Cubical stacks

Introduction

The goal of this note is to explore an ∞-stack model of type theory based on cubical sets.

The model will be best described in a purely syntactical way as a method to build internal models of cubical type theory. We start with a (maybe non fibrant) family of cubical sets Cov, F indexed over a type Cov (which corresponds to the collection of basic coverings of the space). We can associate to this a family of (judgemental) monads on fibrant types $D_c(A) = A^{F(c)}$ with a unit map $m_c\ A : A \to D_c(A)$ (the constant map). These maps D_c and m_c satisfy some remarkable properties: D_c is a strict/judgmental monad and $m_c\ A$ a judgmental natural transformation $A \to D_c(A)$. The family Cov, F is called a «covering family» if, and only if, the following property is satisfied: each monad D_c is idempotent, i.e. each map $m_c\ (D_c(A)) : D_c(A) \to D_c^2(A)$ is path equal to $D_c(m_c\ A)$[1].

In this case, we see $D_c(A)$ as an abstract formulation of the type of *descent data* for the covering c, and we define the property of being a stack for A as the type (which is fibrant and a proposition)

$$\mathsf{isStack}\ A \quad = \quad \Pi(c : \mathsf{Cov})\mathsf{isEquiv}\ (m_c\ A)$$

We define then a cubical stack to be a (fibrant) type with a proof that it is a stack.

A simple result which follows from the idempotency assumption is that, for being a stack, it is enough for each map m_c to have a left inverse.

Since there is such a left inverse for the type $\Sigma(X : \mathsf{U})\mathsf{isStack}X$, the type of stacks is itself a stack. We also show that being a stack is preserved by dependent products and sums, and by the path and glueing operations.

It follows that cubical stacks form a model of cubical type theory, and this model can be described as an internal model.

The first part of this note follows this purely syntactical approach. We then explain how to build a concrete example of such family Cov, F which is enough to get a counter-model to countable choice.

1 Abstract descent data

Given a family Cov, F and c in Cov, the type $D_c(A) = A^{F(c)}$ is the type of *descent data* associated to the «covering» c. It is fibrant, with a composition (which does not use any composition for $F(c)$)

$$(\mathsf{comp}^i\ D_c(A)\ [\psi \mapsto u]\ u_0)\ x = \mathsf{comp}^i\ A\ [\psi \mapsto u\ x]\ (u_0\ x)$$

We have a canonical map $m_c : A \to D_c(A)$ defined by $m_c\ A\ a = \lambda(x : F(c))a$. The map D_c defines a *strict/judgemental* monad. We say that A is a c-stack if, and only if, m_c is an equivalence.

We define Cov, F to be a *covering* family if, and only if, each D_c is idempotent, i.e. $D_c(m_c)$ and $m_c : D_c(A) \to D_c^2(A)$ are path equal. In this case, we have the following result, which will be fundamental for showing that the universe of stacks is a stack.

Theorem 1.1 *If $m_c : A \to D_c(A)$ has a left inverse l then A is a c-stack.*

[1] If there is no ambiguity we omit the A in $m_c\ A$. One simple example of this situation is when Cov has only one inhabitant and $F(c)$ is the interval $\mathbb{I}$. The other examples are obtained working in a presheaf extension of the cubical set model.

Proof. It is enough to prove that l is also a right inverse. But we have $l \circ m_c$ path equal to 1 hence $D_c(l \circ m) = D_c(l) \circ D_c(m)$ is path equal to 1. Since D_c is idempotent, we get that $D_c(l) \circ m$ is path equal to the identity, but $D_c(l) \circ m$ is strictly equal to $m \circ l$, hence the result. □

For any c, c' in Cov the type $D_c(D_{c'})(A)$ and $D_{c'}(D_c(A))$ are (strictly) isomorphic. Using this isomorphism, we can prove.

Theorem 1.2 *If A is a c'-stack then so is $D_c(A)$.*

2 Internal description of the stack model

We define then the type $\mathsf{isStack}\ A$ to be $\Pi(c : \mathsf{Cov})\mathsf{isEquiv}\ m_c\ A$. This is a *proposition* which expresses that A is a *stack*.

We define $L_c : (D_c\ \mathsf{U}) \to \mathsf{U}$ such that we have $L_c\ (m_c\ A) = D_c\ A$ as a judgmental equality[2]. An element of $D_c\ \mathsf{U}$ is a family $A : F(c) \to \mathsf{U}$ and $L_c\ A$ is simply $\Pi\ F(c)\ A$, which has a composition

$$(\mathsf{comp}^i\ (L_c\ A)\ [\psi \mapsto u]\ u_0)\ x = \mathsf{comp}^i\ (A\ x)\ [\psi \mapsto u\ x]\ (u_0\ x)$$

2.1 Path type

We have judgemental isomorphisms between $D_c\ (\mathsf{Path}\ A\ a\ b)$ and $\mathsf{Path}\ (D_c\ A)\ (m_c\ a)\ (m_c\ b)$, and the canonical map

$$\mathsf{Path}\ A\ a\ b\ \to D_c\ (\mathsf{Path}\ A\ a\ b)$$

corresponds to m_c via this isomorphism.

Proposition 2.1 *If A is a stack and $a, b : A$ then $\mathsf{Path}\ A\ a\ b$ is a stack.*

2.2 Product type

If $A : \mathsf{U}$ and $B : A \to \mathsf{U}$, we have judgmental isomorphisms between $D_c\ (\Pi\ A\ B)$ and $\Pi\ A\ (D_c \circ B)$ and the canonical map

$$\Pi\ A\ B \to \Pi\ A\ (D_c \circ B) \qquad w \longmapsto \lambda(a : A) m_c\ (w\ a)$$

corresponds to m_c via this isomorphism.

Proposition 2.2 *If we have $\Pi(a : A)\mathsf{isStack}\ (B\ a)$ then $\Pi\ A\ B$ is a stack.*

2.3 Sum type

We have judgmental isomorphisms between $D_c\ (\Sigma\ A\ B)$ and $\Sigma\ (D_c\ A)\ (L_c \circ (D_c\ B))$ and the canonical map

$$\Sigma\ A\ B \to \Sigma\ (D_c\ A)\ (L_c \circ (D_c\ B)) \qquad (a, b) \longmapsto (m_c\ a, m_c\ b)$$

corresponds to m_c via this isomorphism.

Proposition 2.3 *If A is a stack and we have $\Pi(a : A)\mathsf{isStack}\ (B\ a)$ then $\Sigma\ A\ B$ is a stack.*

2.4 Glueing type

Let G be $\mathsf{Glue}\ [\psi \mapsto (T, w)]\ A$. There is a judgmental isomorphism between $D_c\ G$ and $\mathsf{Glue}\ [\psi \mapsto (D_c(T), D_c(w))]\ D_c(A)$ and the canonical map

$$G \quad \longrightarrow \quad \mathsf{Glue}\ [\psi \mapsto (D_c(T), D_c(w))]\ D_c(A)$$

corresponds to m_c via this isomorphism.

Proposition 2.4 *We can build $\vdash \mathsf{glue}\ (\psi \mapsto p)\ q : \mathsf{isStack}\ (\mathsf{Glue}\ [\psi \mapsto (T, w)]\ A)$ given $\psi \vdash p : \mathsf{isStack}\ T$ and $\vdash q : \mathsf{isStack} A$ so that $\mathsf{glue}\ [1 \mapsto p]\ q = p : \mathsf{isStack}\ T$.*

[2]To simplify the notations, we don't write explicitly the index of the universes.

2.5 Universe

So far, we did not use the idempotency hypothesis, but this will be used for the universe.

If A is a stack, we have path equality between $L_c\ (m_c\ A)$ and A by *univalence*. It then follows that L_c is a left inverse of m_c on types that are stacks. We can thus state, by Theorem 1.1.

Theorem 2.5 $\Sigma(X : U)\mathsf{isStack}\ X$ *is a stack.*

2.6 Internal model

We can now define an internal translation which provides a new model of cubical type theory (and hence of univalence). This is a purely syntactical process. We define, where pf denotes the proof that the first component is a stack[3] (for a type A, $[A].\pi_2$ will be a proof that $[\![A]\!]$ is a stack)

$$
\begin{array}{lcl}
[x] & = & x \\
[M\ N] & = & [M]\ [N] \\
[\lambda(x : A)\ M] & = & \lambda(x : [\![A]\!])[M] \\
[M.\pi_1] & = & [M].\pi_1 \\
[M.\pi_2] & = & [M].\pi_2 \\
[M, N] & = & [M], [N] \\
[\langle i\rangle M] & = & \langle i\rangle [M] \\
[M\ r] & = & [M]\ r \\
[\mathsf{glue}\ (\psi \mapsto M)\ N] & = & \mathsf{glue}\ (\psi \mapsto [M])\ [N] \\
 & & \\
[\Pi(x : A)\ B] & = & (\Pi(x : [\![A]\!])[\![B]\!], pf) \\
[\Sigma(x : A)\ B] & = & (\Sigma(x : [\![A]\!])[\![B]\!], pf) \\
[\mathsf{Path}\ A\ M\ N] & = & (\mathsf{Path}\ [\![A]\!]\ [M]\ [N], pf) \\
[\mathsf{Glue}\ (\psi \mapsto (T, w))\ A] & = & (\mathsf{Glue}\ (\psi \mapsto ([\![T]\!], [w]))\ [\![A]\!], pf) \\
[\mathsf{U}] & = & (\Sigma(X : \mathsf{U})\mathsf{isStack}\ X, pf) \\
 & & \\
[\![A]\!] & = & [A].\pi_1
\end{array}
$$

We then have that if $x_1 : A_1, \ldots, x_n : A_n \vdash^I M : A$ then $x_1 : [\![A_1]\!], \ldots, x_n : [\![A_n]\!] \vdash^I [M] : [\![A]\!]$.

3 Examples of covering family

The first example will be to take Cov to be the unit type and $F(0)$ to be $\mathbb{I}$. Using this, we get a proof that the constant map $A \to A^{\mathbb{I}}$ is an equivalence for each *definable* type A, which is defined by induction on A. (This proof does not proceed by choosing a particular end point of the interval.)

For the second class of examples, we fix a topological space, given by a meet semi-lattice of basic open and a notion of covering. We can then consider the model of cubical type theory where a type is interpreted as a presheaf $\Gamma(I|V)$ indexed by a finite set I of names/directions (used to represent cubes) and V is a basic open. The maps are of the form $f : J|W \to I|V$ with $f : J \to I$ and $W \subseteq V$. We define composition structure so that it commutes not only with name substitution («uniformity» condition), but also to open restriction.

To simplify, we assume only coverings of the form $V = V_0, V_1$ with $V_{01} = V_0 \wedge V_1$ non empty. We define $\mathsf{Cov}(I|V)$ to be exactly the set of covering of V. If V_0, V_1 is a covering of V and $W \subseteq V$, we take $W \cap V_0, W \cap V_1$ to be a covering of W. (In the concrete example, we always have that $W \cap V_0$ and $W \cap V_1$ are non empty.)

For $c = V_0, V_1$ in $\mathsf{Cov}(V)$, we associate the following presheaf $F(c)$. It is defined in the following way for $W \subseteq V$

1. if $W \nsubseteq V_0$ and $W \nsubseteq V_1$ we have $F(c)(I|W) = \emptyset$

[3]For the definition to be non ambiguous we need the proof that $[\![\mathsf{Glue}\ (\psi \mapsto (T, w)]\ A]\!]$ is a stack to be such that it strictly equal to $[T].\pi_2$ if $\psi = 1$, cf. Proposition 2.4.

2. if $W \subseteq V_0$ and $W \nsubseteq V_{01}$ we have $F(c)(I|W) = \{0\}$

3. if $W \subseteq V_1$ and $W \nsubseteq V_{01}$ we have $F(c)(I|W) = \{1\}$

4. if $W \subseteq V_{01}$ we define $F(c)(I|W) = \mathbb{I}(I) = \mathsf{dM}(I)$

The fact that this defines a family of idempotent monads follow from the fact that can build a proof of the type $\Pi(x\ y : F(c))\mathsf{Path}\ F(c)\ x\ y$[4].

For this, it is enough to define a de Morgan formula $\theta(k, i, j)$ such that

$$\theta(0, i, j) = i \quad \theta(1, i, j) = j \quad \theta(k, 1, 1) = 1 \quad \theta(k, 0, 0) = 0$$

We can take for instance $\theta(k, i, j) = ((1 - k) \wedge i) \vee (k \wedge j) \vee (i \wedge j)$.

Given this formula, and r, s in $F(c)(I|W)$ we can define a path connecting r and s as $\langle k\rangle\theta(k, r, s)$, which will be uniform w.r.t. both I and W.

We explain in this example why $D_c(A)$ can be seen as a «descent data». Let A be a presheaf. An element of $D_c(A)(I|W)$ for $W \subseteq V$ will be given by a_0, a_1, a_{01} with a_0 in $A(I|W \cap V_0)$ and a_1 in $A(I|W \cap V_1)$ and a_{01} in $A(I, i|W \cap V_{01})$ (with i fresh for I) connecting $a_0|W \cap V_{01}$ to $a_1|W \cap V_{01}$.

4 An example of a space

For being concrete, we present an example which is enough to get independence of countable choice from univalence. This describes $X = [0, 1)$ (unit interval of real numbers r such that $0 \leqslant r < 1$) with the following formal presentation, where the space is given by a decidable meet semilattice of basic open. It is the semilattice generated by X_n, representing $[0, 1/2^n)$, and R_n, representing $(0, 1/2^n)$, and $R_{n+1} = X_{n+1} \wedge R_n$. The basic covering is that X_n is covered by X_{n+1}, R_n.

Stacks over this space form a counter-example to countable choice [5].

A covering of a basic open V is given by an indexed set (in this case) V_0, V_1 of subopen. We have the following situations

1. $V = X_n$ and $(V_0, V_1) = (X_{n+1}, R_n)$ (principal case)

2. $V = X_{n+1}$ and $(V_0, V_1) = (X_{n+1}, R_{n+1})$

3. $V = R_{n+1}$ and $(V_0, V_1) = (R_{n+1}, R_{n+1})$

5 Data types and dependent elimination

In this concrete example, the *constant* presheaf $\mathbb{N}$ has a stack structure, using the fact that all covering are *connected*: if we have compatible local data n_0 on V_0 and n_1 on V_1 we should have $n_0 = n_1$ on V_{01} and, since V_{01} is non trivial, this implies $n_0 = n_1$, which is the unique way to glue n_0 and n_1. Because of this, we don't have any problem in interpreting dependent elimination in the stack model in this case.

This corresponds to the following hypothesis on the family Cov, F: for all c in Cov, the map $N \to D_c(N)$ is an equivalence.

6 Generalization to disjoint coverings

We can also consider *disjoint* covering, e.g. for spaces corresponding to Boolean algebra where the constant presheaf N is not a stack. If $c = V_1, \ldots, V_n$ is a partition of V, we have $F(c)^2$ strictly isomorphic to $F(c)$ and we can define $D_c(A)$ and m_c in such a way that $A = D_c(A)$ on V_i and $m_c\ a = a$ on V_i with strict equality. In order to get dependent elimination with the required judgemental equality, we require to have the inverse l of m_c such that $m_c \circ l : D_c(A) \to D_c(A)$ is *strictly* the identity map (which we can ensure by induction on the type).

[4] Note that, since $F(c)$ is not fibrant, this cannot be expressed as the fact that $F(c)$ is a proposition. Also, this condition on $F(c)$ is *sufficient* but not *necessary* to get a covering family. It will be enough instead to only have a zigzag of paths between any two elements of $F(c)$.

[5] We can indeed define $A_1(n)$ to be 1 on R_0, while $A_0(n)$ if 1 on X_n. We have $A_0(n) \vee A_1(n)$ for all n, but $A_0(n) + A_1(n)$ is only 1 on X_l for $n < l$.