Constructive Stacks?

Munich, 20 December 2019

Goal

Generalize sheaf models of Intuitionistic Logic to Univalent Type Theory

Reminder: main issue

The problem is how to model *universes*

The collection of sheaves don't form a sheaf

If we define $F(V)$ to be the collection of all $\mathcal{U}$-sheaves on V then F is a presheaf which is not a sheaf in general, since glueing will only be defined up to isomorphism

This basic fact was the motivation for the notion of *stacks*

Reminder: what is a sheaf of sets?

Small category $\mathcal{C}$, objects $X, Y, Z, \ldots$ and J Grothendieck topology on $\mathcal{C}$

F presheaf: collection of sets $F(X)$ with restriction maps $u \mapsto uf$

S in $\mathsf{J}(X)$: we can form the set $D_S(F)(X)$

An element of this set is a family $u(f)$ in $F(Y)$ for $f : Y \to X$ in S which is *compatible*: $u(f)g = u(fg)$ if $g : Z \to Y$

We have a map $\eta_F : F(X) \to D_S(F)(X)$ natural in X, S

The presheaf F is a *sheaf* if each map η_F is a bijection

Reminder: what is a sheaf of sets?

If $\mathsf{J}(X)$ contains only the trivial sieve

Then we have a patch function

$D_S(F)(X) \to F(X)$

$u \longmapsto u(1_X)$

What should be a stack?

If now $F(X)$ is a presheaf of spaces/types

The equality $u(fg) \rightarrow u(f)g$ may only be given as a *path* equality $u(f,g)$

We should then ask a cocycle condition at the next level

and then higher equalities $u(f_1, \ldots, f_n)$

The compatible descent data still form a space $D_S(F)$

We require the map $F \rightarrow D_S(F)$ to be an *equivalence*

What should be a stack?

Theorem: *The collection of stacks form a new model of univalent type theory with higher inductive types*

Even for the trivial topology, this provides *new* models of type theory

Equivalence coincides with pointwise equivalence

What should be a stack?

$D_S F(X) \to F(X)$

$u \longmapsto u(1_X)$

This might not be natural anymore

$u(1_X)f$ may not be strictly equal to $u(f)$

What should be a stack?

How to organize these definitions?

Key fact: *we have constructive models of univalence*

Hence these models relativize automatically to *presheaf* models

We define stacks from these models using *left exact modalities*

How to define these left exact modalities?

Content of the talk

Part 1: abstract notion of descent data

frame/point-free space	model of univalent type theory
nucleus	lex modality
prenucleus	abstract notion of descent data
$x \leqslant p(x)$	well-pointed endofunctor D, η
$p(1) = 1 \quad p(x \wedge y) = p(x) \wedge p(y)$	lex endofunctor
$x = p(x)$	η is an equivalence
frame of fixpoints	new model of univalent type theory

cf. Martín Escardó *Joins in the frame of nucleus*, 2003

Content of the talk

Part 2: examples of lex operations

Lex operations

We express in type theory the notion of endomorphism of *tribes*

Functor that preserves

terminal objects, fibrations, base change of fibrations and anodyne maps

We have a map $E : \mathcal{U} \to \mathcal{U}$ which defines a strict functor

Lex operations

A type theoretic function $T \to A$ is a *fibration* if it is strictly isomorphic, as a map over A, to some projection map $\Sigma_A B \to A$.

We express that E preserves fibrations by giving a map $L : E(\mathcal{U}) \to \mathcal{U}$

In this way from $B : A \to \mathcal{U}$ we can define

$\tilde{E}(B) = L \circ E(B) : E(A) \to \mathcal{U}$

and we express that $E(\Sigma_A B) \to E(A)$ is isomorphic to $\Sigma_{E(A)} \tilde{E}(B) \to E(A)$, naturally in A

Lex operations

The map $E(1) \to 1$ should be a strict isomorphism

E also should preserve *equivalences*

This corresponds to the preservation of anodyne maps

If E is a lex operation we have a natural transformation $\eta_A : A \to E(A)$

This natural transformation is furthermore uniquely determined

Lex operations

We require $L \circ \eta_{\mathcal{U}} = E$

This implies that the (strict) gap map of the commuting diagram

$$\begin{tikzcd} T \arrow[r, "\eta_T"] \arrow[d, "\pi_B"'] & E(T) \arrow[d, "E(\pi_B)"] \\ A \arrow[r, "\eta_A"'] & E(A) \end{tikzcd}$$

where $T = \Sigma_A B$, is the map $\eta_{Ba} : Ba \to E(Ba)$ over A

E-modal types

We say that a type A is E-*modal* if the map $\eta_A : A \to E(A)$ is an equivalence

Family of E-modal types

Theorem: *If B is a family of types over A then this is a family of E-modal types iff the strict commuting diagram*

$$\begin{tikzcd} T \arrow[r, "\eta_T"] \arrow[d, "\pi_B"'] & E(T) \arrow[d, "E(\pi_B)"] \\ A \arrow[r, "\eta_A"'] & E(A) \end{tikzcd}$$

where $T = \Sigma_A B$, is a homotopy pullback diagram.

Family of E-modal types

Corollary: *Families of E-modal types are closed by composition*

Example

If R is a type then $E(A) = A^R$

We can define $L : E(\mathcal{U}) \to \mathcal{U}$ by $L(B) = \Pi_R B$

E preserves fibrations and equivalences

The map $\eta_A : A \to A^R$ is defined by $\eta_A\ a\ x = a$

Example

Consider a (cubical) presheaf model over a small category $\mathcal{C}$

We define $E(A)(X)$ to be the set of families $u(f)$ in $A(Y)$ for $f : Y \to X$

$E(A)(X) = \prod_{f:Y \to X} A(Y)$

E preserves fibrations and equivalences

The map $\eta_A : A \to E(A)$ is defined by $(\eta_A\ a)(f) = a f$

Abstract descent data

Definition: *An* abstract notion of descent data *is a lex operation* D, η *such that there is a path between* $\eta_{D(A)}$ *and* $D(\eta_A)$

Furthermore this path should be natural in A along fibrations

Well-pointed endofunctor up to homotopy

A is a *stack* for D if A is D-modal i.e. $\eta_A : A \to D(A)$ is an equivalence

Example

In general $D(A) = A^R$ may not be a notion of descent data

But this is the case if R is a *proposition*

Abstract descent data

This notion of abstract descent data can be seen as a higher version of the notion of *prenucleus* on a frame, i.e. a map such that $x \leqslant p(x)$ and $p(1) = 1$ and $p(x \wedge y) = p(x) \wedge p(y)$

The fixpoints of p form a frame

There is a least nucleus j such that $p \leqslant j$ and p and j have the same fixpoints

We are going to see a higher version of these results

First we show that the D-modal types form a model of type theory

Family of stacks

Proposition 1: *Family of stacks are preserved by* D

Family of stacks

$T \to A$ family of stacks

$$\begin{tikzcd}
D(T) \arrow[r, "\eta_{D(T)}"] \arrow[d, "D(\pi_B)"'] & D^2(T) \arrow[d, "D^2(\pi_B)"] \\
D(A) \arrow[r, "\eta_{D(A)}"'] & D^2(A)
\end{tikzcd}$$

should be homotopy pull-back

Family of stacks

We know that this is the case for

$$\begin{tikzcd} D(T) \arrow[r, "D(\eta_T)"] \arrow[d, "D(\pi_B)"'] & D^2(T) \arrow[d, "D^2(\pi_B)"] \\ D(A) \arrow[r, "D(\eta_A)"'] & D^2(A) \end{tikzcd}$$

since D is lex and B is a family of stacks

Family of stacks

Proposition 2: A *is a stack iff* η_A *has a left homotopy inverse*

We call such a left inverse a *patch* function

Theorem: *The type* $\mathcal{U}_S = \Sigma(X : \mathcal{U})\mathsf{isStack}(X)$ *is a stack*

We have a family of stacks $\pi_1 : \mathcal{U}_S \to \mathcal{U}$

Hence by Proposition 1, $D(\pi_1)$ is a family of $\mathcal{U}$-stacks over $D(\mathcal{U}_S)$

In this way we build a patch function $D(\mathcal{U}_S) \to \mathcal{U}_S$, using $L \circ \eta_U = D$

Application: left exact modality

$D(A)$ may not be a stack in general

We define M as a HIT

$$
\begin{array}{lll}
\mathsf{inc} & : & A \to M(A) \\
\mathsf{patch} & : & D(M(A)) \to M(A) \\
\mathsf{linv} & : & \Pi(x : M(A))\ \mathsf{patch}(\eta_{M(A)} x) =_{M(A)} x
\end{array}
$$

Theorem: *The pair* M, isStack *defines a left exact modality*

This corresponds to the nucleus associated to a prenucleus obtained by (maybe) transfinite iteration

Application: left exact modality

Note that A is D-modal iff A is M-modal

Corresponds to the fact that, if j is the nucleus generated by a prenucleus p then $p(x) = x$ iff $j(x) = x$

Application: left exact modality

We then get a model of univalent type theory

A *type* now a pair A, p where p is a proof that A is a stack

We can even interpret HIT, e.g. N is interpreted by

$$\begin{array}{lcl}
\mathsf{zero} & : & N \\
\mathsf{succ} & : & N \to N \\
\mathsf{patch} & : & D(N) \to N \\
\mathsf{linv} & : & \Pi(x : N)\ \mathsf{patch}(\eta_N x) =_N x
\end{array}$$

How to define a notion of descent data

Consider a (cubical) presheaf model over a small category $\mathcal{C}$

We have defined $E(A)(X) = \prod_{f:Y\to X} A(Y)$

This defines a lex operation with a natural transformation $\eta : A \to E(A)$

$(\eta\ a)(f) = af$ in $A(Y)$ for $f : Y \to X$ and a in $A(X)$

In general, this might not define a *well-pointed* notion of descent data

How to define a notion of descent data

We define $D(A)$ from $E(A)$

An element u of $D(A)$ is now a family $u(i_1,\dots,i_n)$ in $E^{n+1}(A)$ which satisfies the *compatibility conditions*

We have $v = u()$ in $E(A)$ and then a path between $\eta\ v$ and $E(\eta)\ v$

$u(0) = \eta\ v$, $u(1) = E(\eta)\ v$

How to define a notion of descent data

Then we express the cocycle conditions between these paths

$i = 0 \to u(i,j) = \eta u(j)$,
$i = j \to u(i,j) = E(\eta)u(i)$,
$j = 1 \to u(i,j) = E^2(\eta)u(i)$

and so on

This defines a new space $D(A)$

We get in this way an abstract notion of descent data

How to define a notion of descent data

If we start instead from $E(A) = A^R$

What is an element an element of $D(A)$?

it should be a map $v : A \to R$ which is constant $v(r_1) = v(r_2)$ and with the cocycle conditions between these paths and so on

it is a *coherently constant* map, as defined in the PhD thesis of Nicolai Kraus,

hence to give such a map is to give an element of $\|R\| \to A$

How to define a notion of descent data

An element of $D^2(A)$ is a double sequence $v(\vec{i})(\vec{j})$

The proof that D is well-pointed is by defining

$v_k(\vec{i})(\vec{j}) = u(k \wedge \vec{i}, k, k \vee \vec{j})$

a path between $D(\eta_A)(u)(\vec{i})(\vec{j})$ and $\eta_{D(A)}(u)(\vec{i})(\vec{j})$

Presheaf models

We get in this way a model of univalent type theory on presheaves that are D-modals

Presheaf models

For the "direct" presheaf model, it might be that each $F(X)$ is contractible as a space but that F has no global point

An example: presheaves over $0 \leqslant 1 \leqslant 2 \leqslant \dots$

Let $F(n)$ be the trivial groupoid on the set $n, n+1, n+2, \dots$

The inclusion $F(n+1) \to F(n)$ is the restriction map

Then each $F(n)$ is contractible but F has no global point

Presheaf models

Another example over $G = \mathbb{Z}/2\mathbb{Z}$

Take the trivial groupoid $A = a \leftrightarrow b$ with a, b swapped by G

Then the map $A \to 1$ is an equivalence as a groupoid map

But A has no global points, so this is not a G-equivalence

Presheaf models

Let A be a family of types over Γ in the presheaf model

Proposition: *If each* $A(X)$ *is a family of contractible types over* $\Gamma(X)$ *then* $D(A)$ *has a section over* Γ

Corollary: *If each* $A(X)$ *is a family of modal contractible types over* $\Gamma(X)$ *then* A *is contractible*

Corollary: *If* A *and* B *are* D*-modals and* $\sigma : A \to B$ *is a pointwise equivalence, then it is an equivalence*

Application

Cf. the work of Matthew Weaver and Dan Licata

A Model of Type Theory with Directed Univalence in Bicubical Sets

Presheaf model

The obstacle there was precisely that a pointwise equivalence might not be in general a global equivalence

Hope: these new models solve this issue

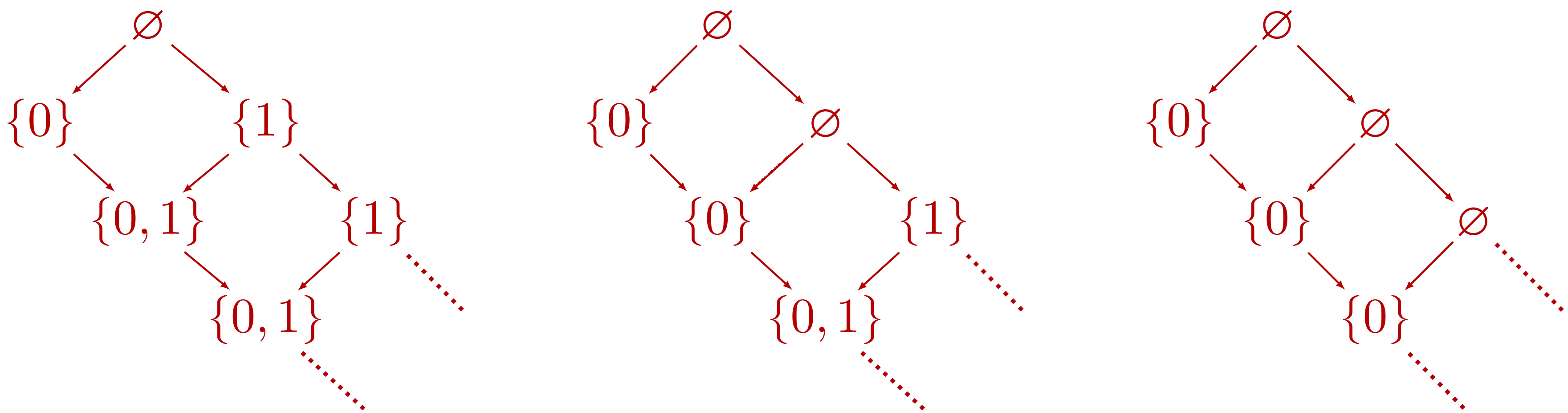
Application

Model of parametrised pointed type

This is the model over the walking retract

Example: Countable choice

We then can define a family of sets (stacks) $A\ n$, e.g. for $A\ 0,\ A\ 1$ and $A\ 2$



Example: Countable choice

$\Pi(n : N) A\ n$ is (a proposition) is *not* globally inhabited and $\|A\ n\|$ *is* globally inhabited *because* of the stack condition

$$\begin{tikzcd}
 & \emptyset \arrow[dl] \arrow[dr] & & & \\
\{\overline{0}\} \arrow[dr] & & \emptyset \arrow[dl] \arrow[dr] & & \\
 & \{\overline{0}\} \arrow[dr] & & \emptyset \arrow[dl] \arrow[dr, dotted, dash] & \\
 & & \{\overline{0}\} \arrow[dr, dotted, dash] & & {} \\
 & & & {} &
\end{tikzcd}$$