Sets, Setoids and Groupoids

Erik Palmgren
Uppsala University

Workshop on Constructive Aspects of Logic and Mathematics
Kanazawa, March 8-12, 2010.

Families of sets

In set theories such as, ZF or CZF, a family of sets can always be represented as a function $\beta : B \longrightarrow A$. The fibers of β (inverses of singletons) represent the sets of the family and A is the index set:

$$B_x = \beta^{-1}(x) = \{b \in B : \beta(b) = x\} \qquad (x \in A)$$

This representation is possible by the replacement scheme, since any family given by a set-theoretic formula (F is the set associated with the index x)

$$(\forall x \in A)(\exists! F)\varphi(x, F)$$

can be turned into a family represented by a function.

Families of sets

In set theories such as, ZF or CZF, a family of sets can always be represented as a function $\beta : B \longrightarrow A$. The fibers of β (inverses of singletons) represent the sets of the family and A is the index set:

$$B_x = \beta^{-1}(x) = \{b \in B : \beta(b) = x\} \qquad (x \in A)$$

This representation is possible by the replacement scheme, since any family given by a set-theoretic formula (F is the set associated with the index x)

$$(\forall x \in A)(\exists! F)\varphi(x, F)$$

can be turned into a family represented by a function.

Setoids

In type theories is 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.

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 is 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.

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

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 $x =_A x'$

Families of setoids

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

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

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.

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

"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.

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

Proof-irrelevant family

(I) For a proof-irrelevant family we require

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

(2) $\phi_q \circ \phi_p =_{\text{ext}} \phi_r$, whenever $p : x =_A y, q : y =_A z, r : x =_A z$. Here $=_{ext}$ 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

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

Proof-irrelevant family

(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 $=_{ext}$ 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-relevant family

(R) Of a proof relevant family we require only

(a) $\phi_{\mathrm{ref}(x)} =_{\mathrm{ext}} \mathrm{id}_{B_x}$

(b) $\phi_{\mathrm{trans}(q,p)} =_{\mathrm{ext}} \phi_q \circ \phi_p$ for $p : x =_A y$ and $q : y =_A z$

(c) $\phi_{\mathrm{sym}(p)} \circ \phi_p =_{\mathrm{ext}} \mathrm{id}_{B_x}$ for $p : x =_A y$,

(d) $\phi_p \circ \phi_{\mathrm{sym}(p)} =_{\mathrm{ext}} \mathrm{id}_{B_y}$ for $p : x =_A y$.

Here $\mathrm{ref}(x) : x =_A x$, a proof object for reflexivity. Moreover the proof objects associated with symmetry and transitivity are $\mathrm{sym}(p) : y =_A x$, for $p : x =_A y$, and $\mathrm{trans}(q, p) : x =_A z$ for $p : x =_A y$ and $q : y =_A z$.

Note: For $p : x =_A x$, the function ϕ_p may be a non-trivial automorphism on B_x.

Proof-relevant family

(R) Of a proof relevant family we require only

(a) $\phi_{\mathrm{ref}(x)} =_{\mathrm{ext}} \mathrm{id}_{B_x}$

(b) $\phi_{\mathrm{trans}(q,p)} =_{\mathrm{ext}} \phi_q \circ \phi_p$ for $p : x =_A y$ and $q : y =_A z$

(c) $\phi_{\mathrm{sym}(p)} \circ \phi_p =_{\mathrm{ext}} \mathrm{id}_{B_x}$ for $p : x =_A y$,

(d) $\phi_p \circ \phi_{\mathrm{sym}(p)} =_{\mathrm{ext}} \mathrm{id}_{B_y}$ for $p : x =_A y$.

Here $\mathrm{ref}(x) : x =_A x$, a proof object for reflexivity. Moreover the proof objects associated with symmetry and transitivity are $\mathrm{sym}(p) : y =_A x$, for $p : x =_A y$, and $\mathrm{trans}(q, p) : x =_A z$ for $p : x =_A y$ and $q : y =_A z$.

Note: For $p : x =_A x$, the function ϕ_p may be a non-trivial automorphism on B_x.

Proof-relevant family

(R) Of a proof relevant family we require only

(a) $\phi_{\mathrm{ref}(x)} =_{\mathrm{ext}} \mathrm{id}_{B_x}$

(b) $\phi_{\mathrm{trans}(q,p)} =_{\mathrm{ext}} \phi_q \circ \phi_p$ for $p : x =_A y$ and $q : y =_A z$

(c) $\phi_{\mathrm{sym}(p)} \circ \phi_p =_{\mathrm{ext}} \mathrm{id}_{B_x}$ for $p : x =_A y$,

(d) $\phi_p \circ \phi_{\mathrm{sym}(p)} =_{\mathrm{ext}} \mathrm{id}_{B_y}$ for $p : x =_A y$.

Here $\mathrm{ref}(x) : x =_A x$, a proof object for reflexivity. Moreover the proof objects associated with symmetry and transitivity are $\mathrm{sym}(p) : y =_A x$, for $p : x =_A y$, and $\mathrm{trans}(q, p) : x =_A z$ for $p : x =_A y$ and $q : y =_A z$.

Note: For $p : x =_A x$, the function ϕ_p may be a non-trivial automorphism on B_x.

Proof-relevant family

(R) Of a proof relevant family we require only

(a) $\phi_{\mathrm{ref}(x)} =_{\mathrm{ext}} \mathrm{id}_{B_x}$

(b) $\phi_{\mathrm{trans}(q,p)} =_{\mathrm{ext}} \phi_q \circ \phi_p$ for $p : x =_A y$ and $q : y =_A z$

(c) $\phi_{\mathrm{sym}(p)} \circ \phi_p =_{\mathrm{ext}} \mathrm{id}_{B_x}$ for $p : x =_A y$,

(d) $\phi_p \circ \phi_{\mathrm{sym}(p)} =_{\mathrm{ext}} \mathrm{id}_{B_y}$ for $p : x =_A y$.

Here $\mathrm{ref}(x) : x =_A x$, a proof object for reflexivity. Moreover the proof objects associated with symmetry and transitivity are $\mathrm{sym}(p) : y =_A x$, for $p : x =_A y$, and $\mathrm{trans}(q, p) : x =_A z$ for $p : x =_A y$ and $q : y =_A z$.

Note: For $p : x =_A x$, the function ϕ_p may be a non-trivial automorphism on B_x.

Consider a function $f : B \to A$ between setoids. Define the fiber of f over x by

$$f^{-1}(x) =_{\mathrm{def}} ((\Sigma z : B)(f(z) =_A x), \sim),$$

where $(z, p) \sim (z', p')$ holds if and only if $z =_B z'$.

For $q : x =_A x'$ let $f^{-1}(q) : f^{-1}(x) \to f^{-1}(x')$ be given by

$$f^{-1}(q)(z, p) = (z, \mathrm{trans}(q, p)).$$

This clearly defines a proof-irrelevant family of setoids.

Consider a function $f : B \to A$ between setoids. Define the fiber of f over x by

$$f^{-1}(x) =_{\mathrm{def}} ((\Sigma z : B)(f(z) =_A x), \sim),$$

where $(z, p) \sim (z', p')$ holds if and only if $z =_B z'$.

For $q : x =_A x'$ let $f^{-1}(q) : f^{-1}(x) \to f^{-1}(x')$ be given by

$$f^{-1}(q)(z, p) = (z, \mathrm{trans}(q, p)).$$

This clearly defines a proof-irrelevant family of setoids.

Example: category of setoids with equality on objects

Let (B, ϕ) be a family of setoids indexed by the setoid A.

- The collection of objects of the category $\mathcal{B}$ is the setoid A.
- An arrow of the category is a triple (a, f, b) where $a, b : |A|$ and $f : B_a \longrightarrow B_b$ is an extensional function.
- Two arrows (a, f, b) and (a', f', b') are equal if there are $p : a =_A a'$ and $q : b =_A b'$ so that

$$\begin{array}{ccc} B_a & \xrightarrow{\ f\ } & B_b \\ {\scriptstyle \phi_p}\downarrow & & \downarrow{\scriptstyle \phi_q} \\ B_{a'} & \xrightarrow[\ f'\ ]{} & B_{b'} \end{array}$$

commutes (extensionally).

Example: category of setoids with equality on objects

Let (B, ϕ) be a family of setoids indexed by the setoid A.

- The collection of objects of the category $\mathcal{B}$ is the setoid A.
- An arrow of the category is a triple (a, f, b) where $a, b : |A|$ and $f : B_a \longrightarrow B_b$ is an extensional function.
- Two arrows (a, f, b) and (a', f', b') are equal if there are $p : a =_A a'$ and $q : b =_A b'$ so that

$$\begin{array}{ccc} B_a & \xrightarrow{f} & B_b \\ {\scriptstyle \phi_p}\downarrow & & \downarrow{\scriptstyle \phi_q} \\ B_{a'} & \xrightarrow[f']{} & B_{b'} \end{array}$$

commutes (extensionally).

Example: category of setoids with equality on objects

Let (B, ϕ) be a family of setoids indexed by the setoid A.

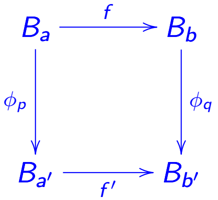
- The collection of objects of the category $\mathcal{B}$ is the setoid A.
- An arrow of the category is a triple (a, f, b) where $a, b : |A|$ and $f : B_a \longrightarrow B_b$ is an extensional function.
- Two arrows (a, f, b) and (a', f', b') are equal if there are $p : a =_A a'$ and $q : b =_A b'$ so that

commutes (extensionally).

Example: category of setoids with equality on objects

Let (B, ϕ) be a family of setoids indexed by the setoid A.

- The collection of objects of the category $\mathcal{B}$ is the setoid A.
- An arrow of the category is a triple (a, f, b) where $a, b : |A|$ and $f : B_a \longrightarrow B_b$ is an extensional function.
- Two arrows (a, f, b) and (a', f', b') are equal if there are $p : a =_A a'$ and $q : b =_A b'$ so that

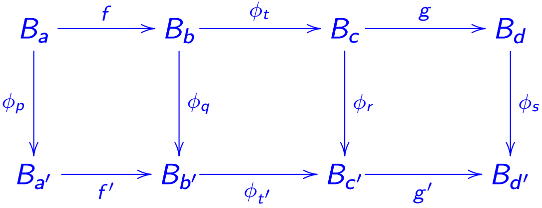


commutes (extensionally).

Arrows (a, f, b) and (c, g, d) are composable if there is $t : b =_A c$. Their composition is $(a, g \circ \phi_t \circ f, d)$. Now the problem arises when proving that composition respects equality of arrows

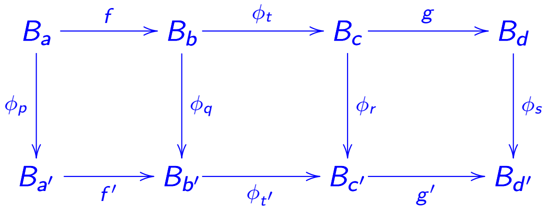
If we have a proof-irrelevant family the center square commutes, proving that composition respects equality of arrows. This is impossible in the proof-relevant version, unless some higher-order structure is required of categories.

Arrows (a, f, b) and (c, g, d) are composable if there is $t : b =_A c$. Their composition is $(a, g \circ \phi_t \circ f, d)$. Now the problem arises when proving that composition respects equality of arrows



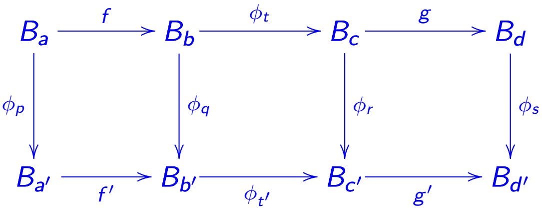
If we have a proof-irrelevant family the center square commutes, proving that composition respects equality of arrows. This is impossible in the proof-relevant version, unless some higher-order structure is required of categories.

Arrows (a, f, b) and (c, g, d) are composable if there is $t : b =_A c$. Their composition is $(a, g \circ \phi_t \circ f, d)$. Now the problem arises when proving that composition respects equality of arrows



If we have a proof-irrelevant family the center square commutes, proving that composition respects equality of arrows. This is impossible in the proof-relevant version, unless some higher-order structure is required of categories.

Arrows (a, f, b) and (c, g, d) are composable if there is $t : b =_A c$. Their composition is $(a, g \circ \phi_t \circ f, d)$. Now the problem arises when proving that composition respects equality of arrows



If we have a proof-irrelevant family the center square commutes, proving that composition respects equality of arrows. This is impossible in the proof-relevant version, unless some higher-order structure is required of categories.

Identity types

For any type A and any $a, b : A$ the type $\mathrm{I}(A, a, b)$of proofs that a and b are identical may be formed.

The introduction rule is

$$\frac{a : A}{\mathrm{r}(a) : \mathrm{I}(A, a, a)}.$$

The elimination rule for I with respect to the family $C(x, y, z)$ type $\quad (x, y : A, z : \mathrm{I}(A, x, y))$ is

$$\frac{a, b : A \qquad c : \mathrm{I}(A, a, b) \qquad d(x) : C(x, x, r(x))\ (x : A)}{\mathrm{J}_{C,a,b}(c, d) : C(a, b, c)}.$$

The associated computation rule is $\mathrm{J}_{C,a,a}(\mathrm{r}(a), d) = d(a)$.

Identity types

For any type A and any $a, b : A$ the type $\mathrm{I}(A, a, b)$of proofs that a and b are identical may be formed.

The introduction rule is

$$\frac{a : A}{\mathrm{r}(a) : \mathrm{I}(A, a, a)}.$$

The elimination rule for I with respect to the family $C(x, y, z)$ type $\quad (x, y : A, z : \mathrm{I}(A, x, y))$ is

$$\frac{a, b : A \qquad c : \mathrm{I}(A, a, b) \qquad d(x) : C(x, x, r(x))\ (x : A)}{\mathrm{J}_{C,a,b}(c, d) : C(a, b, c)}.$$

The associated computation rule is $\mathrm{J}_{C,a,a}(\mathrm{r}(a), d) = d(a)$.

Identity types

For any type A and any $a, b : A$ the type $\mathrm{I}(A, a, b)$of proofs that a and b are identical may be formed.

The introduction rule is

$$\frac{a : A}{\mathrm{r}(a) : \mathrm{I}(A, a, a)}.$$

The elimination rule for I with respect to the family $C(x, y, z)\ \mathrm{type} \quad (x, y : A, z : \mathrm{I}(A, x, y))$ is

$$\frac{a, b : A \qquad c : \mathrm{I}(A, a, b) \qquad d(x) : C(x, x, r(x))\ (x : A)}{\mathrm{J}_{C,a,b}(c, d) : C(a, b, c)}.$$

The associated computation rule is $\mathrm{J}_{C,a,a}(\mathrm{r}(a), d) = d(a)$.

Identity types gives projective setoids

For any type A , $\mathrm{I}(A, \cdot, \cdot)$ is the finest equivalence relation on A. Indeed, if $\approx$ is an equivalence on relation on A then by I-elimination:

$$\mathrm{I}(A, x, y) \Longrightarrow x \approx y.$$

For any type A, define $A^* = (A, \mathrm{I}(A, \cdot, \cdot))$. The setoid A^* is projective, that is, if $g : B \longrightarrow A^*$ is any surjective function between setoids then there is a function $f : A^* \longrightarrow B$ such that $g \circ f = \mathrm{id}_{A^*}$.

Or in other words, choice functions exists on A^*.

Identity types gives projective setoids

For any type A , $\mathrm{I}(A,\cdot,\cdot)$ is the finest equivalence relation on A. Indeed, if $\approx$ is an equivalence on relation on A then by I-elimination:

$$\mathrm{I}(A,x,y) \Longrightarrow x \approx y.$$

For any type A, define $A^* = (A, \mathrm{I}(A,\cdot,\cdot))$. The setoid A^* is projective, that is, if $g : B \longrightarrow A^*$ is any surjective function between setoids then there is a function $f : A^* \longrightarrow B$ such that $g \circ f = \mathrm{id}_{A^*}$.

Or in other words, choice functions exists on A^*.

Identity types gives projective setoids

For any type A , $\mathrm{I}(A, \cdot, \cdot)$ is the finest equivalence relation on A. Indeed, if $\approx$ is an equivalence on relation on A then by I-elimination:

$$\mathrm{I}(A, x, y) \Longrightarrow x \approx y.$$

For any type A, define $A^* = (A, \mathrm{I}(A, \cdot, \cdot))$. The setoid A^* is projective, that is, if $g : B \longrightarrow A^*$ is any surjective function between setoids then there is a function $f : A^* \longrightarrow B$ such that $g \circ f = \mathrm{id}_{A^*}$.

Or in other words, choice functions exists on A^*.

Reindexing maps

A typical use of I-elimination is to derive a rule for substituting equals for equals in a proposition, or equivalently under the propositions-as-types principle, to derive a reindexing operation for families.

For $B(x)$ type $(x : A)$, define $C(x,y,z) = B(x) \to B(y)$. Then $d(x) = \mathrm{id}_{B(x)} = \lambda p : B(x).p : C(x,x,\mathrm{r}(x))$. Hence for $c : \mathrm{I}(A,a,b)$

$$\mathrm{J}_{C,a,b}(c,(x)\mathrm{id}_{B(x)}) : C(a,b,c) = B(a) \to B(b). \tag{1}$$

Define

$$\mathrm{R}_{B,a,b}(c,q) = \mathrm{J}_{C,a,b}(c,(x)\mathrm{id}_{B(x)})(q) : B(b)$$

for $q : B(a)$. Clearly $\mathrm{R}_{B,a,a}(\mathrm{r}(a),q) = q$.

Reindexing maps

A typical use of I-elimination is to derive a rule for substituting equals for equals in a proposition,or equivalently under the propositions-as-types principle, to derive a reindexing operation for families.

For $B(x)$ type $\quad (x : A)$, define $C(x, y, z) = B(x) \to B(y)$. Then $d(x) = \mathrm{id}_{B(x)} = \lambda p : B(x).p : C(x, x, \mathrm{r}(x))$. Hence for $c : \mathrm{I}(A, a, b)$

$$\mathrm{J}_{C,a,b}(c, (x)\mathrm{id}_{B(x)}) : C(a, b, c) = B(a) \to B(b). \tag{1}$$

Define

$$\mathrm{R}_{B,a,b}(c, q) = \mathrm{J}_{C,a,b}(c, (x)\mathrm{id}_{B(x)})(q) : B(b)$$

for $q : B(a)$. Clearly $\mathrm{R}_{B,a,a}(\mathrm{r}(a), q) = q$.

Reindexing maps

A typical use of I-elimination is to derive a rule for substituting equals for equals in a proposition,or equivalently under the propositions-as-types principle, to derive a reindexing operation for families.

For $B(x)$ type $\quad (x : A)$, define $C(x, y, z) = B(x) \to B(y)$. Then $d(x) = \mathrm{id}_{B(x)} = \lambda p : B(x).p : C(x, x, \mathrm{r}(x))$. Hence for $c : \mathrm{I}(A, a, b)$

$$\mathrm{J}_{C,a,b}(c, (x)\mathrm{id}_{B(x)}) : C(a, b, c) = B(a) \to B(b). \tag{1}$$

Define

$$\mathrm{R}_{B,a,b}(c, q) = \mathrm{J}_{C,a,b}(c, (x)\mathrm{id}_{B(x)})(q) : B(b)$$

for $q : B(a)$. Clearly $\mathrm{R}_{B,a,a}(\mathrm{r}(a), q) = q$.

Reindexing maps

A typical use of I-elimination is to derive a rule for substituting equals for equals in a proposition,or equivalently under the propositions-as-types principle, to derive a reindexing operation for families.

For $B(x)\ \mathrm{type} \quad (x : A)$, define $C(x, y, z) = B(x) \rightarrow B(y)$. Then $d(x) = \mathrm{id}_{B(x)} = \lambda p : B(x).p : C(x, x, \mathrm{r}(x))$. Hence for $c : \mathrm{I}(A, a, b)$

$$\mathrm{J}_{C,a,b}(c, (x)\mathrm{id}_{B(x)}) : C(a, b, c) = B(a) \rightarrow B(b). \tag{1}$$

Define

$$\mathrm{R}_{B,a,b}(c, q) = \mathrm{J}_{C,a,b}(c, (x)\mathrm{id}_{B(x)})(q) : B(b)$$

for $q : B(a)$. Clearly $\mathrm{R}_{B,a,a}(\mathrm{r}(a), q) = q$.

Using the I-elimination rule one constructs operations for proofs of symmetry and transitivity

- $\mathrm{id}_a = \mathrm{r}(a) : \mathrm{I}(A, a, a)$
- $c^{-1} : \mathrm{I}(A, b, a) \qquad (a, b : A, c : \mathrm{I}(A, a, b))$,
  where $c^{-1} = J_{C,a,b}(c, \mathrm{r})$ and $C(x, y, z) = \mathrm{I}(A, y, x)$,
- $w \circ z : \mathrm{I}(A, a, u) \qquad (a, b, u : A, z : \mathrm{I}(A, a, b), w : \mathrm{I}(A, b, u))$,
  where $w \circ z = \mathrm{J}_{C,a,b}(z, d)(w)$, $C(x, y, z') = \mathrm{I}(A, y, u) \to \mathrm{I}(A, x, u)$ and $d(x) = \lambda s : \mathrm{I}(A, x, u).s$.

Groupoid structure on types

These operations satisfy the *groupoid laws* with $\mathrm{r}(x)$ as identity in the sense that the following equalities hold:

(G1) $\mathrm{I}_{\mathrm{I}(A,x,y)}(\mathrm{r}(y) \circ z, z)$,

(G2) $\mathrm{I}_{\mathrm{I}(A,x,y)}(z \circ \mathrm{r}(x), z)$,

(G3) $\mathrm{I}_{\mathrm{I}(A,x,x)}(z \circ z^{-1}, \mathrm{r}(x))$,

(G4) $\mathrm{I}_{\mathrm{I}(A,x,x)}(z^{-1} \circ z, \mathrm{r}(x))$,

(G5) $\mathrm{I}_{\mathrm{I}(A,x,v)}((z \circ w) \circ p, z \circ (w \circ p))$.

M. Hofmann and T. Streicher 1993-1996: The groupoid interpretation of type theory.

The reindexing operation R is functorial in the sense that

(R1) $\mathrm{I}_{B(a)}(\mathrm{R}(\mathrm{r}(a), w), w)$ holds for $a : A$, $w : B(a)$,

(R2) $\mathrm{I}_{B(c)}(\mathrm{R}(t, \mathrm{R}(s, w)), \mathrm{R}((t \circ s), w))$ holds for $a, b, c : A$ and $s : \mathrm{I}(A, a, b)$ and $t : \mathrm{I}(A, b, c)$ and $w : B(a)$.

Proof relevant families of projective setoids

For a family of types $B : (A)\mathrm{type}$ we obtain a standard family of setoids as follows.

Define $A^* = (A, \mathrm{I}(A, \cdot, \cdot))$ and $B^*(a) = (B(a), \mathrm{I}(B(a), \cdot, \cdot))$ and define $\phi_p : B^*(a) \to B^*(b)$, by $\phi_p(x) = \mathrm{R}_{B,a,b}(p, x)$ for $p : \mathrm{I}(A, a, b)$.

I-elimination gives

$$\mathrm{I}(\mathrm{I}(A, a, b), p, q) \Longrightarrow \phi_p =_{\mathrm{ext}} \phi_q.$$

The groupoid laws G1 – G5 gives with $\mathrm{ref(x)} = \mathrm{r}(x)$, $\mathrm{sym}(p) = p^{-1}$ and $\mathrm{trans}(q, p) = q \circ p$ the following theorem:

Theorem. For any family of types $B : (A)\mathrm{type}$ the standard family of setoids (A^*, B^*) is proof-relevant.

Proof relevant families of projective setoids

For a family of types $B : (A)\mathrm{type}$ we obtain a standard family of setoids as follows.

Define $A^* = (A, \mathrm{I}(A, \cdot, \cdot))$ and $B^*(a) = (B(a), \mathrm{I}(B(a), \cdot, \cdot))$ and define $\phi_p : B^*(a) \to B^*(b)$, by $\phi_p(x) = \mathrm{R}_{B,a,b}(p, x)$ for $p : \mathrm{I}(A, a, b)$.

I-elimination gives

$$\mathrm{I}(\mathrm{I}(A, a, b), p, q) \Longrightarrow \phi_p =_{\mathrm{ext}} \phi_q.$$

The groupoid laws G1 – G5 gives with $\mathrm{ref}(\mathrm{x}) = \mathrm{r}(x)$, $\mathrm{sym}(p) = p^{-1}$ and $\mathrm{trans}(q, p) = q \circ p$ the following theorem:

Theorem. For any family of types $B : (A)\mathrm{type}$ the standard family of setoids (A^*, B^*) is proof-relevant.

Proof relevant families of projective setoids

For a family of types $B : (A)\mathrm{type}$ we obtain a standard family of setoids as follows.

Define $A^* = (A, \mathrm{I}(A, \cdot, \cdot))$ and $B^*(a) = (B(a), \mathrm{I}(B(a), \cdot, \cdot))$ and define $\phi_p : B^*(a) \to B^*(b)$, by $\phi_p(x) = \mathrm{R}_{B,a,b}(p, x)$ for $p : \mathrm{I}(A, a, b)$.

I-elimination gives

$$\mathrm{I}(\mathrm{I}(A, a, b), p, q) \Longrightarrow \phi_p =_{\mathrm{ext}} \phi_q.$$

The groupoid laws G1 – G5 gives with $\mathrm{ref}(\mathrm{x}) = \mathrm{r}(x)$, $\mathrm{sym}(p) = p^{-1}$ and $\mathrm{trans}(q, p) = q \circ p$ the following theorem:

Theorem. For any family of types $B : (A)\mathrm{type}$ the standard family of setoids (A^*, B^*) is proof-relevant.

Uniqueness of identity proofs?

The identity proofs of A are said to be unique in case

$$(\forall z, w : \mathrm{I}(A, a, b))\mathrm{I}_{\mathrm{I}(A,a,b)}(z, w) \qquad (\mathrm{UIP}_A)$$

holds. We say that UIP holds if for each type A satisfies UIP_A. Hofmann and Streicher 1995 showed that this need not hold for general types by exhibiting a groupoid model of type theory.

Theorem. Let $A : \mathrm{type}$ be fixed. Then UIP_A holds if and only if the standard family of setoids (A^*, B^*) is proof-irrelevant, for any family of types $B : (A)\mathrm{type}$ over A.

Uniqueness of identity proofs?

The identity proofs of A are said to be unique in case

$$(\forall z, w : \mathrm{I}(A, a, b))\mathrm{I}_{\mathrm{I}(A,a,b)}(z, w) \qquad (\mathrm{UIP}_A)$$

holds. We say that UIP holds if for each type A satisfies UIP_A. Hofmann and Streicher 1995 showed that this need not hold for general types by exhibiting a groupoid model of type theory.

Theorem. Let $A : \mathrm{type}$ be fixed. Then UIP_A holds if and only if the standard family of setoids (A^*, B^*) is proof-irrelevant, for any family of types $B : (A)\mathrm{type}$ over A.

Decidable identity types

Theorem (Hedberg 1995). If $(\forall x, y : A)(\mathrm{I}(A, x, y) \vee \neg \mathrm{I}(A, x, y))$, then

$$(\forall x, y : A)(\forall u, v : \mathrm{I}(A, x, y))\mathrm{I}(\mathrm{I}(A, x, y), u, v).$$

Thus UIP is always true in classical extensions of type theory.

Examining the proof one can see that a somewhat stronger result follows:

Theorem. Let $x : A$ be fixed. If $(\forall y : A)(\mathrm{I}(A, x, y) \vee \neg \mathrm{I}(A, x, y))$, then

$$(\forall y : A)(\forall u, v : \mathrm{I}(A, x, y))\mathrm{I}(\mathrm{I}(A, x, y), u, v).$$

Decidable identity types

Theorem (Hedberg 1995). If $(\forall x, y : A)(\mathrm{I}(A, x, y) \vee \neg\mathrm{I}(A, x, y))$, then

$$(\forall x, y : A)(\forall u, v : \mathrm{I}(A, x, y))\mathrm{I}(\mathrm{I}(A, x, y), u, v).$$

Thus UIP is always true in classical extensions of type theory.

Examining the proof one can see that a somewhat stronger result follows:

Theorem. Let $x : A$ be fixed. If $(\forall y : A)(\mathrm{I}(A, x, y) \vee \neg\mathrm{I}(A, x, y))$, then

$$(\forall y : A)(\forall u, v : \mathrm{I}(A, x, y))\mathrm{I}(\mathrm{I}(A, x, y), u, v).$$

Decidable identity types

Theorem (Hedberg 1995). If $(\forall x, y : A)(\mathrm{I}(A, x, y) \vee \neg \mathrm{I}(A, x, y))$, then

$$(\forall x, y : A)(\forall u, v : \mathrm{I}(A, x, y))\mathrm{I}(\mathrm{I}(A, x, y), u, v).$$

Thus UIP is always true in classical extensions of type theory.

Examining the proof one can see that a somewhat stronger result follows:

Theorem. Let $x : A$ be fixed. If $(\forall y : A)(\mathrm{I}(A, x, y) \vee \neg \mathrm{I}(A, x, y))$, then

$$(\forall y : A)(\forall u, v : \mathrm{I}(A, x, y))\mathrm{I}(\mathrm{I}(A, x, y), u, v).$$

Streicher 1993 suggested to supplement the standard J operator with an additional elimination operator K given by the rules for
$D(x,z)\ \mathrm{type} \quad (x : A, z : \mathrm{I}(A,x,x))$

$$\frac{c : \mathrm{I}(A,a,a) \qquad d(x) : D(x, r(x)) \quad (x : A)}{\mathrm{K}_{D,a}(c,d) : D(a,c)}$$

and where $\mathrm{K}_{D,a}(r(a),d) = d(a)$.

It seems possible to justify the rule by following standard meaning explanations with closed terms.

Theorem (Streicher). Using K the principle UIP can be proved.

Streicher 1993 suggested to supplement the standard J operator with an additional elimination operator K given by the rules for
$D(x,z)\ \mathrm{type} \quad (x : A, z : \mathrm{I}(A,x,x))$

$$\frac{c : \mathrm{I}(A,a,a) \qquad d(x) : D(x, r(x)) \quad (x : A)}{\mathrm{K}_{D,a}(c,d) : D(a,c)}$$

and where $\mathrm{K}_{D,a}(r(a),d) = d(a)$.

It seems possible to justify the rule by following standard meaning explanations with closed terms.

Theorem (Streicher). Using K the principle UIP can be proved.

Streicher 1993 suggested to supplement the standard J operator with an additional elimination operator K given by the rules for
$D(x,z)$ type $\quad (x : A, z : \mathrm{I}(A,x,x))$

$$\frac{c : \mathrm{I}(A,a,a) \qquad d(x) : D(x,r(x)) \quad (x : A)}{\mathrm{K}_{D,a}(c,d) : D(a,c)}$$

and where $\mathrm{K}_{D,a}(r(a),d) = d(a)$.

It seems possible to justify the rule by following standard meaning explanations with closed terms.

Theorem (Streicher). Using K the principle UIP can be proved.

The rules J and K may be combined into a single elimination rule: for $C(x, y, u, v)$ type $\quad (x : A, y : A, u : \mathrm{I}(A, x, y), v : \mathrm{I}(A, x, y))$ we have

$$\frac{c : \mathrm{I}(A, a, b) \quad c' : \mathrm{I}(A, a, b) \qquad d(x) : C(x, x, r(x), r(x)) \quad (x : A)}{\mathrm{J}^2_{C,a,b}(c, c', d) : C(a, b, c, c')}$$

with $\mathrm{J}^2_{C,a,a}(r(a), r(a), d) = d(a)$.

Theorem. J^2 is equivalent to the combination of J and K.

However neither K nor J^2 follow the usual pattern of elimination rules.

The rules J and K may be combined into a single elimination rule: for $C(x, y, u, v)\ \mathrm{type} \quad (x : A, y : A, u : \mathrm{I}(A, x, y), v : \mathrm{I}(A, x, y))$ we have

$$\frac{c : \mathrm{I}(A, a, b) \quad c' : \mathrm{I}(A, a, b) \qquad d(x) : C(x, x, r(x), r(x)) \quad (x : A)}{\mathrm{J}^2_{C,a,b}(c, c', d) : C(a, b, c, c')}$$

with $\mathrm{J}^2_{C,a,a}(r(a), r(a), d) = d(a)$.

Theorem. J^2 is equivalent to the combination of J and K.

However neither K nor J^2 follow the usual pattern of elimination rules.

The Hofmann–Streicher groupoid model of type theory suggests that groupoids could be used as fundamental objects of mathematics instead of sets (or setoids). The type-theoretic version of a groupoid is an E-category where all morphisms are invertible. To be explicit:

A groupoid $A = (|A|, \mathrm{Hom}, \mathrm{id}, \circ, (\ )^{-1})$ consists of

- a type $|A|$,
- a setoid $\mathrm{Hom}(a, b)$ of morphisms for any $a, b : |A|$,
- an identity morphism $\mathrm{id}_a \in \mathrm{Hom}(a, a)$ for each $a : |A|$,
- a composition operation $\circ : \mathrm{Hom}(b, c) \times \mathrm{Hom}(a, b) \longrightarrow \mathrm{Hom}(a, c)$ for all $a, b, c : |A|$,
- an inversion $(\ )^{-1} : \mathrm{Hom}(a, b) \longrightarrow \mathrm{Hom}(b, a)$ for $a, b : |A|$,

satisfying standard identities.

The Hofmann–Streicher groupoid model of type theory suggests that groupoids could be used as fundamental objects of mathematics instead of sets (or setoids). The type-theoretic version of a groupoid is an E-category where all morphisms are invertible. To be explicit:

A groupoid $A = (|A|, \mathrm{Hom}, \mathrm{id}, \circ, (\ )^{-1})$ consists of

- a type $|A|$,
- a setoid $\mathrm{Hom}(a, b)$ of morphisms for any $a, b : |A|$,
- an identity morphism $\mathrm{id}_a \in \mathrm{Hom}(a, a)$ for each $a : |A|$,
- a composition operation $\circ : \mathrm{Hom}(b, c) \times \mathrm{Hom}(a, b) \longrightarrow \mathrm{Hom}(a, c)$ for all $a, b, c : |A|$,
- an inversion $(\ )^{-1} : \mathrm{Hom}(a, b) \longrightarrow \mathrm{Hom}(b, a)$ for $a, b : |A|$,

satisfying standard identities.

The Hofmann–Streicher groupoid model of type theory suggests that groupoids could be used as fundamental objects of mathematics instead of sets (or setoids). The type-theoretic version of a groupoid is an E-category where all morphisms are invertible. To be explicit:

A groupoid $A = (|A|, \mathrm{Hom}, \mathrm{id}, \circ, (\ )^{-1})$ consists of

- a type $|A|$,
- a setoid $\mathrm{Hom}(a, b)$ of morphisms for any $a, b : |A|$,
- an identity morphism $\mathrm{id}_a \in \mathrm{Hom}(a, a)$ for each $a : |A|$,
- a composition operation $\circ : \mathrm{Hom}(b, c) \times \mathrm{Hom}(a, b) \longrightarrow \mathrm{Hom}(a, c)$ for all $a, b, c : |A|$,
- an inversion $(\ )^{-1} : \mathrm{Hom}(a, b) \longrightarrow \mathrm{Hom}(b, a)$ for $a, b : |A|$,

satisfying standard identities.

The Hofmann–Streicher groupoid model of type theory suggests that groupoids could be used as fundamental objects of mathematics instead of sets (or setoids). The type-theoretic version of a groupoid is an E-category where all morphisms are invertible. To be explicit:

A groupoid $A = (|A|, \mathrm{Hom}, \mathrm{id}, \circ, (\ \ )^{-1})$ consists of

- a type $|A|$,
- a setoid $\mathrm{Hom}(a, b)$ of morphisms for any $a, b : |A|$,
- an identity morphism $\mathrm{id}_a \in \mathrm{Hom}(a, a)$ for each $a : |A|$,
- a composition operation $\circ : \mathrm{Hom}(b, c) \times \mathrm{Hom}(a, b) \longrightarrow \mathrm{Hom}(a, c)$ for all $a, b, c : |A|$,
- an inversion $(\ \ )^{-1} : \mathrm{Hom}(a, b) \longrightarrow \mathrm{Hom}(b, a)$ for $a, b : |A|$,

satisfying standard identities.

The Hofmann–Streicher groupoid model of type theory suggests that groupoids could be used as fundamental objects of mathematics instead of sets (or setoids). The type-theoretic version of a groupoid is an E-category where all morphisms are invertible. To be explicit:

A groupoid $A = (|A|, \mathrm{Hom}, \mathrm{id}, \circ, (\ \ )^{-1})$ consists of

- a type $|A|$,
- a setoid $\mathrm{Hom}(a, b)$ of morphisms for any $a, b : |A|$,
- an identity morphism $\mathrm{id}_a \in \mathrm{Hom}(a, a)$ for each $a : |A|$,
- a composition operation $\circ : \mathrm{Hom}(b, c) \times \mathrm{Hom}(a, b) \longrightarrow \mathrm{Hom}(a, c)$ for all $a, b, c : |A|$,
- an inversion $(\ \ )^{-1} : \mathrm{Hom}(a, b) \longrightarrow \mathrm{Hom}(b, a)$ for $a, b : |A|$,

satisfying standard identities.

Any setoid $A = (|A|, =_A)$ becomes a (discrete) groupoid by letting $\mathrm{Hom}(a, b) = ((a =_A b), \sim)$ and $p \sim q$ hold for all $p, q : a =_A b$.

A type A yields a canonical groupoid $A^\star = (A, \mathrm{Hom}, \mathrm{id}, \circ, (\ )^{-1})$ where $\mathrm{Hom}(a, b) = (\mathrm{I}(A, a, b), \mathrm{I}_{\mathrm{I}(A,a,b)}(\cdot, \cdot)$ and $\mathrm{id}, \circ, (\ )^{-1}$ are operations defined on page 15.

Any group may be regarded as a groupoid with one object.

Any setoid $A = (|A|, =_A)$ becomes a (discrete) groupoid by letting $\mathrm{Hom}(a, b) = ((a =_A b), \sim)$ and $p \sim q$ hold for all $p, q : a =_A b$.

A type A yields a canonical groupoid $A^\star = (A, \mathrm{Hom}, \mathrm{id}, \circ, (\ \ )^{-1})$ where $\mathrm{Hom}(a, b) = (\mathrm{I}(A, a, b), \mathrm{I}_{\mathrm{I}(A,a,b)}(\cdot, \cdot)$ and $\mathrm{id}, \circ, (\ \ )^{-1}$ are operations defined on page 15.

Any group may be regarded as a groupoid with one object.

Any setoid $A = (|A|, =_A)$ becomes a (discrete) groupoid by letting $\mathrm{Hom}(a, b) = ((a =_A b), \sim)$ and $p \sim q$ hold for all $p, q : a =_A b$.

A type A yields a canonical groupoid $A^\star = (A, \mathrm{Hom}, \mathrm{id}, \circ, (\ \ )^{-1})$ where $\mathrm{Hom}(a, b) = (\mathrm{I}(A, a, b), \mathrm{I}_{\mathrm{I}(A,a,b)}(\cdot, \cdot)$ and $\mathrm{id}, \circ, (\ \ )^{-1}$ are operations defined on page 15.

Any group may be regarded as a groupoid with one object.

The notion of functor and natural transformation are defined in the expected way. We have the following correspondences:

groupoid	setoid
functor	extensional function
natural transformation	proof of equality of functions
equivalence of groupoids	proof of isomorphism of setoids
functor $G \to \mathrm{Groupoids}$	proof-irr. family of setoids

The notion of functor and natural transformation are defined in the expected way. We have the following correspondences:

groupoid	setoid
functor	extensional function
natural transformation	proof of equality of functions
equivalence of groupoids	proof of isomorphism of setoids
functor $G \to \mathrm{Groupoids}$	proof-irr. family of setoids

The notion of functor and natural transformation are defined in the expected way. We have the following correspondences:

groupoid	setoid
functor	extensional function
natural transformation	proof of equality of functions
equivalence of groupoids	proof of isomorphism of setoids
functor $G \to \mathrm{Groupoids}$	proof-irr. family of setoids

The notion of functor and natural transformation are defined in the expected way. We have the following correspondences:

groupoid	setoid
functor	extensional function
natural transformation	proof of equality of functions
equivalence of groupoids	proof of isomorphism of setoids
functor $G \to \mathrm{Groupoids}$	proof-irr. family of setoids

The notion of functor and natural transformation are defined in the expected way. We have the following correspondences:

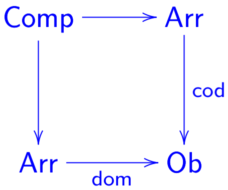
groupoid	setoid
functor	extensional function
natural transformation	proof of equality of functions
equivalence of groupoids	proof of isomorphism of setoids
functor $G \rightarrow \mathrm{Groupoids}$	proof-irr. family of setoids

2-pullback and ordinary 1-pullback

$$\begin{array}{ccc} P & \longrightarrow & A_2 \\ \downarrow & \overset{\sigma}{\Rightarrow} & \downarrow \\ A_1 & \longrightarrow & X \end{array} \qquad\qquad \begin{array}{ccc} P & \longrightarrow & A_2 \\ \downarrow & & \downarrow \\ A_1 & \longrightarrow & X \end{array}$$

Return to the example category $\mathcal{B}$ given by a family B, ϕ, A of setoids

For a proof-**irrelevant** family the composable arrows of $\mathcal{B}$ is obtained by a pullback



Ob, Arr, Comp are setoids.

However for a proof-**relevant** family the composable arrows of $\mathcal{B}$ is obtained by a 2-pullback

$$\begin{array}{ccc} \mathrm{Comp} & \longrightarrow & \mathrm{Arr} \\ \downarrow & \overset{\sigma}{\Rightarrow} & \downarrow \mathrm{cod} \\ \mathrm{Arr} & \underset{\mathrm{dom}}{\longrightarrow} & \mathrm{Ob} \end{array}$$

Ob, Arr, Comp are groupoids, and cod and dom are functors.

Comp is the groupoid indicated by the previously displayed diagram

$$\begin{array}{ccccccc}
B_a & \xrightarrow{f} & B_b & \xrightarrow{\phi_t} & B_c & \xrightarrow{g} & B_d \\
\phi_p \downarrow & & \downarrow \phi_q & & \downarrow \phi_r & & \downarrow \phi_s \\
B_{a'} & \xrightarrow[f']{} & B_{b'} & \xrightarrow[\phi_{t'}]{} & B_{c'} & \xrightarrow[g']{} & B_{d'}
\end{array}$$

Conclusion

- Proof-relevant families of setoids appear in abundance in standard Martin-Löf type theory. Every family of types $B : (A)\mathrm{type}$ gives a such a family (A^*, B^*). However they seem difficult to use for certain purposes, e.g. construction of categories with equality on objects.
- For such purposes the standard proof-irrelevant families are suitable. They are not easy to construct in standard type theory. It seems roughly that we need to construct extensional collapses of the types. This procedure is familiar from set theory, and indeed, one way of constructing such families is to use Aczel's model construction for the constructive set theory CZF. (But then why use type theory?)
- Another possibility is to try to use proof-relevant families, inspired by the Hofmann-Streicher model. This seems to involve developing some new ways (or getting used to) thinking of basic mathematical objects as groupoids.

Conclusion

- Proof-relevant families of setoids appear in abundance in standard Martin-Löf type theory. Every family of types $B : (A)\mathrm{type}$ gives a such a family (A^*, B^*). However they seem difficult to use for certain purposes, e.g. construction of categories with equality on objects.
- For such purposes the standard proof-irrelevant families are suitable. They are not easy to construct in standard type theory. It seems roughly that we need to construct extensional collapses of the types. This procedure is familiar from set theory, and indeed, one way of constructing such families is to use Aczel's model construction for the constructive set theory CZF. (But then why use type theory?)
- Another possibility is to try to use proof-relevant families, inspired by the Hofmann-Streicher model. This seems to involve developing some new ways (or getting used to) thinking of basic mathematical objects as groupoids.

Conclusion

- Proof-relevant families of setoids appear in abundance in standard Martin-Löf type theory. Every family of types $B : (A)\mathrm{type}$ gives a such a family (A^*, B^*). However they seem difficult to use for certain purposes, e.g. construction of categories with equality on objects.
- For such purposes the standard proof-irrelevant families are suitable. They are not easy to construct in standard type theory. It seems roughly that we need to construct extensional collapses of the types. This procedure is familiar from set theory, and indeed, one way of constructing such families is to use Aczel's model construction for the constructive set theory CZF. (But then why use type theory?)
- Another possibility is to try to use proof-relevant families, inspired by the Hofmann-Streicher model. This seems to involve developing some new ways (or getting used to) thinking of basic mathematical objects as groupoids.

Conclusion

- Proof-relevant families of setoids appear in abundance in standard Martin-Löf type theory. Every family of types $B : (A)\mathrm{type}$ gives a such a family (A^*, B^*). However they seem difficult to use for certain purposes, e.g. construction of categories with equality on objects.
- For such purposes the standard proof-irrelevant families are suitable. They are not easy to construct in standard type theory. It seems roughly that we need to construct extensional collapses of the types. This procedure is familiar from set theory, and indeed, one way of constructing such families is to use Aczel's model construction for the constructive set theory CZF. (But then why use type theory?)
- Another possibility is to try to use proof-relevant families, inspired by the Hofmann-Streicher model. This seems to involve developing some new ways (or getting used to) thinking of basic mathematical objects as groupoids.

References

- Errett Bishop and Douglas S. Bridges. *Constructive Analysis.* Springer 1985.
- Alexandre Buisse, Peter Dybjer. Categories of families and E-categories.
- Michael Hedberg. A coherence theorem for Martin-Löf's type theory. *J. Funct. Programming* 8 (1998), no. 4, 413–436.
- Martin Hofmann and Thomas Streicher. The groupoid interpretation of type theory. In: G. Sambin and J. Smith (eds.) *Twenty-five years of constructive type theory (Venice, 1995),* 83–111, Oxford Logic Guides, 36, Oxford Univ. Press, New York, 1998.

References (cont.)

- Ieke Moerdijk and Erik Palmgren. Type Theories, Toposes and Constructive Set Theory: Predicative Aspects of AST, *Annals of Pure and Applied Logic* 114(2002), 155 - 201.
- Bengt Nordström, Kent Peterson and Jan M. Smith. *Programming in Martin-Löf type theory.* Oxford University Press 1990.
- Erik Palmgren. *Groupoids and local cartesian closure.* Uppsala University, Department of Mathematics Report 2003:21. `www.math.uu.se/research/pub`

References (cont.)

- Thomas Streicher. *Investigations into Intensional Type Theory.* Habilitation Thesis, Ludwig-Maximilians Universität, Munich 1993. `www.mathematik.tu-darmstadt.de/~streicher/`