On Setoid Models of Type Theory (work in progress)

Erik Palmgren
Department of Mathematics
Stockholm University

October 11, 2012 [1]

[1] Part of this material was presented at the logic seminar in Stockholm March, 2012.

Models of type theory

One standard framework for interpreting dependent type theory is a category with attributes (Cartmell) or equivalently category with families (Dybjer).

- This modeling usually takes place in set theory.
- An important example is Hofmann's model (Hofmann 1997), which is built from the syntax and judgements of intensional type theory, and is used to translate proofs in extensional type theory into proofs in the intensional theory.
- However, it is of interest to do the modeling in type theory itself, e.g. for the purpose of formal verification, and for foundational reasons. Some work has been done by Dybjer 1995 and onwards.

Models of type theory

One standard framework for interpreting dependent type theory is a category with attributes (Cartmell) or equivalently category with families (Dybjer).

- This modeling usually takes place in set theory.
- An important example is Hofmann's model (Hofmann 1997), which is built from the syntax and judgements of intensional type theory, and is used to translate proofs in extensional type theory into proofs in the intensional theory.
- However, it is of interest to do the modeling in type theory itself, e.g. for the purpose of formal verification, and for foundational reasons. Some work has been done by Dybjer 1995 and onwards.

Models of type theory

One standard framework for interpreting dependent type theory is a category with attributes (Cartmell) or equivalently category with families (Dybjer).

- This modeling usually takes place in set theory.
- An important example is Hofmann's model (Hofmann 1997), which is built from the syntax and judgements of intensional type theory, and is used to translate proofs in extensional type theory into proofs in the intensional theory.
- However, it is of interest to do the modeling in type theory itself, e.g. for the purpose of formal verification, and for foundational reasons. Some work has been done by Dybjer 1995 and onwards.

Definition

1. A category with attributes (cwa) consists of the data
   (a) A category $\mathcal{C}$ with a terminal object 1.
   This is the called the category of contexts and substitutions.
   (b) A functor $T : \mathcal{C}^{\mathrm{op}} \longrightarrow \mathrm{Set}$.
   This functor is intended to assign to each context Γ a set $T(\Gamma)$ of types in the context and tells how substitutions act on these types.
   For $f : B \longrightarrow \Gamma$ and $\sigma \in T(\Gamma)$ we write
   $$\sigma\{f\} \text{ for } T(f)(\sigma).$$
   (c) For each $\sigma \in T(\Gamma)$, an object $\Gamma.\sigma$ in $\mathcal{C}$ and a morphism
   $$\mathrm{p}(\sigma) = \mathrm{p}_\Gamma(\sigma) : \Gamma.\sigma \longrightarrow \Gamma \text{ in } \mathcal{C}.$$
   This tells that each context can be extended by a type in the context, and that there is a projection from the extended context to the original one.

Definition

1. A category with attributes (cwa) consists of the data

(a) A category $\mathcal{C}$ with a terminal object 1.
This is the called the category of contexts and substitutions.

(b) A functor $T : \mathcal{C}^{\mathrm{op}} \longrightarrow \mathrm{Set}$.
This functor is intended to assign to each context Γ a set $T(\Gamma)$ of types in the context and tells how substitutions act on these types.
For $f : B \longrightarrow \Gamma$ and $\sigma \in T(\Gamma)$ we write

$$\sigma\{f\} \text{ for } T(f)(\sigma).$$

(c) For each $\sigma \in T(\Gamma)$, an object $\Gamma.\sigma$ in $\mathcal{C}$ and a morphism

$$\mathrm{p}(\sigma) = \mathrm{p}_\Gamma(\sigma) : \Gamma.\sigma \longrightarrow \Gamma \text{ in } \mathcal{C}.$$

This tells that each context can be extended by a type in the context, and that there is a projection from the extended context to the original one.

Definition

1. A category with attributes (cwa) consists of the data
   (a) A category $\mathcal{C}$ with a terminal object 1.
   This is the called the category of contexts and substitutions.
   (b) A functor $T : \mathcal{C}^{\mathrm{op}} \longrightarrow \mathrm{Set}$.
   This functor is intended to assign to each context Γ a set $T(\Gamma)$ of types in the context and tells how substitutions act on these types.
   For $f : B \longrightarrow \Gamma$ and $\sigma \in T(\Gamma)$ we write
   $$\sigma\{f\} \text{ for } T(f)(\sigma).$$
   (c) For each $\sigma \in T(\Gamma)$, an object $\Gamma.\sigma$ in $\mathcal{C}$ and a morphism
   $$\mathrm{p}(\sigma) = \mathrm{p}_\Gamma(\sigma) : \Gamma.\sigma \longrightarrow \Gamma \text{ in } \mathcal{C}.$$
   This tells that each context can be extended by a type in the context, and that there is a projection from the extended context to the original one.

(d) The final datum tells how substitutions interact with context extensions: For each $f : B \longrightarrow \Gamma$ and $\sigma \in T(\Gamma)$, there is a morphism $\mathrm{q}(f,\sigma) = \mathrm{q}_\Gamma(f,\sigma) : B.(T(f)(\sigma)) \longrightarrow \Gamma.\sigma$ in $\mathcal{C}$ such that

$$\begin{tikzcd}
B.(\sigma\{f\}) \arrow[r, "\mathrm{q}(f{,}\sigma)"] \arrow[d, "\mathrm{p}(\sigma\{f\})"'] & \Gamma.\sigma \arrow[d, "\mathrm{p}(\sigma)"] \\
B \arrow[r, "f"'] & \Gamma
\end{tikzcd}$$

is a pullback, and furthermore

(d.1) $\mathrm{q}(1_\Gamma, \sigma) = 1_{\Gamma.\sigma}$

(d.2) $\mathrm{q}(f \circ g, \sigma) = \mathrm{q}(f,\sigma) \circ \mathrm{q}(g, \sigma\{f\})$ for $A \xrightarrow{g} B \xrightarrow{f} \Gamma$.

(d) The final datum tells how substitutions interact with context extensions: For each $f : B \longrightarrow \Gamma$ and $\sigma \in T(\Gamma)$, there is a morphism $\mathsf{q}(f,\sigma) = \mathsf{q}_\Gamma(f,\sigma) : B.(T(f)(\sigma)) \longrightarrow \Gamma.\sigma$ in $\mathcal{C}$ such that

$$\begin{tikzcd}
B.(\sigma\{f\}) \arrow[r, "\mathsf{q}(f,\sigma)"] \arrow[d, "\mathsf{p}(\sigma\{f\})"'] & \Gamma.\sigma \arrow[d, "\mathsf{p}(\sigma)"] \\
B \arrow[r, "f"'] & \Gamma
\end{tikzcd}$$

is a pullback, and furthermore

(d.1) $\mathsf{q}(1_\Gamma, \sigma) = 1_{\Gamma.\sigma}$

(d.2) $\mathsf{q}(f \circ g, \sigma) = \mathsf{q}(f,\sigma) \circ \mathsf{q}(g, \sigma\{f\})$ for $A \xrightarrow{g} B \xrightarrow{f} \Gamma$.

(d) The final datum tells how substitutions interact with context extensions: For each $f : B \longrightarrow \Gamma$ and $\sigma \in T(\Gamma)$, there is a morphism $\mathsf{q}(f,\sigma) = \mathsf{q}_\Gamma(f,\sigma) : B.(T(f)(\sigma)) \longrightarrow \Gamma.\sigma$ in $\mathcal{C}$ such that

$$\begin{tikzcd} B.(\sigma\{f\}) \arrow[r, "\mathsf{q}(f,\sigma)"] \arrow[d, "\mathsf{p}(\sigma\{f\})"'] & \Gamma.\sigma \arrow[d, "\mathsf{p}(\sigma)"] \\ B \arrow[r, "f"'] & \Gamma \end{tikzcd}$$

is a pullback, and furthermore

(d.1) $\mathsf{q}(1_\Gamma,\sigma) = 1_{\Gamma.\sigma}$

(d.2) $\mathsf{q}(f \circ g,\sigma) = \mathsf{q}(f,\sigma) \circ \mathsf{q}(g,\sigma\{f\})$ for $A \xrightarrow{g} B \xrightarrow{f} \Gamma$.

Example:

In set theory (ZF or CZF) we may construct a cwa from any set U which contains a singleton set and is Σ-*closed* in the sense that if $A \in U$ and $F : A \longrightarrow U$ is any function then the following Σ-set belongs to U

$$\Sigma_{x \in A} F(x) = \{\langle x, y \rangle : x \in A, y \in F(x)\}.$$

Then we can take $\mathcal{C}$ to be the full subcategory of sets with objects in U. It is small. Moreover

$$T =_{\mathrm{def}} \mathrm{Set}(\cdot, U) : \mathcal{C}^{\mathrm{op}} \longrightarrow \mathrm{Set}.$$

For $\Gamma \in \mathcal{C}$ and $\sigma \in T(\Gamma)$

$$\Gamma.\sigma =_{\mathrm{def}} \Sigma_{x \in \Gamma} \sigma(x).$$

Example (cont): The set-theoretic interpretation of the pullback diagram in (d) is then

$$\begin{tikzcd}
\Sigma_{y\in B}.\sigma(f(y)) \arrow[r, "\mathrm{q}"] \arrow[d, "\mathrm{p}"'] & \Sigma_{x\in\Gamma}.\sigma(x) \arrow[d, "\mathrm{p}"] \\
B \arrow[r, "f"'] & \Gamma
\end{tikzcd}$$

where $\mathrm{p}(\langle u, v\rangle) = u$ and $\mathrm{q}(\langle y, s\rangle) = \langle f(y), s\rangle$.

By assuming that the universe U is closed under further constructions one can verify axioms for type theoretic constructions like Π, W, I-types etc.

Indirect model

In view of the fact that intensional Martin-Löf type theory interprets the universe V of CZF (Aczel 1978, 1986) we get an indirect intepretation of the extensional M-L type theory in the intensional one.

Aczel's interpretation has been formalized in various proof assistants: LEGO (N.P. Mendler , 1990), Agda 1 (M.Takeyama mid 1990s), Coq (P. and Wilander 2011). In the latter interpretation a full faithful functor

$$V \longrightarrow \mathrm{Setoids}$$

is explicitly constructed.

However, we are interested in more direct interpretations.

Setoids

In type theories the notion of set is usually understood in the sense of Bishop as a type together with an equivalence relation, also called a setoid

$$A = (|A|, =_A)$$

where $|A|$ is a type and $=_A$ is an equivalence relation on $|A|$.

An extensional function $f : A \longrightarrow B$ between setoids is a function $|A| \longrightarrow |B|$ which respects the equivalence relations, i.e.

$$(\forall x, y : |A|)[x =_A y \Longrightarrow f(x) =_B f(y)]$$

Two such functions f and g are extensionally equal ($f =_{\mathrm{ext}} g$) if $(\forall x : |A|)(f(x) =_B g(x))$.

Setoids

In type theories the notion of set is usually understood in the sense of Bishop as a type together with an equivalence relation, also called a setoid

$$A = (|A|, =_A)$$

where $|A|$ is a type and $=_A$ is an equivalence relation on $|A|$.
An extensional function $f : A \longrightarrow B$ between setoids is a function $|A| \longrightarrow |B|$ which respects the equivalence relations, i.e.

$$(\forall x, y : |A|)[x =_A y \Longrightarrow f(x) =_B f(y)]$$

Two such functions f and g are extensionally equal ($f =_{\text{ext}} g$) if $(\forall x : |A|)(f(x) =_B g(x))$.

The setoids and extensional functions in a (constructive) type theory form an e-category **Setoids** and its properties can be described in the same abstract way as the category **Sets** in a (constructive) set theory.

The properties of the category reflects the possibilities and limitations of constructions in the background theory.

The setoids and extensional functions in a (constructive) type theory form an e-category **Setoids** and its properties can be described in the same abstract way as the category **Sets** in a (constructive) set theory.

The properties of the category reflects the possibilities and limitations of constructions in the background theory.

The notion of e-category is a variant of the standard notion of category, but where no equality relation is required on objects.

An e-category $\mathcal{C}$ consists of a type $\mathrm{Ob}\,\mathcal{C}$ of objects, together with a setoid $\mathcal{C}(A, B)$ of morphisms for every pair of objects A and B. The composition is an extensional function

$$\circ : \mathcal{C}(B, C) \times \mathcal{C}(A, B) \longrightarrow \mathcal{C}(A, C)$$

which satisfies the usual monoid laws.

An e-functor is a functor where the object part is just a function between types. There is no equality of objects to respect.

The notion of e-category is a variant of the standard notion of category, but where no equality relation is required on objects.

An e-category $\mathcal{C}$ consists of a type $\mathrm{Ob}\,\mathcal{C}$ of objects, together with a setoid $\mathcal{C}(A, B)$ of morphisms for every pair of objects A and B. The composition is an extensional function

$$\circ : \mathcal{C}(B, C) \times \mathcal{C}(A, B) \longrightarrow \mathcal{C}(A, C)$$

which satisfies the usual monoid laws.

An e-functor is a functor where the object part is just a function between types. There is no equality of objects to respect.

Families of setoids

In dependent type theories, such as Martin-Löf type theory, the notion of a family of types is fundamental.

$$B(x)\ \mathrm{type} \quad (x : A).$$

But ...

What do we mean by a family of setoids indexed by a setoid?

- A is an index setoid
- B_x setoid for each $x : |A|$
- B_x and $B_{x'}$ should be "equal" if $x =_A x'$

Families of setoids

In dependent type theories, such as Martin-Löf type theory, the notion of a family of types is fundamental.

$$B(x)\ \mathrm{type} \quad (x : A).$$

But ...

What do we mean by a family of setoids indexed by a setoid?

- A is an index setoid
- B_x setoid for each $x : |A|$
- B_x and $B_{x'}$ should be "equal" if $x =_A x'$

Families of setoids

In dependent type theories, such as Martin-Löf type theory, the notion of a family of types is fundamental.

$$B(x)\ \mathrm{type} \quad (x : A).$$

But ...

What do we mean by a family of setoids indexed by a setoid?

- A is an index setoid
- B_x setoid for each $x : |A|$
- B_x and $B_{x'}$ should be "equal" if $x =_A x'$

"Equality" of B_x and B'_x is stated by saying:

$\phi_p : B_x \longrightarrow B_{x'}$ bijection for each proof-object $p : x =_A x'$
The bijections should be "compatible" with the proof objects p.

There are then two principal choices:

(I) proof-irrelevant family: ϕ_p is independent of p

(R) proof-relevant family: ϕ_p may depend on p

The first is the most well-behaved version and is entirely what we expect from set theory.

"Equality" of B_x and B'_x is stated by saying:

$\phi_p : B_x \longrightarrow B_{x'}$ bijection for each proof-object $p : x =_A x'$
The bijections should be "compatible" with the proof objects p.

There are then two principal choices:

(I) proof-irrelevant family: ϕ_p is independent of p

(R) proof-relevant family: ϕ_p may depend on p

The first is the most well-behaved version and is entirely what we expect from set theory.

"Equality" of B_x and B'_x is stated by saying:

$\phi_p : B_x \longrightarrow B_{x'}$ bijection for each proof-object $p : x =_A x'$
The bijections should be "compatible" with the proof objects p.

There are then two principal choices:

(I) proof-irrelevant family: ϕ_p is independent of p

(R) proof-relevant family: ϕ_p may depend on p

The first is the most well-behaved version and is entirely what we expect from set theory.

Proof-irrelevant families

(I) For a proof-irrelevant family we require

(1) $\phi_p =_{\mathrm{ext}} \mathrm{id}_{B_x}$ whenever $p : x =_A x$,

(2) $\phi_q \circ \phi_p =_{\mathrm{ext}} \phi_r$, whenever $p : x =_A y, q : y =_A z, r : x =_A z$. Here $=_{\mathrm{ext}}$ is extensional equality of functions between setoids.

(Compare to definition in Problem 3.2 of Bishop–Bridges 1985.)

From (1) and (2) follows independence of ϕ_p on p

$\phi_p =_{\mathrm{ext}} \phi_r$ for $p, r : x =_A y$

Proof-irrelevant families

(I) For a proof-irrelevant family we require

(1) $\phi_p =_{\mathrm{ext}} \mathrm{id}_{B_x}$ whenever $p : x =_A x$,

(2) $\phi_q \circ \phi_p =_{\mathrm{ext}} \phi_r$, whenever $p : x =_A y, q : y =_A z, r : x =_A z$. Here $=_{\mathrm{ext}}$ is extensional equality of functions between setoids.

(Compare to definition in Problem 3.2 of Bishop–Bridges 1985.)

From (1) and (2) follows independence of ϕ_p on p

$\phi_p =_{\mathrm{ext}} \phi_r$ for $p, r : x =_A y$

Proof-irrelevant families (cont.)

The proof-irrelevant families of setoids over a setoid A correspond exactly to e-functors B from the discrete e-category $A^{\#}$ into **Setoids.**

Thus $B_x = B(x)$ and $\phi_p = B(p)$ in the notation above.

The objects of the e-category $A^{\#}$ is $|A|$, and the setoid of morphisms $A^{\#}(x, y)$ is the type of proofs p in $x =_A y$. Any two proof p, r are identified. The proof object for transitivity gives the composition, reflexivity gives the identity morphism, and symmetry gives an inverse operation.

Proof-relevant families

One drawback of proof-irrelevant families is that there are too few of them. However every family of $B(x)$ of types over a type $x : A$ gives rise to a proof-relevant family of (projective) setoids

$$\underline{B(x)} = (B(x), Id_{B(x)}(\cdot, \cdot))$$

which can be described as a functor

$$\underline{B} : A^g \longrightarrow \mathbf{Setoids}$$

Here $A^g = (A, Id_A(\cdot, \cdot), r, s, t)$ is the groupoid that arises from the type A (Hofmann and Streicher). The morphism part is given by the standard substitution operation associated identity type.

Proof-relevant families

Theorem (P.): The following are equivalent:

(a) The type A satisfies UIP, i.e.$\forall a : A, \forall p : Id_A(a, a), Id(p, r(a))$

(b) Streicher's K-axiom holds for A

(c) For every family of types B over A, $\underline{B}$ is a proof irrelevant family.

Families and categories of setoid

For any proof-irrelevant $F : A^\# \longrightarrow \mathrm{Setoid}$ we may construct an ordinary category $\mathcal{C} = \mathcal{C}(F)$ whose objects are $\mathcal{C}_0 = A$ and whose arrows $\mathcal{C}_1$ is the setoid consisting of (a, b, f) where $a, b \in A$ and $f : F(a) \longrightarrow F(b)$ is an extensional function, and where the equality $(a, b, f) \sim (a', b', f')$ holds iff there are $p : a =_A a'$ and $q : b =_A b'$ so that the diagram commutes:

$$\begin{tikzcd} F(a) \arrow[r, "f"] \arrow[d, "F(p)"'] & F(b) \arrow[d, "F(q)"] \\ F(a') \arrow[r, "g"'] & F(b') \end{tikzcd} .$$

The proof-irrelevance comes in when verifying that composition is extensional.

Families and categories of setoid

In case $F : A^g \longrightarrow \mathrm{Setoid}$ is a proof-relevant family the $\mathcal{C}_0$, $\mathcal{C}_1$ and $\mathcal{C}_2$ get however natural groupoid structure, and $\mathcal{C}$ can be regarded as a category object in a higher e-category of groupois.

The collection of small setoids (i.e. belonging to some type-theoretic universe) will naturally be such an "almost category".

It seems therefore difficult to use the original cwa definition inside type theory without first reconstructing the universe of setoids.

Henceforth: family of setoids = proof-irrelevant family of setoids.

ecwas - cwas in type theory

Referring back to Definition 1, in order for the equations (d.1) and (d.2) to make sense, we need the *object equalities*

$$\Gamma.(\sigma\{1_\Gamma\}) = \Gamma.\sigma \qquad \mathsf{p}(\sigma\{1_\Gamma\}) = \mathsf{p}(\sigma)$$

and

$$A.(\sigma\{f \circ g\}) = A.(\sigma\{f\}\{g\})$$

and moreover

$$\mathsf{p}(\sigma\{f \circ g\}) = \mathsf{p}(\sigma\{f\}\{g\}).$$

They follow from the functoriality of T and by requiring the object equality $\Gamma.\sigma = \Gamma.\sigma'$ and $\mathsf{p}(\sigma) = \mathsf{p}(\sigma')$ whenever $\sigma = \sigma'$.

This notion of cwa is not appropriate for categories $\mathcal{C}$ that lack object equality, like e-categories.

ecwas - cwas in type theory

Referring back to Definition 1, in order for the equations (d.1) and (d.2) to make sense, we need the *object equalities*

$$\Gamma.(\sigma\{1_\Gamma\}) = \Gamma.\sigma \qquad \mathsf{p}(\sigma\{1_\Gamma\}) = \mathsf{p}(\sigma)$$

and

$$A.(\sigma\{f \circ g\}) = A.(\sigma\{f\}\{g\})$$

and moreover

$$\mathsf{p}(\sigma\{f \circ g\}) = \mathsf{p}(\sigma\{f\}\{g\}).$$

They follow from the functoriality of T and by requiring the object equality $\Gamma.\sigma = \Gamma.\sigma'$ and $\mathsf{p}(\sigma) = \mathsf{p}(\sigma')$ whenever $\sigma = \sigma'$.

This notion of cwa is not appropriate for categories $\mathcal{C}$ that lack object equality, like e-categories.

ecwas - cwas in type theory

Referring back to Definition 1, in order for the equations (d.1) and (d.2) to make sense, we need the *object equalities*

$$\Gamma.(\sigma\{1_\Gamma\}) = \Gamma.\sigma \qquad \mathsf{p}(\sigma\{1_\Gamma\}) = \mathsf{p}(\sigma)$$

and

$$A.(\sigma\{f \circ g\}) = A.(\sigma\{f\}\{g\})$$

and moreover

$$\mathsf{p}(\sigma\{f \circ g\}) = \mathsf{p}(\sigma\{f\}\{g\}).$$

They follow from the functoriality of T and by requiring the object equality $\Gamma.\sigma = \Gamma.\sigma'$ and $\mathsf{p}(\sigma) = \mathsf{p}(\sigma')$ whenever $\sigma = \sigma'$.

This notion of cwa is not appropriate for categories $\mathcal{C}$ that lack object equality, like e-categories.

We modify the structure slightly to the setting of e-categories:

Definition 2. An e-category with attributes (ecwa) consists of the following data (a) - (d):

(a) An e-category $\mathcal{C}$ with a terminal object 1.

(b) An e-functor $T : \mathcal{C}^{\mathrm{op}} \longrightarrow \mathrm{Setoids}$.

(c) There is an e-functor $\Delta_\Gamma : T(\Gamma)^\# \longrightarrow \mathcal{C}/\Gamma$. For $\sigma \in \mathrm{Ob}\ T(\Gamma)$, write $\Delta_\Gamma(\sigma) = (\mathsf{p}(\sigma) : \Gamma.\sigma \longrightarrow \Gamma)$.

Recall that $S^\#$ denotes the discrete e-category induced by a setoid S. $\mathcal{C}/\Gamma$ denotes the slice e-category of $\mathcal{C}$ over Γ.

We modify the structure slightly to the setting of e-categories:

Definition 2. An e-category with attributes (ecwa) consists of the following data (a) - (d):

(a) An e-category $\mathcal{C}$ with a terminal object 1.

(b) An e-functor $T : \mathcal{C}^{\mathrm{op}} \longrightarrow \mathrm{Setoids}$.

(c) There is an e-functor $\Delta_\Gamma : T(\Gamma)^\# \longrightarrow \mathcal{C}/\Gamma$. For $\sigma \in \mathrm{Ob}\ T(\Gamma)$, write $\Delta_\Gamma(\sigma) = (\mathsf{p}(\sigma) : \Gamma.\sigma \longrightarrow \Gamma)$.

Recall that $S^\#$ denotes the discrete e-category induced by a setoid S. $\mathcal{C}/\Gamma$ denotes the slice e-category of $\mathcal{C}$ over Γ.

Thus for any proof object t of $\sigma =_{T(\Gamma)} \sigma'$, $\Delta_\Gamma(t) : \Gamma.\sigma \longrightarrow \Gamma.\sigma'$ is an isomorphism such that

$$\begin{tikzcd}
\Gamma.\sigma \arrow[rr, "\Delta_\Gamma(t)"] \arrow[dr, "p(\sigma)"'] & & \Gamma.\sigma' \arrow[dl, "p(\sigma')"] \\
& \Gamma &
\end{tikzcd}$$

commutes. Moreover, $\Delta_\Gamma(t)$ is independent of t and

$$\Delta_\Gamma(t) = 1_{\Gamma.\sigma} \qquad (t \text{ proof of } \sigma =_{T(\Gamma)} \sigma)$$

$$\Delta_\Gamma(s \circ t) = \Delta_\Gamma(s) \circ \Delta_\Gamma(t) \qquad (t \text{ pf. of } \sigma =_{T(\Gamma)} \sigma' \text{ and } s \text{ pf. of } \sigma' =_{T(\Gamma)} \sigma'')$$

(A particular feature of the slice of an e-category is that equalities of objects over the base turn into isomorphism.)

Thus for any proof object t of $\sigma =_{T(\Gamma)} \sigma'$, $\Delta_\Gamma(t) : \Gamma.\sigma \longrightarrow \Gamma.\sigma'$ is an isomorphism such that

$$\begin{tikzcd}
\Gamma.\sigma \arrow[rr, "\Delta_\Gamma(t)"] \arrow[dr, "p(\sigma)"'] & & \Gamma.\sigma' \arrow[dl, "p(\sigma')"] \\
& \Gamma &
\end{tikzcd}$$

commutes. Moreover, $\Delta_\Gamma(t)$ is independent of t and

$$\Delta_\Gamma(t) = 1_{\Gamma.\sigma} \qquad (t \text{ proof of } \sigma =_{T(\Gamma)} \sigma)$$

$$\Delta_\Gamma(s \circ t) = \Delta_\Gamma(s) \circ \Delta_\Gamma(t) \qquad (t \text{ pf. of } \sigma =_{T(\Gamma)} \sigma' \text{ and } s \text{ pf. of } \sigma' =_{T(\Gamma)} \sigma'')$$

(A particular feature of the slice of an e-category is that equalities of objects over the base turn into isomorphism.)

(d) For each $f : B \longrightarrow \Gamma$ and $\sigma \in T(\Gamma)$, there is a morphism $\mathrm{q}(f,\sigma) : B.\sigma\{f\} \longrightarrow \Gamma.\sigma$ in $\mathcal{C}$ such that

$$\begin{tikzcd} B.\sigma\{f\} \arrow[r, "\mathrm{q}(f,\sigma)"] \arrow[d, "\mathrm{p}(\sigma\{f\})"'] & \Gamma.\sigma \arrow[d, "\mathrm{p}(\sigma)"] \\ B \arrow[r, "f"'] & \Gamma \end{tikzcd}$$

is a pullback, and moreover these morphisms satisfy

(d.1) $\mathrm{q}(1_\Gamma, \sigma) \circ \Delta_\Gamma(t) = 1_{\Gamma.\sigma}$ where t is any pf. for $T(1_\Gamma)(\sigma) = \sigma$.

(d.2) $\mathrm{q}(f \circ g, \sigma) \circ \Delta_A(t) = \mathrm{q}(f,\sigma) \circ \mathrm{q}(g, \sigma\{f\})$ for $A \xrightarrow{g} B \xrightarrow{f} \Gamma$ and where t is any pf. for $\sigma\{f \circ g\} =_{TA} \sigma\{f\}\{g\}$.

Note the type correcting isomorphisms $\Delta(t)$.

Further in condition (d), note that if $f = f' : B \longrightarrow \Gamma$, s is a proof of $\sigma = \sigma' \in T(\Gamma)$ and t is a proof of $\sigma\{f\} = \sigma'\{f'\}$, then by the pullback properly,

$$\mathsf{q}(f, \sigma) \circ \Delta_B(t) = \Delta_\Gamma(s) \circ \mathsf{q}(f', \sigma').$$

Interpretation

The definition of ecwas suggests introducing the following judgements about types

- $\Gamma \vdash \sigma\ \mathrm{type}$ meaning $\sigma \in T(\Gamma)$
- $\Gamma \vdash \sigma = \sigma'$ meaning $\sigma =_{T(\Gamma)} \sigma'$ where $\sigma, \sigma' \in T(\Gamma)$.

Define $E(\Gamma, \sigma)$, the elements of σ in the context Γ, to be the setoid of sections of $\mathrm{p}(\sigma) : \Gamma.\sigma \longrightarrow \Gamma$.

Note that if r is a proof for $\sigma =_{T(\Gamma)} \sigma'$, then $M \in E(\Gamma, \sigma)$ implies $\Delta_\Gamma(r) \circ M \in E(\Gamma, \sigma')$.

Interpretation

The definition of ecwas suggests introducing the following judgements about types

- $\Gamma \vdash \sigma \text{ type}$ meaning $\sigma \in T(\Gamma)$
- $\Gamma \vdash \sigma = \sigma'$ meaning $\sigma =_{T(\Gamma)} \sigma'$ where $\sigma, \sigma' \in T(\Gamma)$.

Define $E(\Gamma, \sigma)$, the elements of σ in the context Γ, to be the setoid of sections of $\mathrm{p}(\sigma) : \Gamma.\sigma \longrightarrow \Gamma$.

Note that if r is a proof for $\sigma =_{T(\Gamma)} \sigma'$, then $M \in E(\Gamma, \sigma)$ implies $\Delta_\Gamma(r) \circ M \in E(\Gamma, \sigma')$.

Now assuming a term M always come with an "original" type σ, written as a pair (M,σ), we introduce the further judgements

- $\Gamma \vdash (M,\sigma) : \sigma'$ meaning

  $M \in E(\Gamma,\sigma)$ and $\sigma =_{T(\Gamma)} \sigma'$.

- $\Gamma \vdash (M,\sigma) = (M',\sigma') : \sigma''$ meaning

  $\Gamma \vdash (M,\sigma) : \sigma''$ and $\Gamma \vdash (M,\sigma') : \sigma''$ and that there is a proof r of $\sigma =_{T(\Gamma)} \sigma'$ such that $\Delta_\Gamma(r) \circ M =_{E(\Gamma,\sigma')} M'$.

Now assuming a term M always come with an "original" type σ, written as a pair (M, σ), we introduce the further judgements

- $\Gamma \vdash (M, \sigma) : \sigma'$ meaning

  $M \in E(\Gamma, \sigma)$ and $\sigma =_{T(\Gamma)} \sigma'$.

- $\Gamma \vdash (M, \sigma) = (M', \sigma') : \sigma''$ meaning

  $\Gamma \vdash (M, \sigma) : \sigma''$ and $\Gamma \vdash (M, \sigma') : \sigma''$ and that there is a proof r of $\sigma =_{T(\Gamma)} \sigma'$ such that $\Delta_\Gamma(r) \circ M =_{E(\Gamma, \sigma')} M'$.

Now assuming a term M always come with an "original" type σ, written as a pair (M, σ), we introduce the further judgements

- $\Gamma \vdash (M, \sigma) : \sigma'$ meaning

  $M \in E(\Gamma, \sigma)$ and $\sigma =_{T(\Gamma)} \sigma'$.
- $\Gamma \vdash (M, \sigma) = (M', \sigma') : \sigma''$ meaning

  $\Gamma \vdash (M, \sigma) : \sigma''$ and $\Gamma \vdash (M, \sigma') : \sigma''$ and that there is a proof r of $\sigma =_{T(\Gamma)} \sigma'$ such that $\Delta_\Gamma(r) \circ M =_{E(\Gamma, \sigma')} M'$.

The rules

$$\boxed{\frac{\Gamma \vdash s : \sigma \qquad \Gamma \vdash \sigma = \tau}{\Gamma \vdash s : \tau}}$$

$$\boxed{\frac{\Gamma \vdash s = t : \sigma \qquad \Gamma \vdash \sigma = \tau}{\Gamma \vdash s = t : \tau}}$$

are immediately justified.

Substitution into terms

For $M \in E(\Gamma, \sigma)$ and $f : B \longrightarrow \Gamma$, define $M\{f\} : B \longrightarrow B.\sigma\{f\}$ as the unique morphism (see diagram below) with $\mathrm{p}(\sigma\{f\})) \circ M\{f\} = 1_B$ and $\mathrm{q}(f, \sigma) \circ M\{f\} = M \circ f$. Thus $M\{f\} \in E(B, \sigma\{f\})$.

$$
\begin{tikzcd}
B \arrow[dr, dotted, "M\{f\}"'] \arrow[drr, "M\circ f"] \arrow[ddr, "1"'] & & \\
& B.\sigma\{f\} \arrow[r, "\mathrm{q}(f,\sigma)"'] \arrow[d, "\mathrm{p}(\sigma\{f\})"] & \Gamma.\sigma \arrow[d, "\mathrm{p}(\sigma)"] \\
& B \arrow[r, "f"] & \Gamma
\end{tikzcd}
$$

Stability under substitution

Thus if

$$\Gamma \vdash (M, \sigma) : \sigma'$$

and $f : B \longrightarrow \Gamma$, we have $M\{f\} \in E(B, \sigma\{f\})$ and $\sigma\{f\} =_{T(B)} \sigma'\{f\}$, so

$$B \vdash (M\{f\}, \sigma\{f\}) : \sigma'\{f\}$$

Thus the rule

$$\boxed{\dfrac{\Gamma \vdash s : \sigma' \qquad f : B \longrightarrow \Gamma}{B \vdash s\{f\} : \sigma'\{f\}}}$$

is justified.

Moreover, if

$$\Gamma \vdash (M, \sigma) = (M', \sigma') : \sigma''$$

we have $\Delta_\Gamma(r) \circ M =_{E(\Gamma,\sigma')} M'$ for some proof r of $\sigma =_{T(\Gamma)} \sigma'$. Then

$$\Delta_\Gamma(r) \circ M \circ f = M' \circ f$$

and hence

$$\Delta_\Gamma(r) \circ \mathrm{q}(f, \sigma) \circ M\{f\} = M' \circ f.$$

So by remark above there is a proof s of $T(f)(\sigma) = T(f)(\sigma')$ such that $\Delta_\Gamma(r) \circ \mathrm{q}(f, \sigma) = \mathrm{q}(f, \sigma') \circ \Delta_B(s)$ and hence

$$\mathrm{q}(f, \sigma') \circ \Delta_B(s) \circ M\{f\} = M' \circ f.$$

It follows by uniqueness that $\Delta_B(s) \circ M\{f\} = M'\{f\}$, so indeed

$$B \vdash (M\{f\}, \sigma\{f\}) = (M'\{f\}, \sigma'\{f\}) : \sigma''\{f\}.$$

This justifies also the rule:

$$\frac{\Gamma \vdash s = t : \sigma'' \qquad f : B \longrightarrow \Gamma}{B \vdash s\{f\} = t\{f\} : \sigma''\{f\}}$$

Moreover, if

$$\Gamma \vdash (M, \sigma) = (M', \sigma') : \sigma''$$

we have $\Delta_\Gamma(r) \circ M =_{E(\Gamma,\sigma')} M'$ for some proof r of $\sigma =_{T(\Gamma)} \sigma'$. Then

$$\Delta_\Gamma(r) \circ M \circ f = M' \circ f$$

and hence

$$\Delta_\Gamma(r) \circ \mathrm{q}(f, \sigma) \circ M\{f\} = M' \circ f.$$

So by remark above there is a proof s of $T(f)(\sigma) = T(f)(\sigma')$ such that $\Delta_\Gamma(r) \circ \mathrm{q}(f, \sigma) = \mathrm{q}(f, \sigma') \circ \Delta_B(s)$ and hence

$$\mathrm{q}(f, \sigma') \circ \Delta_B(s) \circ M\{f\} = M' \circ f.$$

It follows by uniqueness that $\Delta_B(s) \circ M\{f\} = M'\{f\}$, so indeed

$$B \vdash (M\{f\}, \sigma\{f\}) = (M'\{f\}, \sigma'\{f\}) : \sigma''\{f\}.$$

This justifies also the rule:

$$\frac{\Gamma \vdash s = t : \sigma'' \qquad f : B \longrightarrow \Gamma}{B \vdash s\{f\} = t\{f\} : \sigma''\{f\}}$$

Moreover, if

$$\Gamma \vdash (M, \sigma) = (M', \sigma') : \sigma''$$

we have $\Delta_\Gamma(r) \circ M =_{E(\Gamma,\sigma')} M'$ for some proof r of $\sigma =_{T(\Gamma)} \sigma'$. Then

$$\Delta_\Gamma(r) \circ M \circ f = M' \circ f$$

and hence

$$\Delta_\Gamma(r) \circ \mathrm{q}(f, \sigma) \circ M\{f\} = M' \circ f.$$

So by remark above there is a proof s of $T(f)(\sigma) = T(f)(\sigma')$ such that $\Delta_\Gamma(r) \circ \mathrm{q}(f, \sigma) = \mathrm{q}(f, \sigma') \circ \Delta_B(s)$ and hence

$$\mathrm{q}(f, \sigma') \circ \Delta_B(s) \circ M\{f\} = M' \circ f.$$

It follows by uniqueness that $\Delta_B(s) \circ M\{f\} = M'\{f\}$, so indeed

$$B \vdash (M\{f\}, \sigma\{f\}) = (M'\{f\}, \sigma'\{f\}) : \sigma''\{f\}.$$

This justifies also the rule:

$$\frac{\Gamma \vdash s = t : \sigma'' \qquad f : B \longrightarrow \Gamma}{B \vdash s\{f\} = t\{f\} : \sigma''\{f\}}$$

Moreover, if

$$\Gamma \vdash (M, \sigma) = (M', \sigma') : \sigma''$$

we have $\Delta_\Gamma(r) \circ M =_{E(\Gamma,\sigma')} M'$ for some proof r of $\sigma =_{T(\Gamma)} \sigma'$. Then

$$\Delta_\Gamma(r) \circ M \circ f = M' \circ f$$

and hence

$$\Delta_\Gamma(r) \circ \mathrm{q}(f, \sigma) \circ M\{f\} = M' \circ f.$$

So by remark above there is a proof s of $T(f)(\sigma) = T(f)(\sigma')$ such that $\Delta_\Gamma(r) \circ \mathrm{q}(f, \sigma) = \mathrm{q}(f, \sigma') \circ \Delta_B(s)$ and hence

$$\mathrm{q}(f, \sigma') \circ \Delta_B(s) \circ M\{f\} = M' \circ f.$$

It follows by uniqueness that $\Delta_B(s) \circ M\{f\} = M'\{f\}$, so indeed

$$B \vdash (M\{f\}, \sigma\{f\}) = (M'\{f\}, \sigma'\{f\}) : \sigma''\{f\}.$$

This justifies also the rule:

$$\frac{\Gamma \vdash s = t : \sigma'' \qquad f : B \longrightarrow \Gamma}{B \vdash s\{f\} = t\{f\} : \sigma''\{f\}}$$

Moreover, if

$$\Gamma \vdash (M, \sigma) = (M', \sigma') : \sigma''$$

we have $\Delta_\Gamma(r) \circ M =_{E(\Gamma,\sigma')} M'$ for some proof r of $\sigma =_{T(\Gamma)} \sigma'$. Then

$$\Delta_\Gamma(r) \circ M \circ f = M' \circ f$$

and hence

$$\Delta_\Gamma(r) \circ \mathsf{q}(f, \sigma) \circ M\{f\} = M' \circ f.$$

So by remark above there is a proof s of $T(f)(\sigma) = T(f)(\sigma')$ such that $\Delta_\Gamma(r) \circ \mathsf{q}(f, \sigma) = \mathsf{q}(f, \sigma') \circ \Delta_B(s)$ and hence

$$\mathsf{q}(f, \sigma') \circ \Delta_B(s) \circ M\{f\} = M' \circ f.$$

It follows by uniqueness that $\Delta_B(s) \circ M\{f\} = M'\{f\}$, so indeed

$$B \vdash (M\{f\}, \sigma\{f\}) = (M'\{f\}, \sigma'\{f\}) : \sigma''\{f\}.$$

This justifies also the rule:

$$\boxed{\dfrac{\Gamma \vdash s = t : \sigma'' \qquad f : B \longrightarrow \Gamma}{B \vdash s\{f\} = t\{f\} : \sigma''\{f\}}}$$

From (Hofmann 1994) we take the following definition, but adapt it in the obvious way to cwas.

A cwa supports Π-types if for $\sigma \in T(\Gamma)$ and $\tau \in T(\Gamma.\sigma)$ there is a type

$$\Pi(\sigma, \tau) \in T(\Gamma),$$

and moreover for every $P \in E(\Gamma.\sigma, \tau)$ there is an element

$$\lambda_{\sigma,\tau}(P) \in E(\Gamma, \Pi(\sigma, \tau)),$$

and furthermore for any $M \in E(\Gamma, \Pi(\sigma, \tau))$ and any $N \in E(\Gamma, \sigma)$ there is an element

$$\mathrm{App}_{\sigma,\tau}(M, N) \in E(\Gamma, \tau\{N\}),$$

such that the following equations hold for any subst. $f : B \longrightarrow \Gamma$:

- (β-red) $\mathrm{App}_{\sigma,\tau}(\lambda_{\sigma,\tau}(P), N) = P\{N\}$,
- (Π-subst) $\Pi(\sigma, \tau)\{f\} = \Pi(\sigma\{f\}, \tau\{\mathsf{q}(f, \sigma)\})$,
- (λ-subst) $\lambda_{\sigma,\tau}(P)\{f\} = \lambda_{\sigma\{f\},\tau\{\mathsf{q}(f,\sigma)\}}(P\{\mathsf{q}(f, \sigma)\})$,
- (App-subst) $\mathrm{App}_{\sigma,\tau}(M, N)\{f\} = \mathrm{App}_{\sigma\{f\},\tau\{\mathsf{q}(f,\sigma)\}}(M\{f\}, N\{f\})$.

From (Hofmann 1994) we take the following definition, but adapt it in the obvious way to cwas.

A cwa **supports Π-types** if for $\sigma \in T(\Gamma)$ and $\tau \in T(\Gamma.\sigma)$ there is a type

$$\Pi(\sigma, \tau) \in T(\Gamma),$$

and moreover for every $P \in E(\Gamma.\sigma, \tau)$ there is an element

$$\lambda_{\sigma,\tau}(P) \in E(\Gamma, \Pi(\sigma, \tau)),$$

and furthermore for any $M \in E(\Gamma, \Pi(\sigma, \tau))$ and any $N \in E(\Gamma, \sigma)$ there is an element

$$\mathrm{App}_{\sigma,\tau}(M, N) \in E(\Gamma, \tau\{N\}),$$

such that the following equations hold for any subst. $f : B \longrightarrow \Gamma$:

- (β-red) $\mathrm{App}_{\sigma,\tau}(\lambda_{\sigma,\tau}(P), N) = P\{N\}$,
- (Π-subst) $\Pi(\sigma, \tau)\{f\} = \Pi(\sigma\{f\}, \tau\{\mathrm{q}(f, \sigma)\})$,
- (λ-subst) $\lambda_{\sigma,\tau}(P)\{f\} = \lambda_{\sigma\{f\},\tau\{\mathrm{q}(f,\sigma)\}}(P\{\mathrm{q}(f, \sigma)\})$,
- (App-subst) $\mathrm{App}_{\sigma,\tau}(M, N)\{f\} = \mathrm{App}_{\sigma\{f\},\tau\{\mathrm{q}(f,\sigma)\}}(M\{f\}, N\{f\})$.

From (Hofmann 1994) we take the following definition, but adapt it in the obvious way to cwas.

A cwa supports Π-types if for $\sigma \in T(\Gamma)$ and $\tau \in T(\Gamma.\sigma)$ there is a type

$$\Pi(\sigma, \tau) \in T(\Gamma),$$

and moreover for every $P \in E(\Gamma.\sigma, \tau)$ there is an element

$$\lambda_{\sigma,\tau}(P) \in E(\Gamma, \Pi(\sigma, \tau)),$$

and furthermore for any $M \in E(\Gamma, \Pi(\sigma, \tau))$ and any $N \in E(\Gamma, \sigma)$ there is an element

$$\mathsf{App}_{\sigma,\tau}(M, N) \in E(\Gamma, \tau\{N\}),$$

such that the following equations hold for any subst. $f : B \longrightarrow \Gamma$:

- (β-red) $\mathsf{App}_{\sigma,\tau}(\lambda_{\sigma,\tau}(P), N) = P\{N\}$,
- (Π-subst) $\Pi(\sigma, \tau)\{f\} = \Pi(\sigma\{f\}, \tau\{\mathsf{q}(f, \sigma)\})$,
- (λ-subst) $\lambda_{\sigma,\tau}(P)\{f\} = \lambda_{\sigma\{f\},\tau\{\mathsf{q}(f,\sigma)\}}(P\{\mathsf{q}(f, \sigma)\})$,
- (App-subst) $\mathsf{App}_{\sigma,\tau}(M, N)\{f\} = \mathsf{App}_{\sigma\{f\},\tau\{\mathsf{q}(f,\sigma)\}}(M\{f\}, N\{f\})$.

From (Hofmann 1994) we take the following definition, but adapt it in the obvious way to cwas.

A cwa supports Π-types if for $\sigma \in T(\Gamma)$ and $\tau \in T(\Gamma.\sigma)$ there is a type

$$\Pi(\sigma, \tau) \in T(\Gamma),$$

and moreover for every $P \in E(\Gamma.\sigma, \tau)$ there is an element

$$\lambda_{\sigma,\tau}(P) \in E(\Gamma, \Pi(\sigma, \tau)),$$

and furthermore for any $M \in E(\Gamma, \Pi(\sigma, \tau))$ and any $N \in E(\Gamma, \sigma)$ there is an element

$$\mathsf{App}_{\sigma,\tau}(M, N) \in E(\Gamma, \tau\{N\}),$$

such that the following equations hold for any subst. $f : B \longrightarrow \Gamma$:

- (β-red) $\mathsf{App}_{\sigma,\tau}(\lambda_{\sigma,\tau}(P), N) = P\{N\}$,
- (Π-subst) $\Pi(\sigma, \tau)\{f\} = \Pi(\sigma\{f\}, \tau\{\mathsf{q}(f, \sigma)\})$,
- (λ-subst) $\lambda_{\sigma,\tau}(P)\{f\} = \lambda_{\sigma\{f\},\tau\{\mathsf{q}(f,\sigma)\}}(P\{\mathsf{q}(f, \sigma)\})$,
- (App-subst) $\mathsf{App}_{\sigma,\tau}(M, N)\{f\} = \mathsf{App}_{\sigma\{f\},\tau\{\mathsf{q}(f,\sigma)\}}(M\{f\}, N\{f\})$.

From (Hofmann 1994) we take the following definition, but adapt it in the obvious way to cwas.

A cwa supports Π-types if for $\sigma \in T(\Gamma)$ and $\tau \in T(\Gamma.\sigma)$ there is a type

$$\Pi(\sigma, \tau) \in T(\Gamma),$$

and moreover for every $P \in E(\Gamma.\sigma, \tau)$ there is an element

$$\lambda_{\sigma,\tau}(P) \in E(\Gamma, \Pi(\sigma, \tau)),$$

and furthermore for any $M \in E(\Gamma, \Pi(\sigma, \tau))$ and any $N \in E(\Gamma, \sigma)$ there is an element

$$\mathrm{App}_{\sigma,\tau}(M, N) \in E(\Gamma, \tau\{N\}),$$

such that the following equations hold for any subst. $f : B \longrightarrow \Gamma$:

- (β-red) $\mathrm{App}_{\sigma,\tau}(\lambda_{\sigma,\tau}(P), N) = P\{N\}$,
- (Π-subst) $\Pi(\sigma, \tau)\{f\} = \Pi(\sigma\{f\}, \tau\{\mathrm{q}(f, \sigma)\})$,
- (λ-subst) $\lambda_{\sigma,\tau}(P)\{f\} = \lambda_{\sigma\{f\},\tau\{\mathrm{q}(f,\sigma)\}}(P\{\mathrm{q}(f, \sigma)\})$,
- (App-subst) $\mathrm{App}_{\sigma,\tau}(M, N)\{f\} = \mathrm{App}_{\sigma\{f\},\tau\{\mathrm{q}(f,\sigma)\}}(M\{f\}, N\{f\})$.

From (Hofmann 1994) we take the following definition, but adapt it in the obvious way to cwas.

A cwa supports Π-types if for $\sigma \in T(\Gamma)$ and $\tau \in T(\Gamma.\sigma)$ there is a type

$$\Pi(\sigma, \tau) \in T(\Gamma),$$

and moreover for every $P \in E(\Gamma.\sigma, \tau)$ there is an element

$$\lambda_{\sigma,\tau}(P) \in E(\Gamma, \Pi(\sigma, \tau)),$$

and furthermore for any $M \in E(\Gamma, \Pi(\sigma, \tau))$ and any $N \in E(\Gamma, \sigma)$ there is an element

$$\mathrm{App}_{\sigma,\tau}(M, N) \in E(\Gamma, \tau\{N\}),$$

such that the following equations hold for any subst. $f : B \longrightarrow \Gamma$:

- (β-red) $\mathrm{App}_{\sigma,\tau}(\lambda_{\sigma,\tau}(P), N) = P\{N\}$,
- (Π-subst) $\Pi(\sigma, \tau)\{f\} = \Pi(\sigma\{f\}, \tau\{\mathrm{q}(f, \sigma)\})$,
- (λ-subst) $\lambda_{\sigma,\tau}(P)\{f\} = \lambda_{\sigma\{f\},\tau\{\mathrm{q}(f,\sigma)\}}(P\{\mathrm{q}(f, \sigma)\})$,
- (App-subst) $\mathrm{App}_{\sigma,\tau}(M, N)\{f\} = \mathrm{App}_{\sigma\{f\},\tau\{\mathrm{q}(f,\sigma)\}}(M\{f\}, N\{f\})$.

From (Hofmann 1994) we take the following definition, but adapt it in the obvious way to cwas.

A cwa supports Π-types if for $\sigma \in T(\Gamma)$ and $\tau \in T(\Gamma.\sigma)$ there is a type

$$\Pi(\sigma, \tau) \in T(\Gamma),$$

and moreover for every $P \in E(\Gamma.\sigma, \tau)$ there is an element

$$\lambda_{\sigma,\tau}(P) \in E(\Gamma, \Pi(\sigma, \tau)),$$

and furthermore for any $M \in E(\Gamma, \Pi(\sigma, \tau))$ and any $N \in E(\Gamma, \sigma)$ there is an element

$$\mathrm{App}_{\sigma,\tau}(M, N) \in E(\Gamma, \tau\{N\}),$$

such that the following equations hold for any subst. $f : B \longrightarrow \Gamma$:

- (β-red) $\mathrm{App}_{\sigma,\tau}(\lambda_{\sigma,\tau}(P), N) = P\{N\}$,
- (Π-subst) $\Pi(\sigma, \tau)\{f\} = \Pi(\sigma\{f\}, \tau\{\mathrm{q}(f, \sigma)\})$,
- (λ-subst) $\lambda_{\sigma,\tau}(P)\{f\} = \lambda_{\sigma\{f\},\tau\{\mathrm{q}(f,\sigma)\}}(P\{\mathrm{q}(f, \sigma)\})$,
- (App-subst) $\mathrm{App}_{\sigma,\tau}(M, N)\{f\} = \mathrm{App}_{\sigma\{f\},\tau\{\mathrm{q}(f,\sigma)\}}(M\{f\}, N\{f\})$.

From (Hofmann 1994) we take the following definition, but adapt it in the obvious way to cwas.

A cwa supports Π-types if for $\sigma \in T(\Gamma)$ and $\tau \in T(\Gamma.\sigma)$ there is a type

$$\Pi(\sigma, \tau) \in T(\Gamma),$$

and moreover for every $P \in E(\Gamma.\sigma, \tau)$ there is an element

$$\lambda_{\sigma,\tau}(P) \in E(\Gamma, \Pi(\sigma, \tau)),$$

and furthermore for any $M \in E(\Gamma, \Pi(\sigma, \tau))$ and any $N \in E(\Gamma, \sigma)$ there is an element

$$\mathsf{App}_{\sigma,\tau}(M, N) \in E(\Gamma, \tau\{N\}),$$

such that the following equations hold for any subst. $f : B \longrightarrow \Gamma$:

- (β-red) $\mathsf{App}_{\sigma,\tau}(\lambda_{\sigma,\tau}(P), N) = P\{N\}$,
- (Π-subst) $\Pi(\sigma, \tau)\{f\} = \Pi(\sigma\{f\}, \tau\{\mathsf{q}(f, \sigma)\})$,
- (λ-subst) $\lambda_{\sigma,\tau}(P)\{f\} = \lambda_{\sigma\{f\},\tau\{\mathsf{q}(f,\sigma)\}}(P\{\mathsf{q}(f, \sigma)\})$,
- (App-subst) $\mathsf{App}_{\sigma,\tau}(M, N)\{f\} = \mathsf{App}_{\sigma\{f\},\tau\{\mathsf{q}(f,\sigma)\}}(M\{f\}, N\{f\})$.

Adapting this to ecwas ... the first part is similar

An ecwa supports Π-types if for $\sigma \in T(\Gamma)$ and $\tau \in T(\Gamma.\sigma)$ there is a type $\Pi(\sigma, \tau) \in T(\Gamma)$, and moreover for every $P \in E(\Gamma.\sigma, \tau)$ there is an element $\lambda_{\sigma,\tau}(P) \in E(\Gamma, \Pi(\sigma, \tau))$, and furthermore for any $M \in E(\Gamma, \Pi(\sigma, \tau))$ and any $N \in E(\Gamma, \sigma)$ there is an element $\mathsf{App}_{\sigma,\tau}(M, N) \in E(\Gamma, \tau\{N\})$, such that the following equations hold for any $f : B \longrightarrow \Gamma$:

- (β-red) $\mathsf{App}_{\sigma,\tau}(\lambda_{\sigma,\tau}(P), N) =_{E(\Gamma,\tau\{N\})} P\{N\}$, [as before]
- (Π-subst) $\Pi(\sigma, \tau)\{f\} =_{T(B)} \Pi(\sigma\{f\}, \tau\{\mathsf{q}(f, \sigma)\})$, [as before]

This part of the definition has type adjustments:

- (λ-subst)

$$\lambda_{\sigma,\tau}(P)\{f\} =_{E(B,\Pi(\sigma,\tau)\{f\})} \Delta_B(t) \circ \lambda_{\sigma\{f\},\tau\{\mathsf{q}(f,\sigma)\}}(P\{\mathsf{q}(f,\sigma)\}),$$

  for any proof t of $\Pi(\sigma\{f\},\tau\{\mathsf{q}(f,\sigma)\}) =_{T(B)} \Pi(\sigma,\tau)\{f\}$,

- (App-subst)

$$\mathsf{App}_{\sigma,\tau}(M,N)\{f\} =_{E(\cdots)} \Delta_B(s)\circ\mathsf{App}_{\sigma\{f\},\tau\{\mathsf{q}(f,\sigma)\}}(\Delta_B(t)\circ M\{f\}, N\{f\})$$

  for any proof s of $\tau\{\mathsf{q}(f,\sigma)\}\{N\{f\}\} =_{T(B)} \tau\{N\}\{f\}$ and any proof t of $\Pi(\sigma,\tau)\{f\} =_{T(B)} \Pi(\sigma\{f\},\tau\{\mathsf{q}(f,\sigma)\})$.

This part of the definition has type adjustments:

- (λ-subst)

  $$\lambda_{\sigma,\tau}(P)\{f\} =_{E(B,\Pi(\sigma,\tau)\{f\})} \Delta_B(t) \circ \lambda_{\sigma\{f\},\tau\{\mathsf{q}(f,\sigma)\}}(P\{\mathsf{q}(f,\sigma)\}),$$

  for any proof t of $\Pi(\sigma\{f\},\tau\{\mathsf{q}(f,\sigma)\}) =_{T(B)} \Pi(\sigma,\tau)\{f\}$,
- (App-subst)

  $$\mathsf{App}_{\sigma,\tau}(M,N)\{f\} =_{E(\cdots)} \Delta_B(s)\circ\mathsf{App}_{\sigma\{f\},\tau\{\mathsf{q}(f,\sigma)\}}(\Delta_B(t)\circ M\{f\}, N\{f\})$$

  for any proof s of $\tau\{\mathsf{q}(f,\sigma)\}\{N\{f\}\} =_{T(B)} \tau\{N\}\{f\}$ and any proof t of $\Pi(\sigma,\tau)\{f\} =_{T(B)} \Pi(\sigma\{f\},\tau\{\mathsf{q}(f,\sigma)\})$.

Furthermore there are the following extensionality conditions on Π, λ and App:

- (Π-cong) if s is a proof of $\sigma =_{T(\Gamma)} \sigma'$ and for $\tau \in T(\Gamma.\sigma)$, $\tau' \in T(\Gamma.\sigma')$ with $\tau =_{T(\Gamma.\sigma)} \tau'\{\Delta(s)\}$, then

$$\Pi(\sigma, \tau) =_{T(\Gamma)} \Pi(\sigma', \tau').$$

- (λ-cong) if $P =_{E(\Gamma.\sigma,\tau)} P'$, then

$$\lambda_{\sigma,\tau}(P) =_{E(\Gamma,\Pi(\sigma,\tau))} \lambda_{\sigma,\tau}(P')$$

- (App-cong) if $M =_{E(\Gamma,\Pi(\sigma,\tau))} M'$ and $N =_{E(\Gamma,\sigma)} N'$ then

$$\Delta_\Gamma(s) \circ \mathrm{App}_{\sigma,\tau}(M, N) =_{E(\Gamma,\tau\{N'\})} \mathrm{App}_{\sigma,\tau}(M', N'),$$

where s is any proof of $\tau\{N\} =_{T(\Gamma)} \tau\{N'\}$.

Furthermore there are the following extensionality conditions on Π, λ and App:

- (Π-cong) if s is a proof of $\sigma =_{T(\Gamma)} \sigma'$ and for $\tau \in T(\Gamma.\sigma)$, $\tau' \in T(\Gamma.\sigma')$ with $\tau =_{T(\Gamma.\sigma)} \tau'\{\Delta(s)\}$, then

$$\Pi(\sigma,\tau) =_{T(\Gamma)} \Pi(\sigma',\tau').$$

- (λ-cong) if $P =_{E(\Gamma.\sigma,\tau)} P'$, then

$$\lambda_{\sigma,\tau}(P) =_{E(\Gamma,\Pi(\sigma,\tau))} \lambda_{\sigma,\tau}(P')$$

- (App-cong) if $M =_{E(\Gamma,\Pi(\sigma,\tau))} M'$ and $N =_{E(\Gamma,\sigma)} N'$ then

$$\Delta_\Gamma(s) \circ \mathrm{App}_{\sigma,\tau}(M,N) =_{E(\Gamma,\tau\{N'\})} \mathrm{App}_{\sigma,\tau}(M',N'),$$

where s is any proof of $\tau\{N\} =_{T(\Gamma)} \tau\{N'\}$.

Furthermore there are the following extensionality conditions on Π, λ and App:

- (Π-cong) if s is a proof of $\sigma =_{T(\Gamma)} \sigma'$ and for $\tau \in T(\Gamma.\sigma)$, $\tau' \in T(\Gamma.\sigma')$ with $\tau =_{T(\Gamma.\sigma)} \tau'\{\Delta(s)\}$, then
$$\Pi(\sigma,\tau) =_{T(\Gamma)} \Pi(\sigma',\tau').$$
- (λ-cong) if $P =_{E(\Gamma.\sigma,\tau)} P'$, then
$$\lambda_{\sigma,\tau}(P) =_{E(\Gamma,\Pi(\sigma,\tau))} \lambda_{\sigma,\tau}(P')$$
- (App-cong) if $M =_{E(\Gamma,\Pi(\sigma,\tau))} M'$ and $N =_{E(\Gamma,\sigma)} N'$ then
$$\Delta_\Gamma(s) \circ \mathrm{App}_{\sigma,\tau}(M,N) =_{E(\Gamma,\tau\{N'\})} \mathrm{App}_{\sigma,\tau}(M',N'),$$
where s is any proof of $\tau\{N\} =_{T(\Gamma)} \tau\{N'\}$.

Existence of internal models

We present some work in progress.

Graded extensional universes (jww Olov Wilander)

A graded graded universe is a family of setoids $F : A \longrightarrow \textbf{Setoids}$ with a grading function $\delta : A \longrightarrow \mathbb{N}$, a lifting function $\ell : A \longrightarrow A$ and natural isomorphism $\phi : F \circ \ell \longrightarrow F$ i.e. $(\phi_a)_{a \in A}$ are bijections such that

$$\begin{tikzcd} F(\ell a) \arrow[r, "\phi_a"] \arrow[d, "F(p)"'] & Fa \arrow[d, "F(q)"] \\ F(\ell b) \arrow[r, "\phi_b"'] & Fb \end{tikzcd} \tag{1}$$

where $a, b \in A$ and $p : \ell a =_A \ell b$, $q : a =_A b$ are arbitrary. Moreover, it is required that

$$\delta a \leq \delta b \Longrightarrow \delta(\ell^{\delta b - \delta a}(a)) = \delta b.$$

We allow the grading to be trivial: $\delta(a) \equiv 0$ and $\ell = id_A$.

The graded universe F is Σ-closed if for any $a \in A$ and any function $f : F(a) \longrightarrow A$, which are *adapted*, i.e. $\forall x \in F(a).\delta(a) = \delta(fx)$, any proof H of this, there is $\hat{\Sigma}(a, f, H) \in A$, with $\delta(\hat{\Sigma}(a, f, H)) = \delta(\ell a)$, and a bijection

$$\psi_{a,f,H} : F(\hat{\Sigma}(a, f, H)) \longrightarrow \Sigma(Fa, F \circ f).$$

These maps should moreover satisfy the following condition: If $p : a =_A a'$ and $f = f' \circ F(p)$, and for any proofs H, H' of adaptedness, then

$$\hat{\Sigma}(a, f, H) =_A \hat{\Sigma}(a', f', H')$$

and moreover a naturality condition

$$\begin{tikzcd} F(\hat{\Sigma}(a, f, H)) \arrow[r, "\psi_{a,f,H}"] \arrow[d, "F(q)"'] & \Sigma(Fa, F \circ f) \arrow[d, "\Sigma_{p,k}"] \\ F(\hat{\Sigma}(a', f', H')) \arrow[r, "\psi_{a',f',H'}"'] & \Sigma(Fa', F \circ f') \end{tikzcd} \tag{2}$$

commutes. Above q is any proof of $\hat{\Sigma}(a, f, H) =_A \hat{\Sigma}(a', f', H')$ and $\Sigma_{p,k}$ is given by $\Sigma_{p,k}(x, y) = (x \cdot p, y \cdot k(x))$ where k is any proof of $f = f' \circ F(p)$

The graded universe F is Σ-closed if for any $a \in A$ and any function $f : F(a) \longrightarrow A$, which are *adapted*, i.e. $\forall x \in F(a).\delta(a) = \delta(fx)$, any proof H of this, there is $\hat{\Sigma}(a, f, H) \in A$, with $\delta(\hat{\Sigma}(a, f, H)) = \delta(\ell a)$, and a bijection

$$\psi_{a,f,H} : F(\hat{\Sigma}(a, f, H)) \longrightarrow \Sigma(Fa, F \circ f).$$

These maps should moreover satisfy the following condition: If $p : a =_A a'$ and $f = f' \circ F(p)$, and for any proofs H, H' of adaptedness, then

$$\hat{\Sigma}(a, f, H) =_A \hat{\Sigma}(a', f', H')$$

and moreover a naturality condition

$$\begin{tikzcd}
F(\hat{\Sigma}(a, f, H)) \arrow[r, "\psi_{a,f,H}"] \arrow[d, "F(q)"] & \Sigma(Fa, F \circ f) \arrow[d, "\Sigma_{p,k}"] \\
F(\hat{\Sigma}(a', f', H')) \arrow[r, "\psi_{a',f',H'}"'] & \Sigma(Fa', F \circ f')
\end{tikzcd} \tag{2}$$

commutes. Above q is any proof of $\hat{\Sigma}(a, f, H) =_A \hat{\Sigma}(a', f', H')$ and $\Sigma_{p,k}$ is given by $\Sigma_{p,k}(x, y) = (x \cdot p, y \cdot k(x))$ where k is any proof of $f = f' \circ F(p)$

The graded universe F is Σ-closed if for any $a \in A$ and any function $f : F(a) \longrightarrow A$, which are *adapted*, i.e. $\forall x \in F(a).\delta(a) = \delta(fx)$, any proof H of this, there is $\hat{\Sigma}(a, f, H) \in A$, with $\delta(\hat{\Sigma}(a, f, H)) = \delta(\ell a)$, and a bijection

$$\psi_{a,f,H} : F(\hat{\Sigma}(a, f, H)) \longrightarrow \Sigma(Fa, F \circ f).$$

These maps should moreover satisfy the following condition: If $p : a =_A a'$ and $f = f' \circ F(p)$, and for any proofs H, H' of adaptedness, then

$$\hat{\Sigma}(a, f, H) =_A \hat{\Sigma}(a', f', H')$$

and moreover a naturality condition

$$\begin{tikzcd}
F(\hat{\Sigma}(a, f, H)) \arrow[r, "\psi_{a,f,H}"] \arrow[d, "F(q)"'] & \Sigma(Fa, F \circ f) \arrow[d, "\Sigma_{p,k}"] \\
F(\hat{\Sigma}(a', f', H')) \arrow[r, "\psi_{a',f',H'}"'] & \Sigma(Fa', F \circ f')
\end{tikzcd} \tag{2}$$

commutes. Above q is any proof of $\hat{\Sigma}(a, f, H) =_A \hat{\Sigma}(a', f', H')$ and $\Sigma_{p,k}$ is given by $\Sigma_{p,k}(x, y) = (x \cdot p, y \cdot k(x))$ where k is any proof of $f = f' \circ F(p)$.

We build an ecwa from the graded universe, which we later wish to formalize in Coq. The graded universe has already been so formalized. Define an e-category $\mathcal{C}$ whose objects are elements $a, b, \ldots \in A$ A

morphism $f : a \longrightarrow b$ is an extensional function $f : F(a) \longrightarrow F(b)$.

Define an e-functor $T : \mathcal{C}^{op} \longrightarrow$ **Setoids** by letting

$$T(a) \equiv ((\Sigma k : N)Ext(F(a), A) \wedge (\forall x \in A)\delta(fx) = k), \sim)$$

(i.e. bounded families) where

$$(k, g, p) \sim (k', g', p') \Leftrightarrow k = k' \;\&\; g =_{\text{ext}} g'$$

and for $f : a \longrightarrow b$

$$T(f)(k, g, p) = (k, g \circ f, p')$$

We build an ecwa from the graded universe, which we later wish to formalize in Coq. The graded universe has already been so formalized. Define an e-category $\mathcal{C}$ whose objects are elements $a, b, \ldots \in A$ A

morphism $f : a \longrightarrow b$ is an extensional function $f : F(a) \longrightarrow F(b)$.

Define an e-functor $T : \mathcal{C}^{op} \longrightarrow$ **Setoids** by letting

$$T(a) \equiv ((\Sigma k : N) Ext(F(a), A) \wedge (\forall x \in A)\delta(fx) = k), \sim)$$

(i.e. bounded families) where

$$(k, g, p) \sim (k', g', p') \Leftrightarrow k = k' \;\&\; g =_{\mathrm{ext}} g'$$

and for $f : a \longrightarrow b$

$$T(f)(k, g, p) = (k, g \circ f, p')$$

We build an ecwa from the graded universe, which we later wish to formalize in Coq. The graded universe has already been so formalized. Define an e-category $\mathcal{C}$ whose objects are elements $a, b, \ldots \in A$ A

morphism $f : a \longrightarrow b$ is an extensional function $f : F(a) \longrightarrow F(b)$.

Define an e-functor $T : \mathcal{C}^{op} \longrightarrow$ **Setoids** by letting

$$T(a) \equiv ((\Sigma k : N) Ext(F(a), A) \wedge (\forall x \in A) \delta(fx) = k), \sim)$$

(i.e. bounded families) where

$$(k, g, p) \sim (k', g', p') \Leftrightarrow k = k' \;\&\; g =_{\mathrm{ext}} g'$$

and for $f : a \longrightarrow b$

$$T(f)(k, g, p) = (k, g \circ f, p')$$

We build an ecwa from the graded universe, which we later wish to formalize in Coq. The graded universe has already been so formalized. Define an e-category $\mathcal{C}$ whose objects are elements $a, b, \ldots \in A$ A

morphism $f : a \longrightarrow b$ is an extensional function $f : F(a) \longrightarrow F(b)$.

Define an e-functor $T : \mathcal{C}^{op} \longrightarrow$ **Setoids** by letting

$$T(a) \equiv ((\Sigma k : N) Ext(F(a), A) \wedge (\forall x \in A)\delta(fx) = k), \sim)$$

(i.e. bounded families) where

$$(k, g, p) \sim (k', g', p') \Leftrightarrow k = k' \;\&\; g =_{\text{ext}} g'$$

and for $f : a \longrightarrow b$

$$T(f)(k, g, p) = (k, g \circ f, p')$$

The *context extension operation:* For

$$\Gamma = a \in \mathcal{C} \text{ and } \sigma = (k, g, p) \in T(\Gamma)$$

let the extended context be

$$\Gamma.\sigma = (s+1, \hat{\Sigma}(\ell^{s-n}(a), \ell^{s-k} \circ g \circ \phi^{s-n}))$$

where $s = \max(\delta(a), k)$. The associated projection $\mathrm{p}(\sigma) : \Gamma.\sigma \longrightarrow \Gamma$ is given by

$$\mathrm{p}(\sigma)(x, y) = x.$$

For $f : B \longrightarrow \Gamma$ we wish to define $\mathsf{q}(f, \sigma) : B.\sigma\{f\} \longrightarrow \Gamma.\sigma$ in $\mathcal{C}$ such that

$$\begin{tikzcd}
B.\sigma\{f\} \arrow[r, "\mathsf{q}(f,\sigma)"] \arrow[d, "\mathsf{p}(\sigma\{f\})"'] & \Gamma.\sigma \arrow[d, "\mathsf{p}(\sigma)"] \\
B \arrow[r, "f"'] & \Gamma
\end{tikzcd}$$

is a pullback. This can be done similarly to the case of sets, but using the setoid sum instead of the set-theoretic sum.

Construction of a graded universe

We construct a sequence of setoids A_n and sequence of setoid families F_n over A_n. Let A_0 be a three element setoid with elements called $\hat{N}_0, \hat{N}_1, \hat{N}$. Define $F_0(\hat{N}_0)$, $F_0(\hat{N}_1)$ and $F_0(\hat{N})$ to be the empty setoid, the one element setoid and the setoid of natural numbers respectively. For $p : x =_{A_0} y$ we let $F_0(p)$ be the identity map. We let

$$A_{n+1} = C(A_n, F_n) \text{ and } F_{n+1} = S(A_n, F_n).$$

The desired graded universe U, T will then be a sum of this family in a straightforward way i.e $U = ((\Sigma n : N)A_n, \ldots)$ and $T(n, a) = F_n(a)$.

The constructions C and S.

For a setoid A and setoid family F over A we construct a setoid $A^* = C(A, F)$ and $F^* = S(A, F)$ a setoid family over A^*.
The set $|A^*|$ is constructed by the following rules

$$\frac{a : |A|}{\ell(a) : |A^*|} \qquad \frac{a : |A| \quad b : |A|}{a + b : |A^*|}$$

$$\frac{a : |A| \quad x : |F|(a) \quad y : |F|(a)}{e(a, x, y) : |A^*|}.$$

$$\frac{a : |A| \quad f : \mathrm{Ext}(F(a), A)}{\sigma(a, f) : |A^*|} \qquad \frac{a : |A| \quad f : \mathrm{Ext}(F(a), A)}{\pi(a, f) : |A^*|}$$

We may in fact construct $|A^*|$ as the disjoint sum

$$|A| + |A| \times |A| + (\Sigma a : |A|)(|F(a)| \times |F(a)|) + (\Sigma a : |A|)Ext(F(a), A) + (\Sigma a : |A|)Ext(F(a), A).$$

The constructions C and S.

For a setoid A and setoid family F over A we construct a setoid $A^* = C(A,F)$ and $F^* = S(A,F)$ a setoid family over A^*.
The set $|A^*|$ is constructed by the following rules

$$\frac{a : |A|}{\ell(a) : |A^*|} \qquad \frac{a : |A| \quad b : |A|}{a + b : |A^*|}$$

$$\frac{a : |A| \quad x : |F|(a) \quad y : |F|(a)}{e(a,x,y) : |A^*|}.$$

$$\frac{a : |A| \quad f : \mathrm{Ext}(F(a), A)}{\sigma(a,f) : |A^*|} \qquad \frac{a : |A| \quad f : \mathrm{Ext}(F(a), A)}{\pi(a,f) : |A^*|}$$

We may in fact construct $|A^*|$ as the disjoint sum

$$|A| + |A| \times |A| + (\Sigma a : |A|)(|F(a)| \times |F(a)|) +$$
$$(\Sigma a : |A|)Ext(F(a), A) + (\Sigma a : |A|)Ext(F(a), A).$$

We now define $=_{A^*}$ assuming A and F are given.
Define $=_{A^*}$ by the 5×5 cases indicated by the sum above

- $\ell(a) =_{A^*} \ell(a')$ iff $a =_A a'$
- $a + b =_{A^*} a' + b'$ iff $a =_A a'$ and $b =_A b'$
- $e(a, x, y) =_{A^*} e(a', x', y')$ iff for some $u : (a =_A a')$ we have $F(u)(x) =_A x'$ and $F(u)(y) =_A y'$.
- $\sigma(a, f) =_{A^*} \sigma(a', f')$ iff for some $u : (a =_A a')$ we have $(\forall x : F(a))(f(x) =_{F(a')} f'(F(u)(x)))$.
- $\pi(a, f) =_{A^*} \pi(a', f')$ iff for some $u : (a =_A a')$ we have $(\forall x : F(a))(f(x) =_{F(a')} f'(F(u)(x)))$.

In the remaining cases, $r =_{A^*} r'$ is false.

We now define $=_{A^*}$ assuming A and F are given.
Define $=_{A^*}$ by the 5×5 cases indicated by the sum above

- $\ell(a) =_{A^*} \ell(a')$ iff $a =_A a'$
- $a + b =_{A^*} a' + b'$ iff $a =_A a'$ and $b =_A b'$
- $e(a, x, y) =_{A^*} e(a', x', y')$ iff for some $u : (a =_A a')$ we have $F(u)(x) =_A x'$ and $F(u)(y) =_A y'$.
- $\sigma(a, f) =_{A^*} \sigma(a', f')$ iff for some $u : (a =_A a')$ we have $(\forall x : F(a))(f(x) =_{F(a')} f'(F(u)(x)))$.
- $\pi(a, f) =_{A^*} \pi(a', f')$ iff for some $u : (a =_A a')$ we have $(\forall x : F(a))(f(x) =_{F(a')} f'(F(u)(x)))$.

In the remaining cases, $r =_{A^*} r'$ is false.

We now define $=_{A^*}$ assuming A and F are given.
Define $=_{A^*}$ by the 5×5 cases indicated by the sum above

- $\ell(a) =_{A^*} \ell(a')$ iff $a =_A a'$
- $a + b =_{A^*} a' + b'$ iff $a =_A a'$ and $b =_A b'$
- $e(a, x, y) =_{A^*} e(a', x', y')$ iff for some $u : (a =_A a')$ we have $F(u)(x) =_A x'$ and $F(u)(y) =_A y'$.
- $\sigma(a, f) =_{A^*} \sigma(a', f')$ iff for some $u : (a =_A a')$ we have $(\forall x : F(a))(f(x) =_{F(a')} f'(F(u)(x)))$.
- $\pi(a, f) =_{A^*} \pi(a', f')$ iff for some $u : (a =_A a')$ we have $(\forall x : F(a))(f(x) =_{F(a')} f'(F(u)(x)))$.

In the remaining cases, $r =_{A^*} r'$ is false.

We now define $=_{A^*}$ assuming A and F are given.
Define $=_{A^*}$ by the 5×5 cases indicated by the sum above

- $\ell(a) =_{A^*} \ell(a')$ iff $a =_A a'$
- $a + b =_{A^*} a' + b'$ iff $a =_A a'$ and $b =_A b'$
- $e(a, x, y) =_{A^*} e(a', x', y')$ iff for some $u : (a =_A a')$ we have $F(u)(x) =_A x'$ and $F(u)(y) =_A y'$.
- $\sigma(a, f) =_{A^*} \sigma(a', f')$ iff for some $u : (a =_A a')$ we have $(\forall x : F(a))(f(x) =_{F(a')} f'(F(u)(x)))$.
- $\pi(a, f) =_{A^*} \pi(a', f')$ iff for some $u : (a =_A a')$ we have $(\forall x : F(a))(f(x) =_{F(a')} f'(F(u)(x)))$.

In the remaining cases, $r =_{A^*} r'$ is false.

Next define $F^*(a)$ by cases on $|A^*|$.

- $F^*(\ell(a)) = F(a)$
- $F^*(a + b) = D(F(a), F(b))$ (binary disjoint union as setoids).
- $F^*(e(a, x, y)) = E(F(a), x, y)$ (extensional identity setoid)
- $F^*(\sigma(a, f)) = S(F(a), F \circ f)$ (general disjoint union setoid)
- $F^*(\pi(a, f)) = P(F(a), F \circ f)$ (general product setoid)

The setoid constructions E, S and P: Suppose B is a setoid and $x, y : |B|$. Define

$$E(B, x, y) = ((x =_B y), \sim),$$

where $r \sim r'$ is true for all r, r'.
Suppose B is a setoid and that G is a family of setoids over B. Define

$$S(B, G) = ((\Sigma x : |B|)|G(x)|, \sim)$$

where $(x, y) \sim (x', y')$ iff $(\exists p : x =_B x')(G(p)(y) =_{G(x')} y')$.
Further define

$$\begin{aligned} P(B, G) &= ((\Sigma g : (\Pi x : |B|)|G(x)|) \\ &\quad (\forall x, x' : |B|)(\forall p : x =_B x')[G(p)(g(x)) =_{G(x')} g(x')], \sim) \end{aligned}$$

where $(g, r) \sim (g', r')$ iff $(\forall x : |B|)[g(x) =_{G(x)} g'(x)]$.

The setoid constructions E, S and P: Suppose B is a setoid and $x, y : |B|$. Define

$$E(B, x, y) = ((x =_B y), \sim),$$

where $r \sim r'$ is true for all r, r'.
Suppose B is a setoid and that G is a family of setoids over B. Define

$$S(B, G) = ((\Sigma x : |B|)|G(x)|, \sim)$$

where $(x, y) \sim (x', y')$ iff $(\exists p : x =_B x')(G(p)(y) =_{G(x')} y')$.
Further define

$$\begin{aligned} P(B, G) \quad = \quad & ((\Sigma g : (\Pi x : |B|)|G(x)|) \\ & (\forall x, x' : |B|)(\forall p : x =_B x')[G(p)(g(x)) =_{G(x')} g(x')], \sim) \end{aligned}$$

where $(g, r) \sim (g', r')$ iff $(\forall x : |B|)[g(x) =_{G(x)} g'(x)]$.

The setoid constructions E, S and P: Suppose B is a setoid and $x, y : |B|$. Define

$$E(B, x, y) = ((x =_B y), \sim),$$

where $r \sim r'$ is true for all r, r'.
Suppose B is a setoid and that G is a family of setoids over B. Define

$$S(B, G) = ((\Sigma x : |B|)|G(x)|, \sim)$$

where $(x, y) \sim (x', y')$ iff $(\exists p : x =_B x')(G(p)(y) =_{G(x')} y')$.
Further define

$$\begin{aligned} P(B, G) \quad = \quad & ((\Sigma g : (\Pi x : |B|)|G(x)|) \\ & (\forall x, x' : |B|)(\forall p : x =_B x')[G(p)(g(x)) =_{G(x')} g(x')], \sim) \end{aligned}$$

where $(g, r) \sim (g', r')$ iff $(\forall x : |B|)[g(x) =_{G(x)} g'(x)]$.

Transportation functions

For $p : a =_{A^*} a'$ we define a transportation function

$$F^*(p) : F^*(a) \longrightarrow F^*(a')$$

according to the 5×5 cases for a and a'. For the 20 cases where a and a' have distinct outer form $a =_A a'$ is empty and we define $|F^*(p)|$ by absurdity elimination.

- For $p : (\ell(a) =_{A^*} \ell(a'))$ let $|F^*(p)| = |F(p)|$,
- For $p = (p_1, p_2) : (a + b =_{A^*} a' + b')$ let $F^*(p) = F(p_1) + F(p_2)$,
- For $p = (p_1, p_2, p_3) : e(a, x, y) =_{A^*} e(a', x', y')$ we have $p_1 : a =_A a'$, $p_2 : F(p_1)(x) =_{F(a)} x'$ and $p_3 : F(p_1)(y) =_{F(a)} y'$. Let $F^*(p)(q) = p_3 \circ \mathrm{ext}_{F(p_1)}(x, y, q) \circ p_2^{-1}$

Transportation functions

For $p : a =_{A^*} a'$ we define a transportation function

$$F^*(p) : F^*(a) \longrightarrow F^*(a')$$

according to the 5×5 cases for a and a'. For the 20 cases where a and a' have distinct outer form $a =_A a'$ is empty and we define $|F^*(p)|$ by absurdity elimination.

- For $p : (\ell(a) =_{A^*} \ell(a'))$ let $|F^*(p)| = |F(p)|$,
- For $p = (p_1, p_2) : (a + b =_{A^*} a' + b')$ let $F^*(p) = F(p_1) + F(p_2)$,
- For $p = (p_1, p_2, p_3) : e(a, x, y) =_{A^*} e(a', x', y')$ we have $p_1 : a =_A a'$, $p_2 : F(p_1)(x) =_{F(a)} x'$ and $p_3 : F(p_1)(y) =_{F(a)} y'$. Let $F^*(p)(q) = p_3 \circ \mathrm{ext}_{F(p_1)}(x, y, q) \circ p_2^{-1}$

Transportation functions (cont.)

- For $p = (p_1, p_2) : \sigma(a, f) =_{A^*} \sigma(a', f')$ we have $p_1 : a =_A a'$ and $p_2 : (\forall x : F(a))(f(x) =_{F(a')} f'(F(p_1)(x)))$. For $(x, y) : |\Sigma(F(a), F \circ f)| = (\Sigma x : |F(a)|)\,|F(f(x))|$, let

$$F^*(p)((x, y)) = (F(p_1)(x), F(p_2(x))(y)).$$

- For $p = (p_1, p_2) : \pi(a, f) =_{A^*} \pi(a', f')$ we have $p_1 : a =_A a'$ and $p_2 : (\forall x : F(a))(f(x) =_A f'(F(p_1)(x)))$. For $(g, e) : |\Pi(F(a), F \circ f)|$ we have $g : (\Pi x : |F(a)|)\,|F(f(x))|$ and

$$e : (\forall x, x' : |F(a)|)(\forall p : x =_{F(a)} x')[(F \circ f)(p)(g(x)) =_{(F \circ f)(x')} g(x')].$$

Transportation functions (cont.)

- For $p = (p_1, p_2) : \sigma(a, f) =_{A^*} \sigma(a', f')$ we have $p_1 : a =_A a'$ and $p_2 : (\forall x : F(a))(f(x) =_{F(a')} f'(F(p_1)(x)))$. For $(x, y) : |\Sigma(F(a), F \circ f)| = (\Sigma x : |F(a)|)\, |F(f(x))|$, let

$$F^*(p)((x, y)) = (F(p_1)(x), F(p_2(x))(y)).$$

- For $p = (p_1, p_2) : \pi(a, f) =_{A^*} \pi(a', f')$ we have $p_1 : a =_A a'$ and $p_2 : (\forall x : F(a))(f(x) =_A f'(F(p_1)(x)))$. For $(g, e) : |\Pi(F(a), F \circ f)|$ we have $g : (\Pi x : |F(a)|)\, |F(f(x))|$ and

$$e : (\forall x, x' : |F(a)|)(\forall p : x =_{F(a)} x')[(F \circ f)(p)(g(x)) =_{(F \circ f)(x')} g(x')].$$

Transportation functions (cont.)

We define $F^*(p)(g,e) = (g', e')$ below. For $u : F(a')$ we have $g(F(p_1^{-1})(u)) : |F(f(F(p_1^{-1})(u)))|$ and hence $F(p_2(F(p_1^{-1})(u)))(g(F(p_1^{-1})(u))) : |F(f'(F(p_1)(F(p_1^{-1})(u))))|$. Now

$$F(p_1)(F(p_1^{-1})(u)) =_{F(a')} u$$

so it is inhabited by, say, $h(a', u)$. Now f' is extensional so

$$(F \circ f')(h(a', u)) = F(\mathrm{ext}_{f'}(F(p_1)(F(p_1^{-1})(u)), u, h(a', u))) : F(f'(F(p_1)(F(p_1^{-1}$$

We let

$$g'(u) = (F \circ f')(h(a', u))(F(p_2(F(p_1^{-1})(u)))(g(F(p_1^{-1})(u)))) : |F(f'(u))|.$$

To define e' it suffices to prove (in any way possible):

$$(\forall u, u' : |F(a')|)(\forall p : u =_{F(a')} u')[(F \circ f')(p)(g'(u)) =_{(F \circ f')(u')} g'(u')].$$

Transportation functions (cont.)

We define $F^*(p)(g,e) = (g', e')$ below. For $u : F(a')$ we have $g(F(p_1^{-1})(u)) : |F(f(F(p_1^{-1})(u)))|$ and hence $F(p_2(F(p_1^{-1})(u)))(g(F(p_1^{-1})(u))) : |F(f'(F(p_1)(F(p_1^{-1})(u))))|$. Now

$$F(p_1)(F(p_1^{-1})(u)) =_{F(a')} u$$

so it is inhabited by, say, $h(a', u)$. Now f' is extensional so

$$(F \circ f')(h(a', u)) = F(\mathrm{ext}_{f'}(F(p_1)(F(p_1^{-1})(u)), u, h(a', u))) : F(f'(F(p_1)(F(p_1^{-}$$

We let

$$g'(u) = (F \circ f')(h(a', u))(F(p_2(F(p_1^{-1})(u)))(g(F(p_1^{-1})(u)))) : |F(f'(u))|.$$

To define e' it suffices to prove (in any way possible):

$$(\forall u, u' : |F(a')|)(\forall p : u =_{F(a')} u')[(F \circ f')(p)(g'(u)) =_{(F \circ f')(u')} g'(u')].$$

Transportation functions (cont.)

We define $F^*(p)(g,e) = (g', e')$ below. For $u : F(a')$ we have $g(F(p_1^{-1})(u)) : |F(f(F(p_1^{-1})(u)))|$ and hence $F(p_2(F(p_1^{-1})(u)))(g(F(p_1^{-1})(u)))) : |F(f'(F(p_1)(F(p_1^{-1})(u))))|$. Now

$$F(p_1)(F(p_1^{-1})(u)) =_{F(a')} u$$

so it is inhabited by, say, $h(a', u)$. Now f' is extensional so

$$(F \circ f')(h(a', u)) = F(\mathrm{ext}_{f'}(F(p_1)(F(p_1^{-1})(u)), u, h(a', u))) : F(f'(F(p_1)(F(p_1^{-$$

We let

$$g'(u) = (F \circ f')(h(a', u))(F(p_2(F(p_1^{-1})(u)))(g(F(p_1^{-1})(u)))) : |F(f'(u))|.$$

To define e' it suffices to prove (in any way possible):

$$(\forall u, u' : |F(a')|)(\forall p : u =_{F(a')} u')[(F \circ f')(p)(g'(u)) =_{(F \circ f')(u')} g'(u')].$$

Using a set-theoretic universe

The above construction of a extensional type-theoretic universe is complicated, mainly because of the transportation functions. A technically simpler solution appears to be to use a set-theoretic universe

$$V = (|V|, =_V)$$

as that in Aczel's model of CZF. (Recall that $|V| = (Wx : \mathrm{U})\mathrm{T}(x)$.)

We build a category $\mathcal{A}_V$ consisting of elements a of V as objects and as morphisms $a \longrightarrow b$ elements f of V which are functions from a to b internally to V.

There is a full and faithful functor $F : \mathcal{A}_V \longrightarrow$ **Setoid** which on objects is

$$F(\sup(a, f)) = (T(a), \sim_f)$$

where $x \sim_f y \iff f(x) =_V f(y)$.

Using a set-theoretic universe

The above construction of a extensional type-theoretic universe is complicated, mainly because of the transportation functions. A technically simpler solution appears to be to use a set-theoretic universe

$$V = (|V|, =_V)$$

as that in Aczel's model of CZF. (Recall that $|V| = (Wx : \mathrm{U})\mathrm{T}(x)$.)

We build a category $\mathcal{A}_V$ consisting of elements a of V as objects and as morphisms $a \longrightarrow b$ elements f of V which are functions from a to b internally to V.

There is a full and faithful functor $F : \mathcal{A}_V \longrightarrow$ **Setoid** which on objects is

$$F(\sup(a, f)) = (T(a), \sim_f)$$

where $x \sim_f y \iff f(x) =_V f(y)$.

Using a set-theoretic universe

The above construction of a extensional type-theoretic universe is complicated, mainly because of the transportation functions. A technically simpler solution appears to be to use a set-theoretic universe

$$V = (|V|, =_V)$$

as that in Aczel's model of CZF. (Recall that $|V| = (Wx : \mathrm{U})\mathrm{T}(x)$.)

We build a category $\mathcal{A}_V$ consisting of elements a of V as objects and as morphisms $a \longrightarrow b$ elements f of V which are functions from a to b internally to V.

There is a full and faithful functor $F : \mathcal{A}_V \longrightarrow$ **Setoid** which on objects is

$$F(\mathrm{sup}(a, f)) = (T(a), \sim_f)$$

where $x \sim_f y \iff f(x) =_V f(y)$.

Using a set-theoretic universe (cont.)

It should be possible to use the Π- and Σ-constructions of CZF for constructing ecwas with the corresponding constructions. This remains to be verified formally.

Some references

- J. Cartmell, Generalised Algebraic theories and Contextual Categories, Annals of Pure and Applied Logic, 32:209-243, 1986.
- M. Hofmann, Extensional Concepts in Intensional Type Theory, Springer 1997.
- M. Hofmann, Syntax and Semantics of Dependent Type Theory. 1994.
- P. Dybjer, Internal type theory, pp 120-134 in TYPES '95, LNCS 1158, 1996.
- A. Buisse and P. Dybjer, Towards formalizing categorical models of type theory in type theory, Electronic Notes in Theoretical Computer Science Volume 196, 22 January 2008, Pages 137-151.