The Biequivalence of Locally Cartesian Closed Categories and Martin-Löf Type Theory with Π, Σ and Extensional Identity Types

Pierre Clairambault, Paris 7
and
Peter Dybjer, Chalmers

Uppsala, 25 August 2010

Categorical logic: key correspondences

- Cartesian closed categories and simply typed lambda calculus
- Hyperdoctrines and first order logic
- Toposes and higher order logic (”intuitionistic type theory”)
- ? and Martin-Löf type theory

Beginning of Seely 1984

"It is well known that for much of the mathematics of topos theory, it is in fact sufficient to use a category $\mathbb{C}$ whose slice categories $\mathbb{C}/A$ are cartesian closed. In such a category, the notion of a 'generalized set', for example an 'A-indexed set', is represented by a morphism $B \to A$ of $\mathbb{C}$, i. e. by an object of $\mathbb{C}/A$. The point about such a category $\mathbb{C}$ is that $\mathbb{C}$ is a $\mathbb{C}$-indexed category, and more, is a hyperdoctrine, so that it has a full first order logic associated with it. This logic has some peculiar aspects. For instance, the types are the objects of $\mathbb{C}$ and terms are the morphisms of $\mathbb{C}$. For a given type A, the predicates with a free variable of type A are morphisms into A, and 'proofs' are morphisms over A. We see here a certain 'ambiguity' between the notions of type, predicate, and term, of object and proof: a term of type A is a morphism into A, which is a predicate over A; a morphism $1 \to A$ can be viewed either as an object of type A or as a proof of the proposition A."

Display maps

The morphism $B \to A$ is called a *display map* when it represents an A-indexed set. Terminology introduced by Taylor 1985.

Beginning of Seely 1984, continued

”For a long time now, it has been conjectured that the logic of such categories is given by the type theory of Martin-Löf [5], since one of the features of Martin-Löf’s type theory is that it formalizes ’ambiguities’ of this sort. However, to the best of my knowledge, no one has ever worked out the details of this relationship, and when the question again arose in the McGill Categorical Logic Seminar in 1981-82, it was felt that making this precise was long overdue.”

Seely's conjecture

R. Seely (1984), Locally cartesian closed categories and type theory:

> *6.3. THEOREM. The categories* ***ML*** *and* ***LCC*** *are equivalent.*

- **ML** is the category of "Martin-Löf theories" with types $\prod_{x \in A} B[x], \sum_{x \in A} B[x]$, and $I(a, b)$. Note it is *extensional* intuitionistic type theory of Martin-Löf (1979, 1984).
- **LCC** is the category of locally cartesian closed categories.

Locally cartesian closed categories

A category $\mathbb{C}$ is *locally cartesian closed (lccc)* iff either of the following equivalent conditions hold:

- all slice categories $\mathbb{C}/A$ are cartesian closed.
- $\mathbb{C}$ has pullbacks and the functor $f^* : \mathbb{C}/B \to \mathbb{C}/A$ has a right adjoint Π_f for $f : A \to B$. (The left adjoint Σ_f always exists.)

Seely's **LCC** is the category of lcccs and lccc-structure preserving functors.

Martin-Löf theories and their associated categories

- A *Martin-Löf theory M* is a dependent type theory with *I*-(extensional identity types), Σ- and Π-types and given by a set of typed type-valued function constants and a set of typed term-valued constants.
- The *category* $C(M)$ *associated with M* has types as objects and arrows with source A and target B are terms of type $A \to B$.

$C(M)$ is an lccc

Similar to showing that the category of sets is an lccc.
For example, pullbacks

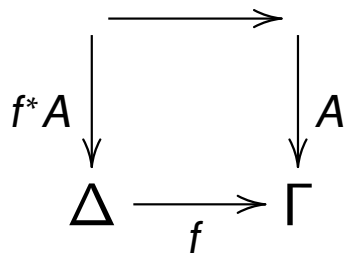
$$\begin{array}{ccc} A \times_C B & \longrightarrow & B \\ \downarrow & & \downarrow g \\ A & \xrightarrow[f]{} & C \end{array}$$

can be defined by

$$A \times_C B = (\Sigma x : A)(\Sigma y : B)\mathrm{I}_C(f(x), g(y))$$

Martin-Löf theory and lccc - correspondences

- Contexts are objects of $\mathbb{C}$.
- Types in context Γ are objects of the slice category $\mathbb{C}/\Gamma$
- Terms of type A are sections of A.
- Type substitution is pullback:



- I-types are equalizers
- Σ-types are (special cases of) left adjoints Σ_f
- Π-types are (special cases of) right adjoints Π_f

Curien

P.-L. Curien (1993), Substitution up to isomorphism:

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

Curien, continued

In the general case, we have to show that the isomorphisms between types arising from substitution are coherent in a sense familiar to category theorists. Due to this coherence problem at the level of types, we are led to:

- *switch to a syntax where substitutions are explicitly present (in traditional presentations substitution is a meta-operation, defined by induction);*
- *include type equality judgements in this modified syntax: we consider here only equalities describing the stepwise performance as substitution.*

Curien, continued

... To our knowledge, the work presented here is the first solution to this problem, which, until very recently, had not even been clearly identified, mainly due to an emphasis on interesting mathematical models rather than on syntactic issues.

Hofmann

M. Hofmann (1994), On the interpretation of type theory in locally cartesian closed categories:

> *Seely argues that substitution should be interpreted as a pullback, so that the interpretation of* $\tau[x := M]$ *becomes the pullback of* τ *along M. ...*
> *The subtle flaw of this idea is that the interpretation of* $\tau[x := M]$ *is already fixed by the clauses of the interpretation, and there is no reason why it should be equal to the chosen pullback of* τ *along M.*
>
> *...*
>
> *Unfortunately, however, it seems impossible to endow an arbitrary lccc with a pullback operation which would satisfy these coherence requirements.*

Hofmann, continued

We show how to construct a model of dependent type theory (category with attributes) from a locally cartesian closed category (lccc).

...

The method we use is a very general procedure due to Bénabou which turns an arbitrary fibration into a split fibration. Our contribution consists of the observation that the cwa obtained thus has not merely a split substitution operation, but is closed under all type formers the original lccc supported. In particular the resulting cwa has Π*-types,* Σ*-types, and (extensional) identity types.*

Bénabou's construction

- Types over Γ are not interpreted as arrows into Γ (display maps), but as functions which map an arrow $\gamma : \Delta \to \Gamma$ into an arrow over Δ. Dependent types are not "display maps", but "families of display maps", one for each substitution instance.
- This is done functorially. Types are interpreted as *functorial families*; they do not only map objects but also arrows of the slice category $\mathbb{C}/\Gamma$. Formally, functorial families are functors $\overrightarrow{A} : \mathbb{C}/\Gamma \to \mathbb{C}^{\to}$ such that $cod \circ \overrightarrow{A} = dom$, which map arrows of $\mathbb{C}/\Gamma$ to pullback squares.
- The technique is reminiscent of the use of presheaf categories for solving coherence problem and in normalization by evaluation (Gordon, Power, Street's proof of MacLane's coherence theorem; Altenkirch, Hofmann, Streicher, and Čubrić, Dybjer, Scott's approach to nbe).

Are **ML** and **LCC** equivalent?

Curien and Hofmann only show how to interpret Martin-Löf theories in lcccs, not that such interpretations give rise to an equivalence of categories, as Seely claimed.
Hofmann conjectured:

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

Giving a precise formulation and proof of this conjecture is the topic of this talk.

Two biequivalences

We shall use cwfs to define an analogue of Seely's category of Martin-Löf theories. The theorem becomes

- **LCC** and $\mathbf{CwF}_{\mathrm{dem}}^{\mathrm{I_{ext}}\Sigma\Pi}$ (democratic cwfs which support extensional identity types, Σ- and Π-types) are biequivalent 2-categories

In fact, we can remove Π-types on the type theory side and the right adjoints on the category side and get the following theorem

- **FL** and $\mathbf{CwF}_{\mathrm{dem}}^{\mathrm{I_{ext}}\Sigma}$ are biequivalent.

where **FL** is the 2-category of categories with finite limits ("left exact categories").

We will focus on the latter. It is essentially as difficult to prove as the first.

What is biequivalence?

Notions of ”abstractly the same”:

- equality of elements (and arrows in a category)
- isomorphism of sets (and objects in a category)
- equivalence of categories
- biequivalence of bicategories
- etc

What is biequivalence?

We need to define

- bicategory (special case of 2-category suffices here)
- weak functor between bicategories
- strong transformation of weak functors
- (invertible) modification of strong transformations

Weak functors

A *weak functor* between 2-categories $\mathcal{B}$ and $\mathcal{B}'$ is a pair (F, ϕ), where $F = (F_0, F_1)$

- $F_0 : \mathcal{B}_0 \to \mathcal{B}'_0$ is a function on 0-cells.
- F_1 is a family of functors

$$F_{1,A,B} : \mathcal{B}(A, B) \to \mathcal{B}'(F_0 A, F_0 B)$$

where $A, B \in \mathcal{B}_0$.

- F_1 preserves identity and composition up to isomorphism. This means that there is an isomorphism (a 2-cell)

$$\phi_A : 1_{F_0 A} \to F_1 1_A$$

in the category $\mathcal{B}'(F_0 A, F_0 A)$ for each A, and moreover an isomorphism

$$\phi_{f,g} : F_1 g \circ F_1 f \to F_1 (g \circ f)$$

in the category $\mathcal{B}'(F_0 A, F_0 C)$ for each $f : \mathcal{B}_1(A, B)$ and $g : \mathcal{B}_1(B, C)$.

Strong transformation

A *strong transformation* between weak functors $(F, \phi), (G, \psi) : \mathcal{B} \to \mathcal{B}'$ is a family of 1-cells

$$\eta_A : \mathcal{B}_1(FA, GA)$$

for $A \in \mathcal{B}_0$, which is weakly natural in the sense that the naturality square commutes up to a natural isomorphism:

$$\eta_f : \mathcal{B}'_2(Ff \circ \eta_A, \eta_B \circ Gf)$$

for each $f : \mathcal{B}_1(A, B)$, satisfying some further conditions with respect to 1 and $\circ$ (not spelled out).

(There is another notion of *weak* transformation where η_f is not required to be iso.)

Modifications

A *modification* between strong transformations $\eta, \theta : (F, \phi) \to (G, \psi)$ is a family of 2-cells

$$m_A : \mathcal{B}_2(\eta_A, \theta_A)$$

satisfying a condition analogous to *naturality* for natural transformations.

Biequivalence

The weak functors $(F, \phi) : \mathcal{B} \to \mathcal{B}'$ and $(G, \psi) : \mathcal{B}' \to \mathcal{B}$ form a *biequivalence* provided $1_{\mathcal{B}} \sim G \circ F \in [\mathcal{B}, \mathcal{B}]$ and $F \circ G \sim 1_{\mathcal{B}'} \in [\mathcal{B}', \mathcal{B}']$ are equivalences inside the 2-categories $[\mathcal{B}, \mathcal{B}]$ and $[\mathcal{B}', \mathcal{B}']$ of weak functors, strong transformations, and modifications.

Proving the biequivalence

We need to provide the following data (and check the appropriate properties):

- **FL**: the 2-category of left exact (lex) categories, lex functors, and natural transformations.
- $\mathbf{CwF}^{\mathrm{I}_{ext}\Sigma}_{\mathrm{dem}}$: the 2-category of cwfs (with Σ-types, extensional identity types, and democracy), weak cwf-morphisms, and cwf-transformations.
- $U : \mathbf{CwF}^{\mathrm{I}_{ext}\Sigma}_{\mathrm{dem}} \to \mathbf{FL}$ is a forgetful 2-functor
- $H : \mathbf{FL} \to \mathbf{CwF}^{\mathrm{I}_{ext}\Sigma}_{\mathrm{dem}}$ is a weak functor based on the Bénabou-Hofmann construction.
- $\eta : 1 \to HU$ and $\epsilon : HU \to 1$: strong transformations, which are inverses wrt invertible modifications ϕ, ψ. This shows that H and U give rise to a biequivalence.

A corrected version of Seely's conjecture

We can extend the above constructions to the case where we add Π-types to cwfs and right adjoints Π_f to lex categories:

THEOREM. The 2-categories $\mathbf{CwF}^{\mathrm{I}_{\mathrm{ext}}\Sigma\Pi}_{\mathrm{dem}}$ and **LCC** are biequivalent.

Just use Seely's technique for relating Π-types and right adjoints in lcccs.

Categories with families (cwfs)

- C, a *category of contexts*. Its objects are called *contexts* and its morphisms are called *substitutions*.
- $T : C^{op} \to \mathbf{Fam}$, a functor where the

  object part maps a context Γ to the family of sets of *terms* $\{a \mid \Gamma \vdash a : A\}$ indexed by the set of *types* $\{A \mid \Gamma \vdash A \ type\}$ in Γ.

  arrow part maps a substitution γ to a pair of functions which perform substitution of γ in types and terms respectively. We write $A[\gamma]$ for *substitution* of γ in a type A and $a[\gamma]$ for *substitution* of γ in the term a.
- A *terminal object* [ ] of C called the *empty context*. The unique arrow $\langle\rangle$ into [ ] is the *empty substitution*.
- A *context comprehension* operation which to an object Γ of C and a type A in Γ associates four components ...

Context comprehension

An operation which to an object Γ of C and a type A in Γ associates four components:

context extension: an object $\Gamma; A$ of C;

weakening: a morphism $\mathrm{p}_{\Gamma,A} : \Gamma; A \to \Gamma$ of C - the *first projection*

assumption: a term $\mathrm{q}_{\Gamma,A} \in \Gamma; A \vdash A[\mathrm{p}_{\Gamma,A}]$ - the *second projection*

substitution extension: for each object Δ in C, morphism $\gamma : \Delta \to \Gamma$, and term $a \in \Delta \vdash A[\gamma]$, there is a unique morphism $\theta = \langle \gamma, a \rangle : \Delta \to \Gamma; A$, such that $p_{\Gamma,A} \circ \theta = \gamma$ and $q_{\Gamma,A}[\theta] = a$. This is the *universal property* of context comprehension.

Cf Lawvere (1970), Equality in hyperdoctrines and the comprehension schema.

The generalized algebraic theory of categories

Sort symbols:

$$\mathrm{Cxt}\ \mathrm{sort}$$

$$\frac{\Delta, \Gamma : \mathrm{Cxt}}{\Delta \to \Gamma\ \mathrm{sort}}$$

Operator symbols:

$$\frac{\Theta, \Delta, \Gamma : \mathrm{Cxt} \qquad \gamma : \Delta \to \Gamma \qquad \delta : \Theta \to \Delta}{\gamma \circ \delta : \Theta \to \Gamma}$$

$$\frac{\Gamma : \mathrm{Cxt}}{\mathrm{id} : \Gamma \to \Gamma}$$

Equations:

$$\begin{aligned} (\gamma \circ \delta) \circ \theta &= \gamma \circ (\delta \circ \theta) \\ \mathrm{id} \circ \gamma &= \gamma \\ \gamma \circ \mathrm{id} &= \gamma \end{aligned}$$

Rules for family-valued functors

Sort symbols:

$$\frac{\Gamma : \mathrm{Cxt}}{\mathrm{Type}(\Gamma)\ \mathrm{sort}}$$

$$\frac{\Gamma : \mathrm{Cxt} \qquad A : \mathrm{Type}(\Gamma)}{\Gamma \vdash A\ \mathrm{sort}}$$

Operator symbols:

$$\frac{\Delta, \Gamma : \mathrm{Cxt} \qquad A : \mathrm{Type}(\Gamma) \qquad \gamma : \Delta \to \Gamma}{A[\gamma] : \mathrm{Type}(\Delta)}$$

$$\frac{\Delta, \Gamma : \mathrm{Cxt} \quad A : \mathrm{Type}(\Gamma) \quad a : \Gamma \vdash A \quad \gamma : \Delta \to \Gamma}{a[\gamma] : \Delta \vdash A[\gamma]}$$

Equations:

$$\begin{aligned} A[(\gamma \circ \delta)] &= A[\gamma][\delta] \\ A[\mathrm{id}] &= A \\ a[(\gamma \circ \delta)] &= a[\gamma][\delta] \end{aligned}$$

Rules for the terminal object

Operator symbols:

$$[\,] : \mathrm{Cxt}$$

$$\frac{\Gamma : \mathrm{Cxt}}{\langle\rangle : \Gamma \to [\,]}$$

Equations

$$\begin{aligned} \langle\rangle \circ \gamma &= \langle\rangle \\ \mathrm{id} &= \langle\rangle \end{aligned}$$

Rules for context comprehension

Operator symbols:

$$\frac{\Gamma : \mathrm{Cxt} \qquad A : \mathrm{Type}(\Gamma)}{\Gamma; A : \mathrm{Cxt}}$$

$$\frac{\Delta, \Gamma : \mathrm{Cxt} \quad A : \mathrm{Type}(\Gamma) \quad \gamma : \Delta \to \Gamma \quad a : \Delta \vdash A[\gamma]}{\langle \gamma, a \rangle : \Delta \to \Gamma; A}$$

$$\frac{\Gamma : \mathrm{Cxt} \qquad A : \mathrm{Type}(\Gamma)}{\mathrm{p} : \Gamma; A \to \Gamma}$$

$$\frac{\Gamma : \mathrm{Cxt} \qquad A : \mathrm{Type}(\Gamma)}{\mathrm{q} : \Gamma; A \vdash A[\mathrm{p}]}$$

Equations:

$$\begin{aligned} \mathrm{p} \circ \langle \gamma, a \rangle &= \gamma \\ \mathrm{q}[\langle \gamma, a \rangle] &= a \\ \langle \delta, a \rangle \circ \gamma &= \langle \delta \circ \gamma, a[\gamma] \rangle \end{aligned}$$

Cwf with Σ-types (including surjective pairing)

- *Formation.* For $A \in \mathrm{Type}(\Gamma)$ and $B \in \mathrm{Type}(\Gamma \cdot A)$ there is a type $\Sigma(A, B) \in \mathrm{Type}(\Gamma)$,
- *Introduction.* For $a : \Gamma \vdash A$ and $b : \Gamma \vdash B[\langle id, a\rangle]$ there is a term $pair(a, b) : \Gamma \vdash \Sigma(A, B)$,
- *Elimination.* For each $a : \Gamma \vdash \Sigma(A, B)$ there are two terms $\pi_1(a) : \Gamma \vdash A$ and $\pi_2(a) : \Gamma \vdash B[\langle id, \pi_1(a)\rangle]$

such that

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

Extensional identity types

A cwf $(\mathbb{C}, T)$ supports *extensional identity types* if and only if:

- *Formation.* For $A \in \mathrm{Type}(\Gamma)$ and $a, a' : \Gamma \vdash A$, there is a type $I_A(a, a')$.
- *Introduction.* For $a : \Gamma \vdash A$, there is a term $refl_{A,a} : \Gamma \vdash I_A(a, a)$.
- *Equality reflection.* $p : \Gamma \vdash I_A(a, a')$ implies
  - $a = a' : \Gamma \vdash A$
  - $p = refl_{A,a} : \Gamma \vdash I_A(a, a')$.

Further equations:

$$\begin{aligned} I_A(a, a')[\delta] &= I_{A[\delta]}(a[\delta], a'[\delta]) \\ refl_{A,a}[\delta] &= refl_{A[\delta],a[\delta]} \end{aligned}$$

Rules for Π

- *Formation.* For $A \in \mathrm{Type}(\Gamma)$ and $B \in \mathrm{Type}(\Gamma \cdot A)$ there is a type $\Pi(A, B) \in \mathrm{Type}(\Gamma)$.
- *Introduction.* For $b : \Gamma; A \vdash B$ there is a term $\lambda(b) : \Gamma \vdash \Pi(A, B)$.
- *Elimination.* For $c : \Gamma \vdash \Pi(A, B)$ and $a : \Gamma \vdash A$ there is a term $\mathrm{ap}(c, a) : \Gamma \vdash B[\langle \mathrm{id}, a \rangle]$.

Equations:

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

Democracy

A cwf $(\mathbb{C}, T)$ is *democratic* iff for each object Γ of $\mathbb{C}$ there is $\overline{\Gamma} \in \mathrm{Type}([\,])$ and an isomorphism $\Gamma \cong_{\gamma_\Gamma} [\,]\cdot\overline{\Gamma}$.

Cwf-morphisms preserving structure on the nose

A notion of strict cwf-morphism was defined in Dybjer 1996. It requires that all data of a cwf is preserved on the nose. If $(\mathbb{C}, T)$ and $(\mathbb{C}', T')$ are two cwfs, then a "strict" cwf-morphism is a pair (F, σ), where

- $F : \mathbb{C} \to \mathbb{C}'$ is a functor and
- $\sigma : T \to T'F$ is a natural transformation between family-valued functors which preserves terminal object and context comprehension on the nose.

Weak cwf-morphism

Here we need a weak version where the data of a cwf is only preserved up to isomorphism. The strong transformations η and ϵ will be families of cwf-morphisms but they do not preserve cwf-structure on the nose.
It is not immediate what it means that "types are preserved up to isomorphism" in a cwf since $\mathrm{Type}(\Gamma)$ is only a set, not a category.

The indexed category of types in context

Let $(\mathbb{C}, T)$ be a cwf with $\mathrm{Type}(\Gamma)$ the set of types over Γ. We construct the functor **Type** : $\mathbb{C}^{op} \to$ **Cat** as follows:

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

Knowing what it means that two types are isomorphic we can formulate a suitable notion of weak cwf-morphism. (The details are verbose.) Note that we must also specify that the type-constructors are preserved up to isomorphism.

Alternative formulation of democracy

The cwf $(\mathbb{C}, T)$ is democratic iff the canonical functor from **Type**([ ]) to $\mathbb{C}$ is an equivalence of categories.

Cf Seely's formulation, where **ML** is the category of categories **Type**([ ]) of closed types, essentially.

$\mathbf{CwF}^{\mathrm{I}_{ext}\Sigma}_{\mathrm{dem}}$ has finite limits

The forgetful 2-functor $U : \mathbf{CwF}^{\mathrm{I}_{ext}\Sigma}_{\mathrm{dem}} \to \mathbf{FL}$ is defined by

$$U(\mathbb{C}, T) = \mathbb{C}$$

U forgets the types and terms. H rebuilds them from the contexts and substitutions!

- We need to build pullbacks in the base category $\mathbb{C}$ of a democratic cwf which supports extensional identity types and Σ-types.
- We also need to show that weak cwf-morphisms map to finite limit preserving functors.
- And we need to show that cwf-transformations map to natural transformations of finite limit categories. (Trivial.)

Bénabou's construction - definition

The weak functor $H : \mathbf{FL} \to \mathbf{CwF}_{\mathrm{dem}}^{\mathrm{I}_{ext}\Sigma}$ is defined by

$$H(\mathbb{C}) = (\mathbb{C}, T_{\mathbb{C}})$$

where $T_{\mathbb{C}}$ rebuilds types and terms from $\mathbb{C}$ using Bénabou's construction.
It is a *weak functor* rather than a 2-functor, since it only preserves identity and composition of 1-cells up to isomorphism.

Bénabou's construction - definition

A *type* over Γ is a *functorial family*, *i.e.* a functor $\overrightarrow{A} : \mathbb{C}/\Gamma \to \mathbb{C}^{\to}$ such that:

(i) $cod \circ \overrightarrow{A} = dom$

(ii) If $\begin{array}{ccc} \Omega & \xrightarrow{\alpha} & \Delta \\ & {\scriptstyle \delta\alpha}\searrow \quad \swarrow{\scriptstyle \delta} & \\ & \Gamma & \end{array}$ is a morphism in $\mathbb{C}/\Gamma$, $\overrightarrow{A}(\alpha)$ is a pullback square, with the naming convention below:

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

Building a cwf by the Benabou construction

Let $\mathbb{C}$ be a category with terminal object. Then we can define $T_{\mathbb{C}}$:

- types and terms
- type substitution and term substitution

and show that $(\mathbb{C}, T_{\mathbb{C}})$ is a cwf by defining

- context comprehension

which supports

- I-types
- Σ-types

All this was proved by Hofmann (1994) using cwas rather than cwfs. In addition we get a *democratic* cwf.

Proving biequivalence

Since $UH = 1$ it suffices to prove that the following strong transformations between weak functors:

$$1 \underset{\epsilon}{\overset{\eta}{\rightleftarrows}} HU$$

are inverse up to invertible modifications. Since $HU(\mathbb{C}, T) = (\mathbb{C}, T^{\mathbb{C}})$, this amounts to proving that

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

is an equivalence of cwfs: that is η and ϵ are weak cwf-morphisms, which are inverse up to invertible modifications.

η: from types to families of display maps

How to get from a type $A \in \mathrm{Type}(\Gamma)$ to the corresponding functorial family $\overrightarrow{A}$ (display map with its substitution behaviour). It's the obvious definition:

$$\begin{aligned} \overrightarrow{A}(\delta) &= \mathrm{p}_{A[\delta]} \\ \overrightarrow{A}(\delta, \gamma) &= \langle \gamma \mathrm{p}, \mathrm{q} \rangle \end{aligned}$$

ϵ: from families of display maps to types

Given a functorial family $\overrightarrow{A}$ over Γ we do the following steps.

- Instantiate to get a display map $\overrightarrow{A}(id) : \Delta \to \Gamma$.
- Use democracy to get $f : [\,]\cdot\overline{\Delta} \vdash \overline{\Gamma}$.
- Build the corresponding type in $\mathrm{Type}([\,]\cdot\overline{\Gamma})$:

$$\Sigma y : \overline{\Delta}.I(f(y), x) \quad (x : \overline{\Gamma})$$

or using cwf-combinators:

$$\Sigma(\overline{\Delta}[\langle\rangle], I(f[\langle\langle\rangle; \mathrm{q}\rangle], \mathrm{q}[\mathrm{p}]))$$

- Use democracy to build the corresponding type in $\mathrm{Type}(\Gamma)$.

Conclusion

- Seely's result is "morally" correct, but
  - the proof is wrong
  - the formulation is wrong (equivalence rather than biequivalence)
- Hofmann's suggestion to consider a "suitable 2-categorical sense" works out, but it was considerable work
  - not only to prove it (many intricate calculations with cwf-combinators)
  - but even to formulate it: what is a good notion of "interpretation of Martin-Löf theories preserving structure up to isomorphism"?

Internal languages

In the end we can conclude that (assuming democracy)

- Martin-Löf type theory with extensional identity types and Σ and Π is the ”internal language of lcccs”.
- Martin-Löf type theory with extensional identity types and Σ is the ”internal language of left exact categories”.