Variation on cubical sets

Oxford, November 9, 2014

This work

From discussions with Cyril Cohen, Simon Huber and Anders Mörtberg

Univalent Foundation

Previous work: a constructive model of dependent type theory together with

$1_a : \mathsf{Id}_A(a,a)$

$\mathsf{transp} : C(a) \to \mathsf{Id}_A(a,x) \to C(x)$

$\mathsf{Id}_{C(a)}(\mathsf{transp}\ u\ 1_a, u)$

$\mathsf{Id}_{(\Sigma x:A)\mathsf{Id}_A(a,x)}((a,1_a),(x,p))$

Univalence Axiom ($+$ Circle)

Dependent type theory: $\Pi, \Sigma, U, N, W(A,B), N_0, N_1, N_2 \ldots$

Computational Interpretation

A type is interpreted by a cubical set with face operations and the simplest possible notion of degeneracy

Simple enough so that we write a Haskell implementation

In particular, we can effectively transport structures along equivalences

Computational Interpretation

However

(1) the computation rule for identity elimination is only justified as a *propositional* equality

(2) some computation rules are not “canonical”; in particular the computation rules for *function extensionality* and for *circle elimination*

This work

We present a *refinement* of the cubical set model

We add *connection* and *diagonal* operations

With *connections* we can reduce the Kan filling operation to a simpler composition operation

With *diagonals* we get a better interpretation of function extensionality and elimination rule over higher inductive type

This work

The system we interpret contains the rules of Martin-Löf type theory with intensional equality

But the justification of the rules for equality is different

In Martin-Löf type theory equality is inductively defined (least reflexive relation)

Here, equality on A is explained by induction on A

The axiom of univalence is justified by this explanation

This work

Still simple enough to be represented in Haskell

We also have experimented with a simple form of higher inductive types (designed by S. Huber)

e.g. pushout, suspension, spheres, propositional truncation

proof that S^1 is equal to the suspension of Bool and transport functions

Constructive models

A *computational model* of univalence is the same as a *constructive model*

For dependent type theory, the computations are done in λ-calculus

For univalence, *nominal* extension of λ-calculus

In particular, we get a justification of *function extensionality* without using function extensionality in the metalogic

Models

Usual models: Kan simplicial sets, Kan cubical sets

They make use in a crucial way of classical logic

Models

(1) B^A is Kan if B is Kan

"Originally, all the work was expressed in term of the extension condition and some rather grisly-looking combinatoric was involved. The Gabriel-Zisman theory of anodyne extensions gives a way to short-circuit most of the pain" (Goerss-Jardine *Simplicial homotopy theory*)

However the argument with anodyne extensions is still *non effective*

(2) if $E \rightarrow B$ Kan fibration and $b_0 \rightarrow b_1$ in B then $E(b_0)$ and $E(b_1)$ are homotopy equivalent

Kripke countermodel showing that this cannot be proved effectively (j.w.w. Marc Bezem)

Cubical sets

if X is a topological space we can consider $X(I)$ set of continuous functions

$$[0,1]^I \rightarrow X$$

for I finite set

Elements of I are called symbols/names/dimensions

Cubical sets

We want a “combinatorial” definition, also effective at function types

Goal: to design a system of *notations* so that

the syntactical description of the model can be seen as an

operational semantics

Cubical sets

Simplest cubical sets

Maps $I \to J$ consists of a disjoint union $I = I_0, I_1, I'$ with an injection $I' \to J$

This defines a category $\mathcal{C}$

A *cubical set* is a presheaf X on $\mathcal{C}^{opp}$

If $f : I \to J$ and $g : J \to K$ we write $fg : I \to K$

Think of f as a name substitution, and define composition following this intuition

Cubical sets

An I-cube is an element of $X(I)$

E.g. from the cube $u(i, j, k)$ we can consider

its face $u(0, j, k)$ or

its edge $u(0, j, 1)$

adding a fresh dimension l correspond to degeneracy

Cubical sets

A *face map* is a map such that where $J = I'$ and $I' \rightarrow J$ is the identity

A face map is *epi* and composition of elementary face maps $(i0)$ or $(i1)$

A map is *strict* if $I_0 = I_1 = \emptyset$

Strict maps correspond to degeneracies

Any map $f : I \rightarrow J$ decomposes uniquely in the form

$$f = \alpha g$$

where α face map and g is strict

Cubical sets

A closed type $\vdash A$ will be interpreted as a presheaf over $\mathcal{C}^{opp}$

A closed type $\vdash_I A$ which may depend on the dimension in I will be interpreted as a presheaf over $\mathcal{C}^{opp}/I$

We can write $A(i_1, \ldots, i_n)$ if $I = i_1, \ldots, i_n$

We have $\vdash_J Af$ if $\vdash_I A$ and $f : I \to J$

E.g. if we have $A(i, j)$ we can consider its face $A(0, j)$

Cubical sets

We want a syntax for describing the presheaf model

We introduce judgement $\vdash_I T$ and $\vdash_I a : T$ with the *restriction rule*

$$\frac{\vdash_I T}{\vdash_J Tf} \qquad \frac{\vdash_I a : T}{\vdash_J af : Tf} \qquad f : I \to J$$

Cubical sets

Similar categories have been considered in the theory of *nominal sets*

Staton (2010), exercise 9.7 in Andy Pitts' book

Nominal Sets. Names and Symmetry in Computer Science

The interval $\mathbf{I}$ is defined as the presheaf

$\mathbf{I}(L) = L + \{0, 1\}$

We also have a natural operation $\mathsf{Path}(X)$

$\mathsf{Path}(X)(L) = X(L, i)$ where i is fresh

However $\mathsf{Path}(X)$ is *not* $X^{\mathbf{I}}$

Cubical sets

Syntactically (where $\iota_i : J \to J, i$ is the inclusion)

$$\frac{\vdash_J T \qquad\qquad \vdash_{J,i} a : T\iota_i}{\vdash_J \langle i\rangle a : \mathsf{Path}(T)}$$

$$\frac{\vdash_J a_0 : T \qquad \vdash_J a_1 : T \qquad \vdash_{J,i} a : T\iota_i \qquad \vdash_J a(i0) = a_0 : T \qquad \vdash_J a(i1) = a_1 : T}{\vdash_J \langle i\rangle a : \mathsf{Id}_T\ a_0\ a_1}$$

Name abstraction

If $\vdash a : T$ then $\langle i\rangle(a\iota_i)$ is a constant path

$(\langle i\rangle a)g = \langle j\rangle ah$ if $g : J \to K$ and $h = (g, i = j) : J, i \to K, j$

Diagonals

We take away the restriction $I' \to J$ injective

We can now consider a map $\{i, j\} \to \{k\}$

Sends a square $u(i, j)$ to its diagonal $u(k, k)$

Now $\mathsf{Path}(X)$ is the same as $X^{\mathbf{I}}$

$\langle i \rangle a$ can be thought of as ordinary λ-abstraction over names

Kan operations

Composition and *Filling*

We consider $\vdash_{J,i} X$

If we have a_{i0} in $X(i0)$ and u_{jb} in $X(jb)$ for all j in J

and we have $a_{i0}(jb) = u_{jb}(i0)$

Then we have a_{i1} in $X(i1)$ such that

$a_{i1}(jb) = u_{jb}(i1)$

For the filling we ask to have a in X such that $a(jb) = u_{jb}$ and $a(i0) = a_{i0}$

Kan operations

"Any open box can be filled"

Kan (1955) version before simplicial sets

If $\vdash_i X$ we don't have effectively an equivalence between $X(i0)$ and $X(i1)$

Kan operations

What would be a syntax for this operation?

$$\frac{\vdash_{J,i} X \qquad \vdash_J a_{i0} : X(i0) \qquad \vdash_{J-j,i} u_{jb} : X(jb)}{\vdash_J \mathsf{comp}^i_{X,\vec{u}}(a_{i0}) : X(i1)}$$

The name i is "bound" in this operation

$$\frac{\vdash_{J,i} X \qquad \vdash_J a_{i0} : X(i0) \qquad \vdash_{J-j,i} u_{jb} : X(jb)}{\vdash_{J,i} \mathsf{fill}^i_{X,\vec{u}}(a_{i0}) : X}$$

We have for instance $\vdash_{J,i} \mathsf{fill}^i_{X,\vec{u}}(a_{i0})(i0) = a_{i0} : X(i0)$

Kan operations

If $g : J \rightarrow K$ we should explain what is

$$\mathsf{comp}^{i}_{X,\vec{u}}(a_{i0})g : X(i1)g$$

It is natural to introduce *new* Kan operations so that we can write

$$\mathsf{comp}^{i}_{X,\vec{u}}(a_{i0})g = \mathsf{comp}^{j}_{Xh,\vec{u}h}(a_{i0}g) : X(i1)g$$

if $g : J \rightarrow K$ and $h = (g, i = j) : J, i \rightarrow K, j$ where j not in K

Kan operations

Introducing these new operations solves the effectivity problem!

For instance if $\vdash_i X$ we can define effectively the equivalence map between $X(i0)$ and $X(i1)$

Any groupoid defines a cubical set with these Kan operations

$B(I)$ free Boolean algebra on I defines a Kan cubical set

Other type theoretic operations

$\Gamma \vdash_I$ is interpreted as a presheaf over $\mathcal{C}^{opp}/I$

$\Gamma \vdash_I T$ and $\Gamma \vdash_I a : T$ gets also a natural interpretation

We have e.g.

$$\frac{\Gamma, x : A \vdash_I t : B}{\Gamma \vdash_I \lambda x.t : (x : A) \to B}$$

Other type theoretic operations

The general restriction rule is

$$\frac{\Gamma \vdash_I a : T}{\Gamma f \vdash_J af : Tf}$$

Kan operations

Kan compositions explain the operation

$$\mathsf{Id}_A\ a_0\ a_1 \to B(a_0) \to B(a_1)$$

For getting the right equality for elimination rule we need a *regularity* condition

$$\mathsf{comp}^i_{A,\vec{u}}(a_{i0}) = a_{i0}$$

whenever $A, \vec{u}$ is independent of i

Kan composition for product

We define Kan composition and Kan filling for types by induction on the type

Simple for $\mathsf{Id}_A\ a_0\ a_1$ since we have Kan operations

For dependent product the definition is complex and not canonical

Essentially because the *composition* operation for $(x : A) \to B$ requires the *filling* operation for A

In particular, the regularity condition does not seem to be preserved

Connections

We change the base category $\mathcal{C}$

A morphism $I \to J$ is now a map $I \to D(J)$ where $D(J)$ is the *free distributive lattice* on J

We still have face maps α

A map $f : I \to J$ is *strict* if it never takes the values $0, 1$

Any map is uniquely the composition of a face map and a strict map

Connections

If we have a line $u(i)$ we can consider the squares $u(i \wedge j)$ and $u(i \vee j)$

The interval is now $\mathbf{I}(J) = D(J)$

Geometrically we think of $i \wedge j$ as $\mathsf{min}(i, j)$ and $i \vee j$ as $\mathsf{max}(i, j)$

Connections

With connections, we can reduce the *filling* operations to the *composition* operations

$$\mathsf{fill}^i_{A,\vec{u}}(a_{i0}) = \mathsf{comp}^j_{Af,\vec{u}f}(a_{i0}\iota_j)$$

where $f : I, i \rightarrow I, i, j$ is defined by $f(i) = i \wedge j$

Regularity condition is now preserved by dependent product

Symmetry

All operations are symmetric in 0 and 1

It is natural to add an operation $1 - i$ on names

The map $I \rightarrow J$ are now maps $I \rightarrow \mathsf{dM}(J)$ where $\mathsf{dM}(J)$ is the free *de Morgan algebra* on J

Gives an interpretation of symmetry of equality which is definitionally involutive

Demo

https://github.com/simhu/cubical

on the branch

connections_hsplit

Design choice: programming language with dependent types

Total fragment

In the total fragment conversion and type-checking are terminating

Some references

"A model of type theory in cubical sets"
M. Bezem, T.C. and S. Huber, proceeding of TYPES 2013

"Variation on cubical sets"
T.C. www.cse.chalmers.se/~coquand/comp.pdf

Names and Symmetry in Computer Science
A.M. Pitts

"An equivalent presentation of the Bezem-Coquand-Huber category of cubical sets"
A. M. Pitts. Preprint arXiv, December 2013.

Some references

The Univalent Foundation Program
Homotopy Type Theory: Univalent foundation of mathematics

V. Voevodsky Univalent foundation home page and
"Experimental library of univalent foundation of mathematics"