The groupoid interpretation of type theory

Martin Hofmann and Thomas Streicher

August 27, 1996

1 Introduction

Many will agree that identity sets are the most intriguing concept of intensional Martin-Löf type theory. For instance, it may appear surprising that their axiomatisation as an inductive family allows one to deduce the usual properties of equality, notably the replacement rule (Leibniz' principle) which allows one to conclude $P(a')$ from $P(a)$ and a proof that a equals a'. Here, unlike in other logical systems, this holds for arbitrary families of sets P not necessarily corresponding to a predicate. This is not in conflict with decidability of type checking since if a equals a' and $p : P(a)$ then one does not in general have $p : P(a')$, but only $\mathit{subst}(s,p) : P(a')$ where s is the *proof* that a equals a' and *subst* is defined from the eliminator for identity sets.

It is now a natural question to ask whether these translation functions $\mathit{subst}(s, \_)$ actually depend upon the nature of the proof s or, more generally, the question whether any two elements of an identity set are equal. We will call $\mathit{UIP}(A)$ (*U*niqueness of *I*dentity *P*roofs) the following property. If a_1, a_2 are objects of type A then for any two proofs p and q of the proposition "a_1 *equals* a_2" there is another proof establishing equality of p and q. More generally, *UIP* will stand for $\mathit{UIP}(A)$ for all types A. Notice that in traditional logical formalism a principle like *UIP* cannot even be sensibly expressed as proofs cannot be referred to by terms of the object language and thus are not within the scope of propositional equality.

The question whether *UIP* is valid in intensional Martin-Löf type theory was open for a while though it was commonly believed that *UIP* is underivable as any attempt for constructing a proof has failed (Coquand 1992; Streicher 1993; Altenkirch 1992). On the other hand, the intuition that a type is determined by its canonical objects might be seen as evidence for the validity of *UIP* as the identity sets have at most one canonical element corresponding to a proof of reflexivity. Indeed, *UIP* is derivable in an extension of type theory based on this intuition, namely type theory augmented with pattern matching as implemented in the ALF system (Coquand 1992; Altenkirch et al. 1994).

In this paper we answer the question of derivability of *UIP* in pure type theory in the negative by exhibiting a counter model. By the above, this model does not validate pattern matching thereby providing a proof that the latter is not conservative over traditional type theory.

The model we give stands in sharp contrast to the abovementioned intuition of types being determined by their canonical inhabitants. In the model a type A will consist of a set $|A|$ of objects together with (possibly empty) sets $A(a_1, a_2)$ of "proofs" that $a_1, a_2 \in |A|$ are propositionally equal. Although a closed term of type A will be modelled as an object of A an open term will not only map objects to objects but also equality proofs to equality proofs. Thus, an open term is not fully determined by its behaviour on closed terms. The principle *UIP* can then be refuted by including a type in which the set $A(a_1, a_2)$ has more than one element for some $a_1, a_2 \in |A|$.

The hard work consists of demonstrating that these mathematical objects can indeed interpret all of Martin-Löf's type theory. It turns out that various additional structure has to be imposed for that purpose. In particular, we need for each type a composition, identities, and inverses; that is to say functions

$$
\begin{aligned}
&\circ : A(a_2, a_3) \times A(a_1, a_2) \to A(a_1, a_3)\\
&id : A(a_1, a_1)\\
&(\_)^{-1} : A(a_1, a_2) \to A(a_2, a_1)
\end{aligned}
$$

for all objects a_1, a_2, a_3 witnessing that propositional equality is an equivalence relation. In order to interpret the various type and term formers it turns out that these operations must satisfy certain equations. Namely, composition must be an associative operation with neutral element id with inverses given by $(\_)^{-1}$. In other words every type will be a *groupoid*, i.e. a category with isomorphisms only. Open terms and dependent types will then be interpreted as certain functors taking account of the fact that propositional equality is preserved by function application.

A posteriori this justifies a view of propositional equality in type theory as a notion of isomorphism. We exploit this view by exhibiting non-standard axioms for propositional equality on universes which contradict *UIP* and pattern matching. These axioms are put to use in a new formalisation of categories in type theory in which isomorphic objects are propositionally equal.

Independently, François Lamarche (1991) has investigated the logical structure of the category of groupoids with the motivation of finding a logical system in which classes of mathematical structures appear as types. He observed that a theory with type dependency arises as a natural candidate for an internal language of the category of groupoids. He gives interpretations of dependent function spaces and sums which agree essentially with ours.

Parts of the material presented in this article have already been published by the authors in (1993, 1994, 1995). The main purpose of the current version is to make the material accessible to a wider audience and to serve as future reference. As opposed to the extended abstract (Hofmann and Streicher 1994) the model construction is described here in full detail and also in more elementary terms. Furthermore, the syntactic extensions to pure type theory which have been sketched in (Hofmann 1995) are worked out here in detail. The application to formalisation of basic category theory and the analysis of interpretations of universes are altogether new.

Acknowledgements

We are indebted to Thorsten Altenkirch, Thierry Coquand, Peter Dybjer, Per Martin-Löf for numerous discussions on equality in type theory and to François Lamarche for explanations and discussions about the groupoid model. The diagrams have been typeset using Paul Taylor's Latex package.

2 Syntax

We work in Martin-Löf's type theory formulated inside a Logical Framework as defined in Ch. 19 & 20 of (Nordström, Petersson, and Smith 1990). However, we will use a slightly different notation as will be explained below. This type theory derives judgements of the following forms:

- A *type* to mean that A is a type,
- $a : A$ to mean that a is an object of type A
- $A = B$ to mean that types A and B are *definitionally equal*,
- $a = a' : A$ to meant that a and a' are *definitionally equal* objects of type A.

All judgements are relative to a list of variable declarations of the form $x_1 : A_1, \ldots, x_n : A_n$ where the variables $x_1, \ldots, x_n$ are distinct and A_i *type* holds under the assumption $x_1 : A_1, \ldots, x_{i-1} : A_{i-1}$. Such lists of assumptions are called contexts and are ranged over by capital Greek letters $\Gamma, \Delta, \ldots$. One writes $\mathcal{J}\ [\Gamma]$ (alternatively $\Gamma \vdash \mathcal{J}$) to indicate that judgement $\mathcal{J}$ holds in context Γ. In the formal presentation (which we include as an appendix) the valid judgements in context, i.e. under assumptions, are defined inductively; context validity is included as an auxiliary judgement. In the informal presentation below we only indicate the relevant part of a context.

If A *type* and B *type* under $[x : A]$ then the dependent function space $(x : A)B$ is a type. If $b : B\ [x : A]$ then $[x : A]b : (x : A)B$. Conversely, if $f : (x : A)B$ and $a : A$ then $f(a) : B[x{:=}a]$. This typed abstraction constitutes the main difference to the presentation in *loc.cit.*

We have β-equality

$$([x : A]b)\ (a) = b[x{:=}a] : B[x{:=}a]$$

and also η-equality

$$[x : A]b(x) = b : (x : A)B$$

provided x is not free in b.

Iterated applications of the form $f(a_1)(a_2)\cdots(a_n)$ are written as $f(a_1, \ldots, a_n)$ where we take the freedom of omitting arguments which can be inferred from later ones.

There is a special type *Set* containing names for certain types, the so-called *sets*, as objects. Whenever A : *Set* then we have $El(A)$ *type*, in particular, we can form the "generic" family $El(A)$ *type* $[A\colon Set]$. It is common to omit the *El* operator, thus writing $a\colon A$ instead of $a\colon El(A)$. Nordström *et al.* write $a \in A$ for $a : El(A)$. We want to reserve the $\in$-symbol for membership in the metatheory.

This machinery allows one to introduce set formers and term forming operations (be they constructors or eliminators) simply as constants together with their definitional equalities. For example, the intensional identity sets are given by the following constants.

$$Id : (A\colon Set)(a_1, a_2\colon A)Set$$

$$refl : (A\colon Set)(a\colon A)Id(A, a, a)$$

$$J : (A\colon Set)(C\colon (a_1, a_2\colon A)(s\colon Id(A, a_1, a_2))Set)(d : (a\colon A)C(a, a, refl(A, a)))\\ (a_1, a_2\colon A)(s\colon Id(A, a_1, a_2))C(a_1, a_2, s)$$

In addition, we impose the definitional equality

$$J(A, C, d, a, a, refl(A, a)) = d(a) : C(a, a, refl(A, a)) \qquad \text{ID-C}$$

for A, C, d, a of appropriate type. Note that J is called idpeel in *loc.cit.*. According to our convention on omitting redundant arguments we will usually write $Id(a_1, a_2)$ and $refl(a)$ instead of $Id(A, a_1, a_2)$ and $refl(A, a)$, respectively.

In addition to identity sets we also use Π-sets, Σ-sets, disjoint union, natural numbers, and a universe. See the appendix for their formal definition. Following common practice, we write $A \to B$ for $\Pi(A, [a\colon A]B)$ and and $A \times B$ for $\Sigma(A, [a\colon A]B)$ if A and B are types or sets (in the case of $\times$).

The notion of equality induced by identity sets is called *propositional equality* (as opposed to *definitional equality*). That is to say, two objects $a_1, a_2 : A$ are *propositionally equal* if $Id(a_1, a_2)$ is inhabited. The main purpose of propositional equality is that it can be assumed in contexts and thus allows for hypothetical equality reasoning. In particular, propositional equality can be established by induction.

Definitional equality, on the other hand, can only be established by pure equational reasoning, i.e. corresponds to the equational theory generated by the postulated equality judgements.

Accordingly, definitional equality is (at least in traditional cases) decidable, whereas propositional equality is not, as soon as one includes natural numbers and Π-sets.

By the congruence rules for definitional equality the latter always entails the propositional one, but not necessarily vice versa.

3 Syntactic considerations on identity sets

The elimination operator J is motivated by the view of $Id(A, \_, \_)$ as an inductively defined family with constructor *refl*. Accordingly, J permits one to define an object of type $(a_1, a_2\colon A)(s\colon Id(A, a_1, a_2)C(a_1, a_2, s)$ by prescribing its behaviour for arguments of canonical form, i.e. $a_1 = a_2 = a$ and $s = refl(A, a)$.

In the presence of Π-sets, this elimination operation J allows one to derive the following replacement rule in the presence of Π-sets.

$$\mathit{subst} : (A\colon \mathit{Set})(P\colon (a\colon A)\mathit{Set})(a_1, a_2\colon A)(s\colon \mathit{Id}(a_1, a_2))P(a_1) \to P(a_2)$$

satisfying

$$\mathit{subst}(\mathit{refl}(a), p) = p$$

See *loc.cit.* for the definition of *subst*.

From *subst* one easily derives symmetry and transitivity of propositional equality as well as congruence with respect to function application:

$$\mathit{sym} : (A\colon \mathit{Set})(a_1, a_2\colon A)\mathit{Id}(a_1, a_2) \to \mathit{Id}(a_2, a_1)$$

$$\begin{aligned}\mathit{trans} : {}&(A\colon \mathit{Set})(a_1, a_2, a_3\colon A)\\ &\mathit{Id}(a_2, a_3) \to \mathit{Id}(a_1, a_2) \to \mathit{Id}(a_1, a_3)\end{aligned}$$

$$\begin{aligned}\mathit{resp} : {}&(A\colon \mathit{Set})(B\colon \mathit{Set})(u\colon (a\colon A)B)\\ &(a_1, a_2\colon A)\mathit{Id}(a_1, a_2) \to \mathit{Id}(u(a_1), u(a_2))\end{aligned}$$

Notice that we supply arguments to *trans* in the applicative order.

We also have the following dependent version of *resp*.

$$\begin{aligned}\mathit{resp}' : {}&(A\colon \mathit{Set})(B\colon (a\colon A)\mathit{Set})(u\colon (a\colon A)B(a))\\ &(a_1, a_2\colon A)(s\colon \mathit{Id}(a_1, a_2))\mathit{Id}(\mathit{subst}(s, u(a_1)), u(a_2))\end{aligned}$$

To derive *resp′* the full power of J is needed; *subst* alone does not suffice.

3.1 Uniqueness of identity proofs (*UIP*)

For most inductive sets it is possible show that arbitrary objects are propositionally equal to canonical ones. For example, the following types are inhabited

$$\begin{aligned}&(n\colon \mathbf{N})\mathit{Id}(n, 0) + \mathit{Id}(n, \mathit{succ}(\mathit{pred}(n)))\\ &(f\colon \Pi(A, B))\mathit{Id}(f, \mathit{fun}([x\colon A]\mathit{apply}(f, x)))\end{aligned}$$

There are several ways of stating an analogous property for identity sets. We introduce the following abbreviations.

$$\mathit{UIP} \stackrel{\text{def}}{=} (A\colon \mathit{Set})(a_1, a_2\colon A)(s_1, s_2\colon \mathit{Id}(a_1, a_2))\mathit{Id}(s_1, s_2)$$

$$\mathit{UIP\_refl} \stackrel{\text{def}}{=} (A\colon \mathit{Set})(a\colon A)(s\colon \mathit{Id}(a, a))\mathit{Id}(s, \mathit{refl}(a))$$

$$\begin{aligned}\mathit{UIP\_tuple} \stackrel{\text{def}}{=} {}&(A\colon \mathit{Set})(a_1, a_2\colon A)(s\colon \mathit{Id}(a_1, a_2))\\ &\mathit{Id}(\Sigma([a_1\colon A]\Sigma([a_2\colon A]\mathit{Id}(a_1, a_2))), \langle a_1, a_2, s\rangle, \langle a_1, a_1, \mathit{refl}(a_1)\rangle)\end{aligned}$$

Using J, one can show that *UIP_tuple* is inhabited and that *UIP_refl*(A) and *UIP*(A) are equivalent for each $A\colon \mathit{Set}$. See (Streicher 1993) for the proofs. He also explains that in the presence of *UIP* the eliminator J can be defined in terms of the derived operator *subst* thereby allowing for a very intuitive axiomatisation of propositional equality in terms of a uniqueness property of identity proofs and a type-theoretic pendant of Leibniz' principle stating that replacement of equal objects preserves validity.

It is also known (Coquand 1992) that an object of *UIP* can be constructed by pattern-matching.

The main result of this paper consists of an interpretation of type theory in which *UIP* (Uniqueness of Identity Proofs) is not inhabited. *A fortiori*, *UIP* is not derivable and, therefore, pattern-matching is not a conservative extension of Martin-Löf type theory.

3.2 Definability of instances of *UIP*

Although *UIP* is not derivable in general, instances $\mathit{UIP}(A)$ for certain sets A are inhabited. Hedberg (1995) has shown that this is in particular the case if A admits a decidable equality, that is to say, if there is a function $\mathit{eq} : A \to A \to \mathbf{N}$ such that $\mathit{Id}(\mathit{eq}(a_1, a_2), 0)$ and $\mathit{Id}(a_1, a_2)$ are equivalent. One can also show that *UIP* is preserved by the set formers Σ and disjoint union. It is also preserved by the identity set former itself, provided one further assumes that *UIP* applied to proofs by reflexivity gives back a proof by reflexivity. Below we will demonstrate that $\mathit{UIP}(\Pi(A, B))$ follows from $\mathit{UIP}(A)$ and $(a{:}\, A)\,\mathit{UIP}(B(a))$ under the assumption of an extensionality axiom. This gives *UIP* for all sets definable without universes.

3.3 Alternatives to *UIP*

Streicher (1993) gives another principle equivalent to *UIP* which in its formulation does not mention propositional equality of identity proofs:

$$\begin{aligned}\mathit{cong\_snd} \stackrel{\text{def}}{=} {} & (A{:}\, \mathit{Set})(B{:}\ (a{:}\, A)\mathit{Set})(a{:}\, A)(b, b'{:}\, B(a)) \\ & (s{:}\, \mathit{Id}(\Sigma(A, B), \langle a, b\rangle\, ,\, \langle a, b'\rangle))\mathit{Id}(B, b, b')\end{aligned}$$

He also introduces an eliminator K for the family $\mathit{Id}(A, a, a)$ $[a{:}\, A]$:

$$\begin{aligned}K : {} & (A{:}\, \mathit{Set})(C{:}\, (a{:}\, A)(s{:}\, \mathit{Id}(A, a, a))\mathit{Set})(d : (a{:}\, A)C(a, \mathit{refl}(A, a))) \\ & (a{:}\, A)(s{:}\, \mathit{Id}(A, a, a))C(a, s)\end{aligned}$$

satisfying $K(d, \mathit{refl}(a)) = d(a)$. Using K an inhabitant of *UIP* may be constructed.

Both alternatives can be directly defined using pattern matching. It is an open problem whether the converse is also true, i.e. whether pattern matching forms a conservative extension of type theory augmented by K (or a constant of type *UIP* together with an appropriate conversion rule).

3.4 Propositional equality as isomorphism

It has been argued in (Hofmann 1995) that intensional type theory augmented by *UIP* together with an extensionality axiom for Π-sets can simulate extensional type theory whilst retaining decidability of type checking.

On the other hand, we will demonstrate below that Martin-Löf's original formulation of identity sets allows for the addition of axioms (inconsistent with *UIP*) expressing a view of propositional equality as a generalised notion of isomorphism. Intuitively, these axioms state that for a universe U and $A, B : U$ the identity set $\mathit{Id}(U, A, B)$ corresponds to the set of isomorphisms between A and B. Such version of identity sets may be useful for a formulation of category theory inside type theory providing a formal underpinning for the common practice of considering isomorphic objects as equal.

4 The groupoid interpretation

Although the principle *UIP* turns out as being non-derivable, certain propositional equalities between objects of identity sets *can* be established using J. If $A : \mathit{Set}$ and $a_1, a_2 : A$ and $s_1, s_2 : \mathit{Id}(a_1, a_2)$ then we write $s_1 =_{prop} s_2$ to mean that $\mathit{Id}(\mathit{Id}(a_1, a_2), s_1, s_2)$ is inhabited.

Proposition 4.1 *1. If $a_1, a_2 : A$ and $s : \mathit{Id}(a_1, a_2)$ then*

$$\begin{aligned}
&\mathit{trans}(s, \mathit{refl}(a_1)) =_{prop} s\\
&\mathit{trans}(\mathit{refl}(a_2), s) =_{prop} s\\
&\mathit{trans}(\mathit{sym}(s), s) =_{prop} \mathit{refl}(a_1)\\
&\mathit{trans}(s, \mathit{sym}(s)) =_{prop} \mathit{refl}(a_2)
\end{aligned}$$

2. If $a_1, a_2, a_3, a_4 : A$ and $s_1 : \mathit{Id}(a_1, a_2)$ and $s_2 : \mathit{Id}(a_2, a_3)$ and $s_3 : \mathit{Id}(a_3, a_4)$ then

$$\mathit{trans}(s_3, \mathit{trans}(s_2, s_1)) =_{prop} \mathit{trans}(\mathit{trans}(s_3, s_2), s_1)$$

3. If $A, B : \mathit{Set}$ and $a_1, a_2, a_3 : A$ and $f : (a{:}A)B$ and $s_1 : \mathit{Id}(a_1, a_2)$ and $s_2 : \mathit{Id}(a_2, a_3)$ then

$$\begin{aligned}
&\mathit{resp}(f, \mathit{refl}(a_1)) =_{prop} \mathit{refl}(f(a_1))\\
&\mathit{resp}(f, \mathit{trans}(s_2, s_1)) =_{prop} \mathit{trans}(\mathit{resp}(f, s_2), \mathit{resp}(f, s_1))
\end{aligned}$$

Proof. All of these follow straightforwardly using J. As an example we derive $\mathit{trans}(\mathit{sym}(s), s) =_{prop} \mathit{refl}(a_1)$ where $s : \mathit{Id}(A, a_1, a_2)$. We put

$$C \stackrel{\text{def}}{=} [a_1, a_2{:}\, A][s{:}\, \mathit{Id}(a_1, a_2)]\mathit{Id}(\mathit{trans}(\mathit{sym}(s), s)\ ,\ \mathit{refl}(a_1))$$

We have

$$J(A, C, d, a_1, a_2, s) : \mathit{Id}(\mathit{trans}(\mathit{sym}(s), s)\ ,\ \mathit{refl}(a_1))$$

where

$$d \stackrel{\text{def}}{=} [a{:}\, A]\mathit{refl}(\mathit{Id}(A, a, a), \mathit{refl}(A, a))$$

The object $d(a)$ has the required type $C(a, a, \mathit{refl}(A, a))$ because both $\mathit{sym}(\mathit{refl}(a))$ and $\mathit{trans}(\mathit{refl}(a), \mathit{refl}(a))$ are definitionally equal to $\mathit{refl}(a)$. This in turn follows from the definition of *sym* and *trans* in terms of *subst*. □

These propositional equalities suggest that one can view a set as a category having as objects the objects of A and in which a morphism from a_1 to a_2 is an object of $\mathit{Id}(A, a_1, a_2)$, or rather an equivalence class of such objects by propositional equality. Composition is then given by transitivity and reflexivity gives the identities. Symmetry, on the other hand, establishes that every such morphism is actually an isomorphism.

A category in which every morphism is an isomorphism is called a *groupoid*. So the identity sets endow every set with a groupoid structure in a natural way. Furthermore, the equations under 4.1 (3) establish that a function f from A to B extends to a functor from A to B with morphism part given by $\mathit{resp}(f, \_)$.

Under this view the principle *UIP* translates into the statement that every such groupoid is in fact a trivial one with at most one morphism between any two objects. This suggests that a refutation of the principle *UIP* can be obtained by way of an interpretation of type theory in which types are interpreted as arbitrary groupoids, provided one succeeds in ascribing appropriate meaning to the type and set formers. We will do exactly this in the rest of this section.

Our metalanguage for the construction of the interpretation is informal set theory augmented with Grothendieck universes or inaccessible cardinals. We use set theory merely for convenience; all our definitions can also be carried out in extensional Martin-Löf type theory with universes which shows that our constructions do not depend upon the consistency of large cardinals. We assume some basic knowledge of category theory, notably the concepts of category, functor, and natural transformation, see (Mac Lane 1971).

4.1 Groupoids

A *groupoid*[1] is a category Γ where all morphisms are isomorphisms.The groupoids together with functors between them form a (large) category *GPD*.

4.1.1 Examples

The products and exponentials of groupoids *qua* categories are groupoids again so that *GPD* is cartesian closed. Recall that the objects of $\Gamma \times \Delta$ are pairs (γ, δ) where $\gamma \in \Gamma$ and $\delta \in \Delta$ and that the objects of $\Gamma \Rightarrow \Delta$ are functors from Γ to Δ.

For every set X the discrete category $\triangle(X)$ with only identities as morphisms is a groupoid—the discrete groupoid over X. If $x \in X$ we write $\star$ rather than id_x for the $\triangle(X)$-morphism from x to x. Notice that we have $\star : x \to y$ iff $x = y$. We remark that $\triangle\{\langle\rangle\}$ is a terminal object in *GPD* denoted $[]$. More

[1] In universal algebra the term groupoid is sometimes used for a set with a binary operation. Our use of the term groupoid is in accordance with homotopy theory and category theory, cf. (Brown 1988).

generally, a groupoid will be called *discrete* if all its morphisms are identities. Note that up to isomorphism discrete groupoids are of the form $\triangle(X)$.

Every group G can be viewed as a one-object groupoid in the obvious way.

Types (at least non-dependent ones) will be interpreted as groupoids, their closed terms as objects of groupoids. The rôle of the morphisms in a groupoid is to give meaning to propositional equality. Composition of these morphisms accounts for transitivity, identity corresponds to reflexivity, and the inverses to symmetry.

Open terms are interpreted as functors between groupoids where the morphism part witnesses the preservation of propositional equality.

Notation. We notationally identify a groupoid with its underlying set of objects thereby writing $\gamma \in \Gamma$ to mean that γ is an object of Γ. We write p^{-1} for the inverse of morphism p.

4.2 Families of groupoids

To obtain a full-fledged interpretation of type theory we need to account for type dependency, that is we have to define a notion of a family of groupoids indexed over a groupoid. This notion should be such that the usual type formers can receive appropriate meaning and in particular such that the homset $\Gamma(-,-)$ arises as a family of groupoids indexed over $\Gamma \times \Gamma$ thus providing meaning for the identity types.

Fortunately, category theory provides us with such a notion of dependency.

A *family of groupoids* indexed over groupoid Γ is a functor $A : \Gamma \to \mathit{GPD}$. Notice that such a functor yields a groupoid $A(\gamma)$ for each $\gamma \in \Gamma$ and moreover a functor $A(p) : A(\gamma) \to A(\gamma')$ whenever $p : \gamma \to \gamma'$. This will serve as interpretation of replacement and more generally of identity elimination. The fact that A itself is a functor ensures that the functors $A(p)$ are compatible with the groupoid structure of Γ, in particular, we have $A(p) \circ A(p^{-1}) = id_{A(\gamma')}$ and $A(p^{-1}) \circ A(p) = id_{A(\gamma)}$, thus all the functors $A(p)$ are actually isomorphisms of groupoids.

4.2.1 Notation

If $p : \gamma \to \gamma'$ and $a \in A(\gamma)$ then we write $p \cdot \_ : A(\gamma) \to A(\gamma')$ for the functor $A(p)$.

We write $\mathit{Ty}(\Gamma)$ for the collection of families of groupoids indexed over Γ. When $f : \Delta \to \Gamma$ is a morphism in GPD and $A \in \mathit{Ty}(\Gamma)$ then the composition $A \circ f$ is an element of $\mathit{Ty}(\Delta)$. We use the notation $A\{f\}$ for this family. In this way Ty extends to a contravariant "collection-valued" functor on GPD.

4.2.2 Example

If Γ is a groupoid then a family of groupoids I_Γ indexed over $\Gamma \times \Gamma$ is defined by $I_\Gamma(\gamma_1, \gamma_2) = \triangle(\Gamma(\gamma, \gamma'))$ and $I_\Gamma((p_1, p_2))(q) = p_2 \circ q \circ p_1^{-1}$ where $p_i : \gamma_i \to \gamma_i'$

and $q \in I_\Gamma(\gamma_1, \gamma_2)$. Notice that

$$I_\Gamma = \hom_\Gamma((\_)^{-1}, \_)$$

where $(\_)^{-1} : \Gamma^{op} \to \Gamma$ is the obvious isomorphism between Γ^{op} and Γ. Notice that the restriction to groupoids is the minimal requirement for making hom covariant in both arguments.

This family I_Γ will be the interpretation of the family of identity set when Γ is the interpretation of a closed type. Below, we will generalise I_Γ to families of groupoids.

4.3 Objects of families

Let $A \in Ty(\Gamma)$ be a family of groupoids over Γ. A *(dependent) object* M of A consists of the following data

- an $A(\gamma)$-object $M(\gamma)$ for each $\gamma \in \Gamma$,
- for each morphism $p : \gamma \to \gamma'$ an $A(\gamma')$-morphism $M(p) : p \cdot (M(\gamma)) \to M(\gamma')$

such that

$$M(id_\gamma) = id_{M(\gamma)}$$

and

$$M(p' \circ p) = M(p') \circ (p' \cdot M(p))$$

Apart from the "adjustment" $p' \cdot \_$ in the second equation required to make the right-hand side typecheck these laws express functoriality of M. After having defined the semantic counterpart of context formation we will be able to identify dependent objects as corresponding to certain functors.

We write $Tm(A)$ for the collection of dependent objects of A. For functor $f : \Delta \to \Gamma$ the operation $\_\{f\} : Ty(\Gamma) \to Ty(\Delta)$ extends to dependent objects. If $A \in Ty(\Gamma)$ and $a \in Tm(A)$ then $a\{f\} \in Ty(A\{f\})$ is given by composing the components of a with f in the obvious way.

4.4 Category-theoretic semantics

Our plan is to organise groupoids and families of groupoids into a model of dependent type theory, namely a *category with families* (CwF). This notion of model was invented by Dybjer, see (Dybjer 1996), and subsequently used by Martin-Löf (Per Martin-Löf 1995). Our reference for CwFs is the survey article (Hofmann 199). Let us review here that a CwF consists of the following data.

- A category $\mathcal{C}$ of contexts and substitutions with terminal object [] corresponding to the empty context.
- A collection-valued functor $Ty : \mathcal{C}^{op} \to \mathcal{S}et$ associating with each context Γ the collection of types depending on it. If $f : \Delta \to \Gamma$ and $A \in Ty(\Gamma)$ one writes $A\{f\}$ for $Ty(f)(A)$. The type $A\{f\}$ corresponds to the substitution of f into A.

- For each $\Gamma \in \mathcal{C}$ and $A \in \mathit{Ty}(\Gamma)$ a collection of terms $\mathit{Tm}(\Gamma, A)$ together with a substitution function $\mathit{Tm}(f, A) : \mathit{Tm}(\Gamma, A) \to \mathit{Tm}(\Delta, A\{f\})$ functorial in $f : \Delta \to \Gamma$ in the obvious sense.

- For each $A \in \mathit{Ty}(\Gamma)$ a so-called *context extension* $\Gamma.A$ which has the property that the homset $\mathcal{C}(\Delta, \Gamma.A)$ and $\{(f, M) \mid f : \Delta \to \Gamma \text{ and } M \in \mathit{Tm}(\Delta, A\{f\})\}$ are isomorphic naturally in Δ.

- Operations corresponding to the desired type, set, and term formers.

We have already defined the category of contexts, namely *GPD*, and the collections *Ty* and *Tm* together with the required substitution operations.

4.5 Context extension

If $A \in \mathit{Ty}(\Gamma)$ is a family of groupoids the context extension $\Gamma.A$ is the total category of the co-fibration obtained by applying the Grothendieck construction to A. In more explicit terms the groupoid $\Gamma.A$ takes the following form.

The objects of $\Gamma.A$ are pairs (γ, a) where $\gamma \in \Gamma$ and $a \in A(\gamma)$. A morphism in $\Gamma.A$ from (γ, a) to (γ', a') is a pair (p, q) where $p \in \Gamma(\gamma, \gamma')$ and $q \in A(\gamma')(p \cdot a, a')$. The composition of $(p, q) : (\gamma, a) \to (\gamma', a')$ and $(p', q') : (\gamma', a') \to (\gamma'', a'')$ is defined as $(p' \circ p, q' \circ (p \cdot q))$. The identity at (γ, a) is (id_γ, id_a). The inverse of $(p, q) : (\gamma, a) \to (\gamma', a')$ is $(p^{-1}, p^{-1} \cdot q^{-1}) : (\gamma', a') \to (\gamma, a)$. The verifications are left to the reader.

The projection sending (γ, a) to γ and (p, q) to p is a morphism of groupoids from $\Gamma.A$ to Γ. It is called the canonical projection associated to A and is denoted $\mathsf{p}_A : \Gamma.A \to \Gamma$.

In order that $\Gamma.A$ indeed captures context extension we need a bijective correspondence between the set

$$\{(f, M) \mid f : \Delta \to \Gamma \text{ and } M \in \mathit{Tm}(A\{f\})\}$$

and the homset $\mathit{GPD}(\Delta, \Gamma.A)$. Given $f : \Delta \to \Gamma$ and $M \in \mathit{Tm}(A\{f\})$ we define $\langle f, M\rangle_A : \Delta \to \Gamma.A$ by

$$\begin{aligned} \langle f, M\rangle_A(\delta) &= (f(\delta), M(\delta)) \\ \langle f, M\rangle_A(p) &= (f(p), M(p)) \end{aligned}$$

To obtain an inverse we first define a semantic analogue to the sequent $\Gamma, x{:}\, A \vdash x : A$ as follows. A dependent object $\mathsf{v}_A \in \mathit{Tm}(A\{\mathsf{p}_A\})$ is given by

$$\begin{aligned} \mathsf{v}_A(\gamma, a) &= a \\ \mathsf{v}_A(p, q) &= q \end{aligned}$$

Notice that $A\{\mathsf{p}_A\} \in \mathit{Ty}(\Gamma.A)$.

Now, if $h : \Delta \to \Gamma.A$ then we have $\mathsf{p}_A \circ h : \Delta \to \Gamma$ and $\mathsf{v}_A\{h\} \in \mathit{Tm}(A\{\mathsf{p}_A \circ h\})$

It is routine that these data establish the required bijective correspondence natural in Δ.

We have established that groupoids and families of groupoids form an instance of a CwF.

It remains to identify dependent function spaces and an appropriate universe *Set*, as well as, interpretations of the set formers, in particular the identity sets. We treat these issues in order.

4.6 Dependent function space

To each dependent object $M \in Tm(A)$ we can associate a functor $\overline{M} : \Gamma \to \Gamma.A$ by $\overline{M}(\gamma) = (\gamma, M(\gamma))$ and $\overline{M}(p) = (p, M(p))$. We have $\mathsf{p}_A \circ S = id_\Gamma$. Conversely, given a section $f : \Gamma \to \Gamma.A$ of p_A, i.e. $\mathsf{p}_A \circ f = id_\Gamma$ then we have $f(\gamma) = (\gamma, M(\gamma))$ and $f(p) = (p, M(p))$ for a uniquely determined $M \in Tm(A)$.

This correspondence enables us to view $Tm(A)$ (for $A \in Ty(\Gamma)$) not merely as a set, but as a groupoid. A morphism τ from term M to term N is an assignment of an $A(\gamma)$-morphism $\tau_\gamma : M(\gamma) \to N(\gamma)$ such that the family $\overline{\tau}_\gamma := (id_\gamma, \tau_\gamma)$ is a natural transformation from $\overline{M}$ to $\overline{N}$, i.e. for every $p : \gamma \to \gamma'$ the following diagram commutes

$$\begin{tikzcd}
\overline{M}(\gamma) \arrow[r, "\overline{M}(p)"] \arrow[d, "\overline{\tau}_\gamma"'] & \overline{M}(\gamma') \arrow[d, "\overline{\tau}'_\gamma"] \\
\overline{N}(\gamma) \arrow[r, "\overline{N}(p)"'] & \overline{N}(\gamma')
\end{tikzcd}$$

Now suppose that $A \in Ty(\Gamma)$ and $B \in Ty(\Gamma.A)$. We wish to define a family $\Pi_{\mathrm{LF}}(A, B) \in Ty(\Gamma)$ together with additional structure to interpret application and abstraction. In order to avoid lengthy and rather unreadable calculations we will only give the definitions and leave the straightforward verifications to the reader.

If $\gamma \in \Gamma$ let $B_\gamma \in Ty(A(\gamma))$ be the family of groupoids over the groupoid $A(\gamma)$ given by

$$\begin{aligned} B_\gamma(a) &= B(\gamma, a) \\ B_\gamma(p)(\_) &= (id_\gamma, p) \cdot \_ \end{aligned}$$

Notice that $B_\gamma = B\{\hat{\gamma}\}$ where $\hat{\gamma} : A(\gamma) \to \Gamma.A$ is the functor sending a to (γ, a) and $p : a \to a'$ to (id_γ, p).

Now we put

$$\Pi_{\mathrm{LF}}(A, B)(\gamma) = Tm(B_\gamma) \quad \text{considered as a groupoid}$$

If $p : \gamma \to \gamma'$ and $M \in Tm(B_\gamma)$ then $(p \cdot M) \in Tm(B'_\gamma)$ is given by

$$\begin{aligned} (p \cdot M)(a \in A(\gamma')) &= (p, id) \cdot M(p^{-1} \cdot a) \\ (p \cdot M)(q : a \to a') &= (p, id) \cdot M(p^{-1} \cdot q) \end{aligned}$$

If $M, M' \in Tm(B_\gamma)$ and $\tau : M \to M'$ is a natural transformation then $p \cdot \tau : p \cdot M \to p \cdot M'$ is defined by

$$(p \cdot \tau)_a = (p, id) \cdot \tau_{p^{-1} \cdot a}$$

for $a \in A(\gamma')$.

4.6.1 Abstraction and application

Suppose that $M \in Tm(B)$. We define its abstraction $\lambda_{A,B}(M) \in Tm(\Pi_{\mathrm{LF}}(A,B))$ on objects by

$$\lambda_{A,B}(M)(\gamma)(a) = M(\gamma, a)$$
$$\lambda_{A,B}(M)(\gamma)(q) = M(id_\gamma, q)$$

If $p : \gamma \to \gamma'$ then we need a natural transformation

$$\lambda_{A,B}(M)(p) : p \cdot \lambda_{A,B}(M)(\gamma) \to \lambda_{A,B}(M)(\gamma')$$

At object $a \in A(\gamma')$ it is given by $M(p, id_a)$.

Conversely, if $M \in Tm(\Pi(A,B))$ we define a dependent object $\lambda^{-1}_{A,B} \in Tm(B)$. Its object part is given by

$$\lambda^{-1}_{A,B}(M)(\gamma, a) = M(\gamma)(a)$$

For the morphism part assume $p : \gamma \to \gamma'$ and $q : p \cdot a \to a'$. We define

$$\lambda^{-1}_{A,B}(M)(p,q) = M(\gamma')(q) \circ (id_{\gamma'}, q) \cdot M(p)_{p \cdot a}$$

We claim that

$$\lambda^{-1}_{A,B}(M)(p,q) : (p,q) \cdot \lambda^{-1}_{A,B}(M)(\gamma, a) \to \lambda^{-1}_{A,B}(M)(\gamma', a')$$

as required. To see this, first note that

$$M(\gamma')(q) : (id_{\gamma'}, q) \cdot M'(\gamma')(p \cdot a) \to M'(\gamma')(a')$$

because $q : p \cdot a \to a'$. On the other hand $M(p) : p \cdot M(\gamma) \to M(\gamma')$, thus

$$M(p)_{p \cdot a} : (p, id_{a'}) \cdot M(\gamma)(a) \to M'(\gamma')(p \cdot a)$$

as $p^{-1} \cdot p \cdot a = a$. The claim follows by $(id_{\gamma'}, q) \circ (p, id_{a'}) = (p,q)$.

Note that we can define application of $M \in Tm(\Pi(A,B))$ to $N \in Tm(A)$ as $\lambda^{-1}_{A,B}(M)\{\overline{N}\}$.

Rather than defining an object of a family $\Pi_{\mathrm{LF}}(A,B)$ we will often define an object of B instead. This will be referred to as "currying".

4.7 The universe of sets

Let $\mathcal{V}$ be a universe in the metalanguage which is closed under dependent function space, dependent sum, and inductive definitions. If our metalanguage is chosen to be axiomatic set theory such universe may be chosen either as a Grothendieck universe (Mac Lane 1971) or V_κ for κ an inaccessible cardinal (Luo 1994). If we use extensional Martin-Löf type theory as a metalanguage then $\mathcal{V}$ will be a type-theroetic universe with the required closure properties.

Call a groupoid Γ $\mathcal{V}$-small, or small for short, if both its collection of objects and its homsets lie in $\mathcal{V}$. Let us write *Gpd* for the groupoid which has as objects the small groupoids and *only* isomorphisms of groupoids as morphisms. The (non-full) inclusion from *Gpd* to *GPD* defines a family $El \in Ty(Gpd)$.

The groupoid *Gpd* together with its associated family *El* serves as the interpretation of the type *Set* and its associated "invisible" *El* operator. Notice that, if $A : \Gamma \to Gpd$ then $El\{A\}$ is actually equal to A. Therefore, it is appropriate to introduce the notation $\mathrm{Se}(\Gamma)$ for the homset $\Gamma \to Gpd$ where $\mathrm{Se}(\Gamma) \subset Ty(\Gamma)$.

4.8 Interpretation of the syntax

The structure exhibited so far is sufficient to interpret the Logical Framework, i.e. the dependent function spaces and the universe *Set*. This means that we have a unique *compositional* assignment $[\![-]\!]$ which maps

- a well-formed contexts Γ to a groupoid $[\![\Gamma]\!]$,
- a type $A\ [\Gamma]$ to a family of groupoids $[\![A\ [\Gamma]]\!] \in Ty([\![\Gamma]\!])$,
- an object $a : A\ [\Gamma]$ to a dependent object $[\![a : A\ [\Gamma]]\!] \in Tm(A\ [\Gamma])$,

in such a way that derivable equality judgements are validated. More explicitly, this means that

- $[\![A\ [\Gamma]]\!] = [\![A'\ [\Gamma]]\!]$, whenever $A = A'\ [\Gamma]$ is derivable,
- $[\![a : A\ [\Gamma]]\!] = [\![a' : A\ [\Gamma]]\!]$, whenever $a = a' : A\ [\Gamma]$.

Notice that if $A : Set\ [\Gamma]$ then we have $[\![A : Set\ [\Gamma]]\!] \in \mathrm{Se}([\![\Gamma]\!])$ by compositionality. The same goes for dependent function spaces which are interpreted by $\Pi_{\mathrm{LF}}(\_,\_)$.

In order to extend this interpretation to a hierarchically[2] structured equational theory such as in particular Martin-Löf set theory we have to assign to each constant of type A an element of $[\![A]\!]$ in such a way that the required definitional equalities are validated.

In defining these semantic constants we will use type-theoretic syntax to denote semantic entities, thereby omitting semantic brackets.

4.9 Dependent function spaces and sums

Since the dependent function space of a small family of groupoids over a small groupoid is again small, we immediately obtain an interpretation of Π-sets.

To interpret Σ-sets we need an element

$$\Sigma : (A\colon Set)(B\colon (a\colon A)Set)Set$$

By currying, this amounts to defining a small family over the groupoid $\Gamma := [A\colon Set, B\colon (a\colon A)Set]$. We will use Γ as a black box and only use the fact that

[2]The type of a constant may depend on previously declared constants.

its two components arise as small families $A \in \mathrm{Se}(\Gamma)$ and $B \in \mathrm{Se}(\Gamma.A)$ by projection.

We have to define a small family $\Sigma(A,B) \in \mathrm{Se}(\Gamma)$. If $\gamma \in \Gamma$ let $B_\gamma \in \mathrm{Se}(A(\gamma))$ be defined as in Section 4.6.

The family $\Sigma(A,B) \in \mathit{Ty}(\Gamma)$ is now defined as follows.

$$\begin{array}{l} \Sigma(A,B)(\gamma) = A(\gamma).B_\gamma \\ p \cdot (a,b) = (p \cdot a\,,\,(p, id_{p\,\cdot\,a}) \cdot b) \end{array}$$

In order to give meaning to pairing and elimination it is sufficient, albeit not necessary, to exhibit an isomorphism between $\Gamma.\Sigma(A,B)$ and $\Gamma.A.B$. But these two groupoids are identical up to restructuring of parentheses. That is to say the isomorphism sends (γ,a,b) to $(\gamma,(a,b))$ and vice versa and similarly for morphisms.

4.10 Identity sets

Before embarking on the precise definition of identity sets we motivate the main idea by assuming that the ambient context is empty. So let A be a groupoid. The interpretation of $\mathit{Id}(A)$ arises as the family $I_A \in \mathit{Ty}(A \times A)$ as defined in 4.2.2. Recall that $I_A(a_1,a_2) = A(a_1,a_2)$ and $I_A(q_1,q_2)(s) = q_2 \circ s \circ q_1^{-1}$. Reflexivity is interpreted as the dependent object which sends $a \in A$ to $\mathit{refl}(a) := id_a \in A(a,a)$. If $q : a \to a'$, then

$$I_A(q,q)(\mathit{refl}(a)) = q \circ id_a \circ q^{-1} = id_{a'} = \mathit{refl}(a')$$

Therefore, *refl* is indeed a dependent object of the family $I_A(a,a)\ [a{:}\,A]$.

For identity elimination let C be a family over the groupoid

$$\Theta := [a_1, a_2{:}\,A, s{:}\,I_A(a_1,a_2)]$$

The groupoid Θ has as objects triples (a_1,a_2,s) where $s \in A(a_1,a_2)$. A Θ-morphism from (a_1,a_2,s) to (a_1',a_2',s') amounts to a pair (q_1,q_2) where $q_i : a_i \to a_i'$ with $I_A(q_1,q_2)(s) = s'$, i.e. $q_2 \circ s = s' \circ q_1$. Notice that, therefore, Θ is (isomorphic to) the arrow category $A^{\to}$.

In order to interpret $J(A,C)$ one has to provide a uniform way of extending an object d of $(a{:}\,A)C(a,a,\mathit{refl}(a))$ to a dependent object $J(A,C,d)$ of C such that $J(A,C,d)(a,a,\mathit{refl}(a)) = d(a)$. The key to the interpretation of $J(A,C)$ is the observation that for any object (a_1,a_2,s) we have

$$(id_{a_1}, s) : (a_1, a_1, id_{a_1}) \to (a_1, a_2, s)$$

because

$$\begin{tikzcd} a_1 \arrow[r, "id_{a_1}"] \arrow[d, "id_{a_1}"'] & a_1 \arrow[d, "s"] \\ a_1 \arrow[r, "s"'] & a_2 \end{tikzcd}$$

commutes. Therefore, we can put

$$J(A,C,d)(a_1,a_2,s) = (id_{a_1},s)\cdot d(a_1)$$

The morphism part of J is defined analogously.

For those familiar with fibrations we remark that the extension $J(A,C,d)$ depends crucially on the morphism part of C, i.e. on the choice of a splitting of C when viewed as a fibration of groupoids. Therefore, it seems unlikely that it can be characterised by a universal property.

4.10.1 The identity set former

To interpret identity sets in full generality we need an element

$$\mathit{Id} \in \mathit{Tm}((A{:}\,\mathit{Set})(a,a'{:}\,A)\mathit{Set})$$

By currying this amounts to defining a small family over the groupoid $[A{:}\,\mathit{Set}, a_1{:}\,A, a_2{:}\,A]$. This groupoid has as objects triples (A,a_1,a_2) where A is a small groupoid and a_1,a_2 are objects of A. A morphism from (A,a_1,a_2) to (A',a_1',a_2') is a triple (p,q_1,q_2) where $p : A \to A'$ is an isomorphism of groupoids, i.e. $p \in \mathit{Gpd}(A,A')$, and $q_i : p(a_i) \to a_i'$ in A'. Note that $\mathit{El}(p)(a_i) = p(a_i)$ thus permitting us to write also $p(x)$ instead of $p\cdot x$.

The family Id over $[A{:}\,\mathit{Set}, a_1{:}\,A, a_2{:}\,A]$ is now given by

$$\begin{aligned}&\mathit{Id}(A,a_1,a_2) = \triangle(A(a_1,a_2))\\&\mathit{Id}(p,q_1,q_2)(s) = q_2\circ p(s)\circ q_1^{-1} \in \mathit{Id}(A',a_1',a_2')\end{aligned}$$

where $p : A \to A'$ and $q_i \in A'(p(a_i),a_i')$ and $s \in \mathit{Id}(A,a_1,a_2) = A(a_1,a_2)$.

4.10.2 Reflexivity

We define a dependent object

$$\mathit{refl} : (A{:}\,\mathit{Set})(a{:}\,A)\mathit{Id}(A,a,a))$$

Again, by currying this amounts to giving a dependent object of the family $\mathit{Id}_{diag} := \mathit{Id}(A,a)\ [A{:}\,\mathit{Set}, a{:}\,A]$ (over the groupoid $[A{:}\,\mathit{Set}, a{:}\,A]$). Let us make the involved groupoids explicit. The groupoid $[A{:}\,\mathit{Set}, a{:}\,A]$ has as objects pairs (A,a) where A is a small groupoid and $a \in A$. A morphism from (A,a) to (A',a') is a pair (p,q) where $p : A \to A'$ and $q \in A'(p(a),a')$. Furthermore, we have $\mathit{Id}_{diag}(A,a) = \triangle(A(a,a))$ and if $s \in A(a,a)$ then $(p,q)\cdot s = q\circ p(s)\circ q^{-1}$.

The object part of *refl* is now given by

$$\mathit{refl}(A,a) = id_a \in A(a,a)$$

Now since Id_{diag} is discrete the definition of the morphism part of *refl* reduces to checking that

$$q\circ p(\mathit{refl}(A,a))\circ q^{-1} = \mathit{refl}(A',a')$$

This in turn is immediate by by functoriality of p and the fact that q^{-1} is an inverse of q.

4.10.3 Identity elimination

We seek a global element of the following groupoid.

$$(A\colon \mathit{Set})(C\colon (a_1, a_2\colon A, s\colon \mathit{Id}(A, a_1, a_2))\mathit{Set})(d : (a\colon A)C(a, a, \mathit{refl}(A, a)))$$
$$(a_1, a_2\colon A)(s\colon \mathit{Id}(A, a_1, a_2))C(a_1, a_2, s)$$

By currying this amounts to defining a dependent object

$$J \in \mathit{Tm}(C(a_1, a_2, s)\ [\Gamma, a_1\colon A, a_2\colon A, s\colon \mathit{Id}(A, a_1, a_2)])$$

where

$$\Gamma = [A : \mathit{Set}, C : (a_1, a_2\colon A, s\colon \mathit{Id}(A, a_1, a_2))\mathit{Set}, d : (a\colon A)C(a, a, \mathit{refl}(A, a))]$$

and A, C, d refer to the respective components of Γ. Note that we have $A \in \mathrm{Se}(\Gamma)$, a family $C \in \mathrm{Se}([\Gamma, a_1, a_2\colon A, s\colon \mathit{Id}(A, a_1, a_2)])$, and a dependent object d of $C(a, a, \mathit{refl}(A, a))\ [\Gamma, a\colon A]$ via projection (and uncurrying in the case of d).

The object part of J is given as follows. Let $u = (\gamma, a_1, a_2, s)$ be an object of $[\Gamma, a_1, a_2\colon A, s\colon \mathit{Id}(A, a_1, a_2)]$. Put

$$f(u) := (id_\gamma, id_{a_1}, s, \star)$$

Notice that

$$f(u) : (\gamma, a_1, a_1, \mathit{refl}(A, a_1)) \to (\gamma, a_1, a_2, s)$$

as $(s, id_{a_1}) \cdot \mathit{refl}(A, a_1) = s \circ \mathit{refl}(A, a_1) \circ id_{a_1}^{-1} = s$. Now recall that $d(\gamma, a_1) \in C(\gamma, a_1, a_1, \mathit{refl}(A, a_1))$; so we are led to define

$$J(u) := f(u) \cdot d(\gamma, a_1) \in C(\gamma, a_1, a_2, s)$$

We come to the morphism part. Let $u = (\gamma, a_1, a_2, s)$ and $u' = (\gamma', a_1', a_2', s')$ be objects of $[\Gamma, a_1, a_2\colon A, s\colon \mathit{Id}(A, a_1, a_2)]$ and let $h = (p, q_1, q_2, \star) : u \to u'$. In other words $p : \gamma \to \gamma'$ and $q_i : p \cdot (a_i) \to a_i'$ and

$$q_2 \circ (p \cdot s) = s' \circ q_1 \tag{1}$$

We have to define a morphism $J(h) : h \cdot J(u) \to J(u')$ in $C(u')$. We claim that

$$J(h) := f(u') \cdot d(p, q_1)$$

has the required property. To see this, first observe that $(p, q_1) : (\gamma, a_1) \to (\gamma', a_1')$ and therefore

$$d(p, q_1) : (p, q_1, q_1, \star) \cdot d(\gamma, a_1) \to d(\gamma', a_1') \tag{2}$$

as $d \in \mathit{Tm}(C(a, a, \mathit{refl}(A, a))\ [\Gamma, a\colon A])$. Notice that

$$(p, q_1, q_1, \star) : (\gamma, a_1, a_1, \mathit{refl}(a_1)) \to (\gamma', a_1', a_1', \mathit{refl}(a_1)')$$

Applying the operation $f(u') \cdot \_$ to (2) and using functoriality yields

$$f(u') \cdot d(p, q_1) : (f(u') \circ (p, q_1, q_1, \star)) \cdot d(\gamma, a_1) \to f(u') \cdot d(\gamma', a_1')$$

Now we calculate as follows.

$$\begin{array}{rll}
& f(u') \circ (p, q_1, q_1, \star) & \\
= & (id_{\gamma'}, id_{a_1'}, s', \star) \circ (p, q_1, q_1, \star) & \\
= & (p, q_1, s' \circ q_1, \star) & \\
= & (p, q_1, q_2 \circ (p \cdot s), \star) & \text{by Eqn. 1} \\
= & (p, q_1, q_2, \star) \circ (id_\gamma, id_{a_1}, s, \star) & \text{since } p \cdot id_{a_1} = id_{a_1'} \\
= & h \circ f(u) &
\end{array}$$

So $J(h) : (h \circ f(u)) \cdot d(\gamma, a_1) \to f(u') \cdot d(\gamma', a_1')$ and therefore $J(h) : h \cdot J(u) \to J(u')$ as required. The verification of the functor laws for J is tedious but straightforward.

Notice that for all $\gamma \in \Gamma$ and $a \in A(\gamma)$ we have

$$f(\gamma, a, a, \mathit{refl}(A, a)) = (id_\gamma, id_a, id_a, \star)$$

Therefore

$$J(\gamma, a, a, \mathit{refl}(A, a)) = f(\gamma, a, a, \mathit{refl}(A, a)) \cdot d(\gamma, a) = d(\gamma, a)$$

and

$$J(p, q, q, \star) = f(\gamma', a', a', \mathit{refl}(A, a')) \cdot d(p, q) = d(p, q)$$

whenever $(p, q) : (\gamma, a) \to (\gamma', a')$. This establishes the validity of the definitional equality required for J.

4.11 Other set formers

The natural numbers $\mathbf{N} \in \mathrm{Se}([])$ are given as the discrete groupoid $\triangle(\mathbb{N})$ over the set of natural numbers. We omit the definition of the associated operations. The disjoint union set $A + B$ is interpreted as the co-product of groupoids, which is constructed as the disjoint union of the underlying sets of objects and morphisms.

In a similar way, we can interpret lists, trees, unit set, empty set, and other datatypes.

4.12 Universes

Let V be a meta-theoretic universe contained in $\mathcal{V}$. We write $\mathit{Gpd}(V)$ for the groupoid of V-small groupoids with isomorphisms as morphisms.

Provided V has appropriate meta-theoretic closure properties the groupoid $\mathit{Gpd}(V)$ can serve as interpretation of a universe closed under the usual set forming operations.

We call a meta-theoretic universe V *impredicative*, if it is closed under impredicative universal quantification, i.e. for $A \in \mathcal{V}$ and $B : A \to V$ the dependent function space $\Pi_{a\in A}B(a)$ is in V. Of course, non-trivial instances of such V are possible only in an intuitionistic metatheory such as an extensional variant of Luo's ECC (Luo 1994) whose consistency is established by various realizability models.

If V is impredicative (and has the usual closure properties) then $Gpd(V)$ is closed under impredicative quantification as well. That is to say, if $A \in \mathcal{V}$ and $B : A \to Gpd(V)$ then $\Pi_{\mathrm{LF}}(A, B) \in Gpd(V)$. This is immediate from the definition of dependent function spaces in the groupoid model.

For subsequent applications it is useful to have universes of small discrete groupoids available. We write $Gpd_{\triangle}(V)$ for the groupoid consisting of V-small discrete groupoids with isomorphisms (or rather bijections) as morphisms. Due to the fact that discrete groupoids are closed under all set forming operations so will be a universe of the form $Gpd_{\triangle}(V)$. Moreover, since identity sets (even of non-discrete groupoids) are discrete, the family of groupoids $Id(A)$ lives in $Gpd_{\triangle}(A)$ provided the groupoid A has V-small homsets.

Of particular interest is the situation where V is impredicative and *Set* is confined to groupoids with V-small homsets. Due to impredicativity of V this interpretation of *Set* is still closed under dependent function space as well as the remaining set forming operations. Now, $Gpd(V)$ and $Gpd_{\triangle}(V)$ are impredicative universes (still contained in *Set*) which moreover contain *all* identity sets. We thus obtain an interpretation of impredicative higher-order logic in which propositional equality is proof-relevant.

Notice, that since universes of the above form are not discrete they cannot be contained in a universe of the form $Gpd_{\triangle}(V)$. Nevertheless, if V' is a metatheoretic universe contained in V then $Gpd_{\triangle}(V)$ contains the *discrete* groupoid consisting of V'-small groupoids. Universes obtained in this way are still closed under all set formers, but do not contain identity sets of non-discrete groupoids.

We have not checked whether these universes can be suffiently narrowed down so as to validate *universe elimination*, cf. Ch. 14 of (Nordström, Petersson, and Smith 1990), but we do not see any principle obstacle.

5 Applications and extension

In this section we will exploit the benefits of the model construction carried out. We derive the promised independence results and investigate some extensions validated by the groupoid model, in particular a set of axioms expressing that equality on certain universes is isomorphism. These extension are put to use in a new type-theoretic formalisation of basic category theory in which isomorphic objects are propositionally equal.

5.1 Independence of *UIP*

Since for a groupoid A the identity set $Id(A, a_1, a_2)$ is interpreted as $\triangle(A(a_1, a_2))$ it will contain more than one object if there is more than one A-morphism from a_1 to a_2. Due to discreteness of identity sets these objects are then not even propositionally equal as propositional and definitional equality coincide for discrete groupoids. More formally, we have the following theorem.

Theorem 5.1 *The type UIP is empty.*

Proof. Suppose that $u \in \mathit{Tm}(\mathit{UIP})$. Let A be the group $\mathbb{Z}_2$ viewed as a one-object groupoid. That is to say, we have a single object $\star \in A$ and two distinct morphisms $id_\star, p \in A(\star, \star)$ where $p \circ p = id_\star$. Then $u(A, \star, \star, p, id_\star)$ would be an element of $\mathit{Id}(\mathit{Id}(A, \star, \star), p, id_\star)$. However, the latter set is empty as $p \neq id_\star$ and identity sets are discrete. $\square$

By soundness of the interpretation the following is now immediate.

Corollary 5.2 *There is no syntactically definable closed term of type UIP.*

5.1.1 Non-definability of K

As *UIP* can be proved using the eliminator K for the family $\mathit{Id}(A, a, a)$ $[a\colon A]$, as given in Section 3.3 it follows that the latter cannot be interpreted in the groupoid model. It is, however, instructive to see directly, why an attempt of interpreting K in the same way as J fails. Let A be a groupoid and $C \in \mathrm{Se}([a\colon A, s\colon \mathit{Id}(A, a, a)])$ and $d : (a\colon A)C(a, \mathit{refl}(a))$. In order to construct a dependent object of C extending d in the same way as we did in the case of J we would have to come up with a morphism in $[a\colon A, s\colon \mathit{Id}(A, a, a)]$ from $(a, \mathit{refl}(a))$ to (a, s) for arbitrary $s : a \to a$. Now such a morphism would amount to a morphism $q : a \to a$ satisfying $s \circ q = q \circ \mathit{refl}(a) = q$. But this implies $s = \mathit{refl}(a)$. So no such morphism exists if $s \neq \mathit{refl}(a)$.

That an interpretation of K cannot be achieved in any other way either can be seen as follows. Let A be a groupoid with an object $a_0 \in A$ such that $A(a_0, a_0)$ contains a non-identity morphism p_0. Let $C \in \mathrm{Se}([a\colon A, \mathit{Id}(a, a)])$ be the family of discrete groupoids given by

$$C(a, p) = \triangle\{\star \mid p = id_a\}$$

So $C(a, p)$ is empty for $p \neq id_a$. To see that this is a family of groupoids assume that we have a morphism $(q, \star) : (a, p) \to (a', p')$, i.e. $q : a \to a'$ and $p' \circ q = q \circ p$. So $p' = id_{a'}$ iff $p = id_a$. Now $d(a) := \star$ is an object of $(a\colon A)C(a, \mathit{refl}(a))$. However, $\mathit{Tm}(C)$ is empty as $C(a_0, p_0)$ is empty.

5.1.2 Non-definability of *cong_snd*

Finally, let us look at the congruence property *cong_snd*. If A is a groupoid and $B \in \mathrm{Se}(A)$ and $a : A$ and $b, b' : B(a)$ then

$$\mathit{Id}(\Sigma(A, B), \langle a, b\rangle\,, \langle a, b'\rangle) = \triangle\{(p, q) \mid p : a \to a,\ q : p \cdot b \to b'\}$$

whereas

$$\mathit{Id}(B(a), b, b') = \triangle\{q \mid q : b \to b'\}$$

So the two groupoids are different and one can easily construct a situation in which the first one is inhabited and the second one is empty.

5.2 Canonicity of identity types

Unlike Π-sets with η-equality or extensional identity sets, the intensional identity sets are not defined by a universal property. Therefore, it is natural to ask how interpretations of identity sets in the groupoid model look like in general.

To answer this question, assume for the moment that we enrich our type theory by another set former $Id' : (A\colon Set)A \to A \to Set$ together with appropriately typed constants refl' and J' satisfying the corresponding definitional equalities. Then using J and J' one can exhibit terms

$$
\begin{aligned}
i &: (A\colon Set)(a_1, a_2\colon A)Id(A, a_1, a_2) \to Id'(A, a_1, a_2)\\
j &: (A\colon Set)(a_1, a_2\colon A)Id'(A, a_1, a_2) \to Id(A, a_1, a_2)
\end{aligned}
$$

in such a way that, moroever, the following two types are inhabited

$$
\begin{aligned}
&(A\colon Set)(a_1, a_2\colon A)(s\colon Id(A, a_1, a_2))Id(j(i(s))\,,\, s)\\
&(A\colon Set)(a_1, a_2\colon A)(s'\colon Id'(A, a_1, a_2))Id(i(j(s'))\,,\, s')
\end{aligned}
$$

It follows that *UIP* holds with respect to Id' if and only if it holds w.r.t. Id because the property of having at most one element is stable under propositional isomorphism. It follows that no interpretation of identity sets satisfying *UIP* is possible in the groupoid model, thus in particular extensional identity sets cannot be interpreted.

A more refined analysis shows that i and j establish an equivalence in the category-theoretic sense and therefore any possible interpretation of the identity set $Id(A, a_1, a_2)$ must be a posetal groupoid whose connected components are in a 1-1 correspondence with $A(a_1, a_2)$.

5.3 Functional extensionality

Despite the intensional character of the groupoid model propositional equality on function spaces is pointwise in the sense that the following type is inhabited in the model.

$$
\begin{aligned}
\mathit{Fun\_Ext} \stackrel{\text{def}}{=} &(A\colon Set)(B\colon (a\colon A)Set)(f, g\colon \Pi(A, B))\\
&((a\colon A)Id(B(a), f(a), g(a))) \to Id(f, g)
\end{aligned}
$$

To see this, let Γ denote the groupoid $[A\colon Set, B\colon (a\colon A)Set\,,\, f, g\colon \Pi(A, B)]$ and let $PE \in \mathrm{Se}(\Gamma)$ denote the family $(a\colon A)Id(B(a), f(a), g(a))$.

Let $\gamma \in \Gamma$. The groupoid $Id(f, g)(\gamma)$ is the discrete groupoid with objects the natural transformations in the sense of Section 4.6. More precisely, an object of $Id(f, g)(\gamma)$ is an assignment τ mapping objects $a \in A(\gamma)$ to $B(\gamma, a)$-morphisms $\tau_a : f(\gamma)(a) \to g(\gamma)(a)$ such that whenever $q : a \to a'$ then the following diagram commutes.

$$\begin{tikzcd}
f(\gamma)(a) \arrow[r, "\tau_a"] & g(\gamma)(a) \\
(id,q)\cdot f(\gamma)(a) \arrow[r, "(id,q)\cdot\tau_a"] \arrow[d, "f(\gamma)(q)"] & (id,q)\cdot g(\gamma)(a) \arrow[d, "g(\gamma)(q)"'] \\
f(\gamma)(a') \arrow[r, "\tau_{a'}"] & g(\gamma)(a')
\end{tikzcd}$$

Now let M be an object of $PE(\gamma)$. By definition of dependent function space $M(a)$ is a morphism from $f(\gamma)(a)$ to $g(\gamma)(a)$ for every $a \in A(\gamma)$. Furthermore, if $q : a \to a'$ then

$$M(q) : (id,q)\cdot M(a) \to M(a') \tag{3}$$

where $\cdot$ refers to the identity set $\mathit{Id}(f(a), g(a))$. By discreteness of identity sets (3) translates into the following diagram.

$$\begin{tikzcd}
f(\gamma)(a) \arrow[r, "M(a)"] & g(\gamma)(a) \\
(id,q)\cdot f(\gamma)(a) \arrow[r, "(id,q)\cdot M(a)"] \arrow[d, "f(\gamma)(q)"] & (id,q)\cdot g(\gamma)(a) \arrow[d, "g(\gamma)(q)"'] \\
f(\gamma)(a') \arrow[r, "M(a')"] & g(\gamma)(a')
\end{tikzcd}$$

This means that the objects of $\mathit{Id}(f,g)(\gamma)$ and $PE(\gamma)$ are the same! Being a dependent function space of a discrete family, PE is discrete itself. So $\mathit{Id}(f,g)(\gamma)$ and $PE(\gamma)$ are isomorphic. One can also show that this isomorphism is natural in γ thus establishing an isomorphism between the families PE and $\mathit{Id}(f,g)$. A more refined analysis shows that one direction of this isomorphism arises as the interpretation of the following proof that equal functions are pointwise equal.

$$\begin{aligned}
\mathit{app_resp} \stackrel{\text{def}}{=}\; & [A\colon \mathit{Set}][B\colon (a\colon A)\mathit{Set}][f,g\colon \Pi(A,B)][s\colon \mathit{Id}(f,g)] \\
& [a\colon A]\mathit{resp}([h\colon \Pi(A,B)]h(a), s) \\
: & (A\colon \mathit{Set})(B\colon (a\colon A)\mathit{Set})(f,g\colon \Pi(A,B))\mathit{Id}(f,g) \to (a\colon A)\mathit{Id}(f(a),g(a))
\end{aligned}$$

This allows us to interpret the following extension of Martin-Löf's type theory.

$$\mathit{fun\_ext} : \mathit{Fun\_Ext}$$

$$\begin{aligned}
\mathit{fun_ext_ax1} : & (A\colon \mathit{Set})(B\colon (a\colon A)\mathit{Set})(f,g\colon \Pi(A,B)) \\
& (s\colon \mathit{Id}(f,g))\mathit{Id}(s, \mathit{fun_ext}(\mathit{app_resp}(s)))
\end{aligned}$$

$$\begin{aligned}
\mathit{fun_ext_ax2} : & (A\colon \mathit{Set})(B\colon (a\colon A)\mathit{Set})(f,g\colon \Pi(A,B)) \\
& (s\colon (a\colon A)\mathit{Id}(f(a),g(a)))\mathit{Id}(s, \mathit{app_resp}(\mathit{fun_ext}(s)))
\end{aligned}$$

The special case of *fun_ext_ax1* where s is an instance of reflexivity was proposed by Turner (Turner 1989) as a possible axiomatisation of functional extensionality. It is easy to see that this special case is equivalent to the general *fun_ext_ax1* using J. Apparently, *fun_ext_ax2* is independent of *fun_ext_ax1* (and Turner's axiom). Note that our two axioms determine the postulated object *fun_ext* uniquely up to propositional equality. Obviously, the axioms *fun_ext_ax1/2* are derivable from *UIP*.

A potential application of functional extensionality is that it allows one to derive $\mathit{UIP}(\Pi(A,B))$ from $(a{:}\,A)\,\mathit{UIP}(B(a))$. Assuming *fun_ext* alone, does not seem to suffice for that purpose.

Notice that functional extensionality allows us to express an identity set of the form $\mathit{Id}(\Pi(A,B))$ in terms of identity sets of the form $\mathit{Id}(B(a))$. A similar decomposition is possible for Σ-sets without any extension to the syntax. Indeed, using Σ-elimination we can establish a canonical isomorphism between $\mathit{Id}(\Sigma(A,B), \langle u_1,u_2\rangle, \langle v_1,v_2\rangle)$ and

$$\Sigma([p{:}\,\mathit{Id}(A,u_1,v_1)]\mathit{Id}(B(v_2), \mathit{subst}(p,u_2), v_2)$$

Analogously, we can decompose identity sets at disjoint unions and natural numbers.

5.4 Universe extensionality

In this section we want to make an extension of type theory taking account of the fact that propositional equality on a universe is isomorphism. To make this more precise we need some notation. We write $\mathit{Iso}(A,B)$ for the set

$$\Sigma([f{:}\,A \to B]\Sigma([g{:}\,B \to A]\mathit{Id}(g \circ f, \mathit{id}) \times \mathit{Id}(f \circ g, \mathit{id}))$$

where composition ($\circ$) and identities (id) are defined as usual in terms of abstraction and application. If $h : \mathit{Iso}(A,B)$ we abbreviate its first component by h and its second component by h^{-1}. Conversely, if $f : A \to B$ and it is clear from the context that f has an inverse in the sense propositional equality then we may write $f : \mathit{Iso}(A,B)$.

Now let U be a universe of discrete groupoids, i.e. of the form $\mathit{Gpd}_{\triangle}(V)$.

It is then clear that if $A, B : U$ then the interpretations of $\mathit{Iso}(A,B)$ and $\mathit{Id}(U,A,B)$ are isomorphic. One direction of the isomorphism is syntactically definable as

$$\begin{aligned} \mathit{id\_iso} &\stackrel{\text{def}}{=} [A,B{:}\,U][s{:}\,\mathit{Id}(A,B)]\mathit{subst}([X{:}\,U]\mathit{Iso}(A,X)\,,\, s, \mathit{id}_A) \\ &\quad : (A,B{:}\,U)\mathit{Id}(A,B) \to \mathit{Iso}(A,B) \end{aligned}$$

Notice that $\mathit{Iso}(A,B)$ and $\mathit{Id}(A,B)$ are *not* isomorphic if $U = \mathit{Gpd}(V)$ because then $\mathit{Iso}(A,B)$ is in one-to-one correspondence with *equivalences* between A and B. Thus $\mathit{Iso}(A,B)$ may be inhabited even if A and B are not isomorphic.

Like in the case of functional extensionality we can now syntactically postulate an inverse to the function *id_iso*:

$$\mathit{iso\_id} : (A, B \colon U)\mathit{Iso}(A, B) \to \mathit{Id}(A, B)$$

$$\mathit{iso\_id\_ax1} : (A, B \colon U)(s \colon \mathit{Id}(A, B))\mathit{Id}(s, \mathit{iso\_id}(\mathit{id\_iso}(s)))$$

$$\mathit{iso\_id\_ax2} : (A, B \colon U)(s \colon \mathit{Iso}(A, B))\mathit{Id}(s, \mathit{id\_iso}(\mathit{iso\_id}(s)))$$

By analogy to functional extensionality we refer to this extension by *universe extensionality.*

We remark that universe extensionality is inconsistent with $\mathit{UIP}(U)$ if U contains the natural numbers. This is so because the set $\mathit{Iso}(\mathbf{N}, \mathbf{N})$ contains two different definable elements f and g. *UIP* together with the above constants would identify f and g and therefore two different natural numbers.

5.5 A new formalisation of category theory

An application of the above extension is a new formalisation of category theory where isomorphic objects are propositionally equal. Let U be a universe of discrete groupoids. We reflect this syntactically by assuming $(A \colon U)\mathit{UIP}(A)$.

A *category with isomorphism as equality* then consists of the following data.

- a set $\mathit{Ob} : \mathit{Set}$ of objects,
- a family of sets $\mathit{Mor} : \mathit{Ob} \to \mathit{Ob} \to U$ of morphisms,
- objects *id* and *comp* of the obvious types corresponding to identity and composition,
- proofs of the traditional axioms stated in terms of propositional equality,
- a proof that for each $A, B \colon \mathit{Ob}$ the sets $\mathit{Id}(\mathit{Ob}, A, B)$ and $\mathit{Iso}(A, B)$ are canonically isomorphic.

This definition deserves some explanation. The set $\mathit{Iso}(A, B)$ for $A, B \colon \mathit{Ob}$ is defined analogously to *Iso* for members of a universe. More precisely, $\mathit{Iso}(A, B)$ is

$$\begin{array}{l}\Sigma[f \colon \mathit{Mor}(A, B)]\Sigma[g \colon \mathit{Mor}(B, A)] \\ \qquad \mathit{Id}(\mathit{comp}(g, f), \mathit{id}(A)) \times \mathit{Id}(\mathit{comp}(f, g), \mathit{id}(B))\end{array}$$

This set being "canonically isomorphic" to $\mathit{Id}(A, B)$ means that the canonical function $\mathit{Id}(A, B) \to \mathit{Iso}(A, B)$ obtained by applying *subst* to $\mathit{id}(A)$ is bijective.

The fact that the homsets $\mathit{Mor}(A, B)$ are discrete enables us to do without further axioms qualifying the behaviour of the assumed proofs of the category equations. In other words, there is only one reason for morphisms to be equal.

If we want to consider the collections of categories as objects of *Set* then we have to restrict *Ob* to a member of a certain not necessarily discrete universe.

Of course, this formalisation of categories does not *per se* require universe extensionality. The point is that universe extensionality is necessary for organising U itself into a category with isomorphism as equality.

More precisely, we can define a category where $Ob \stackrel{\mathrm{def}}{=} U$ and $Mor(A, B) \stackrel{\mathrm{def}}{=} A \to B$.

Furthermore, using functional extensionality in an essential way, we can show that categories with isomorphism as equality are closed under formation of functor categories. The crucial point here is to establish a one-to-one correspondence between between natural isomorphism between two functors F and G (between categories C and D) and proofs that F and G are equal. This is achieved by decomposing the set $Id(FUNC(C, D), F, G)$ according to the rules set out at the end of Section 5.3. Here the set of functors $FUNC(C, D)$ is defined as usual by grouping together object and morphism part as an object of a Σ-set.

The components of a natural isomorphism now correspond to a proof that the object parts of F and G are pointwise equal, thus equal by functional extensionality. The naturality condition, on the other hand, is required for the proof that the two morphism parts are propositionally equal. The details are messy, but have been machine-checked using a proof assistant (LEGO).

We emphasise that the abovementioned decomposition of $Id(F, G)$ suggests the usual definition of isomorphism between functors. It would be interesting to see whether this analogy can be exploited for finding appropriate notions of isomorphism for other mathematical structures (e.g. models of typed lambda calculus).

Notice that, obviously, $UIP(FUNC(C, D))$ is not valid as there is in general more than one natural isomorphism between any two functors. This implies that categories and functors do not form a category with equality as isomorphism since its homsets are not discrete. Of course, categories and functors still can be organised into a category in the traditional sense. However, it might be interesting to view equivalent categories as propositionally equal. This, however, would require "2-level groupoids" in which we have morphisms between morphisms and accordingly the identity sets are not necessarily discrete. We do not know whether such structures (or even infinite-level generalisations thereof) can be sensibly organised into a model of type theory.

A Syntax of type theory

A.1 General rules

$$\frac{}{[]\ ctxt}\ \textsc{Emp}$$

$$\frac{A\ type\ [\Gamma]}{\Gamma, x{:}\, A\ ctxt}\ \textsc{Compr}$$

$$\frac{\Gamma, x{:}\, A, \Delta\ ctxt}{x : A\ [\Gamma, x{:}\, A, \Delta]}\ \textsc{Var}$$

In rule COMPR x is a fresh variable.

Rules expressing that definitional equality is a congruence relation with respect to all subsequent type and term forming operations.

A.2 Rules for the Logical Framework

$$\frac{A\ \mathit{type}\,[\Gamma] \quad B\ \mathit{type}\,[\Gamma, x\colon A]}{(x\colon A)B\ \mathit{type}\,[\Gamma]}\ \textsc{Fun}$$

$$\frac{A\ \mathit{type}\,[\Gamma] \quad B\ \mathit{type}\,[\Gamma, x\colon A] \quad t : B\ [\Gamma, x\colon A]}{[x\colon A]t : (x\colon A)B\ [\Gamma]}\ \textsc{lam}$$

$$\frac{A\ \mathit{type}\,[\Gamma] \quad B\ \mathit{type}\,[\Gamma, x\colon A] \quad t : (x\colon A)B\ [\Gamma] \quad s : A\ [\Gamma]}{t(s) : B[x{:=}s]\ [\Gamma]}\ \textsc{app}$$

$$\frac{A\ \mathit{type}\,[\Gamma] \quad B\ \mathit{type}\,[\Gamma, x\colon A] \quad t : B\ [\Gamma, x\colon A] \quad s : A\ [\Gamma]}{([x\colon A]t)(s) = t[x{:=}s] : B[x{:=}s]\ [\Gamma]}\ \beta$$

$$\frac{A\ \mathit{type}\,[\Gamma] \quad B\ \mathit{type}\,[\Gamma, x\colon A] \quad t : (x\colon A)B\ [\Gamma]}{[x\colon A]\,t(x) = t : (x\colon A)B\ [\Gamma]}\ \eta$$

$$\frac{\Gamma\ \mathit{ctxt}}{\mathit{Set}\ \mathit{type}\,[\Gamma]}\ \textsc{Set}$$

$$\frac{A : \mathit{Set}\ [\Gamma]}{\mathit{El}(A)\ \mathit{type}\,[\Gamma]}\ \textsc{El}$$

We henceforth omit the *El*-operator and use the conventions on abstraction, application, and omission of redundant arguments set out in Section 2. We abbreviate $(x\colon A)B$ by $A \to B$ if x does not occur in B.

A.3 Martin-Löf's set theory

Martin-Löf's set theory is defined as the extension of the Logical Framework by the following constants and definitional equations understood in every valid context. The definitional equalities hold under the proviso that their components are well-typed.

A.4 Π-sets

$$\Pi : (A\colon \mathit{Set})(A \to \mathit{Set}) \to \mathit{Set}$$

$$\mathit{fun} : (A\colon \mathit{Set})(B\colon A \to \mathit{Set})((a\colon A)B(a)) \to \Pi(A, B)$$

$$\mathit{app} : (A\colon \mathit{Set})(B\colon A \to \mathit{Set})(\Pi(A, B)) \to (a\colon A)B(a)$$

$$\mathit{app}(\mathit{fun}(b), a) = b(a) : B(a)$$

$$\mathit{fun}([x\colon A]\,\mathit{app}(t, x)) = t : \Pi(A, B)$$

Alternatively, we can replace *app* by the so-called *funsplit* operator (Nordström, Petersson, and Smith 1990). Then the second η-like equality only holds propositionally.

We take the freedom of writing $f(a)$ for $app(f, a)$.

A.5 Σ-sets

$$\Sigma : (A\colon Set)(A \to Set) \to Set$$

$$pair : (A\colon Set)(B\colon A \to Set)(a\colon A)B(a) \to \Sigma(A, B)$$

$$\begin{aligned}E : {} & (A\colon Set)(B\colon A \to Set)(C : \Sigma(A, B) \to Set) \\ & ((a\colon A)(b\colon B(a))C(pair(a, b))) \to (c\colon \Sigma(A, B))C(c)\end{aligned}$$

$$E(d, pair(a, b)) = d(a, b) : C(pair(a, b))$$

When $A, B : Set$ then we abbreviate $\Sigma([x\colon A]B)$ by $A \times B$. We also write $\langle a, b\rangle$ for $pair(a, b)$.

A.6 Identity sets

$$Id : (A\colon Set)A \to A \to Set$$

$$refl : (A\colon Set)(a\colon A)Id(A, a, a)$$

$$\begin{aligned}J : {} & (A\colon Set)(C\colon (a_1, a_2\colon A)Id(A, a_1, a_2) \to Set) \\ & ((a\colon A)C(a, a, refl(A, a))) \to (a_1, a_2\colon A)(s\colon Id(A, a_1, a_2))C(a_1, a_2, s)\end{aligned}$$

$$J(d, refl(a)) = d(a) : C(refl(a))$$

A.7 Natural numbers and disjoint unions

$\mathbf{N} : Set$
$0 : \mathbf{N}$
$succ : \mathbf{N} \to \mathbf{N}$

$R : (C : \mathbf{N} \to Set)C(0) \to ((n{:}\,\mathbf{N})C(n) \to C(succ(n))) \to (n{:}\,\mathbf{N})C(n)$

$R(d, e, 0) = d : C(0)$
$R(d, e, succ(n)) = e(n, R(d, e, n)) : C(succ(n))$

$+ : Set \to Set \to Set$
$i : (A, B{:}\, Set)A \to A + B$
$j : (A, B{:}\, Set)B \to A + B$

$D : (A, B{:}\, Set)(C{:}\,(A + B) \to Set)$
$\quad ((a{:}\,A)C(i(a))) \to ((b{:}\,B)C(j(b))) \to (c{:}\,A + B)C(c)$
$D(d, e, i(a)) = d(a) : C(i(a))$
$D(d, e, j(b)) = e(b) : C(j(b))$

A.8 Universes

A universe U in *Set* is defined by two constants $U : Set$ and $T : U \to Set$

Closure properties of the universe under certain type and term formers are expressed by reintroducing them with *Set* replaced by U. If the desired type former is available for *Set* as well then it suffices to reintroduce the type former for U and relate it to the corresponding type former on the level of *Set* by an appropriate equality axiom for T. The associated term formers can then be inherited from *Set*. For example, closure under impredicative universal quantification is defined by

$\hat{\Pi} : (A{:}\, Set)(A \to U) \to U$
$\widehat{fun} : (A{:}\, Set)(B{:}\, A \to U)((a{:}\, A)T(B(a))) \to T(\hat{\Pi}(B))$
$\widehat{app} : (A{:}\, Set)(B{:}\, A \to U)(t{:}\, T(\hat{\Pi}(B))) \to (a{:}\, A)T(B(a))$
β and η equations

If we have Π-sets then $\widehat{fun}$ and $\widehat{app}$ together with their equations can be replaced by the single set equation

$$T(\hat{\Pi}(A, B)) = \Pi(A, [a{:}\, A]T(B(a)))$$

References

Altenkirch, T. (1992, January). An open question concerning inductive equality. E-mail message to the Edinburgh LEGO club.

Altenkirch, T., V. Gaspes, B. Nordström, and B. von Sydow (1994). *A User's Guide to ALF*. Sweden: Chalmers University of Technology. Available under ftp://ftp.cs.chalmers.se/pub/users/alti/alf.ps.Z.

Brown, R. (1988). *Topology*. Ellis Horwood.

Coquand, T. (1992). Pattern matching with dependent types. In *Workshop on Logical Frameworks, Båstad*. Preliminary Proceedings.

Dybjer, P. (1996). Internal type theory. In *Proc. BRA TYPES workshop, Torino, June 1995, Springer LNCS*. To appear.

Hedberg, M. (1995). Uniqueness and internal decidability in type theory. Manuscript, Chalmers University, Gothenburg.

Hofmann, M. (199?). Syntax and semantics of dependent types. In P. Dybjer and A. M. Pitts (Eds.), *Semantics and Logics of Computation*. Cambridge University Press.

Hofmann, M. (1993, July). A model of intensional Martin-Löf type theory in which unicity of identity proofs does not hold. unpublished note, available on email request.

Hofmann, M. (1995). *Extensional Concepts in Intensional Type Theory*. Ph. D. thesis, Univ. of Edinburgh.

Hofmann, M. and T. Streicher (1994). A groupoid model refutes uniqueness of identity proofs. In *Proceedings of the 9th Symposium on Logic in Computer Science (LICS), Paris*.

Lamarche, F. (1991(?)). A Proposal about Foundations I. Manuscript.

Luo, Z. (1994). *Computation and Reasoning*. Oxford University Press.

Mac Lane, S. (1971). *Categories for the Working Mathematician*. Springer.

Nordström, B., K. Petersson, and J. M. Smith (1990). *Programming in Martin-Löf's Type Theory, An Introduction*. Clarendon Press, Oxford.

Per Martin-Löf (1995). Tarskian semantics for type theory. Talk given at the symposium "25 years of constructive type theory", Venice.

Streicher, T. (1993). *Semantical Investigations into Intensional Type Theory*. Habilitationsschrift, LMU München.

Turner, D. (1989, May). A new formulation of constructive type theory. In P. Dybjer (Ed.), *Proceedings of the Workshop on Programming Logic*, pp. 258–294. Programming Methodology Group, Univ. of Göteborg.