad \Delta \vdash \gamma : \Gamma}{\Delta \vdash \mathrm{snd}(A,B,c)[\gamma] = \mathrm{snd}(A[\gamma], B[\gamma^+], c[\gamma]) : B[\langle \gamma, \mathrm{fst}(A,c)[\gamma]\rangle_A]}$$

$$\frac{\Gamma \vdash A \qquad \Gamma.A \vdash B \qquad \Gamma \vdash a : A \qquad \Gamma \vdash b : B[\langle \mathtt{id}_\Gamma, \mathrm{fst}(A,c)\rangle_A] \qquad \Delta \vdash \gamma : \Gamma}{\Delta \vdash \mathrm{pair}(A,B,a,b)[\gamma] = \mathrm{pair}(A[\gamma], B[\gamma^+], a[\gamma], b[\gamma]) : \Sigma(A,B)[\gamma]}$$

Rules for I-types

$$\frac{\Gamma \vdash a = a' : A \qquad \Gamma \vdash b = b' : A}{\Gamma \vdash \mathrm{I}(A,a,b) = \mathrm{I}(A,a',b')} \qquad \frac{\Gamma \vdash a = a' : A}{\Gamma \vdash \mathrm{r}(a) = \mathrm{r}(a') : \mathrm{I}(A,a,a')} \qquad \frac{\Gamma \vdash c : \mathrm{I}(A,a,a')}{\Gamma \vdash a = a' : A}$$

$$\frac{\Gamma \vdash c : \mathrm{I}(A,a,a')}{\Gamma \vdash c = \mathrm{r}(a) : \mathrm{I}(A,a,a')} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash a' : A \qquad \Delta \vdash \gamma : \Gamma}{\Delta \vdash \mathrm{I}(A,a,a')[\gamma] = \mathrm{I}(A[\gamma], a[\gamma], a'[\gamma])}$$

Rules for N_1

$$\frac{}{1 \vdash \mathrm{N}_1 = \mathrm{N}_1} \qquad \frac{}{1 \vdash 0_1 = 0_1 : \mathrm{N}_1} \qquad \frac{1 \vdash a : \mathrm{N}_1}{1 \vdash a = 0_1 : \mathrm{N}_1}$$

Rules for Π-types

$$\frac{\Gamma \vdash A = A' \qquad \Gamma.A \vdash B = B'}{\Gamma \vdash \Pi(A, B) = \Pi(A', B')} \qquad \frac{\Gamma \vdash A = A' \qquad \Gamma.A \vdash b = b' : B}{\Gamma \vdash \lambda(A, b) = \lambda(A', b') : \Pi(A, B)}$$

$$\frac{\Gamma \vdash A = A' \qquad \Gamma.A \vdash B = B' \qquad \Gamma \vdash c = c' : \Pi(A, B) \qquad \Gamma \vdash a = a' : A}{\Gamma \vdash \mathrm{app}(A, B, c, a) = \mathrm{app}(A', B', c', a') : B[\langle \mathtt{id}_\Gamma, a\rangle_A]}$$

$$\frac{\Gamma.A \vdash b : B \qquad \Gamma \vdash a : A}{\Gamma \vdash \mathrm{app}(A, B, \lambda(A, b), a) = b[\langle \mathtt{id}_\Gamma, a\rangle_A] : B[\langle \mathtt{id}_\Gamma, a\rangle_A]}$$

$$\frac{\Gamma \vdash c : \Pi(A, B)}{\Gamma \vdash \lambda(A, \mathrm{app}(c[p], q)) = c : \Pi(A, B)} \qquad \frac{\Gamma \vdash A \qquad \Gamma.A \vdash B \qquad \Delta \vdash \gamma : \Gamma}{\Delta \vdash \Pi(A, B)[\gamma] = \Pi(A[\gamma], A[\gamma^+])}$$

$$\frac{\Gamma \vdash c : \Pi(A, B) \qquad \Delta \vdash \gamma : \Gamma}{\Delta \vdash \lambda(A, b)[\gamma] = \lambda(A[\gamma], b[\gamma^+]) : \Pi(A, B)[\gamma]}$$

$$\frac{\Gamma \vdash c : \Pi(A, B) \qquad \Gamma \vdash a : A \qquad \Delta \vdash \gamma : \Gamma}{\Delta \vdash \mathrm{app}(c, a)[\gamma] = \mathrm{app}(c[\gamma], a[\gamma]) : B[\langle \gamma, a[\gamma]\rangle_A]}$$

It is straightforward to extend the definition of the term model $\mathcal{T}$ with $\mathrm{I}, \mathrm{N}_1, \Sigma$, and Π-types to form a cwf $\mathcal{T}^{\mathrm{I},\mathrm{N}_1,\Sigma,\Pi}$ supporting these type constructors. Although there are no grammatical construct and no inference rules corresponding to democracy we can prove the following:

Lemma 4.17. *The cwf* $\mathcal{T}^{\mathrm{I},\mathrm{N}_1,\Sigma,\Pi}$ *is democratic.*

Proof. For any well-formed context $\Gamma \vdash$ we define a type $\overline{\Gamma}$ by induction on the inference rules. For $1 \vdash$, we have $\overline{1} = \mathrm{N}_1 \in \mathrm{Ty}(1)$. For $\Gamma.A \vdash$, we set $\overline{\Gamma.A} = \Sigma(\overline{\Gamma}, A[\gamma_\Gamma^{-1}])$. Constructing the required isomorphism is immediate by induction using Lemma 4.3. □

It is straightforward to extend the interpretation functor and prove its uniqueness (among *strict* cwf-morphisms). It is also easy to check that it preserves democracy.

Theorem 4.18. $\mathcal{T}^{\mathrm{I},\mathrm{N}_1,\Sigma,\Pi}$ *is the free democratic cwf supporting* I, Σ, Π *on one object.*

We do not detail the proof of this theorem: in essence, it is a simplified version of the proof of Theorem 4.19 where all key isomorphisms are replaced by identities. Instead, we go on to prove that just as $\mathcal{T}$, besides being free in the category of strict cwf-morphisms (preserving structure), $\mathcal{T}^{\mathrm{I},\mathrm{N}_1,\Sigma,\Pi}$ is also bifree in the 2-category of *pseudo* cwf-morphisms (preserving structure).

4.4. **Bifreeness of $\mathcal{T}^{\mathrm{I},\mathrm{N}_1,\Sigma,\Pi}$.** We now prove the key result:

Theorem 4.19. $\mathcal{T}^{\mathrm{I},\mathrm{N}_1,\Sigma,\Pi}$ *is the bifree democratic cwf supporting* I, Σ, Π *on one object.*

This means that $\mathcal{T}^{\mathrm{I},\mathrm{N}_1,\Sigma,\Pi}$ is bi-initial in the 2-category $\mathbf{CwF}_d^{\mathrm{I},\Sigma,\Pi,o}$ where objects are democratic cwfs which support I, Σ, Π, and a base type o, and where morphisms preserve these type formers up to coherent isomorphisms.

4.4.1. *Existence of φ and ψ.* We resume our inductive proof from Section 3.4.1, treating the additional inference rules for $\mathrm{I}, \mathrm{N}_1, \Sigma$ and Π. We will first treat the type formation rules, then the type substitution rules. The rules for conversion and substitution on terms are straightforward, and not detailed.

Type formation rules. We start with the type formation rules for $\mathrm{N}_1, \mathrm{I}, \Sigma$ and Π.

Unit type: Since F preserves democracy and the terminal object it follows that:

$$1.\overline{1} \cong 1 \cong F1 \cong 1.\overline{F1} \cong 1.F(\mathrm{N}_1)[\varphi_1]$$

Write $\psi_{\overline{1}}$ for this type isomorphism.

Identity type: Assume $\Gamma \vdash a, b, a', b' : A$. By induction hypothesis, we have $\psi_A : \overline{[\![A]\!]} \cong_{\overline{[\![\Gamma]\!]}} FA[\varphi_\Gamma]$. We know I-types preserve isomorphisms in the indexed category (Lemma A.7) yielding (over $\overline{[\![\Gamma]\!]}$):

$$\begin{aligned}\psi_{\mathrm{I}(A,a,b)} : \overline{[\![\mathrm{I}(A,a,b)]\!]} &= \mathrm{I}(\overline{[\![A]\!]}, \overline{[\![a]\!]}, \overline{[\![b]\!]}) \\ &\cong \mathrm{I}(FA[\varphi_\Gamma], \{\psi_A\}(\overline{[\![a]\!]}), \{\psi_A\}(\overline{[\![b]\!]})) \\ &= \mathrm{I}(FA[\varphi_\Gamma], F(a)[\varphi_\Gamma], F(b)[\varphi_\Gamma])\end{aligned}$$

We also have $\psi_{\mathrm{I}(A,a',b')}$ defined likewise. But $\psi_{\mathrm{I}(A,a,b)}$ and $\psi_{\mathrm{I}(A,a',b')}$ are two parallel type isomorphisms whose domain is an identity type – so $\mathsf{q}[\psi_{\mathrm{I}(A,a,b)}]^{-1}, \mathsf{q}[\psi^{-1}_{\mathrm{I}(A,a',b')}] \in \mathrm{Tm}_{\mathcal{C}}(\overline{[\![\Gamma]\!]}.F(\mathrm{I}(A,a,b))[\varphi_\Gamma], \mathrm{I}(\overline{[\![A]\!]}[\mathsf{p}], \overline{[\![a]\!]}[\mathsf{p}], \overline{[\![b]\!]}[\mathsf{p}])$. It follows by the elimination rule for identity types in a cwf that these are both equal to the reflexivity term, and that $\psi_{\mathrm{I}(A,a,b)} = \psi_{\mathrm{I}(A,a',b')}$.

Σ-types: Assume that we have $\Gamma \vdash A = A'$ and $\Gamma.A \vdash B = B'$. By induction we have the isomorphisms $\psi_A = \psi_{A'} : \overline{[\![A]\!]} \cong_{\overline{[\![\Gamma]\!]}} FA[\varphi_\Gamma]$ and $\psi_B = \psi_{B'} : \overline{[\![B]\!]} \cong_{\overline{[\![\Gamma.A]\!]}} FB[\varphi_{\Gamma.A}]$. We let:

$$\begin{aligned}\psi_{\Sigma(A,B)} = \ \overline{[\![\Gamma.\Sigma(A,B)]\!]} &\xrightarrow{\Sigma(\psi_A,\psi_B)} \overline{[\![\Gamma]\!]}.\Sigma(FA[\varphi_\Gamma], FB[\rho^{-1}_{\Gamma,A} \circ {\varphi_\Gamma}^+]) \\ &\xrightarrow{\mathbf{T}(\varphi_\Gamma)(s^{-1}_{A,B})} \overline{[\![\Gamma]\!]}.F(\Sigma(A,B))[\varphi_\Gamma]\end{aligned}$$

It is clear by construction that $\psi_{\Sigma(A,B)} = \psi_{\Sigma(A',B')}$.

Π-types: Consider $\Gamma \vdash A = A'$ and $\Gamma.A \vdash B = B'$. Define $\psi_{\Pi(A,B)}$ as follows:

$$\begin{aligned}\overline{[\![\Gamma.\Pi(A,B)]\!]} &\xrightarrow{\Pi(\psi_A^{-1}, \mathbf{T}(\psi_A^{-1})(\psi_B))} \overline{[\![\Gamma]\!]}.\Pi(FA[\varphi_\Gamma], FB[\rho^{-1}_{\Gamma,A} \circ \varphi^+_\Gamma]) \\ &\xrightarrow{\mathbf{T}(\varphi_\Gamma)(i^{-1}_{A,B})} \overline{[\![\Gamma]\!]}.F(\Pi(A,B))[\varphi_\Gamma]\end{aligned}$$

It is clear by construction that $\psi_{\Pi(A,B)} = \psi_{\Pi(A',B')}$.

Type substitution rules. We now deal with the inference rules pertaining to the compatibility of the types I, Σ and Π with substitution. There is no inference rule for compatibility of N_1 with substitution.

In order to deal with compatibility under substitution, it is convenient to start with a few lemmas. In particular, we will use heavily the fact that $\theta_{A,\gamma}$ can be characterised with a universal property.

Lemma 4.20. *Let $\gamma : \Gamma \to \Delta$. The type morphism $\theta_{A,\gamma}$ is the only type morphism to make the following diagram commute:*

$$\begin{tikzcd}
F\Gamma.F(A[\gamma]) \arrow[r, "\rho^{-1}_{\Gamma,A[\gamma]}"] & F(\Gamma.A[\gamma]) \arrow[r, "F(\gamma^+)"] & F(\Delta.A) \arrow[d, "\rho_{\Delta,A}"'] \\
F\Gamma.FA[F\gamma] \arrow[u, "\theta_{A,\gamma}"'] \arrow[rr, "(F\gamma)^+"'] & & F\Delta.FA
\end{tikzcd}$$

Proof. The diagram commutes by virtue of Lemma A.3. Moreover, by definition of type substitution the following diagram is a pullback:

$$\begin{tikzcd}
F\Gamma.FA[F\gamma] \arrow[r, "(F\gamma)^+"] \arrow[d, "\mathtt{p}"'] \arrow[dr, phantom, "\lrcorner", very near start] & F\Delta.FA \arrow[d, "\mathtt{p}"] \\
F\Gamma \arrow[r, "F\gamma"'] & F\Delta
\end{tikzcd}$$

Because θ is an isomorphism and the diagram above commutes, the following is also a pullback:

$$\begin{tikzcd}
F\Gamma.F(A[\gamma]) \arrow[r, "\rho_{\Delta,A}\circ F(\gamma^+)\circ\rho^{-1}_{\Gamma,A[\gamma]}"] \arrow[d, "\mathtt{p}"'] \arrow[dr, phantom, "\lrcorner", very near start] & F\Delta.FA \arrow[d, "\mathtt{p}"] \\
F\Gamma \arrow[r, "F\gamma"'] & F\Delta
\end{tikzcd}$$

Thus it follows that there is a unique type morphism $F\Delta.FA[F\gamma] \to F\Delta.F(A[\gamma])$ that makes the diagram of the lemma commute by the universal property of pullbacks. $\square$

Using that, we deduce two lemmas on the compatibility of Σ-types and Π-types under substitution.

Lemma 4.21 (Compatibility of Σ-types with substitution)**.** *For any $A \in \mathrm{Ty}_{\mathcal{C}}(\Delta), B \in \mathrm{Ty}_{\mathcal{C}}(\Delta.A)$ and $\gamma : \Gamma \to \Delta$, the following diagram of type isomorphisms over $F\Gamma$ commutes.*

$$\begin{tikzcd}
F(\Sigma(A,B))[F\gamma] \arrow[r, "\theta_{\Sigma(A,B),\gamma}"] \arrow[d, "\mathbf{T}(F\gamma)(s_{A,B})"'] & F(\Sigma(A,B)[\gamma]) \arrow[d, "s_{A[\gamma],B[\gamma^+]}"] \\
\Sigma(FA, FB[\rho^{-1}_{\Delta,A}])[F\gamma] \arrow[r, "\Sigma(\theta_{A,\gamma},\mathbf{T}(\rho^{-1}_{\Gamma,A[\gamma]}\circ\theta_{A,\gamma})(\theta_{B,\gamma^+}))"'] & \Sigma(F(A[\gamma]), F(B[\gamma^+])[\rho^{-1}_{\Gamma,A[\gamma]}])
\end{tikzcd}$$

(It is well-typed because of the diagram of Lemma 4.20)

Proof. The diagram amounts to showing that $\theta_{\Sigma(A,B),\gamma} = s^{-1}_{A[\gamma],B[\gamma^+]} \circ \Sigma(\theta_{A,\gamma}, \mathbf{T}(\rho^{-1}_{\Gamma,A[\gamma]} \circ \theta_{A,\gamma})(\theta_{B,\gamma^+})) \circ T(F\gamma)(s_{A,B})$. Hence by Lemma 4.20 it is enough to show that the right hand side makes the corresponding diagram commute – which is an involved calculation. $\square$

Lemma 4.22 (Compatibility of Π-types with substitution)**.** *For any* $A \in \mathrm{Ty}_{\mathcal{C}}(\Delta), B \in \mathrm{Ty}_{\mathcal{C}}(\Delta.A)$ *and* $\gamma : \Gamma \to \Delta$*, the following diagram of type isomorphisms over* $F\Gamma$ *commutes.*

$$\begin{tikzcd}
F(\Pi(A,B))[F\gamma] \arrow[r, "\theta_{\Pi(A,B),\gamma}"] \arrow[d, "T(F\gamma)(i_{A,B})"'] & F(\Pi(A,B)[\gamma]) \arrow[d, "i_{A[\gamma],B[\gamma^+]}"] \\
\Pi(FA, FB[\rho^{-1}_{\Delta,A}])[F\gamma] \arrow[r, "\Pi(\theta^{-1}_{A,\gamma}, \mathbf{T}(\rho^{-1}_{\Gamma,A[\gamma]})(\theta_{B,\gamma^+}))"'] & \Pi(F(A[\gamma]), F(B[\gamma^+])[\rho^{-1}_{\Gamma,A[\gamma]}])
\end{tikzcd}$$

Again, it is well-typed by Lemma 4.20.

Proof. The (quite involved) proof appears in Appendix C. □

We now resume the inductive proof, and check the inference rules for stability of types under substitution. We only have to handle the cases for I, Σ and Π since N_1 has no substitution rule.

I-types: Assume we have $\Delta \vdash a, a' : A$ and $\Gamma \vdash \gamma : \Delta$. Because identity types are extensional, they can be at most one isomorphism between identity types, hence $\psi_{\mathrm{I}(A,a,a')[\gamma]} = \psi_{\mathrm{I}(A[\gamma],a[\gamma],a'[\gamma])}$.

Σ-types: Assume we have $\Delta.A \vdash B$ and $\Gamma \vdash \gamma : \Delta$. We want to prove equality of $\psi_{\Sigma(A,B)[\gamma]}$ and $\psi_{\Sigma(A[\gamma],B[\gamma^+])}$. Since $\mathbf{T}(\varphi_\Gamma)(s_{A[\gamma],B[\gamma^+]})$ is an isomorphism, it is equivalent to show the equality of $\mathbf{T}(\varphi_\Gamma)(s_{A[\gamma],B[\gamma^+]}) \circ \psi_{\Sigma(A,B)[\gamma]}$ and $\mathbf{T}(\varphi_\Gamma)(s_{A[\gamma],B[\gamma^+]}) \circ \psi_{\Sigma(A[\gamma],B[\gamma^+])}$.

Calculating yields:

$$\mathbf{T}(\varphi_\Gamma)(s_{A[\gamma],B[\gamma^+]}) \circ \psi_{\Sigma(A,B)[\gamma]}$$

$$= \mathbf{T}(\varphi_\Gamma)(s_{A[\gamma],B[\gamma^+]}) \circ \mathbf{T}(\varphi_\Gamma)(\theta_{\Sigma(A,B),\gamma}) \circ \mathbf{T}(\overline{[\![\gamma]\!]})(\psi_{\Sigma(A,B)})$$

(functoriality of $\mathbf{T}(\varphi_\Gamma)$)

$$= \mathbf{T}(\varphi_\Gamma)(s_{A[\gamma],B[\gamma^+]} \circ \theta_{\Sigma(A,B),\gamma}) \circ \mathbf{T}(\overline{[\![\gamma]\!]})(\psi_{\Sigma(A,B)})$$

(Lemma 4.21)

$$= \mathbf{T}(\varphi_\Gamma)(\Sigma(\theta_{A,\gamma}, \mathbf{T}(\rho^{-1}_{\Gamma,A[\gamma]} \circ \theta_{A,\gamma})(\theta_{B,\gamma^+})) \circ \mathbf{T}(F\gamma)(s_{A,B})) \circ \mathbf{T}(\overline{[\![\gamma]\!]})(\psi_{\Sigma(A,B)})$$

(induction hypothesis on γ)

$$= \mathbf{T}(\varphi_\Gamma)(\Sigma(\theta_{A,\gamma}, \mathbf{T}(\rho^{-1}_{\Gamma,A[\gamma]} \circ \theta_{A,\gamma})(\theta_{B,\gamma^+}))) \circ \mathbf{T}(\overline{[\![\gamma]\!]})(\mathbf{T}(\varphi_\Delta)(s_{A,B}) \circ \psi_{\Sigma(A,B)})$$

$$= \mathbf{T}(\varphi_\Gamma)(\Sigma(\theta_{A,\gamma}, \mathbf{T}(\rho^{-1}_{\Gamma,A[\gamma]} \circ \theta_{A,\gamma})(\theta_{B,\gamma^+}))) \circ \mathbf{T}(\overline{[\![\gamma]\!]})(\Sigma(\psi_A, \psi_B))$$

(functoriality of $\Sigma(\cdot,\cdot)$ – Lemmas 4.4 and 4.5)

$$= \Sigma(\mathbf{T}(\varphi_\Gamma)(\theta_{A,\gamma}) \circ \mathbf{T}(\overline{[\![\gamma]\!]})(\psi_A),$$
$$\mathbf{T}(\mathbf{T}(\overline{[\![\gamma]\!]})(\psi_A))(\mathbf{T}(\varphi^+_\Gamma)(\mathbf{T}(\rho^{-1}_{\Gamma,A[\gamma]} \circ \theta_{A,\gamma})(\theta_{B,\gamma^+}))) \circ \mathbf{T}(\overline{[\![\gamma^+]\!]})(\psi_B))$$

(definition of $\psi_{A[\gamma]}$ and functoriality of $\mathbf{T}$)

$$= \Sigma\left(\psi_{A[\gamma]}, \mathbf{T}(\rho^{-1}_{\Gamma,A[\gamma]} \circ \theta_{A,\gamma} \circ \varphi^+_\Gamma \circ \mathbf{T}(\overline{[\![\gamma]\!]})(\psi_A))(\theta_{B,\gamma^+}) \circ \mathbf{T}(\overline{[\![\gamma^+]\!]})(\psi_B)\right)$$

(definition of $\varphi_{\Gamma.A[\gamma]}$ + cwf calculations)

$$= \Sigma\left(\psi_{A[\gamma]}, \mathbf{T}(\varphi_{\Gamma.A[\gamma]})(\theta_{B,\gamma^+}) \circ \mathbf{T}(\overline{[\![\gamma^+]\!]})(\psi_B)\right)$$

$$= \Sigma(\psi_{A[\gamma]}, \psi_{B[\gamma^+]})$$

Π-types.: The reasoning is analogous to the case of Σ above, using Lemmas 4.7, 4.8 and 4.22.

Term formation rules. The term formation rules are those for the introduction of 0_1, $\mathrm{r}(-)$, $\mathrm{pair}, \mathrm{fst}, \mathrm{snd}, \lambda(-)$ and app.

Unit: We need to prove that $\{\psi_{\overline{1}}\}(0_1) = F0_1[\varphi_1]$, where $0_1 \in \mathrm{Tm}_{\mathcal{C}}(1, \overline{1})$ is defined in the proof of Lemma 4.11. This is straightforward by the universal property of the terminal object.

Reflexivity: Assume that $\Gamma \vdash a = a' : A$. We need to check that

$$\{\psi_{\mathrm{I}(A,a,a')}\}(\mathrm{r}(\overline{[\![a]\!]})) = F(\mathrm{r}(a))[\varphi_\Gamma]$$

By preservation of I-types we have an iso

$$f : F(\mathrm{I}(A, a, a))[\varphi_\Gamma] \cong \mathrm{I}(FA[\varphi_\Gamma], Fa[\varphi_\Gamma], Fa[\varphi_\Gamma]),$$

and by the reflection rule we must have $\{f\}(\{\psi_{\mathrm{I}(A,a,a')}\}(\mathrm{r}(\overline{[\![a]\!]}))) = \{f\}(F\mathrm{r}(a)[\varphi_\Gamma])$ as they are both inhabitants of the identity type.

First projection: Assume we have $\Gamma \vdash A = A'$, $\Gamma \vdash c = c' : \Sigma(A, B)$ from which we deduce $\Gamma \vdash \mathrm{fst}(A, c) = \mathrm{fst}(A', c') : A$. First, we note that $F(\mathrm{fst}(c)) = \mathrm{fst}(\{s_{A,B}\}(Fc))$ by Proposition 4.12. Then, we calculate:

$$\begin{aligned}
F(\mathrm{fst}(c))[\varphi_\Gamma] &= \mathrm{fst}(\{s_{A,B}\}(F(c)))[\varphi_\Gamma] \\
&\qquad \text{(definition } \{\cdot\} + \text{interaction fst/substitution)} \\
&= \mathrm{fst}(\mathtt{q}[s_{A,B} \circ \langle \mathtt{id}, F(c)\rangle][\varphi_\Gamma]) \\
&\qquad \text{(definition functorial action of } \mathbf{T}) \\
&= \mathrm{fst}(\{\mathbf{T}(\varphi_\Gamma)(s_{A,B})\}(F(c)[\varphi_\Gamma])) \\
&\qquad \text{(induction hypothesis on } c) \\
&= \mathrm{fst}(\{\mathbf{T}(\varphi_\Gamma)(s_{A,B})\}(\{\psi_{\Sigma(A,B)}\}(\overline{[\![c]\!]}))) \\
&\qquad \text{(functoriality of } \{\cdot\}) \\
&= \mathrm{fst}(\{\mathbf{T}(\varphi_\Gamma)(s_{A,B}) \circ \psi_{\Sigma(A,B)}\}(\overline{[\![c]\!]})) \\
&\qquad \text{(Unfolding definition of } \psi_{\Sigma(A,B)}) \\
&= \mathrm{fst}(\{\Sigma(\psi_A, \psi_B)\}(\overline{[\![c]\!]})) \\
&\qquad \text{(Lemma 4.4)} \\
&= \mathtt{q}[\psi_A \circ \langle \mathtt{p}, \mathrm{fst}(\mathtt{q})\rangle \circ \langle \mathtt{id}, \overline{[\![c]\!]}\rangle\rangle] \\
&= \{\psi_A\}(\mathrm{fst}(\overline{[\![c]\!]}))
\end{aligned}$$

Second projection: Assume we have $\Gamma \vdash A = A', \Gamma.A \vdash B = B', \Gamma \vdash c = c' : \Sigma(A, B)$ from which we deduce:

$$\Gamma \vdash \mathrm{snd}(A, B, c) = \mathrm{snd}(A', B', c') : B[\langle \mathtt{id}_\Gamma, \mathrm{fst}(A, c)\rangle_A]$$

The calculation follows the same pattern as the one for first projection: we first apply preservation of the combinators by Proposition 4.12, then calculate.

$$
\begin{aligned}
F(\mathrm{snd}(c))[\varphi_\Gamma] &= \{\theta_{B,\langle \mathtt{id}_\Gamma, \mathrm{fst}(c)\rangle_A}\}(\mathrm{snd}(\{s_{A,B}\}(Fc)))[\varphi_\Gamma] \\
&\quad \text{(propagation of } \varphi_\Gamma \text{ and definition of } \mathbf{T}(\varphi_\Gamma)(\cdot)) \\
&= \{\mathbf{T}(\varphi_\Gamma)(\theta_{B,\langle \mathtt{id}_\Gamma, \mathrm{fst}(A,c)\rangle_A})\}(\mathrm{snd}(\{\mathbf{T}(\varphi_\Gamma)(s_{A,B})\}(Fc[\varphi_\Gamma]))) \\
&\quad \text{(I.H. on } c\text{, and definition of } \psi_{\Sigma(A,B)}) \\
&= \{\mathbf{T}(\varphi_\Gamma)(\theta_{B,\langle \mathtt{id}_\Gamma, \mathrm{fst}(A,c)\rangle_A})\}(\mathrm{snd}(\{\Sigma(\psi_A,\psi_B)\}(\overline{[\![c]\!]}))) \\
&\quad \text{(unfolding the functorial action of } \Sigma) \\
&= \{\mathbf{T}(\varphi_\Gamma)(\theta_{B,\langle \mathtt{id}_\Gamma, \mathrm{fst}(A,c)\rangle_A})\}(\mathtt{q}[\psi_B \circ \langle\langle \mathtt{id}, \mathrm{fst}(\overline{[\![c]\!]})\rangle, \mathrm{snd}(\overline{[\![c]\!]})\rangle]) \\
&\quad \text{(definition } \mathbf{T}(\cdot)(\cdot)) \\
&= \{\mathbf{T}(\varphi_\Gamma)(\theta_{B,\langle \mathtt{id}_\Gamma, \mathrm{fst}(A,c)\rangle_A})\}(\{\mathbf{T}(\overline{[\![\langle \mathtt{id}_\Gamma, \mathrm{fst}(A,c)\rangle_A]\!]})(\psi_B)\}(\mathrm{snd}(\overline{[\![c]\!]}))) \\
&\quad \text{(folding definition } \psi_{B[\langle \mathtt{id}_\Gamma, \mathrm{fst}(A,c)\rangle_A]}) \\
&= \{\psi_{B[\langle \mathtt{id}_\Gamma, \mathrm{fst}(A,c)\rangle_A]}\}(\mathrm{snd}(\overline{[\![c]\!]}))
\end{aligned}
$$

Pairing: Assume we have $\Gamma \vdash A = A', \Gamma.A \vdash B = B', \Gamma \vdash a = a' : A'$, and $\Gamma \vdash b = b' : B[\langle \mathtt{id}_\Gamma, \mathrm{fst}(A,c)\rangle_A]$. From that, we deduce:

$$\Gamma \vdash \mathrm{pair}(A,B,a,b) = \mathrm{pair}(A',B',a',b') : \Sigma(A,B)$$

We start by unfolding the definition of $\psi_{\Sigma(A,B)}$, then calculate:

$$
\begin{aligned}
\{\psi_{\Sigma(A,B)}\}(\mathrm{pair}(\overline{[\![a]\!]},\overline{[\![b]\!]})) &= \{\mathbf{T}(\varphi_\Gamma)(s^{-1}_{A,B})\}\left(\{\Sigma(\psi_A,\psi_B)\}(\mathrm{pair}(\overline{[\![a]\!]},\overline{[\![b]\!]}))\right) \\
&\quad \text{(Unfolding the definition of } \Sigma(\psi_A,\psi_B)) \\
&= \{\mathbf{T}(\varphi_\Gamma)(s^{-1}_{A,B})\}\left(\mathrm{pair}(\{\psi_A\}(\overline{[\![a]\!]}), \mathtt{q}[\psi_B \circ \langle\langle \mathtt{id}, \overline{[\![a]\!]}\rangle, \overline{[\![b]\!]}\rangle])\right) \\
&= \{\mathbf{T}(\varphi_\Gamma)(s^{-1}_{A,B})\}\left(\mathrm{pair}(\{\psi_A\}(\overline{[\![a]\!]}), \{\mathbf{T}(\langle \mathtt{id}, \overline{[\![a]\!]}\rangle)(\psi_B)\}(\overline{[\![b]\!]}))\right) \\
&\quad \text{(definition of } \psi_{B[\langle \mathtt{id},a\rangle]})) \\
&= \{\mathbf{T}(\varphi_\Gamma)(s^{-1}_{A,B})\}\left(\mathrm{pair}(\{\psi_A\}(\overline{[\![a]\!]}), \{\mathbf{T}(\varphi_\Gamma)(\theta^{-1}_{B,\langle \mathtt{id}_\Gamma,a\rangle_A})\}(\{\psi_{B[\langle \mathtt{id},a\rangle]}\}(\overline{[\![b]\!]})))\right) \\
&\quad \text{(induction hypothesis on } a \text{ and } b) \\
&= \{\mathbf{T}(\varphi_\Gamma)(s^{-1}_{A,B})\}\left(\mathrm{pair}(Fa[\varphi_\Gamma], \{\mathbf{T}(\varphi_\Gamma)(\theta^{-1}_{B,\langle \mathtt{id}_\Gamma,a\rangle_A})\}(Fb[\varphi_\Gamma]))\right) \\
&\quad \text{(Lemma 3.2)} \\
&= \{s^{-1}_{A,B}\}\left(\mathrm{pair}(Fa, \{\theta^{-1}_{B,\langle \mathtt{id}_\Gamma,a\rangle_A}\}(Fb))\right)[\varphi_\Gamma] \\
&\quad \text{(Proposition 4.12)} \\
&= F(\mathrm{pair}(a,b))[\varphi_\Gamma]
\end{aligned}
$$

Abstraction: Assume we have $\Gamma \vdash A = A'$, $\Gamma.A \vdash b = b' : B$, from which we deduce $\Gamma \vdash \lambda(A,b) = \lambda(A',b') : \Pi(A,B)$.

To limit notational overhead, we omit the first argument of lambda abstractions: we often write $\lambda(b)$ in place of $\lambda(A, b)$.

We first unfold the definition of $\psi_{\Pi(A,B)}$, and then calculate:

$$\begin{aligned}
\{\psi_{\Pi(A,B)}\}(\lambda(\overline{[\![b]\!]})) &= \{\mathbf{T}(\varphi_\Gamma)(i^{-1}_{A,B})\}\left(\{\Pi(\psi_A^{-1}, \mathbf{T}(\psi_A^{-1})(\psi_B))\}(\lambda(\overline{[\![b]\!]}))\right) \\
&\quad \text{(unfolding } \Pi(-,-) \text{ and long simplifications)} \\
&= \{\mathbf{T}(\varphi_\Gamma)(i^{-1}_{A,B})\}\left(\lambda((\{\psi_B\}(\overline{[\![b]\!]}))[\psi_A^{-1}])\right) \\
&\quad \text{(induction hypothesis on } b\text{)} \\
&= \{\mathbf{T}(\varphi_\Gamma)(i^{-1}_{A,B})\}\left(\lambda((Fb[\varphi_{\Gamma.A}][\psi_A^{-1}])\right) \\
&\quad \text{(definition of } \varphi_{\Gamma.A}\text{)} \\
&= \{\mathbf{T}(\varphi_\Gamma)(i^{-1}_{A,B})\}\left(\lambda(Fb[\rho^{-1}_{\Gamma,A} \circ \varphi_\Gamma^+])\right) \\
&\quad \text{(cwf simplification)} \\
&= \{i^{-1}_{A,B}\}(\lambda(Fb[\rho^{-1}_{\Gamma,A}]))[\varphi_\Gamma] \\
&\quad \text{(Lemma 4.16)} \\
&= F(\lambda(b))[\varphi_\Gamma]
\end{aligned}$$

Application: Assume that we have $\Gamma \vdash A = A', \Gamma.A \vdash B = B', \Gamma \vdash c = c' : \Pi(A, B)$, and $\Gamma \vdash a = a' : A$, from which we deduce:

$$\Gamma \vdash \mathrm{app}(A, B, c, a) = \mathrm{app}(A', B', c', a') : B[\langle \mathtt{id}_\Gamma, a\rangle_A]$$

As in the previous case, we now drop the A and B annotations in calculations. First we use that F preserves Π-type (using also that $\theta_{A,\mathtt{id}} = \mathtt{id}$, which is one of the coherence laws for pseudo cwf-morphisms), then calculate:

$$\begin{aligned}
F(\mathrm{app}(c, a))[\varphi_\Gamma] &= \{\theta_{B,\langle \mathtt{id}_\Gamma, a\rangle_A}\}\left(\mathrm{app}(\{i_{A,B}\}(Fc), Fa)\right)[\varphi_\Gamma] \\
&\quad \text{(pushing the substitution by } \varphi_\Gamma \text{ inside)} \\
&= \{\mathbf{T}(\varphi_\Gamma)(\theta_{B,\langle \mathtt{id}_\Gamma, a\rangle_A})\}\left(\mathrm{app}(\{\mathbf{T}(\varphi_\Gamma)(i_{A,B})\}(Fc[\varphi_\Gamma]), Fa[\varphi_\Gamma])\right) \\
&\quad \text{(induction hypothesis on } c \text{ and } a\text{)} \\
&= \{\mathbf{T}(\varphi_\Gamma)(\theta_{B,\langle \mathtt{id}_\Gamma, a\rangle_A})\}\left(\mathrm{app}(\{\mathbf{T}(\varphi_\Gamma)(i_{A,B})\}(\{\psi_{\Pi(A,B)}\}(\overline{[\![c]\!]}), \{\psi_A\}(\overline{[\![a]\!]}))\right) \\
&\quad \text{(definition of } \psi_{\Pi(A,B)}\text{)} \\
&= \{\mathbf{T}(\varphi_\Gamma)(\theta_{B,\langle \mathtt{id}_\Gamma, a\rangle_A})\}\left(\mathrm{app}(\{\Pi(\psi_A^{-1}, \mathbf{T}(\psi_A^{-1})(\psi_B))\}(\overline{[\![c]\!]}), \{\psi_A\}(\overline{[\![a]\!]}))\right) \\
&\quad \text{(calculation of } \Pi(\psi_A^{-1}, \mathbf{T}(\psi_A^{-1})(\psi_B))\text{)} \\
&= \{\mathbf{T}(\varphi_\Gamma)(\theta_{B,\langle \mathtt{id}_\Gamma, a\rangle_A})\}\left(\mathtt{q}\left[\mathbf{T}(\psi_A^{-1})(\psi_B) \circ \left\langle \langle \mathtt{id}, \{\psi_A\}(\overline{[\![a]\!]})\rangle, \mathrm{app}(\overline{[\![c]\!]}, \overline{[\![a]\!]})\right\rangle\right]\right) \\
&\quad \text{(cwf simplification)} \\
&= \{\mathbf{T}(\varphi_\Gamma)(\theta_{B,\langle \mathtt{id}_\Gamma, a\rangle_A})\}\left(\mathtt{q}\left[\psi_B \circ \langle \psi_A^{-1} \circ \langle \mathtt{id}, \{\psi_A\}(\overline{[\![a]\!]})\rangle, \mathrm{app}(\overline{[\![c]\!]}, \overline{[\![a]\!]})\rangle\right]\right) \\
&\quad \text{(cwf simplification)}
\end{aligned}$$

$$
\begin{aligned}
&= \{\mathbf{T}(\varphi_\Gamma)(\theta_{B,\langle \mathtt{id}_\Gamma, a\rangle_A})\}\left(\mathtt{q}\left[\psi_B \circ \langle\langle \mathtt{id}, \overline{[\![a]\!]}\rangle, \mathrm{app}(\overline{[\![c]\!]}, \overline{[\![a]\!]})\rangle\right]\right) \\
&\qquad \text{(folding definitions)} \\
&= \{\mathbf{T}(\varphi_\Gamma)(\theta_{B,\langle \mathtt{id}_\Gamma, a\rangle_A})\}\left(\{\mathbf{T}(\langle \mathtt{id}, \overline{[\![a]\!]}\rangle)(\psi_B)\}(\mathrm{app}(\overline{[\![c]\!]}, \overline{[\![a]\!]}))\right) \\
&\qquad \text{(definition of } \psi_{B[\langle \mathtt{id}_\Gamma, a\rangle_A}) \\
&= \{\psi_{B[\langle \mathtt{id}_\Gamma, a\rangle_A]}\}(\mathrm{app}(\overline{[\![c]\!]}, \overline{[\![a]\!]})))
\end{aligned}
$$

Conversion, and substitution on terms. The last rules left to check are the conversion rules, and the substitution on terms. We do not detail them, as they are all immediate consequences of the corresponding rules for equality and the substitution on terms in the cwf structure.

4.4.2. *Uniqueness of φ and ψ.* We resume the uniqueness proof from Section 3.4.2.

Unit type. Since $1.\mathrm{N}_1 \cong 1$, uniqueness follows from the terminal object universal property.

Identity types. We need to show $\psi'_{\mathrm{I}(A,a,a')} = \psi_{\mathrm{I}(A,a,a')} : \Gamma.\mathrm{I}(A,a,a') \to \Gamma.F(\mathrm{I}(A,a,a'))[\varphi_\Gamma]$. By post-composing with the coherence isomorphism $F(\mathrm{I}(A,a,a')) \cong_{F\Gamma} \mathrm{I}(FA, Fa, Fa')$, we get a morphism between identity types. In an extensional type theory, identity types are either empty or singletons, thus there is at most one morphism between two identity types (which is an isomorphism). This implies that $\psi_{\mathrm{I}(A,a,a')} = \psi'_{\mathrm{I}(A,a,a')}$.

Σ-types. By induction, we assume that $\varphi_{\Gamma.A.B} = \varphi'_{\Gamma.A.B}$. By naturality of φ', we have $\varphi'_{\Sigma(A,B)} = F(\chi^{-1}_{A,B}) \circ \varphi'_{\Gamma.A.B} \circ \chi_{A,B} = \varphi_{\Gamma.\Sigma(A,B)}$. Because φ is also natural, we can derive a similar equation, hence $\psi_{\Sigma(A,B)} = \psi'_{\Sigma(A,B)}$.

Π-types. By induction we assume $\varphi_{\Gamma.A.B} = \varphi'_{\Gamma.A.B}$. It also follows from induction hypothesis that $\varphi_\Gamma = \varphi'_\Gamma$, $\psi_A = \psi'_A$ and $\psi_B = \psi'_B$.

Let $\mathtt{ev}_{A,B}$ be the evaluation map, a morphism in $\Gamma.A$:

$$\mathtt{ev}_{A,B} = \langle \mathtt{p}, \mathrm{app}(\mathtt{q}, \mathtt{q}[\mathtt{p}])\rangle : \Pi(A,B)[\mathtt{p}] \to B$$

Proposition A.5 entails:

Lemma 4.23. *Take $B \in \mathrm{Ty}_{\mathcal{C}}(\Gamma.A)$ in any cwf $\mathcal{C}$ with Π-types. The only automorphism f of $\Pi(A,B)$ (in $\mathbf{T}\Gamma$) such that $T(\mathtt{p})(f) : \Gamma.A.\Pi(A,B)[\mathtt{p}] \cong \Gamma.A.\Pi(A,B)[\mathtt{p}]$ satisfies $\mathtt{ev}_{A,B} \circ T(\mathtt{p})(f) = \mathtt{ev}_{A,B}$ is the identity.*

We will exploit this, and show that $\psi^{-1}_{\Pi(A,B)} \circ \psi'_{\Pi(A,B)}$ satisfies the condition. But first, we prove that the ψ component of a pseudo cwf-transformation from $\overline{[\![-]\!]}$ to F automatically preserves evaluation, in the following sense.

Lemma 4.24. *Let (φ, ψ) be any pseudo cwf-transformation from $\overline{[\![-]\!]}$ to F. Then, we have:*

$$\mathtt{ev}'_{A,B} \circ \mathbf{T}(\mathtt{p})(\psi_{\Pi(A,B)}) = \mathtt{ev}_{A,B} : \overline{[\![\Gamma.A.\Pi(A,B)[\mathtt{p}]]\!]} \to \overline{[\![\Gamma.A.B]\!]}$$

where we use an alternative evaluation map:

$$\mathtt{ev}'_{A,B} \quad = \quad \varphi^{-1}_{\Gamma.A.B} \circ F(\mathtt{ev}_{A,B}) \circ \rho^{-1}_{\Gamma.A,\Pi(A,B)[\mathtt{p}]} \circ \theta_{\Pi(A,B),\mathtt{p}} \circ \varphi^{+}_{\Gamma.A}$$

$$: \quad \overline{[\![\Gamma.A]\!]}.F(\Pi(A,B))[\varphi_\Gamma \circ \mathtt{p}] \to \overline{[\![\Gamma.A.B]\!]}$$

Proof. We calculate:

$$\begin{aligned}
& F(\mathtt{ev}_{A,B}) \circ \rho^{-1}_{\Gamma.A,\Pi(A,B)[\mathtt{p}]} \circ \theta_{\Pi(A,B),\mathtt{p}} \circ \varphi^{+}_{\Gamma.A} \circ T(\mathtt{p})(\psi_{\Pi(A,B)}) \\
= {} & F(\mathtt{ev}_{A,B}) \circ \rho^{-1}_{\Gamma.A,\Pi(A,B)[\mathtt{p}]} \circ \varphi^{+}_{\Gamma.A} \circ T(\varphi_{\Gamma.A})(\theta_{\Pi(A,B),\mathtt{p}}) \circ T(\mathtt{p})(\psi_{\Pi(A,B)}) \\
& \qquad \text{(Lemma B.2)} \\
= {} & F(\mathtt{ev}) \circ \rho^{-1}_{\Gamma.A,\Pi(A,B)[\mathtt{p}]} \circ \varphi^{+}_{\Gamma.A} \circ \psi_{\Pi(A,B)[\mathtt{p}]} \\
& \qquad \text{(Coherence of pseudo-cwf transformations)} \\
= {} & F(\mathtt{ev}) \circ \varphi_{\Gamma.A.\Pi(A,B)[\mathtt{p}]} \\
& \qquad \text{(Naturality of } \varphi\text{)} \\
= {} & \varphi_{\Gamma.A.B} \circ \mathtt{ev}_{A,B}
\end{aligned}$$

Importantly, this is proved not with the inductive definition of (φ, ψ), but only using general properties of pseudo cwf-transformations. □

Using that both $\psi_{\Pi(A,B)}$ and $\psi'_{\Pi(A,B)}$ satisfy the property above, their equality follows easily. We calculate:

$$\begin{aligned}
& \mathtt{ev}_{A,B} \circ \mathbf{T}(\mathtt{p})(\psi^{-1}_{\Pi(A,B)}) \circ \mathbf{T}(\mathtt{p})(\psi'_{\Pi(A,B)}) \\
& \qquad \text{(Lemma 4.24 on } \psi_{\Pi(A,B)}\text{)} \\
= {} & \mathtt{ev}'_{A,B} \circ \mathbf{T}(\mathtt{p})(\psi'_{\Pi(A,B)}) \\
& \qquad \text{(Lemma 4.24 on } \psi'_{\Pi(A,B)}\text{)} \\
= {} & \mathtt{ev}_{A,B}
\end{aligned}$$

Hence, $\psi_{\Pi(A,B)} = \psi'_{\Pi(A,B)}$ by Lemma 4.23.

4.5. **The free lccc.** Let **LCC** be the 2-category of lcccs. Since biequivalences preserve bi-initiality, the biequivalence of [6] $\mathbf{CwF}^{\Sigma,\Pi,I}_{d} \simeq \mathbf{LCC}$ allows us to turn the bi-initial cwf into a bi-initial LCCC:

Theorem 4.25. *The category of contexts of* $\mathcal{T}^{\mathrm{I},\mathrm{N}_1,\Sigma,\Pi}$ *is a bifree lccc on one object, that is, it is bi-initial in* $\boldsymbol{LCC}^{o}$.

5. Conclusion

We have shown that a version of Martin-Löf Type Theory gives rise to the *free* cwf, with and without $\mathrm{I}, \mathrm{N}_1, \Sigma$ and Π. We have proved this freeness result both in a 1-categorical sense (with respect to *strict cwf-morphisms*), and in a 2-categorical sense (with respect to *pseudo cwf-morphisms*). It follows that the category of contexts of our type theory $\mathcal{T}^{\mathrm{I},\mathrm{N}_1,\Sigma,\Pi}$ is a bifree lccc. We also proved that equality is undecidable in $\mathcal{T}^{\mathrm{I},\Pi}$ (improving slightly on the folklore result), hence showing undecidability of equality in a bifree lccc. There is only one bifree lccc up to equivalence, so in that sense $\mathcal{T}^{\mathrm{I},\mathrm{N}_1,\Sigma,\Pi}$ is *the* bifree lccc. However, note that the undecidability statement is only about our particular presentation of the bifree lccc, and not about an arbitrary one. We could introduce a notion of recursively presented lccc and ask the more general question whether an arbitrary such recursively presented bifree lccc has undecidable equality, but we will leave that for future work.

References

[1] Henk P. Barendregt. Lambda calculi with types. In Samson Abramsky, Dov Gabbay, and Tom Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, pages 118–310. Oxford University Press, 1992.
[2] Jean Benabou. Fibered categories and the foundations of naive category theory. *J. Symb. Log*, 50(1):10–37, 1985.
[3] John Cartmell. Generalized algebraic theories and contextual categories. *Annals of Pure and Applied Logic*, 32:209–243, 1986.
[4] Simon Castellan, Pierre Clairambault, and Peter Dybjer. Undecidability of equality in the free locally cartesian closed category. In Thorsten Altenkirch, editor, *13th International Conference on Typed Lambda Calculi and Applications, TLCA 2015, July 1-3, 2015, Warsaw, Poland*, volume 38 of *LIPIcs*, pages 138–152. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.
[5] 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.
[6] 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.
[7] Pierre-Louis Curien. Substitution up to isomorphism. *Fundamenta Informaticae*, 19(1,2):51–86, 1993.
[8] 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.
[9] Martin Hofmann. Interpretation of type theory in locally cartesian closed categories. In *Proceedings of CSL*. Springer LNCS, 1994.
[10] Martin Hofmann. *Extensional concepts in intentional type theory*. PhD thesis, University of Edinburgh, 1995.
[11] Martin Hofmann. Syntax and semantics of dependent types. In Andrew Pitts and Peter Dybjer, editors, *Semantics and Logics of Computation*. Cambridge University Press, 1997.
[12] Per Martin-Löf. Constructive mathematics and computer programming. In *Logic, Methodology and Philosophy of Science, VI, 1979*, pages 153–175. North-Holland, 1982.
[13] Per Martin-Löf. Substitution calculus. Notes from a lecture given in Göteborg, November 1992.
[14] Erik Palmgren and Steve J. Vickers. Partial horn logic and cartesian categories. *Annals of Pure and Applied Logic*, 145(3):314 – 353, 2007.
[15] Robert Seely. Locally cartesian closed categories and type theory. *Math. Proc. Cambridge Philos. Soc.*, 95(1):33–48, 1984.
[16] Thomas Streicher. Semantics of type theory. In *Progress in Theoretical Computer Science*, number 12. Basel: Birkhaeuser Verlag, 1991.
[17] 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.

Appendix A. Combinators for pseudo cwf-morphisms and results from [6]

A.1. **Pseudo cwf-morphisms.** We first give the full definition of pseudo cwf-morphisms along with the coherence and naturality laws that were only sketched in the main text, and we recall a few important results concerning their manipulations.

First, we recall a notation from [6]. Let $\mathcal{C}$ be a cwf. In Section 3.1, we introduced a functor:

$$\boldsymbol{T} : \mathcal{C}^{\mathrm{op}} \to \mathrm{Cat}$$

which in particular associates, to any object Γ of $\mathcal{C}$, a category $\boldsymbol{T}(\Gamma)$. We introduced the notation $f : A \cong_\Gamma B$ to mean that f is an invertible map from A to B in $\boldsymbol{T}\Gamma$. Then, for $a : \Gamma \vdash A$, we also introduced $\{f\}(a) : \Gamma \vdash B$ for the *coercion* of a to type B. In that case, whenever $b = \{f\}(a)$, we introduce the new notation

$$b =_f a$$

meaning that a and b are the same up to coercion.

We are now in position to give the full definition of a pseudo cwf-morphism.

Definition A.1. A **pseudo cwf-morphism** from $(\mathcal{C}, T)$ to $(\mathcal{C}', T')$ is a pair (F, σ) where:

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

The following preservation properties must be satisfied:

- Substitution is preserved: For each substitution $\gamma : \Delta \to \Gamma$ in $\mathcal{C}$ and $A \in \mathrm{Ty}(\Gamma)$, there is an isomorphism of types $\theta_{A,\gamma} : \sigma_\Gamma(A)[F\gamma] \to \sigma_\Delta(A[\gamma])$ such that substitution in 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: $F1$ is terminal, let $!_F : 1 \to F1$ be the isomorphism.
- Context comprehension is preserved: The context $F(\Gamma.A)$, along with the projections $F(\mathtt{p}_{\Gamma,A})$ and $\{\theta_{A,\mathtt{p}}^{-1}\}(\sigma_{\Gamma.A}^{A[\mathtt{p}]}(\mathtt{q}_{\Gamma,A}))$, is a context comprehension of $F\Gamma$ and $\sigma_\Gamma(A)$. Note that the universal property of context comprehension provides a unique isomorphism $\rho_{\Gamma,A} : F(\Gamma.A) \to F\Gamma.\sigma_\Gamma(A)$ which preserves projections in the following sense:

$$F(\mathtt{p}_A) = \mathtt{p}_{\sigma_\Gamma(A)}\rho_{\Gamma,A} \tag{a}$$

$$\sigma_{\Gamma.A}^{A[\mathtt{p}]}(\mathtt{q}_A) =_{\theta_{A,\mathtt{p}}} \mathtt{q}_{\sigma_\Gamma A}[\rho_{\Gamma,A}] \tag{b}$$

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 pseudonatural 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 $\boldsymbol{\sigma}_\Gamma(f) = \rho_{\Gamma,B}F(f)\rho_{\Gamma,A}^{-1}$ on a morphism $f : A \to B$.

More explicitely, pseudonaturality of $\boldsymbol{\sigma}$ amounts to the following coherence and naturality laws.

- *Identity.* For all $A \in \mathrm{Ty}(\Gamma)$, we have $\theta_{A,\mathtt{id}} = \mathtt{id}_{F\Gamma.\sigma_\Gamma(A)}$,

- *Coherence.* For all $\delta : \Xi \to \Delta$ and $\gamma : \Delta \to \Gamma$, the following diagram commutes.

$$\begin{tikzcd} F\Xi.\sigma_\Gamma(A)[F(\gamma\delta)] \arrow[rr, "\theta_{A,\gamma\delta}"] \arrow[dr, "\boldsymbol{T'}(F\delta)(\theta_{A,\gamma})"'] & & F\Xi.\sigma_\Xi(A[\gamma\delta]) \\ & F\Xi.\sigma_\Delta(A[\gamma])[F(\delta)] \arrow[ur, "\theta_{A[\gamma],\delta}"'] & \end{tikzcd}$$

- *Naturality.* For all $\delta : \Delta \to \Gamma$ in $\mathcal{C}$, $A, B \in \mathrm{Ty}(\Gamma)$ and $f : A \to B$ in $\boldsymbol{T}(\Gamma)$, the following diagram commutes in $\boldsymbol{T'}(F\Delta)$.

$$\begin{tikzcd} \sigma_\Gamma(A)[F\delta] \arrow[r, "\theta_{A,\delta}"] \arrow[d, "\boldsymbol{T'}(F\delta)(\boldsymbol{\sigma}_\Gamma(f))"] & \sigma_\Delta(A[\delta]) \arrow[d, "\boldsymbol{\sigma}_\Delta(\boldsymbol{T}(\delta)(f))"] \\ \sigma_\Gamma(B)[F\delta] \arrow[r, "\theta_{B,\delta}"] & \sigma_\Delta(B[\delta]) \end{tikzcd}$$

This concludes the full definition of pseudo cwf-morphisms. We now recall a few key properties of those that are used in the course of the paper. For the proofs we refer to [6].

Firstly, it follows from the definition above that substitution extension is preserved up to isomorphism.

Proposition A.2. *All pseudo cwf-morphisms (F, σ) from $(\mathcal{C}, T)$ to $(\mathcal{C}', T')$ preserve substitution extension in the sense that, if $\delta : \Delta \to \Gamma$ in $\mathcal{C}$ and $a : \Delta \vdash A[\delta]$, then*

$$F(\langle \delta, a\rangle) = \rho^{-1}_{\Gamma,A}\langle F\delta, \{\theta^{-1}_{A,\delta}\}(\sigma^{A[\delta]}_\Delta(a))\rangle$$

Proof. Proposition 3.2 of [6]. ☐

It is convenient to also have a specialized version of the proposition above, in the case where the substitution has the form $\langle \gamma\mathtt{p}, \mathtt{q}\rangle : \Delta.A[\gamma] \to \Gamma.A$, *i.e.* it is the *lifting* of a substitution $\gamma : \Delta \to \Gamma$ in the presence of an additional type $A \in \mathrm{Ty}(\Gamma)$. We get:

Lemma A.3. *Let $(F, \sigma) : (\mathcal{C}, T) \to (\mathcal{C}', T')$ be a pseudo cwf-morphism with families of isomorphisms θ and ρ. Then for any $\delta : \Delta \to \Gamma$ in $\mathcal{C}$ and type $A \in \mathrm{Ty}(\Gamma)$, we have:*

$$F(\langle \delta\mathtt{p}, \mathtt{q}\rangle) = \rho^{-1}_{\Gamma,A}\langle F(\delta)\mathtt{p}, \mathtt{q}\rangle\theta^{-1}_{A,\delta}\rho_{\Delta,A[\delta]}$$

Proof. Lemma A.2 of [6]. ☐

Finally, we mention a technical lemma stating a compatibility between substitutions and coercions.

Lemma A.4. *Let $(\mathcal{C}, T)$ be a cwf. Let $\delta : \Delta \to \Gamma$ be a substitution, $f : A \cong_\Gamma A'$ an isomorphism of types in $\mathrm{Ty}(\Gamma)$ and $a : \Gamma \vdash A$ be a term. Then:*

$$(\{f\}(a))[\delta] = \{\boldsymbol{T}(\delta)(f)\}(a[\delta])$$

Proof. Lemma A.1 of [6]. ☐

A.2. **Preservation of type formers.** Now, we recall some material from [6] about the preservation of Π-types by pseudo cwf-morphisms. In particular, we recall a more abstract characterization of preservation of Π-types, based on a universal property satisfied by Π-types: that of a *dependent product diagram*.

If $g : A \to B$ and $f : B \to C$ are morphisms in a category $\mathcal{C}$, a dependent product of g along f is a diagram of the form:

$$\begin{tikzcd}
 & P \arrow[r] \arrow[d] \arrow[dl, bend right, "ev"'] \arrow[dr, phantom, very near start, "\lrcorner"] & D \arrow[d, "\Pi_f(g)"] \\
A \arrow[r, "g"] & B \arrow[r, "f"] & C
\end{tikzcd}$$

which is universal among all such diagrams in g and f in the following sense:

$$\begin{tikzcd}
 & P' \arrow[r] \arrow[d, dotted] \arrow[ddl, bend right] \arrow[dd, bend right] \arrow[dr, phantom, very near start, "\lrcorner"] & D' \arrow[d, dotted] \arrow[dd, bend right] \\
 & P \arrow[r] \arrow[d] \arrow[dl, bend right] \arrow[dr, phantom, very near start, "\lrcorner"] & D \arrow[d] \\
A \arrow[r] & B \arrow[r] & C
\end{tikzcd}$$

It follows from the universal property that dependent products of g along f are unique up to isomorphism. First, we observe that Π-types indeed yield dependent product diagrams.

Proposition A.5. *Let* $(\mathcal{C}, T)$ *be a cwf supporting* Π*-types, let* Γ *be a context in* $\mathcal{C}$*, let* $A \in \mathrm{Ty}(\Gamma)$ *and* $B \in \mathrm{Ty}(\Gamma.A)$*, then the following diagram is a dependent product diagram, where* $ev_{A,B} = \langle \mathtt{p}_{\Pi(A,B)[\mathtt{p}_A]}, \mathrm{app}(\mathtt{q}_{\Pi(A,B)[\mathtt{p}_A]}, \mathtt{q}_A[\mathtt{p}_{\Pi(A,B)[\mathtt{p}_A]}]) \rangle$.

$$\begin{tikzcd}
 & \Gamma.A.\Pi(A,B)[\mathtt{p}_A] \arrow[r, "\langle \mathtt{p}_A \mathtt{p}_{\Pi(A,B)[\mathtt{p}_A]}, \mathtt{q}_{\Pi(A,B)[\mathtt{p}_A]} \rangle"] \arrow[d, "\mathtt{p}_{\Pi(A,B)[\mathtt{p}_A]}"] \arrow[dl, bend right, "ev_{A,B}"'] \arrow[dr, phantom, very near start, "\lrcorner"] & \Gamma.\Pi(A,B) \arrow[d, "\mathtt{p}_{\Pi(A,B)}"] \\
\Gamma.A.B \arrow[r, "\mathtt{p}_B"] & \Gamma.A \arrow[r, "\mathtt{p}_A"] & \Gamma
\end{tikzcd}$$

It is referred to as the chosen *dependent product of* $\mathtt{p}_B$ *along* $\mathtt{p}_A$*.*

Proof. Proposition 4.6 in [6]. □

With this in place, we can recall from [6] the characterization of preservation of Π-types in terms of preservation of dependent product diagrams.

Lemma A.6. *Let* $(\mathcal{C}, T)$ *and* $(\mathcal{C}', T')$ *be cwfs supporting* Π*-types, and* (F, σ) *be a pseudo cwf-morphism from* $(\mathcal{C}, T)$ *to* $(\mathcal{C}', T')$*. Then* (F, σ) *preserves* Π*-types if and only if the image of any chosen dependent product diagram is a dependent product diagram.*

Proof. Lemma 4.7 in [6]. □

A.3. **Preservation of isomorphisms by type formers.** We recall only one lemma from [6] which is not covered by the development in the main text: that identity types preserve isomorphisms.

Lemma A.7. *For any* $A, A' \in \mathrm{Ty}_{\mathcal{C}}(\Gamma)$*,* $B, B' \in \mathrm{Ty}_{\mathcal{C}}(\Gamma.A)$*, if* $f : A \cong_\Gamma A'$ *and* $a, a' \in \Gamma \vdash A$*, then* $\mathrm{I}(A, a, a') \cong_\Gamma \mathrm{I}(A', \{f\}(a), \{f\}(a'))$

Proof. It is one case of the Lemma B.1 from [6]. □

Appendix B. On pseudo cwf-transformations (erratum for [6])

In [6], pseudo cwf-transformations (2-cells in the 2-category of cwfs) are defined as follows.

Definition B.1. [Pseudo cwf-transformation – version of [6]] Let F and G be cwf-morphisms from $(\mathcal{C}, T)$ to $(\mathcal{C}', T')$. A *pseudo cwf-transformation* from F to G is a pair (φ, ψ) where $\varphi : F \Rightarrow G$ is a natural transformation, and $\psi_A : FA \to GA[\varphi_\Delta]$ is a morphism in $\mathbf{T}'(F\Delta)$ for each Δ in $\mathcal{C}$ and $A \in \mathrm{Ty}_{\mathcal{C}}(\Delta)$. Furthermore, ψ_A is natural in A and the following diagram commutes:

$$\begin{tikzcd}
FA[F\gamma] \arrow[r, "\mathbf{T}'(F\gamma)(\psi_A)"] \arrow[d, "\theta^F_{A,\gamma}"] & GA[\varphi_\Delta F(\gamma)] \arrow[d, "\mathbf{T}'(\varphi_\Gamma)(\theta^G_{A,\gamma})"] \\
F(A[\gamma]) \arrow[r, "\psi_{A[\gamma]}"'] & G(A[\gamma])[\varphi_\Gamma]
\end{tikzcd}$$

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

When working on the present paper, we discovered a shortcoming of this definition: the component ψ is not constrained enough by φ. This causes a mismatch with the 2-cells in **LCC** (where only the φ remains), and as a consequence the family of cwf-transformations ϵ used in the biequivalence (see [6]) fails to satisfy the required condition of pseudonatural transformations.

What is missing from the definition of pseudo cwf-transformation is the following coherence diagram which must commute for the biequivalence to hold:

$$\begin{tikzcd}
F(\Delta.A) \arrow[rr, "\varphi_{\Delta.A}"] \arrow[d, "\rho^F_{\Delta,A}"'] & & G(\Delta.A) \arrow[d, "\rho^G_{\Delta,A}"] \\
F\Delta.FA \arrow[r, "\psi_A"] & F\Delta.FA[\varphi_\Delta] \arrow[r, "\varphi^+_\Delta"] & G\Delta.GA
\end{tikzcd}$$

This diagram shows that ψ can be defined from φ. Hence we could simply define pseudo cwf-transformations as natural transformations $\varphi : F \Rightarrow G$. However, we have not done so, because pseudo cwf-transformations are most naturally presented with the ψ, reflecting the second component of cwfs and cwf-morphisms. Moreover, in our proof of bifreeness, the construction of the unique cwf-transformation between the interpretation and an arbitrary pseudo cwf-functor naturally constructs φ and ψ by mutual induction.

With the addition of the coherence diagram above, the naturality requirement and the coherence diagram of Definition B.1 become redundant, as we establish here. The following lemma is a mild generalization of Lemma 5.6 of [6].

Lemma B.2. *Let $F, G : \mathcal{C} \to \mathcal{C}'$ be pseudo cwf-morphisms, and let (φ, ψ) be a pseudo cwf-transformation from F to G, in the sense of Definition 3.4. Then, (φ, ψ) is also a pseudo cwf-transformation in the sense of Definition B.1,* i.e. *it satisfies the coherence law:*

$$\begin{tikzcd}
FA[F\gamma] \arrow[r, "\mathbf{T}'(F\gamma)(\psi_A)"] \arrow[d, "\theta_{A,\gamma}"] & GA[\varphi_\Delta F(\gamma)] \arrow[d, "\mathbf{T}'(\varphi_\Gamma)(\theta'_{A,\gamma})"] \\
F(A[\gamma]) \arrow[r, "\psi_{A[\gamma]}"'] & G(A[\gamma])[\varphi_\Gamma]
\end{tikzcd}$$

Proof. We first check that ψ_A is natural in A. More explicitely, recall from [6] that each cwf-transformation $F : \mathcal{C} \to \mathcal{C}'$ induces, for any context Γ of $\mathcal{C}$, a functor:

$$F_\Gamma : \mathbf{T}(\Gamma) \to \mathbf{T}'(F\Gamma)$$

Its action on types is obvious. Recall that a morphism $f : A \to B$ in $\mathbf{T}(\Gamma)$ is a morphism $f : \Gamma.A \to \Gamma.B$ in $\mathcal{C}$ such that $\mathtt{p} \circ f = \mathtt{p}$. Its action on F_Γ is simply:

$$\rho^F_{\Gamma,B} \circ F(f) \circ (\rho^F_{\Gamma,A})^{-1} : F\Gamma.FA \to F\Gamma.FB$$

The naturality of ψ_A in A follows directly from the naturality of φ.

We also need to check that the coherence law of Definition B.1 holds. We follow the proof of Lemma 5.6 in [6], and consider the following composition of squares:

$$\begin{tikzcd}
F\Gamma.G(A[\gamma])[\varphi_\Gamma] \arrow[r, "\varphi^+_\Gamma"] \arrow[d, "\mathtt{p}"] \arrow[dr, phantom, "\lrcorner", very near start] & G\Gamma.G(A[\gamma]) \arrow[rr, "\rho^G_{\Delta,A} \circ G(\gamma^+) \circ (\rho^G_{\Gamma,A[\gamma]})^{-1}"] \arrow[d, "\mathtt{p}"] \arrow[drr, phantom, "\lrcorner", very near start] & & G\Delta.G(A) \arrow[d, "\mathtt{p}"] \\
F\Gamma \arrow[r, "\varphi_\Gamma"'] & G\Gamma \arrow[rr, "G\gamma"'] & & G\Delta
\end{tikzcd}$$

The left hand side square is a pullback – the standard substitution pullback of $\mathtt{p}_{G(A[\gamma])}$ along φ_Γ. In [6], it is noted that the right hand side square is also a pullback, as the image of a substitution pullback through G; which is there assumed to preserve pullbacks. For us though F does not in general preserve pullbacks, but it preserves this one. Indeed, by the commutation property of Lemma 4.20 it is straightforward to prove it to be isomorphic to the substitution pullback of $\mathtt{p}_{GA}$ along $G\gamma$.

Therefore, the composition of the two squares is a pullback as well. Once we have established this, the proof follows exactly as in the proof of Lemma 5.6 in [6]. We exploit that the two paths $\mathbf{T}(\varphi_\Gamma)(\theta^G_{A,\gamma}) \circ \mathbf{T}(F\gamma)(\psi_A)$ and $\psi_{A[\gamma]} \circ \theta^F_{A,\gamma}$ of the coherence diagram behave in the same way with respect to this pullback, and therefore are equal by the universal property. The calculations are given in detail in [6], so we do not repeat them here. $\square$

Thus all pseudo cwf-transformations in the sense of Definition 3.4 are also pseudo cwf-transformations in the sense of Definition B.1. Moreover, all pseudo cwf-transformations used in the biequivalence [6] trivially obey this stronger condition. In fact, all the pseudo cwf-transformations (φ, ψ) used in the biequivalence were defined by their φ component, whereas the ψ component was defined *a posteriori* via the equation of Definition 3.4.

Appendix C. Proof of Lemma 4.22

The proof of Lemma 4.22 uses the notion of dependent product diagram and in particular the corresponding universal property – both paths around the diagram will be proved to preserve the structure of some dependent product diagrams. Their equality will immediately follow from the uniqueness component of the universal property. We now inspect in turn all four morphisms of the diagram of Lemma 4.22, and prove that they preserve dependent product structure.

In the remainder of this section, we consider cwfs $\mathcal{C}, \mathcal{C}'$ supporting Π-types, F from $\mathcal{C}$ to $\mathcal{C}'$ preserving Π-types, a substitution $\gamma : \Gamma \to \Delta$ of $\mathcal{C}$ and types $A \in \mathrm{Ty}_{\mathcal{C}}(\Delta), B \in \mathrm{Ty}_{\mathcal{C}}(\Delta.A)$. We first note:

Lemma C.1. *The following is a dependent product diagram.*

$$\begin{tikzcd}
 & F\Delta.FA.F(\Pi(A,B))[\mathtt{p}] \arrow[dl, bend right, "\mathtt{ev}_{F(\Pi(A,B))}"'] \arrow[r, "\mathtt{p}^+"] \arrow[d, "\mathtt{p}"] \arrow[dr, phantom, "\lrcorner", very near start] & F\Delta.F\Pi(A,B) \arrow[d, "\mathtt{p}"] \\
F\Delta.FA.FB[\rho^{-1}_{\Delta,A}] \arrow[r, "\mathtt{p}"] & F\Delta.FA \arrow[r, "\mathtt{p}"] & F\Delta
\end{tikzcd}$$

where $\mathtt{ev}_{F(\Pi(A,B))} = \rho^+_{\Delta,A} \circ \rho_{\Delta.A,B} \circ F(\mathtt{ev}_{A,B}) \circ \rho^{-1}_{\Delta.A,\Pi(A,B)[\mathtt{p}]} \circ \theta_{\Pi(A,B),\mathtt{p}} \circ (\rho^{-1}_{\Delta,A})^+$.

This means that there is a unique isomorphism to the chosen dependent product diagram of FA and $FB[\rho^{-1}_{\Delta,A}]$, which is given by the morphism:

$$i_{A,B} : F(\Pi(A,B)) \to \Pi(FA, FB[\rho^{-1}_{\Delta,A}])$$

involved in the definition of pseudo cwf-morphisms preserving Π-types. The fact that it yields a morphism of dependent product diagrams means that we also have:

$$\mathtt{ev}_{FA,FB[\rho^{-1}_{\Delta,A}]} \circ \mathbf{T}(\mathtt{p})(i_{A,B}) = \mathtt{ev}_{F(\Pi(A,B))}$$

Proof. By Lemma A.6, the image by F of the chosen dependent product diagram of A and B is a dependent product diagram. From this diagram we can obtain the diagram above by applying structural isomorphisms ρ and θ on the nodes. Being obtained by transporting a dependent product diagram along isomorphisms, it is itself a dependent product diagram. Its evaluation morphism, $\mathtt{ev}_{F(\Pi(A,B))}$, is obtained by going through the isomorphisms. The fact that $i_{A,B}$ corresponds to the canonical dependent product diagram morphism is a direct verification, using the universal property of dependent product diagrams. $\square$

From that follows immediately:

Lemma C.2. *The following is a morphism between two dependent product diagrams:*

$$
\begin{tikzcd}
& F\Gamma.FA[F\gamma].F(\Pi(A,B))[(F\gamma)\circ \mathtt{p}] \arrow[r, "\mathtt{p}^+"] \arrow[dl, "\mathbf{T}(F\gamma)(\mathtt{ev}_{F(\Pi(A,B))})"'] \arrow[d, "\mathtt{p}"] \arrow[dd, bend left, "\mathbf{T}((F\gamma)\circ\mathtt{p})(i_{A,B})"] & F\Gamma.F\Pi(A,B)[F\gamma] \arrow[d, "\mathtt{p}"] \arrow[dd, bend left, "\mathbf{T}(F\gamma)(i_{A,B})"] \\
F\Gamma.FA[F\gamma].FB[\rho^{-1}_{\Delta,A}][(F\gamma)^+] \arrow[r, "\mathtt{p}"] & F\Gamma.FA[F\gamma] \arrow[r, "\mathtt{p}"] & F\Gamma \\
& F\Gamma.FA[F\gamma].\Pi(FA,FB[\rho^{-1}_{\Delta,A}])[(F\gamma)\circ\mathtt{p}] \arrow[r, "\mathtt{p}^+"] \arrow[dl, "\mathtt{ev}_{FA[F\gamma],FB[(F\gamma)^+]}"'] \arrow[d, "\mathtt{p}"] & F\Gamma.\Pi(FA,FB[\rho^{-1}_{\Delta,A}])[F\gamma] \arrow[d, "\mathtt{p}"] \\
F\Gamma.FA[F\gamma].FB[\rho^{-1}_{\Delta,A}][(F\gamma)^+] \arrow[r, "\mathtt{p}"] & F\Gamma.FA[F\gamma] \arrow[r, "\mathtt{p}"] & F\Gamma
\end{tikzcd}
$$

where all arrows not explicitely displayed are identities.

Proof. This diagram is obtained by pulling back that of Lemma C.1 along $F\gamma$ – it is straightforward that this operation preserves dependent product diagrams. $\square$

Thus we have proved that the left hand side and the right hand side (instantiating Lemma C.1 with $A[\gamma], B[\gamma^+]$) maps of Lemma 4.22 correspond as required to morphisms of dependent product diagrams. This remains to be done for the upper and lower maps. We start with the lower map.

Lemma C.3. *The following is a morphism between two dependent product diagrams:*

$$
\begin{tikzcd}
& F\Gamma.FA[F\gamma].\Pi(FA,FB[\rho^{-1}_{\Delta,A}])[(F\gamma)\circ\mathtt{p}] \arrow[r, "\mathtt{p}^+"] \arrow[dl, "\mathtt{ev}_{FA[F\gamma],FB[(F\gamma)^+]}"'] \arrow[d, "\mathtt{p}"] \arrow[dd, bend left, "\mathbf{T}(\mathtt{p})(\Pi(\theta^{-1}_{A,\gamma},\mathbf{T}(\rho^{-1}_{\Gamma,A[\gamma]})(\theta_{B,\gamma^+})))\circ\theta^+_{A,\gamma}"'] & F\Gamma.\Pi(FA,FB[\rho^{-1}_{\Delta,A}])[F\gamma] \arrow[d, "\mathtt{p}"] \arrow[dd, bend left, "\Pi(\theta^{-1}_{A,\gamma},\mathbf{T}(\rho^{-1}_{\Gamma,A[\gamma]})(\theta_{B,\gamma^+}))"'] \\
F\Gamma.FA[F\gamma].FB[\rho^{-1}_{\Delta,A}][(F\gamma)^+] \arrow[r, "\mathtt{p}"] \arrow[dd, bend left, "\mathbf{T}(\rho^{-1}_{\Gamma,A[\gamma]})(\theta_{B,\gamma^+})\circ\theta^+_{A,\gamma}"'] & F\Gamma.FA[F\gamma] \arrow[r, "\mathtt{p}"] \arrow[dd, bend left, "\theta_{A,\gamma}"] & F\Gamma \\
& F\Gamma.F(A[\gamma]).\Pi(F(A[\gamma]),F(B[\gamma^+])[\rho^{-1}_{\Gamma,A[\gamma]}])[\mathtt{p}] \arrow[r, "\mathtt{p}^+"] \arrow[dl, "\mathtt{ev}_{F(A[\gamma]),F(B[\gamma^+])}"'] \arrow[d, "\mathtt{p}"] & F\Gamma.\Pi(F(A[\gamma]),F(B[\gamma^+])[\rho^{-1}_{\Gamma,A[\gamma]}]) \arrow[d, "\mathtt{p}"] \\
F\Gamma.F(A[\gamma]).F(B[\gamma^+])[\rho^{-1}_{\Gamma,A[\gamma]}] \arrow[r, "\mathtt{p}"] & F\Gamma.F(A[\gamma]) \arrow[r, "\mathtt{p}"] & F\Gamma
\end{tikzcd}
$$

where all arrows not explicitely displayed are identities.

Proof. The only non-trivial equality to prove is that the morphism preserves evaluation, *i.e.* that

$$\mathtt{ev} \circ \mathbf{T}(\mathtt{p})(\Pi(\theta^{-1}_{A,\gamma}, \mathbf{T}(\rho^{-1}_{\Gamma,A[\gamma]})(\theta_{B,\gamma^+}))) \circ \theta^+_{A,\gamma} = \mathbf{T}(\rho^{-1}_{\Gamma,A[\gamma]})(\theta_{B,\gamma^+}) \circ \theta^+_{A,\gamma} \circ \mathtt{ev}$$

which is a direct (if somewhat intricate) calculation on cwf combinators. Note that both $\mathtt{ev}$ are evaluation morphisms for chosen dependent product diagrams, *i.e.* $\langle \mathtt{p}, \mathrm{app}(\mathtt{q}, \mathtt{q}[\mathtt{p}])\rangle$. $\square$

Finally, the last thing we have to prove is that the upper morphism of the diagram of Lemma 4.22, *i.e.* $\theta_{\Pi(A,B),\gamma}$, induces as well a canonical morphism between dependent product diagrams.

Lemma C.4. *The following is a morphism between two dependent product diagrams:*

$$\begin{tikzcd}
 & F\Gamma.FA[F\gamma].F(\Pi(A,B))[(F\gamma)\circ\mathtt{p}] \arrow[r, "\mathtt{p}^+"] \arrow[dl, "\mathbf{T}((F\gamma)^+)(\mathtt{ev}_{F(\Pi(A,B))})"'] \arrow[d, "\mathtt{p}"] \arrow[dd, bend left, "\mathbf{T}(\mathtt{p})(\theta_{\Pi(A,B),\gamma})\circ\theta^+_{A,\gamma}"] & F\Gamma.F\Pi(A,B)[F\gamma] \arrow[d, "\mathtt{p}"] \arrow[dd, bend left, "\theta_{\Pi(A,B),\gamma}"] \\
F\Gamma.FA[F\gamma].FB[\rho^{-1}_{\Delta,A}][(F\gamma)^+] \arrow[r, "\mathtt{p}"] \arrow[dd, bend right, "\mathbf{T}(\rho^{-1}_{\Gamma,A[\gamma]})(\theta_{B,\gamma^+})\circ\theta^+_{A,\gamma}"'] & F\Gamma.FA[F\gamma] \arrow[r, "\mathtt{p}"] \arrow[dd, bend right, "\theta_{A,\gamma}"'] & F\Gamma \\
 & F\Gamma.F(A[\gamma]).F(\Pi(A,B)[\gamma])[\mathtt{p}] \arrow[r, "\mathtt{p}^+"] \arrow[dl, "\mathtt{ev}_{F(\Pi(A[\gamma],B[\gamma^+]))}"'] \arrow[d, "p"] & F\Gamma.F(\Pi(A,B)[\gamma]) \arrow[d, "\mathtt{p}"] \\
F\Gamma.F(A[\gamma]).F(B[\gamma^+])[\rho^{-1}_{\Gamma,A[\gamma]}] \arrow[r, "\mathtt{p}"] & F\Gamma.F(A[\gamma]) \arrow[r, "\mathtt{p}"] & F\Gamma
\end{tikzcd}$$

where all arrows not explicitely displayed are identities.

Proof. Recall that $\theta_{\Pi(A,B),\gamma}$ can be characterised as the unique morphism between two candidate substitution pullbacks: one computed in $\mathcal{C}$ and transported via F, the other computed in $\mathcal{C}'$. The proof that $\theta_{\Pi(A,B),\gamma}$ respects evaluation consists in redoing the same reasoning, but with the whole dependent product diagram rather than just the type.

The following diagram represents the dependent product diagrams for $\Pi(A,B)$ and $\Pi(A[\gamma], B[\gamma^+])$, along with the morphisms relating them together.

$$\begin{tikzcd}
 & & \Delta.A.\Pi(A,B)[\mathtt{p}] \arrow[rr] \arrow[dd] \arrow[ddll, bend right] & & \Delta.\Pi(A,B) \arrow[dd] & \\
 & & & \Gamma.A[\gamma].\Pi(A,B)[\gamma\circ\mathtt{p}] \arrow[ul] \arrow[rr] \arrow[dd] \arrow[ddll, bend right] & & \Gamma.\Pi(A,B)[\gamma] \arrow[ul] \arrow[dd] \\
\Delta.A.B \arrow[rr] & & \Delta.A \arrow[rr] & & \Delta & \\
 & \Gamma.A[\gamma].B[\gamma^+] \arrow[ul] \arrow[rr] & & \Gamma.A[\gamma] \arrow[ul] \arrow[rr] & & \Gamma \arrow[ul]
\end{tikzcd}$$

The front and back faces are both dependent product diagrams. We now map this diagram to $\mathcal{C}'$ via F, and silently apply the canonical isomorphisms of the pseudo cwf-morphism structure, to obtain (the bottom part of) the following diagram. We do not annotate the arrows to avoid cluttering the diagram too much, but they can be recovered by carefully following the construction of the diagram.

$$
\begin{tikzcd}
 & & & F\Gamma.FA[F\gamma].F(\Pi(A,B))[(F\gamma)\circ \mathtt{p}] \arrow[rr] \arrow[dl] \arrow[dd, dotted, "\mathbf{T}(\mathtt{p})(\theta_{\Pi(A,B),\gamma})"] \arrow[dddd, bend left] \arrow[ddddll, bend right] & & F\Gamma.F(\Pi(A,B))[F\gamma] \arrow[dl] \arrow[dd, dotted, "\theta_{\Pi(A,B),\gamma}"] \arrow[dddd, bend left] \\
 & & F\Delta.FA.F(\Pi(A,B))[\mathtt{p}] \arrow[rr] \arrow[dd] \arrow[ddll, bend right] & & F\Delta.F(\Pi(A,B)) \arrow[dd] & \\
 & & & F\Gamma.FA[F\gamma].F(\Pi(A,B)[\gamma])[\mathtt{p}] \arrow[ul] \arrow[rr] \arrow[dd] \arrow[ddll, bend right] & & F\Gamma.F(\Pi(A,B)[\gamma]) \arrow[ul] \arrow[dd] \\
F\Delta.FA.FB[\rho^{-1}_{\Delta,A}] \arrow[rr] & & F\Delta.FA \arrow[rr] & & F\Delta & \\
 & F\Gamma.F(A[\gamma]).F(B[\gamma^+])[\rho^{-1}_{\Gamma,A[\gamma]}] \arrow[ul] \arrow[rr] & & F\Gamma.F(A[\gamma]) \arrow[ul] \arrow[rr] & & F\Gamma \arrow[ul]
\end{tikzcd}
$$

The top part of the diagram is obtained (up to an obvious isomorphism) by pulling back the dependent product diagram in the back along $F\gamma$. By the universal property of dependent products, the two morphisms from the top dependent product diagram to the one in the back factor uniquely through the two dotted arrows. But for the right hand side one, that exactly means that the condition of Lemma 4.20 is satisfied and that the right hand side dotted map is $\theta_{\Pi(A,B),\gamma}$. Similarly, the left hand side dotted map is necessarily $\mathbf{T}(\mathtt{p})(\theta_{\Pi(A,B),\gamma})$. Therefore, it preserves the evaluation maps, since it was constructed by the universal property of dependent product diagrams.

Annotating the morphisms following their construction, it becomes apparent that the commutation we have proved is exactly the statement of the lemma. □

To wrap things up, we note that by the lemmas above all four morphisms of Lemma 4.22 correspond to canonical morphisms between dependent product diagrams: $i_{A[\gamma],B[\gamma^+]}$ by Lemma C.1, $\mathbf{T}(F\gamma)(i_{A,B})$ by Lemma C.2, $\Pi(\theta^{-1}_{A,\gamma}, \mathbf{T}(\rho^{-1}_{\Gamma,A[\gamma]})(\theta_{B,\gamma^+}))$ by Lemma C.3, and $\theta_{\Pi(A,B),\gamma}$ by Lemma C.4. The corresponding morphisms between dependent product diagrams are composable, and by uniqueness of the universal property it follows that the two paths of the diagram of Lemma 4.22 coincide.