Where Are The Open Sets?

Comparing HoTT with classical Topology.

Chris Grossack (they/them)

§0

Some Disclaimers.

§0 Some Disclaimers.

↳ this is a 30m talk, so I'll necessarily be somewhat vague

§0

Some Disclaimers.

↳ this is a 30m talk, so I'll necessarily be somewhat vague

↳ (but I'll always link to more comprehensive references)

So what's the plan?

So what's the plan?

↳ HoTT is <u>synthetic</u> homotopy theory

So what's the plan?

↳ HoTT is <u>synthetic</u> homotopy theory

↳ how do we know that our theorems are true for a classical topologist?

So what's the plan?

↳ HoTT is <u>synthetic</u> homotopy theory

↳ how do we know that our theorems are true for a classical topologist?

↳ HoTT is a <u>logic</u>, so we want to get <u>semantics</u>.

HoTT

$$\text{HoTT} \xrightarrow{\text{semantics}} \text{Simplicial sets}$$

$$
\begin{tikzcd}
\text{HoTT} \arrow[r, squiggly, "\text{semantics}"] & \text{Simplicial Sets} \arrow[d, "\text{Quillen Equivalence}"] \\
& \text{Topological Spaces}
\end{tikzcd}
$$

The Quillen Equivalence says that any category-theoretic property of simplicial sets (sSet) up to homotopy is <u>also true</u> of topological spaces (Top) up to homotopy.

But, through our semantics, HoTT will tell us true facts about sSet!

Let's get started!

§ 1

Model Categories

§ 1

Model Categories

(a sketch)

Recall the classical notion of homotopy:

Recall the classical notion of homotopy:

$f, g : X \longrightarrow Y$ are <u>homotopic</u> if

Recall the classical notion of homotopy:

$f, g : X \longrightarrow Y$ are <u>homotopic</u> if

$\exists\, H : X \times [0,1] \longrightarrow Y$ so that

Recall the classical notion of homotopy:

$f, g : X \longrightarrow Y$ are <u>homotopic</u> if

$\exists\, H : X \times [0,1] \longrightarrow Y$ so that

- $H(-,0) = f(-)$

Recall the classical notion of homotopy:

$f, g : X \longrightarrow Y$ are <u>homotopic</u> if

$\exists\, H : X \times [0,1] \longrightarrow Y$ so that

- $H(-, 0) = f(-)$
- $H(-, 1) = g(-)$

Such an H is called a <u>homotopy</u> from f to g.

Such an H is called a <u>homotopy</u> from f to g.

written $H: f \simeq g$

Then we say X and Y are

homotopy equivalent if we have

maps $f : X \longrightarrow Y$

Then we say X and Y are <u>homotopy equivalent</u> if we have maps $f: X \longrightarrow Y$, $g: Y \longrightarrow X$

Then we say X and Y are <u>homotopy equivalent</u> if we have maps $f: X \to Y$, $g: Y \to X$

so that

- $f \circ g \sim id_Y$

Then we say X and Y are <u>homotopy equivalent</u> if we have maps $f: X \longrightarrow Y$, $g: Y \longrightarrow X$ so that

- $f \circ g \sim id_Y$
- $g \circ f \sim id_X$

This looks like some kind of isomorphism...

This looks like some kind of isomorphism...

If only we could make

$\sim$ into equality...

This looks like some kind of isomorphism...

If only we could make $\sim$ into equality...

But what if we simply __forced__ that to be true?

Def$^{\underline{n}}$

Def$^{\underline{n}}$

- $\mathcal{W}$ is the class of (Weak) Homotopy Equivalences in Top

$\underline{\text{Def}^{\underline{n}}}$

- $\mathcal{W}$ is the class of (Weak) Homotopy Equivalences in Top
- $\text{Top}[\mathcal{W}^{-1}]$ is the category we get by freely adding inverses for each $f \in \mathcal{W}$.

Good news :

Good news:

(weak) homotopy equivalence

is now isomorphism!

Good news:
(weak) homotopy equivalence
is now isomorphism!

↳ this lets us do category
theory more directly, and
is very useful™.

Bad News:

$\mathrm{Top}[\mathcal{W}^{-1}]$ is <u>awful</u> to work with!

Bad News:

$\mathrm{Top}[\mathcal{W}^{-1}]$ is <u>awful</u> to work with!

↳ basically no limits or colimits, for instance.

Bad News:

$\mathrm{Top}[\mathcal{W}^{-1}]$ is <u>awful</u> to work with!

↳ basically no limits or colimits, for instance.

↳ More importantly, the arrows are <u>extremely</u> complicated.

Good News Again:

Good News Again:

There is a notion of a Model Structure that will let us compute with $Top[w^{-1}]$

Good News Again:

There is a notion of a <u>Model Structure</u> that will let us compute with $\mathrm{Top}[w^{-1}]$

(that is, with spaces up-to-homotopy)

Bad News Again:

I'm not going to tell you what a model structure is.

Bad News Again:

I'm not going to tell you what a model structure is.

(See "Homotopy Theories and Model Categories" by Dwyer & Spalinski)

Briefly, the idea is to choose <u>Fibrations</u>

Briefly, the idea is to choose Fibrations and Cofibrations

Briefly, the idea is to choose Fibrations and Cofibrations in our category, which satisfy certain axioms related to each other and to the weak Equivalences $\mathcal{W}$.

Then we can compute in $\mathrm{Top}[w^{-1}]$ by making computations in Top, restricting attention to the objects that are both <u>fibrant</u> and co<u>fibrant</u>.

Then we can compute in $Top[w^{-1}]$ by making computations in Top, restricting attention to the objects that are both <u>fibrant</u> and co<u>fibrant</u>.

(Again, see Dwyer & Spalinski for details)

But what does this have to do with simplicial sets?

But what does this have to do with simplicial sets?

More importantly... what <u>is</u> a simplicial set?

The idea is to get a <u>combinatorial model</u> for topological spaces.

The idea is to get a <u>combinatorial model</u> for topological spaces.

↳ glue together

The idea is to get a <u>combinatorial model</u> for topological spaces.

↳ glue together points

•

The idea is to get a <u>combinatorial model</u> for topological spaces.

↳ glue together

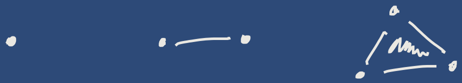
points   arcs

•   •—•

The idea is to get a <u>combinatorial model</u> for topological spaces.

↳ glue together

points   arcs   triangles



The idea is to get a <u>combinatorial model</u> for topological spaces.

↳ glue together

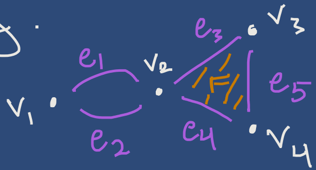
points arcs triangles tetrahedra

The idea is to get a <u>combinatorial model</u> for topological spaces.

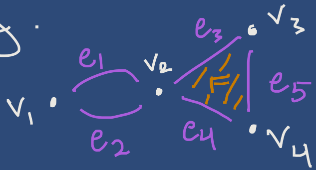
↳ glue together

points arcs triangles tetrahedra ...

eg.

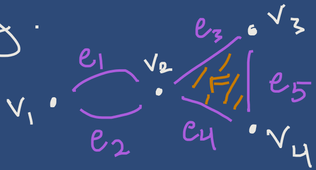


eg.



0-cells: $\{v_1, v_2, v_3, v_4\}$

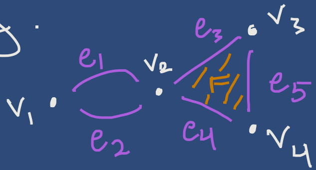
e.g.



0-cells: $\{v_1, v_2, v_3, v_4\}$

1-cells: $\{e_1, e_2, e_3, e_4, e_5\}$

eg.

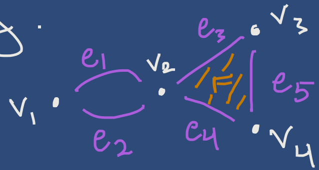


0-cells: $\{v_1, v_2, v_3, v_4\}$

1-cells: $\{e_1, e_2, e_3, e_4, e_5\}$

2-cells: $\{F\}$

eg.



0-cells: $\{v_1, v_2, v_3, v_4\}$

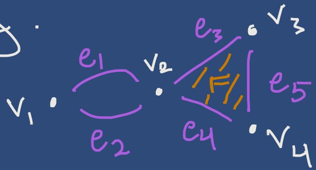
1-cells: $\{e_1, e_2, e_3, e_4, e_5\}$

2-cells: $\{F\}$

We also have

<u>face maps</u>

eg.



0-cells: $\{v_1, v_2, v_3, v_4\}$

1-cells: $\{e_1, e_2, e_3, e_4, e_5\}$

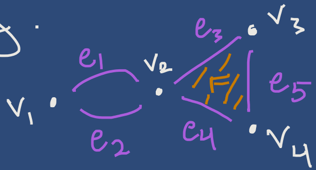
2-cells: $\{F\}$

We also have <u>face maps</u>

eg:

- $d_1 e_1 = v_2$
- $d_0 e_1 = v_1$

eg.



0-cells: $\{v_1, v_2, v_3, v_4\}$

1-cells: $\{e_1, e_2, e_3, e_4, e_5\}$

2-cells: $\{F\}$

We also have face maps

eg:

- $d_1 e_1 = v_2$
- $d_0 e_1 = v_1$

- $d_0 F = e_3$
- $d_1 F = e_4$
- $d_2 F = e_5$

In general, we package this information into a diagram

In general, we package this information into a diagram

X_0

In general, we package this information into a diagram

$$X_0 \leftleftarrows X_1$$

In general, we package this information into a diagram

$$\begin{tikzcd} X_0 & X_1 \arrow[l, shift left] \arrow[l, shift right] & X_2 \arrow[l, shift left=2] \arrow[l] \arrow[l, shift right=2] \end{tikzcd}$$

In general, we package this information into a diagram

$$X_0 \leftleftarrows X_1 \;\substack{\longleftarrow\\ \longleftarrow\\ \longleftarrow}\; X_2 \;\substack{\longleftarrow\\ \longleftarrow\\ \longleftarrow\\ \longleftarrow}\; X_3$$

In general, we package this information into a diagram

$$X_0 \leftleftarrows X_1 \;\substack{\longleftarrow\\ \longleftarrow\\ \longleftarrow}\; X_2 \;\substack{\longleftarrow\\ \longleftarrow\\ \longleftarrow\\ \longleftarrow}\; X_3 \;\cdots$$

In general, we package this information into a diagram

$$\begin{tikzcd}
X_0 \arrow[r, shift left=1] & X_1 \arrow[l, shift left=3] \arrow[l, shift right=1] \arrow[r, shift left=2] \arrow[r, shift right=2] & X_2 \arrow[l, shift left=4] \arrow[l] \arrow[l, shift right=4] \arrow[r, shift left=4] \arrow[r] \arrow[r, shift right=4] & X_3 \arrow[l, shift left=6] \arrow[l, shift left=2] \arrow[l, shift right=2] \arrow[l, shift right=6] & \cdots
\end{tikzcd}$$

In general, we package this information into a diagram

$$
\begin{tikzcd}
X_0 \arrow[r, leftarrow, shift left=2] \arrow[r] \arrow[r, leftarrow, shift right=2] & X_1 \arrow[r, leftarrow, shift left=4] \arrow[r, shift left=2] \arrow[r, leftarrow] \arrow[r, shift right=2] \arrow[r, leftarrow, shift right=4] & X_2 \arrow[r, leftarrow, shift left=6] \arrow[r, shift left=4] \arrow[r, leftarrow, shift left=2] \arrow[r] \arrow[r, leftarrow, shift right=2] \arrow[r, shift right=4] \arrow[r, leftarrow, shift right=6] & X_3 \quad \cdots
\end{tikzcd}
$$

Conveniently realized as a functor into Sets!

In general, we package this information into a diagram

$$
\begin{tikzcd}
X_0 \arrow[r, shift left=2] & X_1 \arrow[l, shift left=4] \arrow[l] \arrow[r, shift left=2] \arrow[r, shift left=6] & X_2 \arrow[l, shift left=4] \arrow[l] \arrow[l, shift right=4] \arrow[r, shift left=2] \arrow[r, shift left=6] \arrow[r, shift right=2] & X_3 \arrow[l, shift left=4] \arrow[l, shift left=8] \arrow[l] \arrow[l, shift right=4] & \cdots
\end{tikzcd}
$$

Conveniently realized as a functor into Sets!

We call this functor category $\underline{\mathrm{sSet}}$.

For more information, see

↳ Friedman's
"An Elementary Illustrated Introduction to Simplicial Sets"

↳ Singh's
"A Survey of Simplicial Sets"

Ok... But why are we talking about this?

Ok... But why are we talking about this?

It turns out there is a model structure on sSet too!

Ok... But why are we talking about this?

It turns out there is a model structure on sSet too!

(Fibrant objects are called <u>Kan Complexes</u>)

Moreover, we have translation functors

Moreover, we have translation functors

$$|\cdot| : sSet \longrightarrow Top$$

Moreover, we have translation functors

$$|\cdot| : sSet \longrightarrow Top \quad (\text{geometric realization})$$

Moreover, we have translation functors

$$|\cdot| : sSet \longrightarrow Top \quad \text{(geometric realization)}$$

$$\mathrm{sing} : Top \longrightarrow sSet$$

Moreover, we have translation functors

$$|\cdot| : sSet \longrightarrow Top \quad \text{(geometric realization)}$$

$$sing : Top \longrightarrow sSet \quad \text{(singular chains)}$$

Moreover, we have translation functors

$$|\cdot| : sSet \longrightarrow Top \quad \text{(geometric realization)}$$

$$\perp$$

$$sing : Top \longrightarrow sSet \quad \text{(singular chains)}$$

Moreover, we have translation functors

$$|\cdot| : sSet \longrightarrow Top \quad \text{(geometric realization)}$$

$$\perp$$

$$\mathrm{sing} : Top \longrightarrow sSet \quad \text{(singular chains)}$$

↳ these play nicely with the model structures on sSet and Top.

Thm

Thm

$\boxed{\mathrm{I}}$ 1.1 and Sing(·) induce functors

Thm

$\boxed{\mathrm{I}}$ $|\cdot|$ and $\mathrm{Sing}(\cdot)$ induce functors

- $\mathbb{L}|\cdot| : sSet[w^{-1}] \longrightarrow Top[w^{-1}]$

Thm $\boxed{\mathrm{I}}$ $|\cdot|$ and $\mathrm{Sing}(\cdot)$ induce functors

- $\mathbb{L}|\cdot| : sSet[w^{-1}] \longrightarrow Top[w^{-1}]$
- $\mathbb{R}\mathrm{Sing} : Top[w^{-1}] \longrightarrow sSet[w^{-1}]$

Thm

$\boxed{\mathrm{I}}$ $|\cdot|$ and $\mathrm{Sing}(\cdot)$ induce functors

- $\mathbb{L}|\cdot| : sSet[w^{-1}] \longrightarrow Top[w^{-1}]$
- $\mathbb{R}\mathrm{Sing} : Top[w^{-1}] \longrightarrow sSet[w^{-1}]$

$\boxed{\mathrm{II}}$ these form an adjoint equivalence

$$Top[w^{-1}] \simeq sSet[w^{-1}]$$

(this is discussed in more detail in the previously mentioned sources)

The punchline is that <u>any</u> question we have about topological spaces up to homotopy

(as long as it is expressible in the language of category theory)

can be answered by looking at $\mathbb{R}\mathrm{Sing}$ of all the spaces involved, and working in sSet instead!

This is great news!

This is great news!

↳ ΔSet is "combinatorial", so it's easier to describe to a computer (i.e. Sage)

This is great news!

↳ sSet is "combinatorial", so it's easier to describe to a computer (i.e. Sage)

↳ sSet is a Topos, giving us access to lots of heavy duty category theoretic machinery.

§ 2

What does this have to do with HoTT?

Lots!

Lots!

But be warned:

this is not for

the faint of heart...

I don't have time to give much detail.

For more, see

I don't have time to give much detail.

For more, see

↳ Riehl's "On the ∞-topos semantics of HoTT" (lectures 1, 2, 3 all on youtube)

I don't have time to give much detail.

For more, see

↳ Riehl's "On the ∞-topos semantics of HoTT" (lectures 1, 2, 3 all on youtube)

↳ Kapulkin & Lumsdaine's "The Simplicial Model of Univalent Foundations"

Morally, the idea is this:

Morally, the idea is this:

$\hookrightarrow$ types in HoTT get interpreted as Kan complexes

Morally, the idea is this:

↳ types in HoTT get interpreted as Kan complexes

↳ Constructions on types get interpreted as constructions on Kan complexes.

eg.

eg.

- dependent types $x:A \vdash B(x)$ type get modeled as fibrations

$$\begin{array}{c} [\![B]\!] \\ \downarrow \\ [\![A]\!] \end{array}$$

eg.

- then Σ and Π types are given by adjunctions to pullback $\Sigma \dashv pr^* \dashv \Pi$

e.g.

- Identity types are interpreted as path spaces

$$\begin{tikzcd} A \arrow[r, tail, "\mathrm{refl}", "\sim"'] \arrow[dr, "\Delta"'] & {[\![x = y]\!]} \arrow[d, two heads] \\ & A \times A \end{tikzcd}$$

eg.

- Identity types are interpreted as path spaces

$$\begin{tikzcd} A \arrow[r, "\mathrm{refl}", "\sim"'] \arrow[dr, "\Delta"'] & {[\![x=y]\!]} \arrow[d, two heads] \\ & A \times A \end{tikzcd}$$

more concretely,

$$\begin{tikzcd} {[\![x=y]\!]} \arrow[d, two heads] \\ A\times A \end{tikzcd} \quad \text{is} \quad \begin{tikzcd} A^{\cdot \to \cdot} \arrow[d, two heads] \\ A\times A \end{tikzcd}$$

eg.

This "explains" path induction!

$$\begin{tikzcd}
A \arrow[r, "p_0"] \arrow[d, "\sim"', "\mathrm{refl}"] & \underline{P} \arrow[d, two heads] \\
A^{\bullet \to \bullet} \arrow[ur, dashed, "J"] \arrow[r, equal] & A^{\bullet \to \bullet}
\end{tikzcd}$$

eg.

This "explains" path induction!

$$\begin{tikzcd}
A \arrow[r, "p_0"] \arrow[d, "s"', "\mathrm{refl}"] & P \arrow[d] \\
A^{\bullet \to \bullet} \arrow[ur, dashed, "J"] \arrow[r, equal] & A^{\bullet \to \bullet}
\end{tikzcd}$$

- $p_0 : P(a, a, \mathrm{refl}_a)$

  says the square commutes

eg.

This "explains" path induction!

$$\begin{tikzcd} A \arrow[r, "p_0"] \arrow[d, tail, "s"', "\mathrm{refl}"] & P \arrow[d, two heads] \\ A^{\cdot \to \cdot} \arrow[r, equal] \arrow[ur, dashed, "J"] & A^{\cdot \to \cdot} \end{tikzcd}$$

$J(x,y,\alpha) : P(x,y,\alpha)$

Says the bottom triangle commutes

eg.

This "explains" path induction!

$$\begin{tikzcd}
A \arrow[r, "p_0"] \arrow[d, "s"', "\mathrm{refl}"] & P \arrow[d] \\
A^{\cdot\to\cdot} \arrow[ur, dashed, "J"] \arrow[r, equal] & A^{\cdot\to\cdot}
\end{tikzcd}$$

$J(a,a,\mathrm{refl}_a) \doteq p_0(a)$

Says the top triangle commutes.

Unfortunately, everything I just told you is a lie.

Unfortunately, everything I just told you is a lie.

↳ in HoTT, we have on-the-nose equality of syntax

Unfortunately, everything I just told you is a lie.

↳ in HoTT, we have on-the-nose equality of syntax

↳ in ΔSet, our structure is only defined up-to-isomorphism so we need to evil-ify sSet somehow, and then do all these constructions in that setting.

so we need to evil-ify sSet somehow, and then do all these constructions in that setting.

↳ the key is called a (weakly) <u>universal fibration</u>

From a universal fibration $\mathcal{U}$ we can turn every construction into a map from (some variant of) $\mathcal{U}$ to itself.

Then each construction is a pullback along a variant, and by choosing a particular pullback in each case, we get equality on-the-nose.

(see the incredibly lucid explanation in Kapulkin & Lumsdaine's "The Simplicial Model of Univalent Foundations" for more)

If you decide to go through with this, here's a good exercise to keep in mind:

If $\mathrm{isContr}(A)$ is inhabited, show the Kan complex $[\![A]\!]$ is contractible.

If $\mathrm{isContr}(A)$ is inhabited,
show the Kan complex $[\![A]\!]$ is
contractible.

(use the ΔSet semantics
of HoTT)

If $\mathrm{isContr}(A)$ is inhabited, show the Kan complex $[\![A]\!]$ is contractible.

Next, show the geometric realization $|[\![A]\!]|$ is contractible.

If $\mathrm{isContr}(A)$ is inhabited, show the Kan complex $[\![A]\!]$ is contractible.

Next, show the geometric realization $|[\![A]\!]|$ is contractible.

(use the properties of $|\cdot| \dashv \mathrm{Sing}$)

Then pat yourself on the back for proving something about an honest topological space using HoTT!

Thank You

^_^