A BASIC INTRODUCTION TO HOMOTOPY TYPE THEORY

EVA BELMONT

Abstract. Homotopy type theory is a new(ish) system of foundations for mathematics that is based on homotopy theory, and has applications to computer-checkable proofs. In these notes, I give an answer to the question

Why does logic have anything to do with topology?!

aimed at those without much background in mathematical foundations.

Warning: I am not even close to being a logician or a category theorist. I learned this for fun to give a casual general-math-audience talk in graduate school, and hope the intuition here might help someone new to the subject. There are many lies and omissions, some of them intentional and most likely many others unintentional. Feel free to contact me about errors, but also take everything here with a grain of salt.

My goal in these notes is to explain why logic has anything to do with homotopy theory. The basic outline of the answer is the following diagram:

$$\begin{tikzcd} \text{Types} \arrow[r] & \infty\text{-groupoids} \arrow[l, bend right, dotted] & \text{Spaces} \arrow[l, "\simeq"'] \end{tikzcd}$$

But don't worry if you don't know what's going on in this diagram: the point of these notes is to explain that. We'll start with types.

1. Types

Type theory[1] is an alternative to the usual foundations of mathematics based on set theory and logic. Set theory is based on the "element-of" relation $a \in A$. In type theory, we have a similar relation

$$\mathtt{a} : \mathtt{A}$$

which is read as "$\mathtt{a}$ is a term of [the type] $\mathtt{A}$". For now you can think of this as vaguely being like $\mathtt{a} \in \mathtt{A}$; we'll get to some of the differences later.

In set theory, if A and B are sets, you can form other related sets like their product and disjoint union. If $\mathtt{A}$ and $\mathtt{B}$ are types, type theory says that the following are also types:

- the product $\mathtt{A} \times \mathtt{B}$,
- the disjoint union $\mathtt{A} \sqcup \mathtt{B}$,
- the function type, denoted $\mathtt{A} \to \mathtt{B}$,

Date: March 17, 2016.

[1] There are many kinds of type theory. In these notes, "type theory" specifically means Martin-Löf type theory with univalence.

- constant types $\mathtt{T}$ and $\mathtt{F}$.[2]

(The constant types correspond to the one-point set and the empty set in set theory.)

One of the important differences between classical logic and type theory comes from the fact that, in type theory, we want the following to be true:

Theorem 1.1 (Curry-Howard Isomorphism)**.** *There is a correspondence*

$$\textit{Types} \rightleftarrows \textit{Propositions}$$

as follows. To a type $\mathtt{A}$ *we associate the proposition "*$\mathtt{A}$ *is nonempty." To a proposition* P *we associate the type whose terms are proofs of* P*.*

Look at one of the compositions:

$$\begin{array}{ccccc} \text{Types} & \longrightarrow & \text{Propositions} & \longrightarrow & \text{Types} \\ \mathtt{A} & \longmapsto & \text{“}\mathtt{A}\text{ is nonempty”} & \longmapsto & \{\text{Proofs that }\mathtt{A}\text{ is nonempty}\} \end{array}$$

That is, we want to identify the type $\mathtt{A}$ with the collection of proofs that $\mathtt{A}$ is nonempty. A term $\mathtt{a} : \mathtt{A}$ clearly leads to a (constructive) proof that $\mathtt{A}$ is nonempty. But what if there are nonconstructive proofs that $\mathtt{A}$ is nonempty, i.e. proofs that don't arise from exhibiting a term of $\mathtt{A}$? The Curry-Howard isomorphism only holds if we work in a setting where all proofs are constructive.

Now I'll describe how type theory modifies classical propositional logic to achieve a setting where all proofs are constructive. Recall classical propositional logic is a system of basic "moves" you can use to get from a collection of logical statements to their consequences. For example, if you assume a proposition A then you are allowed to conclude the proposition $B \implies A$ for any proposition B. Among these moves are the *law of excluded middle*, which allows you to conclude $A \vee \neg A$ for any proposition A, and *double negation elimination*, which says that $\neg\neg A$ implies A for any proposition A. Every time you use proof by contradiction, you are implicitly invoking this: you show $\neg A$ is false, i.e. $\neg\neg A$ is true, and want to use this to conclude A. Intuitionistic logic is what you get when you take classical logic and remove the law of the excluded middle and double negation elimination. ("I don't *dislike* him, but I don't like him either.")

So let's assume we've made this modification and the Curry-Howard isomorphism holds. One reason people want this is that it says that type-checking is the same as proof-checking: checking that $\mathtt{a}$ is a term of $\mathtt{A}$ corresponds, under the Curry-Howard isomorphism, to checking that $\mathtt{a}$ is a valid proof (example) of the proposition "$\mathtt{A}$ is nonempty". In particular we get a correspondence

[2] The curious reader might be asking, "what *are* these types"? Of course, if you're thinking of types as sets, like I told you, then it is perfectly clear what these are. But type theory isn't actually defined in terms of set theory, right? The answer is that these things are defined in terms of universal properties... Read [V+13] to learn more.

terms of $\longleftrightarrow$	proofs of
$\mathtt{A} \times \mathtt{B}$	$A \wedge B$
$\mathtt{A} \sqcup \mathtt{B}$	$A \vee B$
$\mathtt{A} \to \mathtt{B}$	$A \implies B$
$\prod_{\mathtt{x:A}} \mathtt{B(x)}$	$\forall x \in A, B(x)$
$\coprod_{\mathtt{x:A}} \mathtt{B(x)}$	$\exists x \in A, B(x)$
$\mathtt{T}$, $\mathtt{F}$	T, F

where we're using the isomorphism to identify types $\mathtt{A}$, $\mathtt{B}$ on the left with the corresponding propositions A, B on the right.

Now let's add another feature to our description of type theory: for all terms $\mathtt{a}$, $\mathtt{b}$ in $\mathtt{A}$, introduce the *identity type* $\mathrm{Id}_{\mathtt{A}}(\mathtt{a}, \mathtt{b})$. Think of this as the collection of all proofs that $\mathtt{a} = \mathtt{b}$ (of course, it may be empty). We want this to satisfy some properties:

(1) there is a canonical term in $\mathrm{Id}_{\mathtt{A}}(\mathtt{a}, \mathtt{a})$ corresponding to the canonical "proof" that $\mathtt{a} = \mathtt{a}$;

(2) there is a function type $\mathrm{Id}_{\mathtt{A}}(\mathtt{a}, \mathtt{b}) \times \mathrm{Id}_{\mathtt{A}}(\mathtt{b}, \mathtt{c}) \to \mathrm{Id}_{\mathtt{A}}(\mathtt{a}, \mathtt{c})$ corresponding to the statement "if $a = b$ and $b = c$ then $a = c$."

The naïve expectation is that $\mathrm{Id}_{\mathtt{A}}(\mathtt{a}, \mathtt{b})$ will be either $\mathtt{T}$ or $\mathtt{F}$—either $\mathtt{a} = \mathtt{b}$ or not. If we're thinking about types as sets, then it's true that there isn't more than one bit of information required to answer the question "are these elements equal." But this is where we start getting into the nuances of type theory, and how it's different from set theory. It turns out that one cannot prove, using the axioms of type theory, that $\mathrm{Id}_{\mathtt{A}}(\mathtt{a}, \mathtt{b})$ is either $\mathtt{T}$ or $\mathtt{F}$. In the language of the Curry-Howard correspondence, this is saying that we want to allow for the possibility that there can be different (constructive) proofs that two propositions are equal. In order to understand this better, we're going to have to update our mental model of what types are, from sets to something more complicated.

To get an idea of what this "something more complicated" might look like, let's run with the idea that $\mathrm{Id}_{\mathtt{A}}(\mathtt{a}, \mathtt{b})$ might be an interesting type with more than one term. That is, there may be multiple terms $\mathtt{x}$, $\mathtt{y}$ of $\mathrm{Id}_{\mathtt{A}}(\mathtt{a}, \mathtt{b})$, and one may ask whether they are equal. That is, we may consider

$$\mathrm{Id}_{\mathrm{Id}_{\mathtt{A}}(\mathtt{a},\mathtt{b})}(\mathtt{x}, \mathtt{y})$$

and that may have interesting elements $\mathtt{z}$, $\mathtt{w}$ in it, so we may consider

$$\mathrm{Id}_{\mathrm{Id}_{\mathrm{Id}_{\mathtt{A}}(\mathtt{a},\mathtt{b})}(\mathtt{x},\mathtt{y})}(\mathtt{z}, \mathtt{w}),$$

and so on, ad infinitum.

Moreover, we have a funny notion of associativity: if we have $\mathtt{x} : \mathrm{Id}_{\mathtt{A}}(\mathtt{a}, \mathtt{b})$, $\mathtt{y} : \mathrm{Id}_{\mathtt{A}}(\mathtt{b}, \mathtt{c})$, $\mathtt{z} : \mathrm{Id}_{\mathtt{A}}(\mathtt{c}, \mathtt{d})$, we might wonder whether $(\mathtt{xy})\mathtt{z} = \mathtt{x}(\mathtt{yz})$ is true, where concatenation refers to the composition in (2) above. But what does that "=" mean? What we actually require is for there to be a term in the type

$$\mathrm{Id}_{\mathrm{Id}_{\mathtt{A}}(\mathtt{a},\mathtt{d})}((\mathtt{xy})\mathtt{z}, \mathtt{x}(\mathtt{yz})).$$

The structure of $\mathrm{Id}_{\mathrm{Id}_{\mathrm{Id}\cdots}}$ looks a lot like the structure of an ∞-category.[3] I won't go into the nuances of how you define an ∞-category (and, like type theory, there are many models, though unlike type theory, many of them end up being equivalent), but the basic idea is as follows. A category $\mathcal{C}$ (a.k.a. 1-category) has as data:

- a collection of objects $\mathrm{ob}\,\mathcal{C}$, and
- for every pair $a, b \in \mathcal{C}$, a set (called the set of morphisms) $\mathrm{Hom}_{\mathcal{C}}(a, b)$ satisfying some axioms that allows one to compose morphisms in an associative and unital way.

A basic example is the category of topological spaces, whose objects are topological spaces and $\mathrm{Hom}(X, Y)$ is the set of continuous maps from X to Y.

A (weak) 2-category has the same data as a 1-category, plus a set of "morphisms between morphisms": for any $f, g \in \mathrm{Hom}(A, B)$, there is a set $\mathrm{Hom}(f, g)$ satisfying some axioms. Instead of requiring strict associativity of morphisms as in a 1-category, we only require there to be an invertible 2-morphism from $(fg)h$ to $f(gh)$.

For example, let X be a topological space, and consider the 2-category of paths in X: the objects are points of X, the morphisms in $\mathrm{Hom}(x, y)$ are continuous maps $f : [0, 1] \to X$ starting at x and ending at y, and the 2-morphisms $\mathrm{Hom}(f, g)$ are homotopies between paths, i.e. continuous maps $h : [0, 1] \times [0, 1] \to X$ such that $h(0, -) = f$ and $h(1, -) = g$. We may define composition of paths as:

$$g \circ f(t) = \begin{cases} f(2t) & t \in [0, \frac{1}{2}] \\ g(2t - 1) & t \in [\frac{1}{2}, 1]. \end{cases}$$

Note that this is only associative up to re-indexing; i.e. $f(gh)$ and $(fg)h$ are not the same map $[0, 1] \to X$: note that $f(gh)$ does f for $t \in [0, \frac{1}{2}]$, g for $t \in [\frac{1}{2}, \frac{3}{4}]$, and h for $t \in [\frac{3}{4}, 1]$, whereas $(fg)h$ does f for $t \in [0, \frac{1}{4}]$, g for $t \in [\frac{1}{4}, \frac{1}{2}]$, and h for $t \in [\frac{1}{2}, 1]$. However, there is a homotopy (2-morphism) from one to the other given by continuously changing the speeds at which these paths are travelled.

You can probably imagine what an n-category is. An ∞-category (weak ∞-groupoid) has collections of n-morphisms for every n, and n-morphisms are invertible up to $(n+1)$-morphisms.

What does this have to do with where we left off in the previous section? Let's define an n-category whose:

- objects are terms of `A`,
- morphism set from `a` to `b` is $\mathrm{Id}_{\mathtt{a}}(\mathtt{a}, \mathtt{b})$,
- 2-morphism set is from `x` to `y` is $\mathrm{Id}_{\mathrm{Id}_{\mathtt{A}}(\mathtt{a},\mathtt{b})}(\mathtt{x}, \mathtt{y})$,
- and so on.

[3] I mean weak $(\infty, 0)$-category here.

Then the associativity required at the end of section 1 is exactly the associativity required in an n-category. This is what I mean by the arrow "Types $\longrightarrow$ ∞-groupoids" at the beginning of the notes.

3. Spaces vs. Types

Next, I'm going to talk about the arrow Spaces $\longrightarrow$ ∞-groupoids. Let X be a space. We first attempt to define the category whose objects are points of x and whose morphisms are paths in X. As we noted in the previous section, this does not satisfy associativity. One fix is to take the morphisms to be paths up to homotopy. This works, and the resulting category is called the fundamental groupoid, written $\prod_1(X)$. In fact, $\pi_1(X, x) = \mathrm{Aut}_{\prod_1(X)}(x)$, and $\prod_1(X)$ encodes the same information about X as is encoded by $\pi_0(X)$ and $\pi_1(X)$. Thus we say that groupoids are in bijection with spaces X such that $\pi_{>1}X = 0$. (This is a higher analogue of the fact that sets ("0-categories") are in bijection with spaces such that $\pi_{>0} = 0$.) Can we get an analogue of this for $\pi_{>n}$?

Let's go back to the 2-category described in section 2: the objects are points of X, the morphisms are paths of X, and the 2-morphisms are homotopies between paths. But we have to be careful: can you compose 2-morphisms in an associative way? A better construction to make an ∞-groupoid, where the objects are points, the morphisms are paths, the 2-morphisms are homotopies between paths, the 3-morphisms are homotopies between homotopies, and so on. This is called the fundamental ∞-groupoid.

To be fair, there's a lot I've swept under the rug about the construction of ∞-groupoids. Grothendieck's homotopy hypothesis is essentially the conjecture that ∞-groupoids can be defined in a way such that

$$\text{Spaces} \longrightarrow \infty\text{-groupoids}$$

sending X to its fundamental ∞-groupoid is a well-defined equivalence. More precisely, the conjecture is that there is an equivalence of $(n+1)$-categories between "homotopy n-types" (what can be seen about spaces by just looking at $\pi_{\leq n}$) and n-groupoids.

But let's pretend we believe this; it's morally true, at least. Going back to the diagram at the top of these notes, this can be summarized by the claim that there is a map

$$\text{Types} \longrightarrow \text{Spaces}. \tag{3.1}$$

That is, take a type `A`; there is a space (or "space-like thing", depending on how much you believe the details omitted above), such that terms of `A` correspond to points in the space, $\mathrm{Id}_{\mathtt{A}}(\mathtt{a}, \mathtt{b})$ consists of paths from `a` to `b` in the space, and so on, with the higher identity types corresponding to homotopies between homotopies.

Early on, we decided that sets were too restrictive as a model for types: they did not display all the behavior that types are allowed to display. Instead, what we're essentially saying here is that spaces form a better model than sets, and that what we've done so far can be summarized as replacing set theory + classical logic, with "space theory" + a form of logic in which the Curry-Howard isomorphism holds.

For example, in set theory, we have the *axiom of extensionality*, which just says that two sets are the same if they have the same elements. One obvious consequence of the above sketch of a model for types is that this does not hold in type theory; instead, types are like spaces: two spaces might have the same points, but if they are glued together in different ways, then those spaces are different.

Unfortunately, the backwards direction on the arrow (3.1) needs more work. We would like to make the statement "types with equivalent ∞-groupoids are equal." This is a statement about two possible kinds of equality between types `A`, `B`:

(1) there are maps $f : \mathtt{A} \rightleftarrows \mathtt{B} : g$ such that $f \circ g = \mathrm{Id}_B$ and $g \circ f = \mathrm{Id}_A$;

(2) there is a term in $\mathrm{Id}_{\mathtt{Type}}(\mathtt{A}, \mathtt{B})$, where `Type` is a type whose terms are all types (this is like the category of categories).

This present discussion is going on at a different level than the rest of the notes: instead of talking about paths between *terms*, we're now talking about paths between *types*

$$\begin{tikzcd} \mathtt{A} \arrow[r, bend left] & \mathtt{B} \arrow[l, bend left] \end{tikzcd}$$

and we're trying to say something about the geometry of the *universe of types* `Type`. The *univalence axiom* (one of Voevodsky's main contributions to this subject) essentially guarantees something like this works.

If you're interested in learning more, check out the following resources. [V+13] is the canonical reference and contains all the details. The beginning chapters are not a bad place to start if you want to learn more about what types really are. The other items are notes at various levels of (in)formality that introduce some of the main concepts. As a sampler: [LP15] talks about path induction, one of the main ideas in homotopy type theory; [Lic13] talks more about the relationship between ∞-groupoids and types; [PW14] is a good in-depth introduction aimed at those with some knowledge of algebraic topology.

References

[Awo14] Steve Awodey, *Recent work in homotopy type theory*, `https://www.andrew.cmu.edu/user/awodey/hott/ams2014.pdf`, 2014.

[Coq14] Thierry Coquand, *Univalent foundation and constructive mathematics*, `http://www.cse.chalmers.se/~coquand/lics18.pdf`, 2014.

[Lic13] Dan Licata, *Homotopy theory in homotopy type theory: Introduction*, `http://homotopytypetheory.org/2013/03/08/homotopy-theory-in-homotopy-type-theory-introduction/`, 2013.

[LP14] James Ladyman and Stuart Presnell, *A primer on homotopy type theory part 1: The formal type theory*.

[LP15] ______, *Identity in homotopy type theory, part I: The justification of path induction*, Philosophia Mathematica **23** (2015), no. 3, 386–406.

[Mac15] Jackson Macor, *A brief introduction to type theory and the univalence axiom*, `https://math.uchicago.edu/~may/REU2015/REUPapers/Macor.pdf`, 2015.

[PW14] Álvaro Pelayo and Michael Warren, *Homotopy type theory and Voevodsky's univalent foundations*, Bulletin of the American Mathematical Society **51** (2014), no. 4, 597–648.

[Shu] Mike Shulman, *From set theory to type theory*, `https://golem.ph.utexas.edu/category/2013/01/from_set_theory_to_type_theory.html`.

[sta16] *Intuitionistic type theory*, `https://plato.stanford.edu/entries/type-theory-intuitionistic/` (entry in *Stanford Encyclopedia of Philosophy*), 2016.

[V+13] Vladimir Voevodsky et al., *Homotopy type theory: Univalent foundations of mathematics*, Institute for Advanced Study (Princeton), The Univalent Foundations Program (2013), 2007–2009.