The Groupoid Interpretation of Type Theory, a Personal Retrospective

Martin Hofmann

LMU Munich

DMV Mini Symposion on HOTT, Hamburg, 3rd July 2015

Prehistory

- Got to know type theory through the proof assistant LEGO in $\sim$ 1990 in a workshop organized by Terry Stroup in Hetzelsdorf, .de. Among the attendants: Randy Pollack, Benjamin Pierce, Thorsten Altenkirch, . . .

  het

- Became fascinated by machine-checked proof, but also annoyed by lack of "extensional concepts" and intrigued by type dependency and intensionality.

Dependent types

- Propositions as types for predicate logic.
- Curry-Howard: a proof of $\varphi \wedge \psi$ is a pair comprising a proof of φ and a proof of ψ. A proof of $\varphi \to \psi$ is a function mapping proofs of φ to proofs of ψ. A proposition is (induces, corresponds to,. . .) the type (set) of its proofs
- Generalising to predicates: a proof of $\forall x{:}A.\varphi(x)$ is a dependent function mapping an element $v \in A$ to an element (=proof) of $\varphi(v)$. We write the type of such dependent functions as $\Pi x{:}A.\varphi(x)$. a proof of $\exists x{:}A.\varphi(x)$ is a dependent pair (v, p) consisting of an element $v \in A$ and an element (=proof) p of $\varphi(v)$. We write the type of such dependent functions as $\Sigma x{:}A.\varphi(x)$.

- To formalise this, we need families of types depending on values: dependent types.
- Other examples of dependent types: vectors, matrices, arrays, universes, ...
- So far (a formal system of dependent types including Π, Σ) this existed before Martin-Löf: de Bruijn's Automath.

Martin-Löf type theory

- Martin-Löf augments basic system of dependent types with *inductively defined types*: natural numbers, lists, trees, well-orderings, ...
- ...and an inductive definition of equality: the identity type:
- For each type A a dependent type $Id_A(x, y)$ where $x, y \in A$ representing equality of x and y

- Reflexivity: a dependent function $\mathit{refl}_A : \Pi x{:}A.\mathit{Id}_A(x,x)$
- An induction principle asserting that this is the only inhabitant:
  - Given a dependent type $C(x,y,p)$ where $x,y{:}A$ and $p{:}\mathit{Id}_A(x,y)$
  - Given a dependent function $h : \Pi x{:}A.C(x,x,\mathit{refl}_A(x))$ obtain a dependent function $J(h) : \Pi x,y{:}A.C(x,y,p)$
  - β-reduction: $J(h)(x,x,\mathit{refl}_A(x)) \rightsquigarrow h(x)$
- Compare with natural numbers $0 : \mathbf{N}$ and $\mathit{Suc} : \mathbf{N} \to \mathbf{N}$. Given a dependent type $C(x)$ where $x : \mathbf{N}$ and $h_0 : C(0)$ and $h_{\mathit{Suc}} : \Pi x{:}\mathbf{N}.C(x) \to C(\mathit{Suc}(x))$ you get $I(h_0,h_{\mathit{Suc}}) : \Pi x{:}\mathbf{N}.C(x)$. One has $I(h_0,h_{\mathit{Suc}})(0) \rightsquigarrow h_0$ and $I(h_0,h_{\mathit{Suc}})(\mathit{Suc}(n)) \rightsquigarrow h_{\mathit{Suc}}(n, I(h_0,h_{\mathit{Suc}})(n))$. Can use that both for primitive recursion (e.g. $C(x) = \mathbf{N}$) and induction (e.g. $C(x) = \mathit{Id}_{\mathbf{N}}(x, \mathit{Suc}^x(0))$).

Properties of propositional equality

- Leibniz principle: if $P : A \to \mathbf{Set}$ then can define $\mathit{subst} : \Pi x, y{:}A.\mathit{Id}(x, y) \to P(x) \to P(y)$ with β-rule $\mathit{subst}(x, x, \mathit{refl}(x), p) \rightsquigarrow p$.
- Symmetry, transitivity of Id:

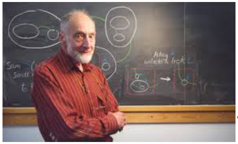
$$\begin{aligned} \mathit{sym} &: \Pi x, y{:}A.\mathit{Id}(x, y) \to \mathit{Id}(y, x) \\ \mathit{trans} &: \Pi x, y, z{:}A.\mathit{Id}(x, y) \to \mathit{Id}(y, z) \to \mathit{Id}(y, x) \end{aligned}$$

- Congruence rules with respect to almost all term formers. In particular $\mathit{resp} : \Pi f : A{\to}B.\Pi x, y{:}A.\mathit{Id}_A(x, y) \to \mathit{Id}_B(f(x), f(y))$.
- Can define elements of Id by induction (recursion), thus e.g. find an inhabitant ("proof") of

$$\Pi A{:}\mathbf{Set}.\Pi l{:}\mathrm{List}(A).\mathit{Id}(\mathrm{rev}(\mathrm{rev}(l)), l)$$

Definitional equality

- Definitional equality is the congruence induced by β-reduction.
- Definitionally equal terms and types are identified ("syntactic congruence")
- Definitional equality is decidable (inhabitance of $Id(x, y)$ a.k.a. *propositional equality* is not).


?

Extensional type theory

- Definitional and propositional equality identified.
- If $Id(u, v)$ is inhabited then $u = v$ may be concluded.
- Definitional equality and thus type checking becomes undecidable.
- Valid typing judgements should be accompanied by a (sufficiently verbose digest of a) derivation.
- Underlying theory of Nuprl and PVS

Id is great

Coming from the Calculus of Constructions (LEGO) where equality is encoded by Leibniz formula, the identity type was quite an innovation for us.

The following type is inhabited, proof using J.

$$\Pi x, y{:}\mathbf{N}.\Pi u{:}\mathrm{Vec}(x).\Pi p{:}\mathit{Id}_{\mathbf{N}}(x, y).\mathit{Id}_{\mathbf{N}}(\mathrm{sum}(u), \mathrm{sum}(\mathit{subst}_{\mathrm{Vec}}(p, u))$$

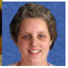
For the first time we could not only define but also reason about dependent functions and data types.

Fortunately, LEGO had support for arbitrary rewrite rules so we could easily put in the identity type even before the Calculus of Inductive Constructions became available.

Extensional concepts

It was known for a long time that propositional equality suffered from some unnecessary (?) defects:

- From $\Pi x{:}A.\mathit{Id}_B(f(x), g(x))$ cannot conclude $\mathit{Id}_{A\to B}(f, g)$
- Cannot redefine propositional equality by quotienting (real numbers, streams, modular arithmetic, ...)
- Cannot conclude $\mathit{Id}_{\mathrm{Prop}}(\varphi, \psi)$ from $\varphi \leftrightarrow \psi$
- Cure: book equality, setoids, OTT, axioms for extensional concepts, setoid interpretation



Uniqueness of identity

- The extensional concepts somehow are orthogonal or even in conflict with the view of propositional equality as an inductive definition.
- But the following are not!
  - Uniqueness of identity proofs (UIP): $\Pi x, y{:}A.\Pi p, q{:}Id_A(x, y).Id_{Id_A(x,y)}(p, q)$
  - Streicher's K: $\Pi C : \Pi x{:}A.Id_A(x, x) \to \mathbf{Set}.(\Pi x{:}A.C(x, \mathit{refl}(x))) \to (\Pi x{:}A.\Pi p{:}Id(x, x).C(x, p))$
  - Congruence for the second projection: $\Pi a{:}A.\Pi b, b'{:}B(a).Id_{\Sigma x:A.B(x)}((a, b), (a, b')) \to Id_{B(a)}(b, b')$

  None of these are inhabited (as we now know), but they are interdefinable and inhabited in an extension of Martin-Löf type theory with pattern matching (Coquand).

"Identity equations"

While not all equality proofs are equal (at least we didn't know how to prove it) some identities *are* provable:

$$\Pi x, y{:}A.\Pi p{:}Id(x, y).Id_{Id(x,y)}(p, trans(refl(x), p))$$

We use

$$C(x, y, p) := Id_{Id(x,y)}(p, trans(refl(x), p))$$

and have

$$h := \lambda x{:}A.refl(refl(x)) : \Pi x{:}A.C(x, x, refl(x))$$

This is because by β-reduction $trans(refl(x), refl(x)) \rightsquigarrow refl(x)$.
Recall the definition of *trans* in terms of J.

Notation and more identity equations

Let us abbreviate: $trans(p, q)$ by qp and $refl(x)$ by id_x. Let us also write $p \sim q$ when $Id_{Id(x,y)}(p, q)$ is inhabited. We have just proved:

$$p\ id_x \sim p$$

In a similar way, we can also prove:

$$id\ \ p \sim p$$

$$p(qr) \sim (pq)r$$

i.e. $Id(trans(trans(r, q), p), trans(r, trans(q, p)))$ is inhabited.
Thus, intuitively, each type forms a category with its members as objects, and *Id*-proofs as morphisms.

Symmetry as an inverse

Recall $sym : \Pi x, y{:}A.Id(x, y) \to Id(y, x)$. Writing $p^{-1} := sym(p)$ we can prove:

$$p\ p^{-1} \sim 1$$

$$p^{-1}\ p \sim 1$$

So, the category of identity proofs is in fact a *groupoid* (a category where all morphisms are isomorphisms). Up to $\sim$, that is.

Going further

One can also show that:

- Every type-theoretic function is a functor between groupoids
- Every dependent type is a groupoid-valued functor or “split fibration”
- In particular the *subst*-maps arising from the Leibniz-principle are compatible with transitivity, etc.

Models of type theory

- The set-theoretic model: types as sets, dependent types as families of sets, functions as set-theoretic functions.

$$[\![Id(x,y)]\!] = \begin{cases} \{\star\}, \text{ if } x = y \\ \emptyset, \text{ otherwise} \end{cases}$$

- Domain-theoretic model: types as domains, dependent types as families of domains, functions as continuous functions
- Realizability model: types as ω-sets, dependent types as families of ω-sets, functions as realizable functions
- Term model: types as types, ...
- *Deliverables* model: types as types together with a predicate, functions as functions together with a proof that the predicates are preserved. Provides support for subset and squash types.
- *Setoid* model: types as types together with a (partial) equivalence relation, functions as functions together with a proof that the equivalences are respected. Provides support for extensional concepts, notably quotient types.

The groupoid interpretation

- Interpret types as groupoids (small categories in which all morphisms are isomorphisms)
- Interpret functions as functors between groupoids
- Interpret dependent types as groupoid-valued functors
- If A is a groupoid then Id_A is the following groupoid valued functor:

  $Id_A(x, y) = A(x, y)$, the A morphisms from x to y with trivial equality

- This means that

$$Id_{Id_A}(p, q) = \begin{cases} \{\star\}, \text{ if } p = q \\ \emptyset, \text{ otherwise} \end{cases}$$

- Choosing A as a nontrivial groupoid, e.g. $A = \{\star\}$ and $A(x, y) = (\mathbb{Z}, +, 0)$, should yield a countermodel to UIP (uniqueness of identity proofs), hence a proof that UIP cannot be uniformly derived by the rules of type theory.
- This works, but there are some interesting points to be looked at:

Functor part of Id_A

- We must also define the functor part of Id_A.
- Given a morphism from (x, y) to (x', y'), i.e., $p : x \to x'$ and $q : y \to y'$, we must define a functor $Id_A(p, q)$ between the (trivial) groupoids $Id_A(x, y)$ and $Id_A(x', y')$. If $r \in Id_A(x, y)$ put

$$Id_A(p, q)(r) = qrp^{-1}$$

- Notice that p^{-1} is the inverse morphism of p in the groupoid A.

$$\begin{tikzcd}
x \arrow[r, "p"] \arrow[d, "r"'] & x' \arrow[d, "qrp^{-1}"] \\
y \arrow[r, "q"'] & y'
\end{tikzcd}$$

The associated term formers

- We must interpret the *refl*-constructor. That's easy:

$$refl(x) = id_x \in Id_A(x,x) = A(x,x)$$

- J is more difficult and crucially relies on the fact that dependent types are interpreted as *split* fibrations. Intuition:
  - Given h and an equality proof $r : Id(x,y)$ we get a morphism from (x,x,id_x) to (x,y,r), namely $(id_x, r, \star)$. Indeed, $r\ id_x = r\ id_x$.
  - Thus, we define $J(h)(x,y,r) := C(id_x, r, \star)(h(x))$

$$h(x) \in C(x,x,id_x) \quad \Longrightarrow \quad C(id_x, r, \star)(h(x)) \in C(x,y,r)$$

$$\begin{tikzcd}(x,x,id_x) \arrow[r, "{(id_x, r, \star)}"] & (x,y,r)\end{tikzcd}$$

Building K ?

In order to interpret K of type
$K(h) : \Pi x{:}A.\Pi p{:}Id_A(x,x).C(x,p)$ given $h : \Pi x{:}A.C(x, refl_A(x))$ we would need a morphism from (x, id_x) to (x, r) for each r from x to x which anmounts to a *single morphism* p from x to x such that

$$\begin{tikzcd}
x \arrow[r, "p"] \arrow[d, "id_x"'] & x \arrow[d, "r"] \\
x \arrow[r, "p"'] & x
\end{tikzcd}$$

But then $r = pp^{-1} = id_x$. So, unless the underlying groupoid is trivial, this does not work.

Remaining work

- Still need to define morphism part and verifications,...
- Still need to interpret dependent Σ-types and Π-types.
- In order to do all that rigorously and systematically it is best to use an abstract notion of model like *categories with attributes* or *categories with families*.

Isomorphism as equality

- G. Kreisel: what more do we know if we prove something with restricted means?
- In the case of Martin-Löf type theory the absence of UIP allows one to treat isomorphism as equality:
- We can soundly assume a constant:

$$iso\_eq : \Pi A, B{:}Set.Bij(A,B) \to Id_{Set}(A,B)$$

  where $Bij(A,B)$ is the type of bijections between A and B.
- We can also assume an equation:
  If $a \in A$ and $f \in Bij(A,B)$ then

$$subst(iso\_eq(f), a) \rightsquigarrow f(a)$$

  "transporting" along $iso\_eq(f)$ is like applying f.
- Clearly, this is in conflict with UIP, but soundly interpretable in the gorupoid model: Interpret *Set* as the groupoid whose objects are (small) sets and where morphisms are bijections.

Application of universe extensionality

- As an application we can define *Set*-valued functors in the naive way:
- such a functor comprises functions $F_0 : \mathrm{Ob} \to \mathit{Set}$ and $F_1 : \Pi x, y{:}\mathrm{Ob}.\mathrm{Mor}(x, y) \to F(x) \to F(y)$.
- and proofs (in terms of *Id*) of the functor laws.
- Ob and Mor refer to some fixed category, e.g. certain maps on finite sets,. . .
- Now from universe extensionality ($\mathrm{iso\_eq}$) and functional extensionality (not described) one can prove that functors are equal iff they are naturally isomorphic.

The important notion of natural isomorphism just sort of pops out!

Non-example: symmetric monoidal categories

- Symmetric monoidal category: category with binary operation $\otimes$ on objects and morphisms ("parallel composition"). Applied in algebra, concurrency, linear logic, . . .
- Strict variant: $A \otimes (B \otimes C) = (A \otimes B) \otimes C$ and $f \otimes (g \otimes h) = (f \otimes g) \otimes h$. Disadvantage: often not satisfied, e.g., $A \otimes B = A \times B$ (cartesian product).
- Non-strict variant: $A \otimes (B \otimes C) \simeq (A \otimes B) \otimes C$ (naturally!) plus MacLane's pentagon: the two ways of $\simeq$-rewriting $((A \otimes B) \otimes C) \otimes D$ to $A \otimes (B \otimes (C \otimes D))$ are equal.
- If we replace $=$ by Id, do strict and non-strict coincide?
- Naturality of the isomorphisms again pops out, but pentagon does not.

$$\begin{tikzcd}
((A \otimes B) \otimes C) \otimes D \arrow[r, "\alpha_{A,B,C} \otimes 1_D"] \arrow[d, "\alpha_{A \otimes B,C,D}"'] & (A \otimes (B \otimes C)) \otimes D \arrow[r, "\alpha_{A,B \otimes C,D}"] & A \otimes ((B \otimes C) \otimes D) \arrow[d, "1_A \otimes \alpha_{B,C,D}"] \\
(A \otimes B) \otimes (C \otimes D) \arrow[rr, "\alpha_{A,B,C \otimes D}"'] & & A \otimes (B \otimes (C \otimes D))
\end{tikzcd}$$

Another application: "suggesting" equations

- Remember that the groupoid structure of Id_A is provable from the syntax.
- So, it's there hidden in Martin-Löf's axioms for equality.
- Turns out, that also the proofs, say of associativity of *trans*, satisfy some equations, etc.
- The syntax of Martin-Löf type theory induces the structure of a *weak* ω-groupoid: never any actual equations, just equivalences satisfying coherence laws that again hold up to isomorphism, ...
- At the time (late 90s) people didn't even know how to define such a thing.

Trying to get people interested. . .

```
Return-Path: <mxh@dcs.ed.ac.uk>
Date: Wed, 18 Nov 1998 15:54:10 GMT
To: baez@math.ucr.edu
Subject: Groupoids & Martin-Loef type theory
From: "Martin Hofmann" <mxh@dcs.ed.ac.uk>

Dear John Baez,

It is with some interest that I follow as a spectator the recent
interest in higher-order category theory by theoretical physicists
notably you and your group.

In my own research on Martin-Loef type theory (don't worry if you
have no clue what that is...) I've come across an intriguing applica
groupoids and [...] I kept wondering whether there might also be a
relationship between Martin-Loef type theory and your applications
groupoids in physics. (Of course, in general "having an interesting
relationship" is not transitive ....).
```

Another attempt 10 years later

…after talking to JB in person

```
Date: Fri, 14 Aug 2009 17:17:15 -0700
Message-ID: <179b05930908141717t6fc69388l442c4330cbec8a16@mail.gmai
Subject: Re: your talk at LICS
From: John Baez <john.c.baez@gmail.com>
To: hofmann@ifi.lmu.de

Hi -

 I found your work on infinity-groupoids from Martin Loef type
 theory to be very inspiring - this is exactly the sort of
 approach to logic that I'm excited by.
```

And in Week 279 (Sept. 2009) of “This weeks finds”:

```
 I also won’t tell you about the new revolution
linking logic to weak -groupoids. For that you’ll have to read thes

15) Martin Hofmann and Thomas Streicher,
 The groupoid interpretation of type theory, in [...]

16) Steve Awodey and Michael A. Warren, Homotopy theoretic
   models of identity types, available as arXiv:0709.0248.

17) Steve Awodey, Pieter Hofstra, Michael A. Warren,
 Martin-Lf Complexes, available as arXiv:0906.4521.

18) Benno van den Berg and Richard Garner, Types are
 weak omega-groupoids [...]
```

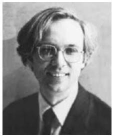
Learning about Homotopy type theory

```
From: Helmut Schwichtenberg <schwicht@mathematik.uni-muenchen.de>
To: hofmann@ifi.lmu.de,
Thomas Streicher <streicher@mathematik.tu-darmstadt.de>
Cc: vladimir@ias.edu, morel@mathematik.uni-muenchen.de
Subject: Vladimir Voevodsky on homotopy lambda calculus
Date: Mon, 09 Nov 2009 17:44:09 +0100

Lieber Herr Hofmann, lieber Herr Streicher,

I write this in English because of the cc above.  Vladimir Voevodsk
is visiting LMU this week (Fabien Morel is his host), and he will g
a talk at our math colloquium next Friday (13.11., 16:15, Room B 00
on homotopy lambda calculus.  In a discussion we had today it was
obvious that there will be many relations to the theses of the two
you.  It would be very nice if you could come to this lecture (mayb
even from Darmstadt?)
```

Quillen model structures

Retrospectively, I know that the *Quillen model structures* invented well before Martin-Löf type theory also contain ω-groupoids in implicit form.

- Fibrations $\Leftrightarrow$ Dependent types
- Cofibrations $\Leftrightarrow$ Isomorphisms up to *Id*
- Factorisation of diagonal $\Leftrightarrow$ Identity type + *refl*
- Diagonal fill-in $\Leftrightarrow$ *J*-elimination rule
- Awodey, Shulman, and others have shown how Quillen model structures yield models of type theory and vice versa. Although some object about degree of rigour...
- Possible advantage of type theory: more explicit syntax, can be completely formalized
- Possible advantage of Quillen model structure: More "semantic", should be easier to discover in actual mathematical structures



Recent use of groupoids: proof-relevant logical relations

In ongoing work (TLCA 13, POPL14,. . .) we (Benton, Nigam, H.) use groupoids (for the time being only setoids, to be honest) to describe the current assumptions on the heap in a functional program with side-effects. One specific curiosity that may be of interest:

- In order to accommodate general recursion we need groupoids relative to complete partial orders ("Scott domains"), but
- There do not exist nontrivial group objects in the category of posets: The inverse operation would have to be both monotone and antitone, thus equal to the identity.
- There do, however, exist nontrivial CPO-enriched groupoids.
  - $A = \{\bot, \top\}$, $A(\bot, \bot) = \{\star\}$, $A(\top, \top) = \mathbb{Z}$. $A(\top, \bot) = A(\top, \bot) = \emptyset$. Order on morphisms: $(\bot, \star\bot) \leq (\top, 0, \top)$.
  - $B = (\mathcal{P}(\mathbb{N}), \subseteq)$, $B(U, V) = Bij(U, V)$, $(U, \varphi, V) \leq (U', \varphi', V') \iff U \leq U', V \leq V', \varphi'|U = \varphi$.
- Possible to define general recursion in the resulting category of CPO-enriched groupoids.