Towards an ω -groupoid model of Type Theory Based on joint work with Ondrej Rypacek

Thorsten Altenkirch

Functional Programming Laboratory School of Computer Science University of Nottingham

August 14, 2012

Background

- In Type Theory for any A: Type and a b: A we can form a new type a = b: Type, the set of proofs that a is equal to b.
- The canonical way to prove an equality is refl : a = a.
- Using the standard eliminator (J) we can show that equality is a congruence.
- Since *refl* is the only constructor we would assume that all equality proofs are equal (*uniqueness of equality proofs*).
- However, this is not provable using the standard eliminator (J).
- This was shown by Hofmann and Streicher using the *Groupoid model* of Type Theory.

Homotopy Type Theory

- Voevodsky proposed an interpretation of Type Theory using Homotopy Theory.
- Types are interpreted as topological spaces and equality proofs as paths (homotopies).
- This interpretation doesn't support uniqueness of equality proofs,
 i.e. (α : a = a) → α = refl is not provabble.
- However, it does support the standard eliminator (J), in particular we can prove: that given a: A for all p: (b: A) × (a = b) we have p = (a, refl).

Dimensions (Homotopy levels)

- We say that a type is *contractible* or 0-dimensional, if it contains exactly one element, i.e. there is (a : A) × (b : A) → a = b.
- A type is n + 1-dimensional, if all its equalities are *n*-dimensional.
- We arrive at the following hierarchy:
 - 0 contractible types
 - 1 propositions
 - 2 sets
 - 3 ???
- We can show that if a type is *n*-dimensional then it is also n + 1-dimensional.
- Uniqueness of equality proofs means that all types are 2-dimensional.

Weak equivalence

- The notion of weak equivalence can be expressed in Type Theory.
- A function *f* : *A* → *B* is a weak equivalence if the type (*a* : *B*) × *f a* = *b* is contractible for *b* : *B*.
- A and B are weakly equivalent.
- Weak equivalence in different dimensions:
 - 0contractible typestrivial1propositionslogical equivalence2setsisomorphism3???weak equivalence
- Univalence axiom (Voevodsky): Weak equivalence is weakly equivalent to equality.
- Univalence implies functional extensionality (Voevodsky): Any two functions which are pointwise equal are equal.

Why are we interested in this?

- We have found a fascinating connection between Type Theory and Homotopy Theory.
- We can use Type Theory to formalize constructions in Homotopy Theory.
- However, most Computer Scientists don't care about Homotopy theory.
- Is there a way to motivate the univalence axiom which has nothing to do with homotopy theory?

Extensionality

Leibniz principle

Any two objects should either have a property which distinguishes them or they should be equal.

- This principle justifies functional extensionality (black box view of functions).
- Isomorphic sets cannot be distinguished in Type Theory hence they should be equal.
- Isomorphism is not the correct notion from dimension 3 because it lacks a coherence property.
- This is fixed by weak equivalence (being a weak equivalence is propositional while being an isomorphism is not.
- Note that the Leibniz principle is not satisfied by Extensional Type Theory.

Open problem

Canonicity

Any closed term inhabiting a datatype (like \mathbb{N}) should be definitionally (strictly) equal to a term in constructor form (starting with a constructor).

- This is justified by Intensional Type Theory due to the normalisation property.
- Assuming univalence destroys canonicity.
- How can we have the univalence principle and keep canonicity?
- How can we eliminate univalence?

Deja vue

• This is reminiscient to the problem of eliminating functional extensionality in Type Theory.

$$ext: (f g: (x:A) \longrightarrow B x) \longrightarrow ((x:A) \longrightarrow f x = g x) \longrightarrow f = g$$

- We have proposed a solution to this problem (LICS 99) which relies on a translation using the *Setoid model*.
- This was later (PLPV 08) refined in joint work with Conor McBride and others (*Observational Type Theory*).
- However, the construction relies on a strong form of proof irrelevance.

Sketch of the construction

- We define a translation from a source type theory to a target type theory.
- The target type theory doesn't have a equality types.
- The source type theory does have equality types and *ext* is inhabited. This is explained by the translation.
- The translation preserves datatypes (like ℕ) and hence canonicity holds.
- The target type theory features a universe of (strictly) proof-irrelevant propositions *Prop*.

Prop

- Prop is a subuniverse of Set, i.e. P : Prop implies P : Set.
- If *P* : *Prop* and *p q* : *P* then *p* and *q* are definitionally equal.
- Prop is closed under Π- and Σ-types. That is
 - If A : Set and $P : A \longrightarrow Prop$ then $(x : A) \longrightarrow P x : Prop$
 - If P : Prop and $Q : P \longrightarrow Prop$ then $(x : P) \times Q x : Prop$.
 - \top : *Prop* and \bot : *Prop*.

The translation

- A closed type A : Set in the source theory is translated as a setoid in the target theory, i.e.
 - A.set : Set a type in the target theory
 - $\sim A : A.set \longrightarrow A.set \longrightarrow Prop$ a relation
 - Proofs of refl, sym, trans.
- A family of types B: A → Set is modelled as a functor from (A.set, ~A) into the category of Setoids, i.e.
 - ► A family *B.fam* : *A.set* → *Setoid* in the target theory.
 - A term subst : $(a \sim A a') \longrightarrow (B.fam a)$.set $\longrightarrow (B.fam a')$.set.
 - Proofs that subst is functorial upto setoid equality.

Translating Π-types

- Given a setoid A : Set and a family of setoids B : A → Set we construct a setoid ∏ A B : Set
 - ► ($\Pi A B$) .set is the set of functions which preserve equality, i.e. (f: (x: A.set) \longrightarrow (B.fam x) .set) × ((p: x~A y) \longrightarrow B.subst p (f x)~(B.fam y) f y
 - Equality $f \sim (\prod A B) g$ is extensional equality, i.e. $(x : A.set) \longrightarrow f x \sim (B.fam x) g x$
- In fact we have to generalize this construction to the case $B: A \longrightarrow Set$ and $C: (a: A) \times B a \longrightarrow Set$ leading to $\Pi B C: A \longrightarrow Set$.

Interpreting equality

- In the source type theory we interpret equality using $\sim A$.
- Equality for Π-types is extensional by definition.
- We can derive the eliminator from *subst* and the fact that equality is definitionally proof-irrelevant.
- N is translated by itself using the definable recursive equality on natural numbers.
- We can also interpret quotient types.
- We can use logical equivalence as the equality for propositions hence we can eliminate univalence for propositions.

Proof - irrelevance

- We need equations between equality proofs at several points of the construction, e.g. when verifying the functor laws for *subst* for Π.
- All these equations hold trivially because of definitional proof-irrelevance.
- In the end we need to derive *J* from *subst*.
- This also requires definitional proof-irrleevance.

Observational Type Theory

- Instead of defining the Source Type Theory by a translation, we can define it directly.
- We define = :A → A → Prop by recursion over the type A and subst : (P : A → Type) → a = b → P a → P b by recursion over the family P : A → Type.
- The other constants don't need to be defined because they live in *Prop*.
- Using a clever trick we can also address the problem that subst P refl is not definitionally equal to the identity.
- For details see our PLPV 2008 paper (jointly with Conor McBride and Wouter Swierstra).

From Setoids to Groupoids

- The setoid construction allows us to interpret types upto dimension 2.
- Replacing setoids by groupoids we can interpret types upto dimension 3.
- We have to use Groupoids enriched over Setoids:
 ∼A: A.set → A.set → Prop is replaced by
 ∼A: A.set → A.set → Setoid.
- The groupoid equations hold up to setoid equality.
- We can define equality of the universe of sets as isomorphism hence we can interpret univalence at dimension 2.
- Carrying out this construction in detail would provide an alternative to Harper's and Licata's proof of canonicity of 2-dimensional Type Theory.
- Our proposal would also address the issue that they have been using an extensional Type Theory at dimension 2.

From Groupoids to ω -Groupoids

- We would like to eliminate the *Prop*-universe.
- If we can construct a groupoid model enriched over itself we should be able to do this.
- And we should be able to interpret univalence at any level.
- This would require to construct an ω -Groupoid model of Type Theory.
- As a first step we need to define what is a ω -Groupoid in Type Theory.

What are weak ω -groupoids?

- There are a number of definitions in the literature, e.g. based on contractible globular operads.
- We need to formalize them in Type Theory ...
- Formalizing the required categorical concepts creates a considerable overhead.
- Also it is not always clear how to represent them in the absence of UIP.
- E.g. what are strict ω -groupoids?

Globular sets

We define a *globular set G* : Glob coinductively:

$$\operatorname{obj}_G$$
 : Set
 hom_G : $\operatorname{obj}_G \to \operatorname{obj}_G \to \infty$ Glob

Given globular sets A, B a morphism f : Glob(A, B) between them is given by

 $\operatorname{obj}_{f}^{\rightarrow}$: $\operatorname{obj}_{A} \to \operatorname{obj}_{B}$ $\operatorname{hom}_{f}^{\rightarrow}$: $\Pi a, b : \operatorname{obj}_{A}$. $\operatorname{Glob}(\operatorname{hom}_{A} ab, \operatorname{hom}_{B}(\operatorname{obj}_{f}^{\rightarrow} a, \operatorname{obj}_{f}^{\rightarrow} b))$

As an example we can define the terminal object in $\mathbf{1}_{\text{Glob}}$: Glob by the equations

$$obj_{1_{Glob}} = 1_{Set}$$

 $nom_{1_{Glob}} x y = 1_{Glob}$

The Identity Globular set

More interestingly, the globular set of identity proofs over a given set *A*, $Id^{\omega} A$: Glob can be defined as follows:

$${
m obj}_{{
m Id}^\omega\,{\it A}} = {\it A}$$

 ${
m hom}_{{
m Id}^\omega\,{\it A}}\,{\it a}\,{\it b} = {
m Id}^\omega\,({\it a}={\it b})$

Globular sets as a presheaf

Our definition of globular sets is equivalent to the usual one as a presheaf category over the diagram:

$$0 \xrightarrow[t_0]{s_0} 1 \xrightarrow[t_1]{s_1} 2 \dots n \xrightarrow[t_n]{s_n} (n+1) \dots$$

with the globular identities:

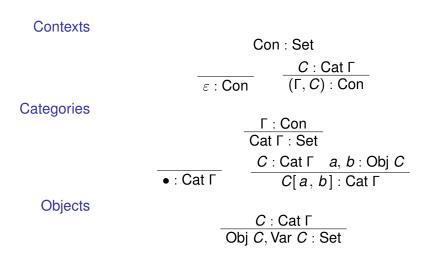
$$t_{i+1} \circ s_i = s_{i+1} \circ t_i$$

 $t_{i+1} \circ t_i = s_{i+1} \circ t_i$

A syntactic approach

- When is a globular set a weak ω -groupoid?
- We define a syntax for objects in a weak ω -groupoid.
- A globular set is a weak ω -groupoid, if we can interpret the syntax.
- This is reminiscient of environment λ-models.

The syntactical framework



Interpretation

An assignment of sets to contexts:

An assignment of globular sets to category expressions:

$$\frac{C: \mathsf{Cat} \ \mathsf{\Gamma} \qquad \gamma : \llbracket \mathsf{\Gamma} \rrbracket}{\llbracket C \rrbracket \ \gamma : \mathsf{Glob}}$$

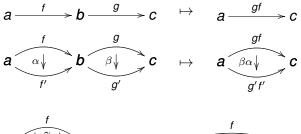
Assignments of elements of object sets to object expressions and variables

$$\begin{tabular}{cccc} \hline C: Cat Γ & A: Obj C & γ: [[\Gamma]] \\ \hline $[A]]$ γ: obj_{[C]]$ γ} \end{tabular}$$

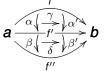
• Subject to some (obvious) conditions such as:

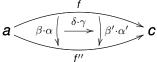
$$\llbracket \bullet \rrbracket \gamma = G$$
$$\llbracket C[a, b] \rrbracket \gamma = \hom_{\llbracket C \rrbracket \gamma} (\llbracket a \rrbracket \gamma) (\llbracket b \rrbracket \gamma)$$

Composition



 \rightarrow





Telescopes

A telescope t: Tel C n is a path of length n from a category C of to one of its (indirect) hom-categories:

$$\frac{C: \operatorname{Cat} \Gamma \quad n: \mathbb{N}}{\operatorname{Tel} C n: \operatorname{Set}}$$

We can turn telescopes into categories:

<u>t : Tel C n</u> C ++ t : Cat Г

Formalizing composition

$$\frac{\alpha: \mathsf{Obj}(t \Downarrow) \qquad \beta: \mathsf{Obj}(u \Downarrow)}{\beta \circ \alpha: \mathsf{Obj}(u \circ t \Downarrow)}$$

is a new constructor of Obj where

$$\frac{t: \text{Tel} (C[a, b]) n \qquad u: \text{Tel} (C[b, c]) n}{u \circ t: \text{Tel} (C[a, c])}$$

is a function on telescopes defined by cases

$$\bullet \circ \bullet C = \bullet \qquad u[a',b'] \circ t[a,b] = (u \circ t)[a' \circ a,b' \circ b]$$

Laws

For example the left unit law in dimension 1:

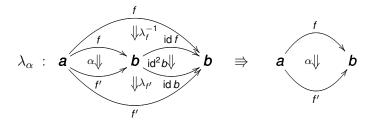
$$\mathsf{id}_b \circ f = f , \qquad (1)$$

and in dimension 2.

$$\mathsf{id}_b^2 \circ \alpha \quad = \quad \alpha \; ,$$

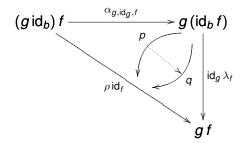
where $id_b^2 = id_{id_b}$

In the strict case the 2nd equation only type-checks due to the first. In the weak case we have to apply the previous isomorphism explicitely.



Coherence

Example:



In summary and full generality:

For any pair of coherence cells with the same domain and target, there must be a mediating coherence cell.

Problem

This definition of coherence is too strong!

Formalizing coherence

x : Obj Chollow x : Set

hollow $(\lambda_{-}) = \top \ldots$

 $\frac{f \ g : \text{Obj} \ C[a, b]}{\cosh p \ q : \text{Obj} \ C[a, b][f, g]} \xrightarrow{q : \text{hollow} \ g}$

hollow (coh pq) = \top

Summary

- To be able to eliminate univalence we want to interpret Type Theory in a weak ω -groupoid in Type Theory.
- As a first step we need to define what is a weak ω -groupoid.
- Our approach is to define a syntax for objects in a weak ω -groupoid.
- A globular set is a weak ω groupoid if we can interpret this syntax.
- See our draft paper for details: A Syntactical Approach to Weak ω-Groupoids

Further work

- We need to fix our definition of coherence!
- The current definition is quite complex can we simplify it?
- Can we actually show that the identity globular set is a weak ω-groupoid, internalizing results by Lumsdaine and Garner/van de Berg?
- What is a model of Type Theory in a weak ω -groupoid.
- Can we use this construction to eliminate univalence?