Formalizing type theory in type theory using nominal techniques

Ulrik Buchholtz

TU Darmstadt

HoTT/UF Workshop, Oxford, September 9, 2017

Outline

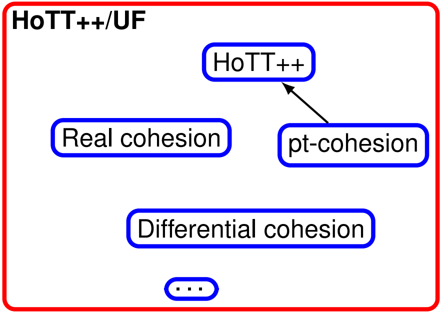
Motivation

1. Age-old problem: what's the best way to reason (& program) with syntax with binders? α-renaming? HOAS? wHOAS? de Bruijn indices? nominal sets?
2. We're not going to settle this today! However, in this talk I'll explore a new approach afforded us by HoTT.
3. This is based on the classifying type $\mathrm{B}\Sigma_\infty$ of the finitary symmetric group Σ_∞.
4. HoTT lets us escape *setoid hell*. Will it also let us escape *weakening hell*?
5. Application: will nominal techniques be useful for letting *HoTT eat itself* (cf. March 2014 blog post by Mike Shulman).

Further applications

Further applications: *Small proofs* (cf. Licata @ Big Proof): $\mathbb{S}$-cohesion, equivariant cohesion, maybe one day real/smooth/differential cohesion, etc.

Further applications: *Small proofs* (cf. Licata @ Big Proof): $\mathbb{S}$-cohesion, equivariant cohesion, maybe one day real/smooth/differential cohesion, etc.



Outline

Synthetic homotopy theory

- In the HoTT book: Whitehead's theorem, $\pi_1(S^1)$, Hopf fibration, etc.
- (Generalized) Blakers-Massey theorem
- Quaternionic Hopf fibration
- Gysin sequence, Whitehead products and $\pi_4(S^3)$ (Brunerie)
- Homology theory (Snowbird group, Graham)
- Serre spectral sequence for any truncated cohomology theory (van Doorn *et al.* following outline by Shulman):

$$H^{-(n-s)}(x : X; H^{-s}(F\,x; Y)) \Rightarrow H^{-n}((x : X) \times F\,x; Y)$$

The homotopy hypothesis

$$\frac{\text{HoTT: types are homotopy types} \quad \text{Grothendieck: homotopy types are } \infty\text{-groupoids}}{\text{Thus: types are } \infty\text{-groupoids}}$$

Elements are objects, paths are morphisms, higher paths are higher morphisms, etc.

It follows that *pointed connected* types A may be viewed as higher groups, with *carrier* $\Omega A = (\mathrm{pt} = \mathrm{pt})$.

Writing G for the carrier, it's common to write BG for the pointed connected type such that $G = \Omega BG$ (BG is the *delooping* of G).

Ordinary groups are thus represented by the pointed, connected, 1-types BG.

It follows that *pointed connected* types A may be viewed as higher groups, with *carrier* $\Omega A = (\mathrm{pt} = \mathrm{pt})$.

Writing G for the carrier, it's common to write BG for the pointed connected type such that $G = \Omega BG$ (BG is the *delooping* of G).

Ordinary groups are thus represented by the pointed, connected, 1-types BG.

Yesterday, Thorsten asked how to define $\mathbb{Z}$? My answer:
$\mathrm{B}\mathbb{Z} :\equiv \mathbb{S}^1$, the usual HIT giving the free ∞-group on one generator (turns out to be a 1-group).

It follows that *pointed connected* types A may be viewed as higher groups, with *carrier* $\Omega A = (\mathrm{pt} = \mathrm{pt})$.

Writing G for the carrier, it's common to write BG for the pointed connected type such that $G = \Omega BG$ (BG is the *delooping* of G).

Ordinary groups are thus represented by the pointed, connected, 1-types BG.

Yesterday, Thorsten asked how to define $\mathbb{Z}$? My answer:
$\mathrm{B}\mathbb{Z} := \mathbb{S}^1$, the usual HIT giving the free ∞-group on one generator (turns out to be a 1-group).

$$0 := \mathrm{idp}, \quad i + j := i.j \quad \text{(path composition)}$$

Table of higher groups

A $(n\text{-})$type G is *k-tuply groupal* if we have a k-fold delooping, $B^kG : \mathrm{Type}^{\geq k}_{\mathrm{pt}}$, such that $G = \Omega^k B^k G$.
(We can also take $k = \omega$ by recording the entire sequence of deloopings.)

$k \setminus n$	0	1	$\cdots$	∞
0	pointed set	pointed groupoid	$\cdots$	pointed ∞-groupoid
1	group	2-group	$\cdots$	∞-group
2	abelian group	braided 2-group	$\cdots$	braided ∞-group
3	— " —	symmetric 2-group	$\cdots$	sylleptic ∞-group
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
ω	— " —	— " —	$\cdots$	connective spectrum

(Cf. forthcoming paper with van Doorn-Rijke)

Today we stick to ordinary groups.

We have equivalences of univalent categories

$$\mathrm{Grp} \simeq \mathrm{Type}_{\mathrm{pt}}^{=1}$$

and

$$\mathrm{AbGrp} \simeq \mathrm{Type}_{\mathrm{pt}}^{=n} \simeq \mathrm{Sp}^{=0}$$

for $n \geq 2$ (formalized in Lean).

An *action* of G on some object of type U is simply a function $X : BG \to U$. The object of the action is $X(\mathrm{pt}) : U$, and it can be convenient to consider evaluation at $\mathrm{pt} : BG$ to be a coercion from actions of type U to U.

If U is a universe of types, then we have actions on types. If X is an action on types, then we can form the:

invariants $X^{hG} :\equiv (x : BG) \to X(x)$, also known as the *homotopy fixed points*

coinvariants $X_{hG} :\equiv (x : BG) \times X(x)$, also known as the *homotopy quotient* $X \mathbin{/\!/} G$.

Automorphism groups

The automorphism group of $u : U$ is simply $(u = u)$ with delooping $\mathrm{BAut}\, u = (v : U) \times \|u = v\|_{-1}$. (This is a 1-group if U is a 1-type.)

An action of G on u is equivalently a homomorphism from G to $\mathrm{Aut}\, u$.

The finite symmetric groups Σ_n are represented by $\mathrm{BAut}[n]$, where $[n]$ is the canonical set with n elements. (Recall the Set is a 1-type.)

More about finite sets

Let $\mathrm{FinSet} := (A : \mathrm{Type}) \times \|\exists n : \mathbb{N}, A = [n]\|_{-1}$.

Then we get an equivalence

$$\mathrm{FinSet} \simeq (n : \mathbb{N}) \times \mathrm{BAut}[n]$$

using the *pigeonhole principle* which implies that $[n] \simeq [m] \to n = m$.

In particular we have the cardinality function $\mathrm{card} : \mathrm{FinSet} \to \mathbb{N}$.

More about finite sets

Let $\mathrm{FinSet} :\equiv (A : \mathrm{Type}) \times \|\exists n : \mathbb{N}, A = [n]\|_{-1}$.

Then we get an equivalence

$$\mathrm{FinSet} \simeq (n : \mathbb{N}) \times \mathrm{BAut}[n]$$

using the *pigeonhole principle* which implies that $[n] \simeq [m] \to n = m$.

In particular we have the cardinality function $\mathrm{card} : \mathrm{FinSet} \to \mathbb{N}$.

NB these are Bishop sets, not Kuratowski sets; see also Yorgey's thesis for an application to the theory of species. See also Shulman's formalization in the HoTT library in Coq.

Outline

Recall the many equivalent ways to present the Schanuel topos:

1. The category of finitely supported nominal sets (Σ_∞-sets).
2. The category of continuous Σ_∞-sets.
3. The category of continuous $\mathrm{Aut}\,\mathbb{N}$-sets.
4. The category of sheaves on $\mathrm{FinSet}_{\mathrm{mon}}^{\mathrm{op}}$ wrt the atomic topology.
5. The category of pullback-preserving functors $\mathrm{FinSet}_{\mathrm{mon}} \to \mathrm{Set}$.

Recall the many equivalent ways to present the Schanuel topos:

1. The category of finitely supported nominal sets (Σ_∞-sets).
2. The category of continuous Σ_∞-sets.
3. The category of continuous $\mathrm{Aut}\,\mathbb{N}$-sets.
4. The category of sheaves on $\mathrm{FinSet}_{\mathrm{mon}}^{\mathrm{op}}$ wrt the atomic topology.
5. The category of pullback-preserving functors $\mathrm{FinSet}_{\mathrm{mon}} \to \mathrm{Set}$.

Focus on first two: in HoTT, we can present a variant as a *slice topos* over $\mathrm{B}\Sigma_\infty$.

From well-scoped de Bruijn and beyond

When representing syntax with binding we have many options:

- Use *names* and quotient by α-equality
- Use *de Bruijn indices*
- Use *well-scoped* de Bruijn indices: index by $\mathbb{N}$ (number of free variables)
- (HoTT) Use *symmetric* well-scoped de Bruijn indices: index by FinSet
- (HoTT) Use *nominal* technique: index by $\mathrm{B}\Sigma_\infty$.

$$\mathbb{N} \xrightarrow{[-]} \mathrm{FinSet} \xrightarrow{i} \mathrm{B}\Sigma_\infty \xrightarrow{j} \mathrm{BAut}\,\mathbb{N} \qquad \mathrm{card} : \mathrm{FinSet} \to \mathbb{N}$$

$B\Sigma_\infty$ is both the homotopy colimit of

$$\mathrm{BAut}[0] \to \mathrm{BAut}[1] \to \cdots$$

and the homotopy coequalizer of

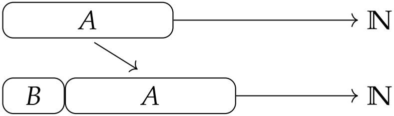
$$\mathrm{id}, (-) + \top : \mathrm{FinSet} \to \mathrm{FinSet}$$

using the equivalence mentioned above.

Constructors:

- $i : \mathrm{FinSet} \to B\Sigma_\infty$ or $i : (n : \mathbb{N}) \to \mathrm{BAut}[n] \to B\Sigma_\infty$,
- $g : (A : \mathrm{FinSet}) \to i(A) = i(A + \top)$.

The shift map is a special case of shifting by an arbitrary finite set B, $iA \mapsto i(B+A)$, illustrated as follows:



Thus we get a map $\mathrm{FinSet} \times \mathrm{B}\Sigma_\infty \to \mathrm{B}\Sigma_\infty$, which we write suggestively as mapping A and X to $A + X$.

Reindexing

If $f : I \to J$ is any function, we get operations

$$\mathrm{Type}^I \underset{f_*}{\overset{f_!}{\rightrightarrows}} \mathrm{Type}^J \quad \mathrm{Type}^I \xleftarrow{\;f^*\;} \mathrm{Type}^J$$

where $f^* Z(i) = Z(f\, i)$,

$$f_! Y(j) = (i : I) \times (f\, i = j) \times Y(i), \quad \text{and}$$
$$f_* Y(j) = (i : I) \to (f\, i = j) \to Y(i).$$

Applying these to the functions between $\top$, $\mathbb{N}$, FinSet and $\mathrm{B}\Sigma_\infty$, we get adjunctions connecting the various kinds of nominal types.
Applying these to the shift maps $B + - : \mathrm{B}\Sigma_\infty \to \mathrm{B}\Sigma_\infty$, we get that the name abstraction operations have both adjoints.

We define $\mathbb{A} : \mathrm{B}\Sigma_\infty \to \mathrm{Type}$ by recursion

$$\begin{aligned} \mathbb{A}\, iA &:= A + \mathbb{N} \\ \mathrm{ap}\, \mathbb{A}\, gA &:= (A + \mathbb{N} \simeq A + (1 + \mathbb{N}) \simeq (A + 1) + \mathbb{N}) \end{aligned}$$

Proposition

For all $X : \mathrm{B}\Sigma_\infty$, $[1]\mathbb{A}\, X \simeq (1 + \mathbb{A})\, X$. *Hence,* $[1]\mathbb{A} \doteq 1 + \mathbb{A}$.

We need to see the generators of Σ_∞ equivariantly.
Define $(- -) : \mathbb{A}\,X \to \mathbb{A}\,X \to X = X$ by induction on X.

(Not yet formalized.)

Then we get $(a\ b)^2 = 1$, $((a\ b)(a\ c))^3 = 1$, $(a\ b)(c\ d) = (c\ d)(a\ b)$ (using fresh name convention).

nominal set $Z : \mathrm{B}\Sigma_\infty \to \mathrm{Set}$

nominal type $Z : \mathrm{B}\Sigma_\infty \to \mathrm{Type}$

element $x \in Z$ means $x : Z(\mathrm{pt})$

action by finite permutation for $\pi \in \mathrm{Aut}[n]$ and $x \in X$ we get $\pi \cdot x$ by transporting to $[n]$, applying π, and transporting back.

equivariant action by transpositions for $a, b : \mathbb{A}\, X$, transport along $(a\ b) : X = X$.

terms with support $Z@A = (X : \mathrm{B}\Sigma_\infty) \to Z(A + X)$

Following Allais-Atkey-Chapman-McBride-McKinna, we introduce a universe of descriptions of scope-safe syntaxes, $\mathrm{Desc} : \mathrm{Set}$:

$$\frac{A : \mathrm{Type}_0 \quad A \text{ has dec.eq.} \quad d : A \to \mathrm{Desc}}{\sigma\, A\, d : \mathrm{Desc}} \qquad \frac{m : \mathbb{N} \quad d : \mathrm{Desc}}{\mathrm{X}\, m\, d : \mathrm{Desc}} \qquad \frac{}{\blacksquare : \mathrm{Desc}}$$

with semantics $[\![-]\!] : \mathrm{Type}^I \to \mathrm{Type}^I$ for any I with $S : I \to I$:

$$\begin{aligned} [\![\sigma\, A\, d]\!]\, Z\, i &:= (a : A) \times [\![d\, a]\!]\, Z\, i \\ [\![\mathrm{X}\, m\, d]\!]\, Z\, i &:= Z\,(S^m\, i) \times [\![d]\!]\, Z\, i \\ [\![\blacksquare]\!]\, Z\, i &:= \top \end{aligned}$$

Terms and semantics in cubical sets model

The terms are then the inductive type family $\mathrm{Tm}\, d : \mathrm{B}\Sigma_\infty \to \mathrm{Type}$:

$$\frac{a : \mathbb{A}\, X}{\mathrm{var}\ a : \mathrm{Tm}\, d\, X} \qquad \frac{z : [\![d]\!]\ (\mathrm{Tm}\, d)\ X}{\mathrm{con}\ z : \mathrm{Tm}\, d\, X}$$

(We can use any I with an atom family $A : I \to \mathrm{Type}$.)

Inductive families of this kind (Dybjer calls them *restricted*) have straight-forward semantics in the cubical models with composition-operators working index-wise.

Nominal kit for generic syntax

We can of course reason about $\mathrm{Tm}\, d : \mathrm{B}\Sigma_\infty \to \mathrm{Type}$ using the (de Bruijn) techniques of Allais et al.

However, we can also work *nominally* using equivalences

$$Z\,(S^m\,X) \simeq (\mathrm{Vec}(\mathbb{A}\,X)\,m \times Z\,X)_{/\sim}.$$

These should obtain whenever Z is a nominal set with finite support.

For generic syntax we can obtain the binding equivalences by proving by induction on d that $[\![-]\!]$ preserves the structure of having finite sets of support *and* binding equivalences.
In the same way can prove that $\mathrm{Tm}\,d\,X$ has decidable equality.

(In the formalization I use sized types to convince Agda these inductions are structural.)

Outline

The un(i)typed λ-calculus can be represented by the description

$$d_\lambda = \sigma\,[2]\,(\lambda b, \text{if } b \text{ then } \mathrm{X}\,1\,\blacksquare \text{ else } \mathrm{X}\,0\,(\mathrm{X}\,0\,\blacksquare)$$

Warmup: untyped lambda calculus

The un(i)typed λ-calculus can be represented by the description

$$d_\lambda = \sigma\,[2]\,(\lambda b, \text{if } b \text{ then } \mathrm{X}\,1\,\blacksquare \text{ else } \mathrm{X}\,0\,(\mathrm{X}\,0\,\blacksquare))$$

A more perspicuous and scalable way to say the same thing:

$$C_\lambda : \mathrm{FinSet}$$
$$C_\lambda := \{\mathrm{lam}, \mathrm{app}\}$$

$$c_\lambda : C_\lambda \to \mathrm{Desc}$$
$$c_\lambda\,\mathrm{lam} := \mathrm{X}\,1\,\blacksquare$$
$$c_\lambda\,\mathrm{app} := \mathrm{X}\,0\,(\mathrm{X}\,0\,\blacksquare)$$

$$d_\lambda := \sigma\,C_\lambda\,c_\lambda$$

Using the binding equivalence

$$\mathrm{Tm}\, d_\lambda\, (S\, X) \simeq (\mathbb{A}\, X \times \mathrm{Tm}\, d_\lambda\, X)_{/\sim}$$

we get a more convenient lam constructor:

$$\mathrm{lam} : \mathbb{A}\, X \to \mathrm{Tm}\, d_\lambda\, X \to \mathrm{Tm}\, d_\lambda\, X.$$

A first test would be the $\lambda\Pi$-calculus:

$$C_{\lambda\Pi} : \mathrm{FinSet}$$
$$C_{\lambda\Pi} := \{\mathrm{lam}, \mathrm{app}, \mathrm{pi}\}$$

$$c_{\lambda\Pi} : C_{\lambda\Pi} \to \mathrm{Desc}$$
$$c_{\lambda\Pi}\,\mathrm{lam} := \mathrm{X}\,1\,\blacksquare$$
$$c_{\lambda\Pi}\,\mathrm{app} := \mathrm{X}\,0\,(\mathrm{X}\,0\,\blacksquare)$$
$$c_{\lambda\Pi}\,\mathrm{pi} := \mathrm{X}\,0\,(\mathrm{X}\,1\,\blacksquare)$$

$$d_{\lambda\Pi} := \sigma\,C_{\lambda\Pi}\,c_{\lambda\Pi}$$

Next steps

- To formalize the standard semantics of the $\lambda\Pi$-calculus (and other dependent type theories), we need to prove that the semantics is well-behaved wrt to substitution.
- Probably(?) the best way is to perform a translation into well-typed syntax with explicit substitutions first (but not set-truncated).
- Longer term goal: The groupoid model of type theory with a universe of sets in $\mathrm{Type}^{\leq 1}$.

Outline

Open problems

- Is there a classically equivalent definition of Σ_∞ that carries the "natural" topology?
- Are there applications of higher-dimensional nominal types?
- What is anyway the "correct" $(\infty, 1)$-analogue of the Schanuel topos? (Should a transposition cost a sign somehow?)
- In directed type theory, there's a nice way to do HOAS-style syntax.
- Let's make HoTT eat itself!