Named variables in categories with families

Erik Palmgren

December 2, 2014

We consider a version of categories with families with named variables in the contexts.

1 Categories with families

We recall the standard version. Categories with families (Dybjer 1996) is one of several equivalent ways of defining categorical semantics for dependent type theories (Hofmann 1997). We recall:

Definition 1.1 A *category with families (cwf)* consists of the following data

(a) A category $\mathcal{C}$ with a terminal object $\top$. This is thought of as the category of contexts and substitutions. For $\Gamma \in \mathcal{C}$ denote by ϵ_Γ the unique morphism $\Gamma \longrightarrow \top$.

(b) For each object Γ of $\mathcal{C}$, a class $\mathrm{Ty}(\Gamma)$ and for each morphism $f : \Delta \longrightarrow \Gamma$, a class function $\mathrm{Ty}(f) : \mathrm{Ty}(\Gamma) \longrightarrow \mathrm{Ty}(\Delta)$, for which use the notation $A\{f\}$ for $\mathrm{Ty}(f)(A)$, to suggest that it is the result of performing the substitution f in the type A. These functions should satisfy, for all $A \in \mathrm{Ty}(\Gamma)$, and $g : \Theta \longrightarrow \Delta$, $f : \Delta \longrightarrow \Gamma$:

- $A\{1_\Gamma\} = A$,
- $A\{f \circ g\} = A\{f\}\{g\}$.

(Thus Ty may be regarded as a functor $\mathcal{C}^{\mathrm{op}} \longrightarrow \mathrm{Class}$.)

(c) For each $A \in \mathrm{Ty}(\Gamma)$, an object $\mathrm{Ext}(\Gamma, A)$ in $\mathcal{C}$, — briefly written $\Gamma.A$ — and a morphism $\mathrm{p}(A) = \mathrm{p}_\Gamma(A) : \Gamma.A \longrightarrow \Gamma$. 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) For each $A \in \mathrm{Ty}(\Gamma)$, there is a class $\mathrm{Tm}(\Gamma, A)$ — thought of as the terms of type A. It should be such that for $f : \Delta \longrightarrow \Gamma$ there is a class function $\mathrm{Tm}(f) : \mathrm{Tm}(\Gamma, A) \to \mathrm{Tm}(\Delta, A\{f\})$, where we write $a\{f\}$ for $\mathrm{Tm}(f)(a)$. It should satisfy the following

- $a\{1_\Gamma\} = a$ for $a \in \mathrm{Tm}(\Gamma, A)$ $(= \mathrm{Tm}(\Gamma, A\{1_\Gamma\}))$.
- $a\{f \circ g\} = a\{f\}\{g\}$ for $a \in \mathrm{Tm}(\Gamma, A)$
  (Note: $\mathrm{Tm}(\Theta, A\{f \circ g\}) = \mathrm{Tm}(\Theta, A\{f\}\{g\})$.)

(e) For each $A \in \mathrm{Ty}(\Delta)$ there is an element $\mathrm{v}_A = \mathrm{v}_{\Gamma,A} \in \mathrm{Tm}(\Delta.A, A\{\mathrm{p}(A)\})$.

(f) For any morphism $f : \Gamma \longrightarrow \Delta$ and $a \in \mathrm{Tm}(\Gamma, A\{f\})$, there is

$$\langle f, a\rangle_A : \Gamma \longrightarrow \Delta.A.$$

This construction should satisfy

- $\mathrm{p}(A) \circ \langle f, a\rangle_A = f$,
- $\mathrm{v}_A\{\langle f, a\rangle_A\} = a$,
  (Note: $\mathrm{v}_A\{\langle f, a\rangle_A\} \in \mathrm{Tm}(\Gamma, A\{\mathrm{p}(A)\}\{\langle f, a\rangle_A\}) = \mathrm{Tm}(\Gamma, A\{f\}))$
- $\langle \mathrm{p}(A) \circ h, \mathrm{v}_A\{h\}\rangle_A = h$ for any $h : \Gamma \longrightarrow \Delta.A$.

and moreover for any $g : \Theta \longrightarrow \Gamma$,

$$\langle f, a\rangle_A \circ g = \langle f \circ g, a\{g\}\rangle_A \tag{1}$$

(Remark: $a\{g\} \in \mathrm{Tm}(\Theta, A\{f\}\{g\}) = \mathrm{Tm}(\Theta, A\{f \circ g\})$.) □

We define for $f : \Theta \longrightarrow \Gamma$ and $A \in \mathrm{Ty}(\Gamma)$ a morphism $\mathrm{q}(f, A) : \Theta.A\{f\} \longrightarrow \Gamma.A$ by

$$\mathrm{q}(f, A) =_{\mathrm{def}} \langle f \circ \mathrm{p}(A\{f\}), \mathrm{v}_{A\{f\}}\rangle_A.$$

The following is a pullback square, satisfying well-known functoriality conditions in the parameter f:

$$\begin{array}{ccc} \Theta.A\{f\} & \xrightarrow{\mathrm{q}(f,A)} & \Gamma.A \\ {\scriptstyle \mathrm{p}(A\{f\})}\downarrow & & \downarrow{\scriptstyle \mathrm{p}(A)} \\ \Theta & \xrightarrow[f]{} & \Gamma \end{array}$$

Lemma 1.1 *(a)* $\mathrm{q}(1_\Gamma, A) = 1_{\Gamma.A}$

(b) $\mathrm{q}(f \circ g, A) = \mathrm{q}(f, A) \circ \mathrm{q}(g, A\{f\})$ *for* $f : \Theta \longrightarrow \Gamma$ *and* $g : \Delta \longrightarrow \Theta$

Proof. (a): $\mathrm{q}(1_\Gamma, A) = \langle 1_\Gamma \circ \mathrm{p}(A\{1_\Gamma\}), \mathrm{v}_{A\{1_\Gamma\}}\rangle_A = \langle \mathrm{p}(A), \mathrm{v}_A\rangle_A = 1_{\Gamma.A}$
(b):

$$\begin{aligned} & \mathrm{q}(f, A) \circ \mathrm{q}(g, A\{f\}) \\ = \ & \langle f \circ \mathrm{p}(A\{f\}), \mathrm{v}_{A\{f\}}\rangle_A \circ \langle g \circ \mathrm{p}(A\{f\}\{g\}), \mathrm{v}_{A\{f\}\{g\}}\rangle_{A\{f\}} \\ = \ & \langle f \circ \mathrm{p}(A\{f\}) \circ \langle g \circ \mathrm{p}(A\{f\}\{g\}), \mathrm{v}_{A\{f\}\{g\}}\rangle_{A\{f\}}, \mathrm{v}_{A\{f\}}\{\langle g \circ \mathrm{p}(A\{f\}\{g\}), \mathrm{v}_{A\{f\}\{g\}}\rangle_{A\{f\}}\}\rangle_A \\ = \ & \langle f \circ g \circ \mathrm{p}(A\{f\}\{g\}), \mathrm{v}_{A\{f\}\{g\}}\}\rangle_A \\ = \ & \langle f \circ g \circ \mathrm{p}(A\{f \circ g\}), \mathrm{v}_{A\{f\circ g\}}\rangle_A = \mathrm{q}(f \circ g, A). \end{aligned}$$

□

1.1

Let $\mathrm{Fm}(\Gamma)$ be the class of families in context $\Gamma \in \mathcal{C}$

$$\{(A, B) : A \in \mathrm{Ty}(\Gamma), B \in \mathrm{Ty}(\Gamma.\sigma)\}$$

For $f : \Theta \longrightarrow \Gamma$ we wish to define a map that performs substitution in a family

$$\mathrm{Fm}(\Gamma) \longrightarrow \mathrm{Fm}(\Theta).$$

To do this we first define for $f : \Theta \longrightarrow \Gamma$ and $A \in \mathrm{Ty}(\Gamma)$ a morphism $\mathrm{q}(f, A) : \Theta.A\{f\} \longrightarrow \Gamma.A$ by

$$\mathrm{q}(f, A) =_{\mathrm{def}} \langle f \circ \mathrm{p}(A\{f\}), \mathrm{v}_{A\{f\}} \rangle_A.$$

Suppose $B \in \mathrm{Ty}(\Gamma.A)$. Then $B\{\mathrm{q}(f, A)\} \in \mathrm{Ty}(\Theta.A\{f\})$. Hence $(A\{f\}, B\{\mathrm{q}(f, A)\}) \in \mathrm{Fm}(\Theta)$, whenever $(A, B) \in \mathrm{Fm}(\Gamma)$. We write

$$\mathrm{Fm}(f)(A, B) = (A\{f\}, B\{\mathrm{q}(f, A)\}). \tag{2}$$

Moreover, Fm is functorial.

Corollary 1.2 *The operation* Fam *is functorial in the sense that*

(a) $\mathrm{Fm}(1_\Gamma) = \mathrm{id}_{\mathrm{Fm}(\Gamma)}$

(b) For $f : \Theta \longrightarrow \Gamma$ *and* $g : \Delta \longrightarrow \Theta$,

$$\mathrm{Fm}(f \circ g) = \mathrm{Fm}(g) \circ \mathrm{Fam}(f). \Box$$

For $(A, B) \in \mathrm{Fm}(\Gamma)$, and $f : \Theta \longrightarrow \Gamma$ we write

$$(A, B)\{f\} = \mathrm{Fm}(f)(A, B).$$

Lemma 1.2 $\mathrm{q}(f, A) \circ \langle g, N \rangle = \langle f \circ g, N \rangle$. $\Box$

For $\Gamma \in \mathcal{C}$ and $A \in \mathrm{Ty}(\Gamma)$ we use the notation

$$\mathrm{Fam}(\Gamma, A) =_{\mathrm{def}} \mathrm{Ty}(\Gamma.A).$$

1.2 Type generated cwfs

A cwf $\mathcal{C}$ is *type generated,* if for every $\Gamma \in \mathcal{C}$ there is a sequence $A_1 \in \mathrm{Ty}(\top)$, $A_2 \in \mathrm{Ty}(\top.A_1)$,..., $A_n \in \mathrm{Ty}(\top.A_1.\ldots.A_{n-1})$, $n \geq 0$ such that

$$\Gamma = \top.A_1.\ldots.A_n. \tag{3}$$

Let $\Theta \in \mathcal{C}$. There is a unique morphism $!_\Theta : \Theta \longrightarrow \top$. For $a_1 \in \mathrm{Tm}(\Theta, A_1\{!_\Theta\})$, we have

$$\langle !_\Theta, a_1 \rangle_{A_1} : \Theta \longrightarrow \top.A_1.$$

For $a_2 \in \mathrm{Tm}(\Theta, A_2\{\langle !_\Theta, a_1 \rangle_{A_1}\})$, we get

$$\langle \langle !_\Theta, a_1 \rangle_{A_1}, a_2 \rangle_{A_2} : \Theta \longrightarrow \top.A_1.A_2.$$

For $a_3 \in \mathrm{Tm}(\Theta, A_3\{\langle \langle !_\Theta, a_1 \rangle_{A_1}, a_2 \rangle_{A_2}\})$, we get

$$\langle \langle \langle !_\Theta, a_1 \rangle_{A_1}, a_2 \rangle_{A_2} a_3 \rangle_{A_3} : \Theta \longrightarrow \top.A_1.A_2.A_3$$

We write

$$\langle a_1, a_2, \ldots, a_n \rangle = \langle \cdots \langle \langle !, a_1 \rangle, a_2, \rangle \ldots, a_n \rangle$$

Note that $\langle\rangle = !$ and $\langle a_1 \rangle = \langle !, a_1 \rangle$.

1.3 Π-type construction

We consider here only one variable binding type construction.

A cwf *supports* Π*-types* if for $A \in \mathrm{Ty}(\Gamma)$ and $B \in \mathrm{Fm}(\Gamma, A)$ there is a type $\Pi_\Gamma(A,B) \in \mathrm{Ty}(\Gamma)$, and moreover for every $b \in \mathrm{Tm}(\Gamma.A, B)$ there is an element $\lambda_{\Gamma,A,B}(b) \in \mathrm{Tm}(\Gamma, \Pi_\Gamma(A,B))$, and furthermore for any $c \in \mathrm{Tm}(\Gamma, \Pi_\Gamma(A,B))$ and any $a \in \mathrm{Tm}(\Gamma, A)$ this is an element $\mathrm{App}_{\Gamma,A,B}(c,a) \in \mathrm{Tm}(\Gamma, B\{\langle 1_\Gamma, a\rangle_A\})$, such that the following equations hold for any $f : \Theta \longrightarrow \Gamma$:

(β-conv) $\mathrm{App}_{\Gamma,A,B}(\lambda_{\Gamma,A,B}(b), a) = b\{\langle 1_\Gamma, a\rangle_A\}$,

(Π-subst) $\Pi_\Gamma(A,B)\{f\} = \Pi_\Theta(A\{f\}, B\{\mathrm{q}(f,A)\})$,

(λ-subst) $\lambda_{\Gamma,A,B}(b)\{f\} = \lambda_{\Theta,A\{f\},B\{\mathsf{q}(f,A)\}}(b\{\mathrm{q}(f,A)\})$,

(App-subst) $\mathrm{App}_{\Gamma,A,B}(c,a)\{f\} = \mathrm{App}_{\Theta,A\{f\},B\{\mathrm{q}(f,A)\}}(c\{f\}, a\{f\})$.

To formulate the η-conversion law we reason as follows. For $c \in \mathrm{Tm}(\Gamma, \Pi_\Gamma(A,B))$:

$$c\{p(A)\} \in \mathrm{Tm}(\Gamma.A, \Pi_\Gamma(A,B)\{p(A)\}) = \mathrm{Tm}(\Gamma.A, \Pi_{\Gamma.A}(A\{p(A)\}, B\{\mathrm{q}(p(A),A)\}))$$

and

$$v_A \in \mathrm{Tm}(\Gamma.A, A\{p(A)\})$$

$$\begin{aligned}\mathrm{App}_{\Gamma.A,A\{p(A)\},B\{\mathrm{q}(p(A),A)\}}(c\{p(A)\}, v_A) &\in \mathrm{Tm}(\Gamma.A, B\{\mathrm{q}(p(A),A)\}\{\langle 1_{\Gamma.A}, v_A\rangle\})\\ &\in \mathrm{Tm}(\Gamma.A, B\{\mathrm{q}(p(A),A) \circ \langle 1_{\Gamma.A}, v_A\rangle\})\\ &\in \mathrm{Tm}(\Gamma.A, B\{\langle p(A), v_A\rangle\}) = \mathrm{Tm}(\Gamma.A, B).\end{aligned}$$

The η-conversion law is thus:

(η-conv) $c = \lambda_{\Gamma,A,B}(\mathrm{App}_{\Gamma.A,A\{p(A)\},B\{\mathrm{q}(p(A),A)\}}(c\{p(A)\}, v_A)) \in \mathrm{Tm}(\Gamma, \Pi_\Gamma(A,B))$.

We say that a cwf *supports extensional* Π*-types* if it *supports* Π*-types* and in addition satisfies (β-conv).

1.4 Bounded cwfs

A small variation of the notion of cwf is that of a *bounded cwf*. Here it is assumed that there are sets T and E (the bounds) such that for all $\Gamma \in \mathcal{C}$, and $A \in \mathrm{Ty}(\Gamma)$,

$$\mathrm{Ty}(\Gamma) \subseteq T \text{ and } \mathrm{Tm}(\Gamma, A) \subseteq E.$$

Any cwf can be turned into bounded cwf by letting T and E be the images of Ty and Tm respectively.

2 Cwfs with named variables

Let V be an infinite discrete set. The set V is considered as the stock of variables. Let $V^\#$ denote the set of finite sequences

$$\mathbf{x} = x_1, x_2, \ldots, x_n = ((\cdots(((), x_1), x_2), \cdots), x_n)$$

from V, where $x_i \neq x_j$ whenever $i \neq j$. Now we introduce a new notion of cwf by modifying the definitions above.

Definition 2.1 A *category with families and named variables (cwfn)* consists of the following data

(a1) A category $\mathcal{C}$ with a terminal object $\top$.

(a2) A function $\mathrm{n} : \mathrm{Ob}(\mathcal{C}) \longrightarrow V^{\#}$ assigning each context a list of different variables intended to be the names of projections. It is required that $\mathrm{n}(\top) = ()$, the empty list. Define

$$\mathrm{Fr}(\Gamma) = \{x \in V : x \text{ is not in } \mathrm{n}(\Gamma)\}.$$

(b) For each object Γ of $\mathcal{C}$, a class $\mathrm{Ty}(\Gamma)$ and for each morphism $f : \Delta \longrightarrow \Gamma$, a class function $\mathrm{Ty}(f) : \mathrm{Ty}(\Gamma) \longrightarrow \mathrm{Ty}(\Delta)$, for which use the notation $A\{f\}$ for $\mathrm{Ty}(f)(A)$, to suggest that it is the result of performing the substitution f in the type A. Ty is a functor $\mathcal{C}^{\mathrm{op}} \longrightarrow \mathrm{Class}$.

(c) For each $A \in \mathrm{Ty}(\Gamma)$, and each $x \in \mathrm{Fr}(\Gamma)$, an object $\mathrm{Ext}(\Gamma, x, A)$ in $\mathcal{C}$, — briefly written $\Gamma.(x : A)$ — and a morphism $\mathrm{p}(x : A) = \mathrm{p}_\Gamma(x : A) : \Gamma.(x : A) \longrightarrow \Gamma$. It is required that

$$\mathrm{n}(\mathrm{Ext}(\Gamma, x, A)) = (\mathrm{n}(\Gamma), x).$$

(d) For each $A \in \mathrm{Ty}(\Gamma)$, there is a class $\mathrm{Tm}(\Gamma, A)$ — thought of as the terms of type A. It should be such that for $f : \Delta \longrightarrow \Gamma$ there is a class function $\mathrm{Tm}(f) : \mathrm{Tm}(\Gamma, A) \to \mathrm{Tm}(\Delta, A\{f\})$, where we write $a\{f\}$ for $\mathrm{Tm}(f)(a)$. It should satisfy the following

- $a\{1_\Gamma\} = a$ for $a \in \mathrm{Tm}(\Gamma, A)$
- $a\{f \circ g\} = a\{f\}\{g\}$ for $a \in \mathrm{Tm}(\Gamma, A)$

(e) For each $A \in \mathrm{Ty}(\Delta)$, and each $x \in \mathrm{Fr}(\Delta)$, there is an element

$$\mathrm{v}_{(x:A)} = \mathrm{v}_{\Delta,(x:A)} \in \mathrm{Tm}(\Delta.(x : A), A\{\mathrm{p}(x : A)\}).$$

(f) For any morphism $f : \Gamma \longrightarrow \Delta$, $A \in \mathrm{Ty}(\Delta)$, and each $x \in \mathrm{Fr}(\Delta)$, and $a \in \mathrm{Tm}(\Gamma, A\{f\})$, there is

$$\langle f, a\rangle_{(x:A)} : \Gamma \longrightarrow \Delta.(x : A).$$

We use the suggestive notation $\langle f, x := a\rangle_A$ for this morphism. It should satisfy

- $\mathrm{p}(x : A) \circ \langle f, x := a\rangle_A = f$,
- $\mathrm{v}_{(x:A)}\{\langle f, x := a\rangle_A\} = a$,
- $\langle \mathrm{p}(x : A) \circ h, x := \mathrm{v}_{(x:A)}\{h\}\rangle_A = h$ for any $h : \Gamma \longrightarrow \Delta.(x : A)$.

and moreover for any $g : \Theta \longrightarrow \Gamma$,

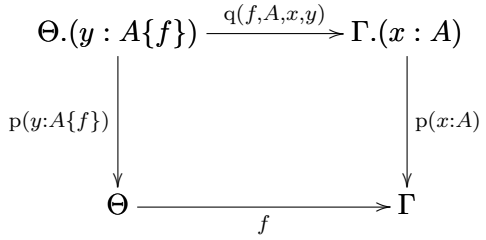
$$\langle f, x := a\rangle_A \circ g = \langle f \circ g, x := a\{g\}\rangle_A \tag{4}$$

□

We define for $f : \Theta \longrightarrow \Gamma$, $A \in \mathrm{Ty}(\Gamma)$, $y \in \mathrm{Fr}(\Theta)$ and $x \in \mathrm{Fr}(\Gamma)$, a morphism $\mathrm{q}(f, A, x, y) : \Theta.(y : A\{f\}) \longrightarrow \Gamma.(x : A)$ by

$$\mathrm{q}(f, A, x, y) =_{\mathrm{def}} \langle f \circ \mathrm{p}(y : A\{f\}), x := \mathrm{v}_{(y:A\{f\})}\rangle_A.$$

The following is a pullback square, satisfying well-known functoriality conditions in the parameter f:



2.1 Π-type construction in a cwfn

A cwfn *supports* Π-*types* if for $A \in \mathrm{Ty}(\Gamma)$, $x \in \mathrm{Fr}(\Gamma)$, and $B \in \mathrm{Ty}(\Gamma.(x : A))$ there is a type $\Pi_\Gamma(x, A, B) \in \mathrm{Ty}(\Gamma)$, and moreover for every $b \in \mathrm{Tm}(\Gamma.(x : A), B)$ there is an element $\lambda_{\Gamma,A,B}(x, b) \in \mathrm{Tm}(\Gamma, \Pi_\Gamma(x, A, B))$, and furthermore for any $c \in \mathrm{Tm}(\Gamma, \Pi_\Gamma(x, A, B))$ and any $a \in \mathrm{Tm}(\Gamma, A)$ there is an element $\mathrm{App}_{\Gamma,x,A,B}(c, a) \in \mathrm{Tm}(\Gamma, B\{\langle 1_\Gamma, x := a\rangle_A\})$, such that the following equations hold for any $f : \Theta \longrightarrow \Gamma$:

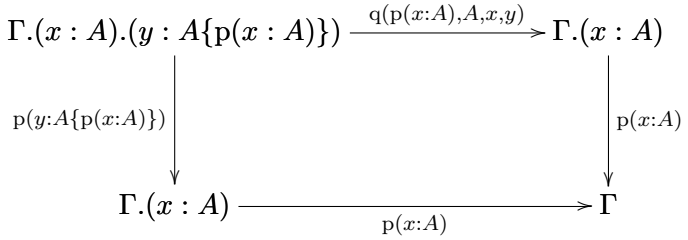
(β-conv) $\mathrm{App}_{\Gamma,A,B}(\lambda_{\Gamma,A,B}(x, b), a) = b\{\langle 1_\Gamma, x := a\rangle_A\}$,

(Π-subst) $\Pi_\Gamma(x, A, B)\{f\} = \Pi_\Theta(y, A\{f\}, B\{\mathrm{q}(f, A, x, y)\})$, provided $y \in \mathrm{Fr}(\Theta)$

(λ-subst) $\lambda_{\Gamma,A,B}(x, b)\{f\} = \lambda_{\Theta,A\{f\},B\{\mathrm{q}(f,A,x,y)\}}(y, b\{\mathrm{q}(f, A, x, y)\})$, provided $y \in \mathrm{Fr}(\Theta)$

(App-subst) $\mathrm{App}_{\Gamma,x,A,B}(c, a)\{f\} = \mathrm{App}_{\Theta,y,A\{f\},B\{\mathrm{q}(f,A,x,y)\}}(c\{f\}, a\{f\})$, provided $y \in \mathrm{Fr}(\Theta)$.

To formulate the η-rule, we consider the pullback square



where $y \in \mathrm{Fr}(\Gamma.x : A)$. For $c \in \mathrm{Tm}(\Gamma, \Pi_\Gamma(x, A, B))$:

$$\begin{array}{rcl} c\{p(x : A)\} & \in & \mathrm{Tm}(\Gamma.x : A, \Pi_\Gamma(x, A, B)\{p(x : A)\}) \\ & = & \mathrm{Tm}(\Gamma.x : A, \Pi_{\Gamma.x:A.y:A\{\mathrm{p}(x:A)\}}(y, A\{p(x : A)\}, B\{\mathrm{q}(p(x : A), A, x, y)\})) \end{array}$$

and

$$v_{(x:A)} \in \mathrm{Tm}(\Gamma.x : A, A\{p(x : A)\})$$

$$\begin{array}{l} \mathrm{App}_{\Gamma.x:A,y,A\{\mathrm{p}(x:A)\},B\{\mathrm{q}(\mathrm{p}(x:A),A,x,y)\}}(c\{p(x : A)\}, v_{(x:A)}) \\ \quad \in \ \mathrm{Tm}(\Gamma.x : A, B\{\mathrm{q}(\mathrm{p}(x : A), A, x, y)\}\{\langle 1_{\Gamma.x:A}, v_{(x:A)}\rangle\}) \\ \quad \in \ \mathrm{Tm}(\Gamma.x : A, B\{\mathrm{q}(\mathrm{p}(x : A), A, x, y) \circ \langle 1_{\Gamma.x:A}, v_{(x:A)}\rangle\}) \\ \quad \in \ \mathrm{Tm}(\Gamma.x : A, B\{\langle \mathrm{p}(x : A), v_{(x:A)}\rangle\}) = \mathrm{Tm}(\Gamma.x : A, B). \end{array}$$

The η-conversion law is thus:

(η-conv)

$$\begin{array}{rcl} c & = & \lambda_{\Gamma,A,B}(x, \mathrm{App}_{\Gamma.x:A,y,A\{\mathrm{p}(x:A)\},B\{\mathrm{q}(\mathrm{p}(x:A),A,x,y)\}}(c\{p(x:A)\}, v_{(x:A)})) \\ & \in & \mathrm{Tm}(\Gamma, \Pi_\Gamma(x, A, B)). \end{array}$$

We say that a cwfn *supports extensional* Π-*types* if it *supports* Π-*types* and in addition satisfies (β-conv).

2.2 Analyzing the syntax

We analyze the syntax as dependent types and relations to see which arguments can be taken as implicit.

- $\Gamma : \mathrm{Ctx} \Longrightarrow \mathrm{Ty}(\Gamma) : \mathrm{Set}$
- $\Gamma, \Delta : \mathrm{Ctx}, f : \mathrm{Sbs}(\Gamma, \Delta), A : \mathrm{Ty}(\Delta) \Longrightarrow \mathrm{Tysubst}(\Gamma, \Delta, f, A) : \mathrm{Ty}(\Gamma) : \mathrm{Set}$
- $\Gamma : \mathrm{Ctx}, A : \mathrm{Ty}(\Gamma) \Longrightarrow \mathrm{Tm}(\Gamma, A) : \mathrm{Set}$
- $\Gamma, \Delta : \mathrm{Ctx}, f : \mathrm{Sbs}(\Gamma, \Delta), A : \mathrm{Ty}(\Delta), a : \mathrm{Tm}(\Delta, A) \Longrightarrow \mathrm{Tmsubst}(\Gamma, \Delta, f, A, a) : \mathrm{Tm}(\Gamma, \mathrm{Tysubst}(\Gamma, \Delta, f, A))$
- $\Gamma : \mathrm{Ctx}, x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Gamma) \Longrightarrow \mathrm{Ext}(\Gamma, x, A) : \mathrm{Ctx}$
- $\Gamma : \mathrm{Ctx}, x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Gamma) \Longrightarrow \mathrm{p}(\Gamma, x, A) : \mathrm{Sbs}(\mathrm{Ext}(\Gamma, x, A), \Gamma)$
- $\Gamma : \mathrm{Ctx}, x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Gamma) \Longrightarrow \mathrm{v}(\Gamma, x, A) : \mathrm{Tm}(\mathrm{Ext}(\Gamma, x, A), \mathrm{Tysubst}(\mathrm{Ext}(\Gamma, x, A), \Gamma, \mathrm{p}(\Gamma, x, A), A))$
- $\Gamma, \Delta : \mathrm{Ctx}, f : \mathrm{Sbs}(\Gamma, \Delta), x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Delta), a : \mathrm{Tm}(\Gamma, \mathrm{Tysubst}(\Gamma, \Delta, f, A)) \Longrightarrow \mathrm{ext}(\Gamma, \Delta, f, x, A, a) : \mathrm{Sbs}(\Gamma, \mathrm{Ext}(\Delta, x, A))$
- $\Gamma : \mathrm{Ctx}, x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Gamma), B : \mathrm{Ty}(\mathrm{Ext}(\Gamma, x, A)) \Longrightarrow \Pi(\Gamma, x, A, B) : \mathrm{Ty}(\Gamma)$
- $\Gamma : \mathrm{Ctx}, x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Gamma), B : \mathrm{Ty}(\mathrm{Ext}(\Gamma, x, A)), b : \mathrm{Tm}(\mathrm{Ext}(\Gamma, x, A), B) \Longrightarrow \lambda(\Gamma, x, A, B, b) \in \mathrm{Tm}(\Gamma, \Pi(\Gamma, x, A, B))$
- $\Gamma : \mathrm{Ctx}, x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Gamma), B : \mathrm{Ty}(\mathrm{Ext}(\Gamma, x, A)), c : \mathrm{Tm}(\Gamma, \Pi(\Gamma, x, A, B)), a : \mathrm{Tm}(\Gamma, A) \Longrightarrow \mathrm{App}(\Gamma, x, A, B, c, a) \in \mathrm{Tm}(\Gamma, \mathrm{Tysubst}(\Gamma, \mathrm{Ext}(\Gamma, x, A), \mathrm{ext}(\Gamma, \Gamma, 1_\Gamma, x, A, a), B))$

Using hidden parameters

- $\Gamma : \mathrm{Ctx} \Longrightarrow \mathrm{Ty}(\Gamma) : \mathrm{Set}$
- $\Gamma, \Delta : \mathrm{Ctx}, f : \mathrm{Sbs}(\Gamma, \Delta), A : \mathrm{Ty}(\Delta) \Longrightarrow \mathrm{Tysubst}(f, A) : \mathrm{Ty}(\Gamma)$
- $\Gamma : \mathrm{Ctx}, A : \mathrm{Ty}(\Gamma) \Longrightarrow \mathrm{Tm}(\Gamma, A) : \mathrm{Set}$
- $\Gamma, \Delta : \mathrm{Ctx}, f : \mathrm{Sbs}(\Gamma, \Delta), A : \mathrm{Ty}(\Delta), a : \mathrm{Tm}(\Delta, A) \Longrightarrow \mathrm{Tmsubst}(f, a) : \mathrm{Tm}(\Gamma, \mathrm{Tysubst}(f, A))$
- $\Gamma : \mathrm{Ctx}, x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Gamma) \Longrightarrow \mathrm{Ext}(\Gamma, x, A) : \mathrm{Ctx}$
- $\Gamma : \mathrm{Ctx}, x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Gamma) \Longrightarrow \mathrm{p}(x, A) : \mathrm{Sbs}(\mathrm{Ext}(\Gamma, x, A), \Gamma)$
- $\Gamma : \mathrm{Ctx}, x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Gamma) \Longrightarrow \mathrm{v}(x, A) : \mathrm{Tm}(\mathrm{Ext}(\Gamma, x, A), \mathrm{Tysubst}(\mathrm{p}(x, A), A))$

- $\Gamma, \Delta : \mathrm{Ctx}, f : \mathrm{Sbs}(\Gamma, \Delta), x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Delta), a : \mathrm{Tm}(\Gamma, \mathrm{Tysubst}(f, A)) \Longrightarrow \mathrm{ext}(f, x, A, a) : \mathrm{Sbs}(\Gamma, \mathrm{Ext}(\Delta, x, A))$
- $\Gamma : \mathrm{Ctx}, x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Gamma), B : \mathrm{Ty}(\mathrm{Ext}(\Gamma, x, A)) \Longrightarrow \Pi(x, A, B) : \mathrm{Ty}(\Gamma)$
- $\Gamma : \mathrm{Ctx}, x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Gamma), B : \mathrm{Ty}(\mathrm{Ext}(\Gamma, x, A)), b : \mathrm{Tm}(\mathrm{Ext}(\Gamma, x, A), B) \Longrightarrow \lambda(x, b) \in \mathrm{Tm}(\Gamma, \Pi(\Gamma, x, A, B))$
- $\Gamma : \mathrm{Ctx}, x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Gamma), B : \mathrm{Ty}(\mathrm{Ext}(\Gamma, x, A)), c : \mathrm{Tm}(\Gamma, \Pi(\Gamma, x, A, B)), a : \mathrm{Tm}(\Gamma, A) \Longrightarrow \mathrm{App}(c, a) \in \mathrm{Tm}(\Gamma, \mathrm{Tysubst}(\mathrm{ext}(1_\Gamma, x, A, a), B))$

With the notations

$$\begin{aligned}
\Gamma \longrightarrow \Delta &= \mathrm{Sbs}(\Gamma, \Delta) \\
A\{f\} &= \mathrm{Tysubst}(f, A) \\
a\{f\} &= \mathrm{Tmsubst}(f, a) \\
\Gamma.(x : A) &= \mathrm{Ext}(\Gamma, x, A) \\
\mathrm{p}(x : A) &= \mathrm{p}(x, A) \\
\mathrm{v}_{(x:A)} &= \mathrm{v}(x, A) \\
\langle f, x := a \rangle_A &= \mathrm{ext}(f, x, A, a) \\
\Pi(A, (x)B) &= \Pi(x, A, B) \\
\lambda((x)b) &= \lambda(x, b)
\end{aligned}$$

we can rewrite the above as

- $\Gamma : \mathrm{Ctx} \Longrightarrow \mathrm{Ty}(\Gamma) : \mathrm{Set}$
- $\Gamma, \Delta : \mathrm{Ctx}, f : \Gamma \longrightarrow \Delta, A : \mathrm{Ty}(\Delta) \Longrightarrow A\{f\} : \mathrm{Ty}(\Gamma) : \mathrm{Set}$
- $\Gamma : \mathrm{Ctx}, A : \mathrm{Ty}(\Gamma) \Longrightarrow \mathrm{Tm}(\Gamma, A) : \mathrm{Set}$
- $\Gamma, \Delta : \mathrm{Ctx}, f : \Gamma \longrightarrow \Delta, A : \mathrm{Ty}(\Delta), a : \mathrm{Tm}(\Delta, A) \Longrightarrow a\{f\} : \mathrm{Tm}(\Gamma, A\{f\})$
- $\Gamma : \mathrm{Ctx}, x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Gamma) \Longrightarrow \Gamma.(x : A) : \mathrm{Ctx}$
- $\Gamma : \mathrm{Ctx}, x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Gamma) \Longrightarrow \mathrm{p}(x, A) : \Gamma.(x : A) \longrightarrow \Gamma$
- $\Gamma : \mathrm{Ctx}, x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Gamma) \Longrightarrow \mathrm{v}(x, A) : \mathrm{Tm}(\Gamma.(x : A), A\{\mathrm{p}(x, A)\})$
- $\Gamma, \Delta : \mathrm{Ctx}, f : \Gamma \longrightarrow \Delta, x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Delta), a : \mathrm{Tm}(\Gamma, A\{f\}) \Longrightarrow \langle f, x := a \rangle_A : \Gamma \longrightarrow \Gamma.(x : A)$
- $\Gamma : \mathrm{Ctx}, x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Gamma), B : \mathrm{Ty}(\Gamma.(x : A)) \Longrightarrow \Pi(A, (x)B) : \mathrm{Ty}(\Gamma)$
- $\Gamma : \mathrm{Ctx}, x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Gamma), B : \mathrm{Ty}(\Gamma.(x : A)), b : \mathrm{Tm}(\Gamma.(x : A), B) \Longrightarrow \lambda((x)b) : \mathrm{Tm}(\Gamma, \Pi(\Gamma, A, (x)B))$
- $\Gamma : \mathrm{Ctx}, x : \mathrm{Fr}(\Gamma), A : \mathrm{Ty}(\Gamma), B : \mathrm{Ty}(\Gamma.(x : A)), c : \mathrm{Tm}(\Gamma, \Pi(\Gamma, A, (x)B)), a : \mathrm{Tm}(\Gamma, A) \Longrightarrow \mathrm{App}(c, a) \in \mathrm{Tm}(\Gamma, B\{\langle 1_\Gamma, x := a \rangle_A\})$

3 Demonstration

We drop several of the subscripts and annotations in these examples. We also write

$$\langle x_1 := a_1, x_2 := a_2, \ldots, x_n := a_n\rangle = \langle \ldots \langle \langle !, x_1 := a_1\rangle, x_2 := a_2\rangle, \ldots, x_n := a_n\rangle$$

Note that $\langle\rangle = !$ and $\langle x_1 := a_1\rangle = \langle !, x_1 := a_1\rangle$.

For a context $\Gamma = \top.x_1 : A_1.x_2 : A_2.\ldots.x_n : A_n$ we note that its identity is

$$1_\Gamma = \langle x_1 := \mathrm{v}_{x_1}, x_2 := \mathrm{v}_{x_2}, \ldots, x_n := \mathrm{v}_{x_n}\rangle.$$

Consider first a simple example with a binary relation R between two types and a function f.

Let $A, B \in \mathrm{Ty}(\top)$. Then $\top.x : A \in \mathcal{C}$ and $\mathrm{p} = ! : \top.x : A \longrightarrow \top$. Thus $B\{\mathrm{p}\} \in \mathrm{Ty}(\top.x : A)$. Form $\top.x : A.y : B\{\mathrm{p}\} :\in \mathcal{C}$.

Let $R \in \mathrm{Ty}(\top.x : A.y : B\{\mathrm{p}\})$ and $f \in \mathrm{Tm}(\top.x : A, B\{\mathrm{p}\})$. Now $1_{\top.x:A} : \top.x : A \longrightarrow \top.x : A$ gives

$$\langle 1_{\top.x:A}, y := f\rangle : \top.x : A \longrightarrow \top.x : A, y : B\{\mathrm{p}\}$$

and hence

$$R\{\langle 1_{\top.x:A}, y := f\rangle\} \in \mathrm{Ty}(\top.x : A).$$

Rewriting the identity and using $f = f\{1_{\top.x:A}\}$ we get

$$R\{\langle x := \mathrm{v}_x, y := f\{x := \mathrm{v}_x\}\rangle\} \in \mathrm{Ty}(\top.x : A).$$

Thus

$$\Pi(A, (x)R\{\langle x := \mathrm{v}_x, y := f\{x := \mathrm{v}_x\}\rangle\}) \in \mathrm{Ty}(\top).$$

Assume furthermore that

$$g \in \mathrm{Tm}(\top.x : A, R\{\langle x := \mathrm{v}_x, y := f\{x := \mathrm{v}_x\}\rangle\}).$$

Rewriting the identity this becomes

$$g\{\langle x := \mathrm{v}_x\rangle\} \in \mathrm{Tm}(\top.x : A, R\{\langle x := \mathrm{v}_x, y := f\{x := \mathrm{v}_x\}\rangle\}).$$

Thus

$$\lambda((x)g\{\langle x := \mathrm{v}_x\rangle\}) \in \Pi(A, (x)R\{\langle x := \mathrm{v}_x, y := f\{x := \mathrm{v}_x\}\rangle\}) \in \mathrm{Ty}(\top),$$

where $(\lambda x : A)b$ is short for $\lambda_{\Gamma,A,B}(x, b)$. Writing $E[a_1, \ldots, a_n]$ for $E\{\langle x_1 := a_1, \ldots, x_n := a_n\}$ and E type or term in the context $\top.x_1 : A_1.x_2 : A_2.\ldots.x_n : A_n$, we can express this as

$$\lambda((x)g[\mathrm{v}_x]) \in \Pi(A, (x)R[\mathrm{v}_x, f[\mathrm{v}_x]]) \in \mathrm{Ty}(\top),$$

We can form $\top.y : B \in \mathcal{C}$ and $\top.y : B.x : A\{p\} \in \mathcal{C}$. Find the variable swapping substitution

$$h : \top.y : B.x : A\{p\} \in \mathcal{C} \longrightarrow \top.x : A.y : B\{\mathrm{p}\}$$

First find the projection

$$h_1 : \top.y : B.x : A\{p\} \in \mathcal{C} \longrightarrow \top.x : A.$$

We have $! : \top.y : B.x : A\{p\} \in \mathcal{C} \longrightarrow \top$ and $\mathrm{v}_x \in \mathrm{Tm}(\top.y : B.x : A\{p\}, A\{p\}\{p\})$. But $p \circ p =!$ and $A\{p\}\{p\} = A\{p \circ p\} = A\{!\}$ so

$$h_1 =_{\mathrm{def}} \langle x := \mathrm{v}_x \rangle = \langle !, x := \mathrm{v}_x \rangle : \top.y : B.x : A\{p\} \longrightarrow \top.x : A$$

We have $\mathrm{v}_y \in \mathrm{Tm}(\top.y : B, B\{!\})$ since $p =! : \top.x : B \longrightarrow \top$. Thus

$$\begin{aligned} \mathrm{v}_y\{p\} \in \mathrm{Tm}(\top.y : B.x : A\{p\}, B\{!\}\{p\}) &= \mathrm{Tm}(\top.y : B.x : A\{p\}, B\{!\}) \\ &= \mathrm{Tm}(\top.y : B.x : A\{p\}, B\{! \circ h_1\}) \\ &= \mathrm{Tm}(\top.y : B.x : A\{p\}, B\{!\}\{h_1\}) \end{aligned}$$

Hence

$$\langle h_1, y := \mathrm{v}_y\{p\} \rangle : \top.y : B.x : A\{p\} \longrightarrow \top.x : A.y : B\{p\}.$$

That is

$$\langle x := \mathrm{v}_x, y := \mathrm{v}_y\{p\} \rangle : \top.y : B.x : A\{p\} \longrightarrow \top.x : A.y : B\{p\}$$

Thus

$$R\{\langle x := \mathrm{v}_x, y := \mathrm{v}_y\{p\} \rangle\} \in \mathrm{Ty}(\top.y : B.x : A\{p\}).$$

With the abbreviation principle above

$$R[\mathrm{v}_x, \mathrm{v}_y\{p\}] \in \mathrm{Ty}(\top.y : B.x : A\{p\}).$$

Furthermore

$$\Pi(A\{p\}, (x)R[\mathrm{v}_x, \mathrm{v}_y\{p\}]) \in \mathrm{Ty}(\top.y : B). \tag{5}$$

This suggests that projections p should be forgotten in writing out the formulas in order to produce the standard variableful presentation. The last would then be

$$\Pi(A, (x)R[\mathrm{v}_x, \mathrm{v}_y]) \in \mathrm{Ty}(\top.y : B).$$

Consider the substitution

$$k =_{\mathrm{def}} \langle y := \mathrm{v}_x \rangle : \top.x : B \longrightarrow \top.y : B$$

and apply it to (5)

$$\Pi(A\{p\}, (x)R[\mathrm{v}_x, \mathrm{v}_y\{p\}])\{\langle y := \mathrm{v}_x \rangle\} \tag{6}$$

The associated pullback square, with top most morphism

$$\mathrm{q}(k, A\{p\}, x, z) = \langle k \circ \mathrm{p}, x := \mathrm{v}_z \rangle = \langle \langle y := \mathrm{v}_x \rangle \circ \mathrm{p}, x := \mathrm{v}_z \rangle = \langle y := \mathrm{v}_x\{\mathrm{p}\}, x := \mathrm{v}_z \rangle$$

is

$$\begin{array}{ccc} \top.x : B.z : A\{p\}\{f\} & \xrightarrow{\langle y := \mathrm{v}_x\{\mathrm{p}\}, x := \mathrm{v}_z \rangle} & \top.y : B.x : A\{p\} \\ \Big\downarrow{\scriptstyle \mathrm{p}} & & \Big\downarrow{\scriptstyle \mathrm{p}} \\ \top.x : B & \xrightarrow[\langle y := \mathrm{v}_x \rangle]{} & \top.y : B \end{array}$$

Now (6) equals

$$(\Pi z : A\{p\}\{\langle y := \mathrm{v}_x \rangle\})R\{\langle x := \mathrm{v}_x, y := \mathrm{v}_y\{p\} \rangle\}\{\langle y := \mathrm{v}_x\{\mathrm{p}\}, x := \mathrm{v}_z \rangle\}$$

provided $z \in \mathrm{Fr}(\top.x : B)$, which is clear. We simplify

$$\begin{aligned}
(\Pi z : A\{p\}\{\langle y := \mathrm{v}_x\rangle\})R\{\langle x := \mathrm{v}_x, y := \mathrm{v}_y\{p\}\rangle\}\{\langle y := \mathrm{v}_x\{\mathrm{p}\}, x := \mathrm{v}_z\rangle\} &= \\
(\Pi z : A\{p \circ \{\langle y := \mathrm{v}_x\rangle\}\}R\{\langle x := \mathrm{v}_x\{\langle y := \mathrm{v}_x\{\mathrm{p}\}, x := \mathrm{v}_z\rangle\}, y := \mathrm{v}_y\{p\}\{\langle y := \mathrm{v}_x\{\mathrm{p}\}, x := \mathrm{v}_z\rangle\}\rangle\} &= \\
(\Pi z : A\{!\})R\{\langle x := \mathrm{v}_z, y := \mathrm{v}_y\{p \circ \langle y := \mathrm{v}_x\{\mathrm{p}\}, x := \mathrm{v}_z\rangle\}\rangle\} &= \\
(\Pi z : A\{!\})R\{\langle x := \mathrm{v}_z, y := \mathrm{v}_y\{\langle y := \mathrm{v}_x\{\mathrm{p}\}\rangle\}\rangle\} &= \\
(\Pi z : A\{!\})R\{\langle x := \mathrm{v}_z, y := \mathrm{v}_x\{\mathrm{p}\}\rangle\} &= \\
(\Pi z : A\{!\})R[\mathrm{v}_z, \mathrm{v}_x\{\mathrm{p}\}] &=
\end{aligned}$$

But $A\{!\} = A\{p\}$ so we have

$$\Pi(A\{p\}, (x)R[\mathrm{v}_x, \mathrm{v}_y\{p\}])\{\langle y := \mathrm{v}_x\rangle\} = \Pi(A\{p\}, (z)R[\mathrm{v}_z, \mathrm{v}_x\{\mathrm{p}\}]),$$

as expected.

4 Projections

Let $\mathcal{C}$ be a cwfn. Define inductively a relation $\succeq$ on the contexts.

- $\Gamma \succeq \Gamma$
- If $\Delta \succeq \Gamma$, $x \in \mathrm{Fr}(\Delta)$, $A \in \mathrm{Ty}(\Delta)$, then $\Delta.(x : A) \succeq \Gamma$.

Whenever $\Delta \succeq \Gamma$ there is a projection $\Delta \longrightarrow \Gamma$ defined as follows

- $\mathrm{p}_{\Gamma,\Gamma} = 1_\Gamma : \Gamma \longrightarrow \Gamma$
- $\mathrm{p}_{\Delta.(x:A),\Gamma} = \mathrm{p}_{\Delta,\Gamma} \circ \mathrm{p}(x : A)$ for $x \in \mathrm{Fr}(\Delta)$, $A \in \mathrm{Ty}(\Delta)$ and $\Delta \succeq \Gamma$.

Note that $\succeq$ is a well-founded partial order on any type generated cwfn.

5 From cwfn to cwf

Suppose that $\mathcal{C}$ is a cwfn with variable set V. Assume furthermore that there is a variable choice function $\mathrm{fr}(\Gamma) \in \mathrm{Fr}(\Gamma)$. Define the data for the cwf $\mathcal{C}'$:

the underlying category is the same as that of $\mathcal{C}$,

$\mathrm{Ty}'(\Gamma) =_{\mathrm{def}} \mathrm{Ty}(\Gamma)$,

$\mathrm{Ext}'(\Gamma, A) =_{\mathrm{def}} \mathrm{Ext}(\Gamma, \mathrm{fr}(\Gamma), A)$,

$\mathrm{p}'_\Gamma(A) =_{\mathrm{def}} \mathrm{p}_\Gamma(\mathrm{fr}(\Gamma) : A)$,

$\mathrm{Tm}'(\Gamma, A) =_{\mathrm{def}} \mathrm{Tm}(\Gamma, A)$,

$\mathrm{v}'_A =_{\mathrm{def}} \mathrm{v}_{(\mathrm{fr}(\Delta):A)}$ for $A \in \mathrm{Ty}(\Delta)$,

$\langle f, a\rangle'_A =_{\mathrm{def}} \langle f, a\rangle'_{(\mathrm{fr}(\Delta):A)}$ for $f : \Gamma \longrightarrow \Delta$, $A \in \mathrm{Ty}(\Delta)$ and $a \in \mathrm{Tm}(\Gamma, A\{f\})$.

Theorem 5.1 *If $\mathcal{C}$ is a cwfn, then $\mathcal{C}'$ is a cwf.* □

Suppose that $\mathcal{C}$ is a cwfn that supports Π-types. Define

$$\Pi'_\Gamma(A,B) = \Pi_\Gamma(\mathrm{fr}(\Gamma), A, B),$$

$$\lambda'_{\Gamma,A,B}(b) = \lambda_{\Gamma,A,B}(\mathrm{fr}(\Gamma), b),$$

$$\mathrm{App}'_{\Gamma,A,B}(c,a) = \mathrm{App}_{\Gamma,\mathrm{fr}(\Gamma),A,B}(c,a).$$

Theorem 5.2 *If $\mathcal{C}$ is a cwfn which support Π-types, then $\mathcal{C}'$ is a cwf which supports Π-types.* □

6 From cwf to cwfn

Suppose that $\mathcal{C}$ is a type generated cwf. Let V be an infinite discrete set, and let $V^\#$ denote the set of finite sequences as in the previous section.

Define a new category $\mathcal{C}^V$ of contexts, whose objects are pairs $(\Gamma;\mathbf{x})$ where $\Gamma \in \mathcal{C}$ and $\mathbf{x} \in V^\#$ and the length of Γ and $\mathbf{x}$ are the same. A context $(\top.A_1.\ldots.A_n; x_1, \ldots, x_n)$ can suggestively be written

$$(x_1 : A_1, \ldots, x_n : A_n).$$

Define $\mathrm{n}((\Gamma;\mathbf{x})) = \mathbf{x}$.

Morphisms $f : (\Gamma;\mathbf{x}) \longrightarrow (\Delta;\mathbf{y})$ are just morphisms $f : \Gamma \longrightarrow \Delta$, and composition is inherited from $\mathcal{C}$. Define

$$\mathrm{Ty}^V((\Gamma;\mathbf{x})) = \mathrm{Ty}(\Gamma)$$

and for $f : (\Gamma;\mathbf{x}) \longrightarrow (\Delta;\mathbf{y})$,

$$\mathrm{Ty}^V(f) = \mathrm{Ty}(f).$$

For extending context we require a freshness condition on variables. For $A \in \mathrm{Ty}^V((\Gamma;\mathbf{x}))$ and $y \in V$ not in $\mathbf{x}$,

$$\mathrm{Ext}^V((\Gamma;\mathbf{x}), y, A) = (\mathrm{Ext}(\Gamma, A), (\mathbf{x}, y)).$$

The set of fresh variables for the enriched context $(\Gamma;\mathbf{x})$ is

$$\mathrm{Fr}((\Gamma;\mathbf{x})) = \{y \in V : y \text{ is not in } \mathrm{n}((\Gamma;\mathbf{x}))\} = \{y \in V : y \text{ is not in } \mathbf{x}\}.$$

Then we can define the projection as

$$\mathsf{p}^V_{(\Gamma;\mathbf{x})}(y : A) = \mathsf{p}^V_\Gamma(A) : \mathrm{Ext}^V((\Gamma,\mathbf{x}), y, A) \longrightarrow (\Gamma;\mathbf{x}).$$

Conditions (a) – (d) are fulfilled. For (e) and (f) we need to take care of the freshness condition, after straightforwardly defining

$$\mathrm{Tm}^V((\Gamma;\mathbf{x}), A) = \mathrm{Tm}(\Gamma, A),$$

$$\mathrm{v}^V_{x:A} = \mathrm{v}_A$$

and

$$\langle f, a\rangle^V_{(x:A)} = \langle f, a\rangle_A.$$

Theorem 6.1 *If $\mathcal{C}$ is a cwf, and V is an infinite discrete set, then $\mathcal{C}^V$ is a cwfn.* □

Suppose that $\mathcal{C}$ is a cwf that supports Π-types. Define

$$\Pi^V_{(\Gamma,\overline{y})}(x, A, B) = \Pi_\Gamma(A, B),$$

$$\lambda^V_{(\Gamma,\overline{y}),A,B}(x, b) = \lambda_{\Gamma,A,B}(b),$$

$$\mathrm{App}^V_{(\Gamma,\overline{y}),x,A,B}(c, a) = \mathrm{App}_{\Gamma,x,A,B}(c, a).$$

Theorem 6.2 *If $\mathcal{C}$ is a cwf which supports Π-types, and V is an infinite discrete set, then $\mathcal{C}^V$ is a cwfn which supports Π-types.* □

7 Martin-Löf's substitution calculus

The first-order part of the Substitution Calculus (Martin-Löf 1992) has the judgment forms

Γ : context

$\Gamma = \Delta$: context

$\gamma : \Delta \longrightarrow \Gamma$

$\gamma = \delta : \Delta \longrightarrow \Gamma$

$\Gamma \Longrightarrow A$: type

$\Gamma \Longrightarrow A = B$: type

$\Gamma \Longrightarrow a : A$

$\Gamma \Longrightarrow a = b : A$

One can add (as in Tasistro 1993) a judgement for context extension

$\Gamma \preceq \Delta$: context

to explicate this notion. There are two rules for context formation

$$\frac{}{() : \text{context}} \qquad \frac{\Gamma : \text{context} \quad \Gamma \Longrightarrow A : \text{type}}{(\Gamma, x : A) : \text{context}}$$

The thinning rules are:

$$\frac{\gamma : \Delta \longrightarrow \Gamma}{\gamma : \Theta \longrightarrow \Gamma}\ (\Theta \text{ is an extension of } \Delta) \qquad \frac{\gamma : \Delta \longrightarrow \Gamma}{\gamma : \Delta \longrightarrow \Theta}\ (\Gamma \text{ is an extension of } \Theta)$$

$$\frac{\gamma : \Delta \Longrightarrow A : \text{type}}{\gamma : \Theta \Longrightarrow A : \text{type}}\ (\Theta \text{ is an extension of } \Delta) \qquad \frac{\gamma : \Delta \Longrightarrow a : A}{\gamma : \Theta \Longrightarrow a : A}\ (\Theta \text{ is an extension of } \Delta)$$

The composition rules are:

$$\frac{\delta : \Theta \longrightarrow \Delta \quad \gamma : \Delta \longrightarrow \Gamma}{\gamma\delta : \Theta \longrightarrow \Gamma}$$

$$\frac{\gamma : \Delta \longrightarrow \Gamma \quad \Gamma \Longrightarrow A : \text{type}}{\Gamma \Longrightarrow A\gamma : \text{type}} \qquad \frac{\gamma : \Delta \longrightarrow \Gamma \quad \Gamma \Longrightarrow a : A}{\Gamma \Longrightarrow a\gamma : A\gamma}$$

Associativity laws:

$$\frac{\theta : \Lambda \longrightarrow \Theta \quad \delta : \Theta \longrightarrow \Delta \quad \gamma : \Delta \longrightarrow \Gamma}{(\gamma\delta)\theta = \gamma(\delta\theta) : \Lambda \longrightarrow \Gamma}$$

$$\frac{\delta : \Theta \longrightarrow \Delta \quad \gamma : \Delta \longrightarrow \Gamma \quad \Gamma \Longrightarrow A : \text{type}}{\Theta \Longrightarrow (A\gamma)\delta = A(\gamma\delta) : \text{type}} \qquad \frac{\delta : \Theta \longrightarrow \Delta \quad \gamma : \Delta \longrightarrow \Gamma \quad \Gamma \Longrightarrow a : A}{\Theta \Longrightarrow (a\gamma)\delta = a(\gamma\delta) : A(\gamma\delta)}$$

Unit laws:

$$\frac{\Gamma : \text{context}}{() : \Gamma \longrightarrow \Gamma}$$

$$\frac{\gamma : \Delta \longrightarrow \Gamma}{()\gamma = \gamma : \Delta \longrightarrow \Gamma} \qquad \frac{\gamma : \Delta \longrightarrow \Gamma}{\gamma() = \gamma : \Delta \longrightarrow \Gamma}$$

$$\frac{\Gamma \longrightarrow A : \text{type}}{\Gamma \longrightarrow A() = A : \text{type}} \qquad \frac{\Gamma \longrightarrow a : A}{\Gamma \longrightarrow a() = a : A}$$

Updating laws:

$$\frac{\gamma : \Delta \longrightarrow \Gamma \quad \Delta \Longrightarrow a : A\gamma}{(\gamma, x = a) : \Delta \longrightarrow \Gamma, x : A}$$

$$\frac{\delta : \Theta \longrightarrow \Delta \quad \gamma : \Delta \longrightarrow \Gamma \quad \Delta \Longrightarrow a : A\gamma}{(\gamma, x = a)\delta = (\gamma\delta, x = a\delta) : \Theta \longrightarrow \Gamma, x : A}$$

$$\frac{\gamma : \Delta \longrightarrow \Gamma \quad \Delta \Longrightarrow a : A\gamma}{(\gamma, x = a) = \gamma : \Delta \longrightarrow \Gamma} \qquad \frac{\gamma : \Delta \longrightarrow \Gamma \quad \Delta \Longrightarrow a : A\gamma}{\Delta \Longrightarrow x(\gamma, x = a) = a : A\gamma}$$

$$\frac{\gamma : \Delta \longrightarrow ()}{\gamma = () : \Delta \longrightarrow ()} \qquad \frac{\gamma : \Delta \longrightarrow \Gamma, x : A}{\gamma = (\gamma, x = a) : \Delta \longrightarrow \Gamma, x : A}$$

$$\frac{\gamma : \Delta \longrightarrow () \quad \gamma : \Delta \longrightarrow ()}{\gamma = \delta : \Delta \longrightarrow ()}$$

$$\frac{\gamma : \Delta \longrightarrow \Gamma, x : A \quad \delta : \Delta \longrightarrow \Gamma, x : A \quad \gamma = \delta : \Delta \longrightarrow \Gamma \quad \Delta \Longrightarrow x\gamma = x\delta : A\gamma}{\gamma = \delta : \Delta \longrightarrow \Gamma, x : A}$$

Variable law:

$$\frac{}{\Gamma \Longrightarrow x : A} \ (x : A \text{ is declared in } \Gamma)$$

References

[1] J. Cartmell. *Generalized algebraic theories and contextual categories.* DPhil. Thesis Oxford 1978.

[2] J. Cartmell. Generalized algebraic theories and contextual categories. *Annals of Pure and Applied Logic,* 32(1986), 209 – 243.

[3] P. Dybjer. Internal Type Theory. In: *Types for Proofs and Programs* (eds. S. Berardi and M. Coppo) Lecture Notes in Computer Science, Vol. 1158, 1996, pp 120 – 134

[4] B. Jacobs. *Categorical Logic and Type Theory.* Elsevier 1999.

[5] M. Hofmann. Syntax and semantics of dependent types. In: *Semantics and Logics of Computation* (eds. A. Pitts and P. Dybjer), Cambridge University Press 1997.

[6] P. Martin-Löf. *The substitution calculus.* Notes from a seminar 1992.

[7] A. Tasistro. Formulation of Martin-Löf's theory of types with explicit substitutions. Licentiate Thesis, Chalmers University and Göteborg University 1993.