Variation on Cubical sets

April 9, 2014

Introduction

In the model presented in [1, 4] a type is interpreted by a nominal set A equipped with two "face" operations: if $u : A$ and x is a symbol we can form $u(x = 0) : A(x = 0)$ and $u(x = 1) : A(x = 1)$ elements independent of x. The unit interval is represented by the nominal set $\mathbf{I}$, whose elements are $0, 1$ and the symbols. The set $\mathbf{I}$ however does not satisfy the Kan filling condition.

In this model, if A represents a type, the path space of A is represented by the affine exponential $\mathbf{I} \rightarrow *A$, which is adjoint to the separated product $B * I$ with $(b, x) \in B * I$ if x is independent of b.

This means in particular that if we have $u : I \rightarrow *A$ then we cannot in general apply u to an arbitrary symbol x.

For certain operations, e.g. the realization of function extensionality, and of the elimination rule for the circle, it seems that we get more natural realizers if we could represent the path space as a real exponential $\mathbf{I} \rightarrow A$.

One way to achieve this is to allow for operations on cubes not only swapping but also arbitrary substitutions.

In term of cubical sets [1], this amounts to work the category of finite sets with maps $I \rightarrow J + 2$ (i.e. the Kleisli category for the monad $I + 2$). This category appears on pages 47–48 in Pursuing Stacks [3] as "in a sense, the smallest test category".

The path space $\mathbf{I} \rightarrow A$ and if $t : A$ and x is a symbol we can form $\langle x \rangle t$ in $\mathbf{I} \rightarrow A$ which behaves now as lambda abstraction.

In this note, we describe a way to define what should be the naturality condition for the Kan filling operation. We also present a generalization of this Kan filling operation, which provides a simple interpretation of identity type and dependent product. We give also a tentative definition of the Kan filling operation for the universe. We conjecture that it is possible to extend this interpretation to the fact that this operation builds types having the Kan property and to an interpretation of the axiom of univalence.

1 Kan condition

A *symbol* is an elemeny of $\mathbf{I}$ distinct from $0, 1$.

Like in [1] we define a *tube* for a type A and symbols $x_1, \dots, x_n$ to be a family of elements $u_{x_i b} : A(x_i = b)$ for $b = 0, 1$ such that the following compatibility condition holds $u_{x_i b}(x_j = c) = u_{x_j c}(x_i = b)$.

We experiment with the following notation: we write $u(x_i b)$ for $u_{x_i b}$ and the compatibility condition in the form $u(x_i b, x_j c) = u(x_j c, x_i b)$. In general we write $w(xb)$ for $w(x = b)$.

With these notations, a $x_1, \dots, x_n$-tube is a family $u_{x_i b}$ such that $u_{x_i b} : A(x_i b)$ and $u_{x_i b}(x_j c) = u_{x_j c}(x_i b) : A(x_i b, x_j c)$.

Given a $x_1, \dots, x_n$-tube $\vec{u}$ we define $\vec{u}(xr)$ where x is a symbol and r an element of $\mathbf{I}$.

There are 3 main cases.

If x is not $x_1, \dots, x_n$ we write $v_{x_i b} = u_{x_i b}(xr(x_i b))$ and define $\vec{u}(xr) = \vec{v}$.

If $x = x_1$ and $r = y$ is a symbol not $x_1, \dots, x_n$ then we define the following $y, x_2, \dots, x_n$-tube. We take $v_{x_1 b} = u_{x_1 b}(yb)$ and $v_{x_i b} = u_{x_i b}(x_1 y)$ if $i \neq 1$. If $r = 0$ or $r = 1$ then $\vec{u}(xr)$ is the $x_2, \dots, x_n$ tube defined by taking $v_{x_i b} = u_{x_i b}(x_1 r)$. We then have $v_{x_i b}(x_j c) = u_{x_i b}(x_1 r)(x_j c) = u_{x_j c}(x_1 r)(x_i b) = v_{x_j c}(x_i b)$ as required.

If $x = x_1$ and $y = x_2$ we define the following $x_2, \dots, x_n$-tube. We take $v_{x_2 b} = u_{x_1 b}(x_2 b)$ and $v_{x_i b} = u_{x_i b}(x_1 x_2)$ if $i \neq 2$.

We then say that A satisfies the Kan property if given another symbol z and given two distinct elements p and q in $\mathbf{I}$ (these elements may be $0, 1$ or symbols) and given $\omega_p : A(zp)$ satisfying $\omega_p(x_ib) = u(x_ib, zp_{ib})$ where $p_{ib} = p(x_ib)$ we can find an element $\omega_q : A(zq)$ satisfying $\omega_q(x_ib) = u(x_ib, zq_{ib})$ where $q_{ib} = q(x_i = b)$. Furthermore this is obtained by an operation

$$\omega_q = \mathsf{comp}_A(p, q, z, \omega_p, \vec{u})$$

In this operation, we consider z to be bound.

In the case where q is a fresh symbol y we can form

$$\mathsf{fill}_A(p, z, \omega_p, \vec{u}) = \langle y \rangle \mathsf{comp}_A(p, y, z, \omega_p, \vec{u})$$

which behaves like a Kan filling operation.

If $n = 0$ we write simply $\mathsf{comp}_A(p, q, z, \omega_p)$ instead of $\mathsf{comp}_A(p, q, z, \omega_p, \vec{u})$ where $\vec{u}$ is the empty vector.

We should then explain how to compute $\omega_q(xr)$ where x is a symbol and r an element of $\mathbf{I}$.

If $p(xr) = q(xr)$ then we should have $\omega_q(xr) = \omega_p(xr)$.

Otherwise, let us write $p' = p(xr)$ and $q' = q(xr)$ and $A' = A(xr)$.

We have already defined the tube $\vec{u}(xr)$.

If $x = x_i$ and $r = 0$ or $r = 1$ then we should have

$$\omega_q(xr) = u_{ir}(zq_{ir})$$

If $x = x_i$ and r is a symbol or if x is not among $x_1, \dots, x_n$ we define

$$\omega_q(xr) = \mathsf{comp}_{A'}(p', q', z, \omega_p(xr), \vec{u}(xr))$$

This covers all cases.

2 Dependent product

The first advantage of this presentation is that the Kan operation for dependent product seems completely canonical.

We define $\mathsf{comp}_{\Pi\ A\ B}(p, q, z, \lambda_p, \vec{w})\ \omega$ by computing

$$\theta = \mathsf{fill}_A(q, z, \omega)$$

and $\omega_p = \theta\ p$ and then

$$\mathsf{comp}_{\Pi\ A\ B}(p, q, z, \lambda_p, \vec{w})\ \omega = \mathsf{comp}_{B\ \theta}(p, q, z, \lambda_p\ \omega_p, \vec{v})$$

where $v(x_ib) = w(x_ib)\ (\theta\ z)(x_ib)$.

3 Identity type

The Kan operation for identity type is similar to the one in [1].

4 Function extensionality

5 Propositional truncation

6 Universe

We define $A_q = \mathsf{comp}_U(p, q, z, A_p, \vec{B})$ by defining it to be the type of elements of the form $(\vec{v}, u)$ where u is an element of A_p and v_{x_ib} is an element of $\vec{B}(zq)$ such that

$$u(x_ib) = \mathsf{comp}_{B_{x_ib}}(q_{ib}, p_{ib}, v_{x_ib})$$

We define then

$$(\vec{v}, u)(x_i b) = v(x_i b)$$

and if y is a symbol

$$(\vec{v}, u)(x_i y) = (\vec{v}(x_i y), u(x_i y))$$

since this should be an element in $\mathsf{comp}_U(p(x_i y), q(x_i y), z, A_p(x_i y), \vec{C})$ Finally if y is a symbol distinct from $x_1, \ldots, x_n$ we define

$$(\vec{v}, u)(yr) = (\vec{v}(yr), u(yr))$$

References

[1] M.Bezem, Th. Coquand and S. Huber. A model of type theory in cubical sets. Preprint, 2013.

[2] M.J. Gabbay and M. Hofmann. Nominal renaming sets. Logic for Programming, Artificial Intelligence, and Reasoning Lecture Notes in Computer Science Volume 5330, 2008, pp 158-173.

[3] A. Grothendieck. Pursuing stacks. Manuscript, 1983.

[4] A. M. Pitts. An Equivalent Presentation of the Bezem-Coquand-Huber Category of Cubical Sets. Manuscript, 17 September 2013.