Groups and higher groups in homotopy type theory

Ulrik Buchholtz

TU Darmstadt

Arbeitstagung Bern-München, December 14, 2017

Outline

Recall the Hofmann-Streicher groupoid model (1995):

- Types A are groupoids.
- Terms $x : A$ are objects.
- Identity types $x =_A y$ are hom-sets (as discrete groupoids).
- Dependent types ($x : A \vdash B : \mathrm{Type}$) are fibrations of groupoids (think "$B : A \to \mathrm{Gpd}$").
- The universe consists of discrete groupoids, aka sets. It is *univalent*.

Recall the Hofmann-Streicher groupoid model (1995):

- Types A are groupoids.
- Terms $x : A$ are objects.
- Identity types $x =_A y$ are hom-sets (as discrete groupoids).
- Dependent types ($x : A \vdash B : \mathrm{Type}$) are fibrations of groupoids (think "$B : A \to \mathrm{Gpd}$").
- The universe consists of discrete groupoids, aka sets. It is *univalent*.
- The propositional truncation of a type, $\|A\|_{-1}$, is the groupoid with the same objects as A, but with a unique isomorphism between any pair of objects.

Groups in the groupoid model

Groups are pointed, connected groupoids. We can formalize *connected* as $\mathrm{isConn}(A) :\equiv \|A\|_{-1} \times [(x, y : A) \to \|x = y\|_{-1}]$.

Hence it's possible to use the groupoid model to do synthetic group theory.

If A is a pointed connected type, the *carrier set* (when viewed as a group) is the loop type $\Omega A := (\text{pt} = \text{pt})$.

The identity is the reflexivity path idp, and the group operation is path concatenation.

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).

We can use *higher inductive types* to define some common groups. For example, the free group on one generator, aka the integers $\mathbb{Z}$, is represented by the higher inductive type generated by one point pt : $B\mathbb{Z}$ and one path g : pt $=$ pt.

Another way to get groups

If $a : A$ is any object of any type A, then the connected component of A containing a is a pointed connected type.
$\mathrm{BAut}(a) :\equiv (x : A) \times \|a = x\|_{-1}$.

The point $\mathrm{pt} : \mathrm{BAut}(a)$ is of course $\mathrm{pt} :\equiv \langle a, - \rangle$.

And the carrier is $\mathrm{Aut}(a) :\equiv (a = a)$.

Some group theory

$BG \to_* BH$	homomorphism $G \to H$
$BG \to BH$	conjugacy class of homomorphisms
$B\mathbb{Z} \to_* BH$	element of H
$B\mathbb{Z} \to BH$	conjugacy class in H
$BG \to A$	A-action of G
$BG \to_* \mathrm{BAut}(a)$	action of G on $a : A$
$X : BG \to \mathrm{Type}$	type with an action of G
$(x : BG) \times X(x)$	quotient
$(x : BG) \to X(x)$	fixed points

A comparison theorem?

Suppose we would like to prove an equivalence

$$\mathrm{Grp} \simeq \mathrm{Type}_{\mathsf{pt}}^{>0}$$

between the type of groups and the type of pointed connected types. We will fail, because nothing in type theory ensures that the identities types are sets!

A comparison theorem?

Suppose we would like to prove an equivalence

$$\mathrm{Grp} \simeq \mathrm{Type}_{\mathsf{pt}}^{>0}$$

between the type of groups and the type of pointed connected types. We will fail, because nothing in type theory ensures that the identities types are sets!

In fact, the pointed connected types correspond to general ∞-groups.

Recall Voevodsky's definition of the homotopy levels:

Level	Name	Definition
-2	contractible	$\mathrm{isContr}(A) := (x : A) \times \big((y : A) \to (x = y)\big)$
1	proposition	$\mathrm{isProp}(A) := (x, y : A) \to \mathrm{isContr}(x = y)$
2	set	$\mathrm{isSet}(A) := (x, y : A) \to \mathrm{isSet}(x = y)$
3	groupoid	$\vdots$
$\vdots$	$\vdots$	$\vdots$
n	n-type	$\vdots$
$\vdots$	$\vdots$	$\vdots$

Recall Voevodsky's definition of the homotopy levels:

Level	Name	Definition
-2	contractible	$\mathrm{isContr}(A) :\equiv (x : A) \times \big((y : A) \to (x = y)\big)$
1	proposition	$\mathrm{isProp}(A) :\equiv (x, y : A) \to \mathrm{isContr}(x = y)$
2	set	$\mathrm{isSet}(A) :\equiv (x, y : A) \to \mathrm{isSet}(x = y)$
3	groupoid	$\vdots$
$\vdots$	$\vdots$	$\vdots$
n	n-type	$\vdots$
$\vdots$	$\vdots$	$\vdots$

(Modulo starting at -2 instead of 0.)

The right comparison theorem

Now we get an equivalence

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

between discrete groups (i.e., whose carrier is a set) and pointed connected 1-types.

And this equivalence lifts to an equivalence of (univalent) 1-categories.

Outline

What's so great about abelian groups?

If sets that are loop types are good, double loop types must be twice as good!

What's so great about abelian groups?

If sets that are loop types are good, double loop types must be twice as good!

And they are! Because of the Eckmann-Hilton argument, they represent abelian groups.

What’s so great about abelian groups?

If sets that are loop types are good, double loop types must be twice as good!

And they are! Because of the Eckmann-Hilton argument, they represent abelian groups.

But triple loop types give nothing new.

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

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

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 and cohomology theories
- Serre spectral sequence for any cohomology theory (van Doorn *et al.* following outline by Shulman)

Let us introduce the type

$$
\begin{aligned}
(n,k)\,\mathrm{Grp} &:= (G : \mathrm{Type}^{\le n}) \times (B^k G : \mathrm{Type}_{\mathsf{pt}}^{\ge k}) \times (G = \Omega^k B^k G) \\
&= \mathrm{Type}_{\mathsf{pt}}^{\ge k, \le n+k}
\end{aligned}
$$

for the type of k*-tuply groupal* n*-groupoids*.

We can also allow k to be infinite, $k = \omega$, but in this case we can't cancel out the G and we must record all the intermediate delooping steps:

$$
\begin{aligned}
(n,\omega)\,\mathrm{Grp} := \big(B^{-}G : (k : \mathbb{N}) \to \mathrm{Type}_{\mathsf{pt}}^{\ge k, \le n+k}\big) \\
\times \big((k : \mathbb{N}) \to B^k G = \Omega B^{k+1} G\big)
\end{aligned}
$$

$k \setminus n$	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

$k \setminus n$	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

decategorication	$(n,k)\,\mathrm{Grp} \to (n-1,k)\,\mathrm{Grp}$,
discrete categorification	$(n,k)\,\mathrm{Grp} \to (n+1,k)\,\mathrm{Grp}$,
looping	$(n,k)\,\mathrm{Grp} \to (n-1,k+1)\,\mathrm{Grp}$
delooping	$(n,k)\,\mathrm{Grp} \to (n+1,k-1)\,\mathrm{Grp}$
forgetting	$(n,k)\,\mathrm{Grp} \to (n,k-1)\,\mathrm{Grp}$
stabilization	$(n,k)\,\mathrm{Grp} \to (n,k+1)\,\mathrm{Grp}$

Theorem (Freudenthal)

If $A : \mathrm{Type}_{\mathsf{pt}}^{>n}$ *with* $n \geq 0$*, then the map* $A \to \Omega\Sigma A$ *is* $2n$*-connected.*

Corollary (Stabilization)

If $k \geq n+2$*, then* $S : (n,k)\,\mathrm{Grp} \to (n,k+1)\,\mathrm{Grp}$ *is an equivalence, and any* $G : (n,k)\,\mathrm{Grp}$ *is an infinite loop space.*

Theorem

There is an equivalence $\mathrm{AbGrp} \simeq (0,k)\,\mathrm{Grp}$ *for* $k \geq 2$*, and this lifts to an equivalence of univalent categories.*

For example, for $G : (0,2)\,\mathrm{Grp}$ an abelian group, we have $B^n G = K(G,n)$, an Eilenberg-MacLane space.

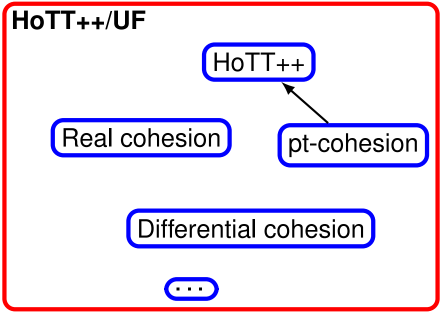
Outline

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. 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*?

Many interesting applications of autophagy: $\mathbb{S}$-cohesion, equivariant cohesion, maybe one day real/smooth/differential cohesion, etc.
All with internally defined models.

Many interesting applications of autophagy: $\mathbb{S}$-cohesion, equivariant cohesion, maybe one day real/smooth/differential cohesion, etc.
All with internally defined models.



Symmetric groups

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

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.)

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

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.

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 $\mathrm{B}\Sigma_\infty$.

$$\begin{tikzcd} \mathbb{N} \arrow[r, "{[-]}"] & \mathrm{FinSet} \arrow[l, bend left, "\mathrm{card}"] \arrow[r, "i"] & \mathrm{B}\Sigma_\infty \arrow[r, "j"] & \mathrm{BAut}\,\mathbb{N} \end{tikzcd}$$

Finitary symmetric group

$\mathrm{B}\Sigma_\infty$ is both the homotopy colimit of

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

and the homotopy coequalizer of

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

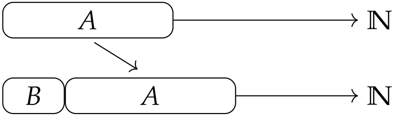
using the equivalence mentioned above.

Constructors:

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

Shift and weakening

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 $A + X$.

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

$$\begin{tikzcd} \mathrm{Type}^I \arrow[r, shift left=4, "f_!"] \arrow[r, shift right=4, "f_*"'] & \mathrm{Type}^J \arrow[l, "f^*" description] \end{tikzcd}$$

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).

Basic nominal theory

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 $x : Z(\mathsf{pt})$

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.

Supports and binding

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

Warmup: untyped lambda calculus

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

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

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 $\mathrm{Type}^{\leq 1}$.

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!