An introduction to cohesive homotopy type theory

Jonas Frey

2022 NORTH AMERICAN ANNUAL MEETING
OF THE ASSOCIATION FOR SYMBOLIC LOGIC
Cornell University
Ithaca, NY, USA
April 710, 2022

Overview

Three Parts

1. Type theory
2. Homotopy Type Theory
3. Cohesive Homotopy Type Theory

Part I – Type Theory

Pre-history: Russell, Simple Types, Lambda Calculus

- According to the *Stanford Encyclopedia of Philosopy*[1]:

  «The theory of types was introduced by Russell in order to cope with some contradictions he found in his account of set theory and was introduced in Appendix B: The Doctrine of Types of Russell 1903[2]*.»*

- Detailed presentation in *Principia Mathematica*, using
  - a type i of individuals, and
  - types $P(A_1, \ldots, A_n)$ of n-ary relations ranging over previously defined types
- Functions are defined as *functional relations*
- Church 1940[3] gives a reformulation using λ-calculus

---

1. T. Coquand. "Type Theory". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Fall 2018. Metaphysics Research Lab, Stanford University, 2018.
2. B. Russell. *The principles of Mathematics. Vol I.*. English. Cambridge: University Press. XXIX u. 534 S. 8° (1903). 1903.
3. A. Church. "A formulation of the simple theory of types". In: *The Journal of Symbolic Logic* (1940).

Higher order logic in simply typed λ-calculus

Here's a variation of Church's system:

- We have *base types* ι (individuals) and o (propositions), and *arrow types* $A \to B$ for all types A and B
- Well-formed terms are presented using *typing judgments*: i.e. expressions of the form

$$\Gamma \vdash t : B\,.$$

  Here $\Gamma \;\equiv\; x_1{:}A_1, \ldots, x_n{:}A_n$ is a list of 'typed' variables called the **context** of the judgments and the judgment asserts that t is a well-formed term of type B possibly containing the variables of the specified types
- Well-formed terms-in-context are formed inductively by the rules

$$\frac{}{x_1{:}A_1, \ldots, x_n{:}A_n \vdash x_i : A_i} \qquad \frac{\Gamma \vdash t : B \to C \qquad \Gamma \vdash u : B}{\Gamma \vdash (t\,u) : C} \qquad \frac{\Gamma, x{:}B \vdash t : C}{\Gamma \vdash (\lambda x{:}B\,.\,t) : B \to C}$$

  from constants for logical operations (including e.g. $\forall_A : (A \to o) \to o$ for all types A).
- 'Logic' is given by a Hilbert-style system on terms of type o representing propositions and predicates

Another point of view on the λ*-calculus*

- Erasing terms from the typing rules gives a natural deduction system for the implicational fragment of intuitionistic propositional logic (IPC)

$$\frac{}{A_1, \ldots, A_n \vdash A_i} \qquad \frac{\Gamma \vdash B \to C \qquad \Gamma \vdash B}{\Gamma \vdash C} \qquad \frac{\Gamma, B \vdash C}{\Gamma \vdash B \to C}$$

- Thus, in absence of constants a type A is inhabited by a closed term iff it is a propositional tautology
- This is an instance of the *propositions as types* paradigm (a.k.a. Curry Howard isomorphism)
- Observe the difference
    - In Church's system types represent collections ('sets') of mathematical objects, and terms represent elements
    - Under the propositons-as-types reading, types are propositions and λ-terms are encodings of proof trees!
- Per Martin-Löf's **dependent type theory** unifies both points of view, and types can be used to represent both propositions and collections of mathematical objects ('sets').

Martin-Löf's Dependent Type Theory

Dependent type theory extends the language of simple type theory by introducing **type families** ('dependent types') written

$$x{:}A \vdash B[x] \ \text{type}.$$

- Depending on the point of view, dependent types can represent both families of sets, and predicates ('families of propositions')
- We can consider types depending on several variables:

  $$x{:}A, y{:}B[x] \vdash C[x, y] \ \text{type}$$

  and more generally

  $$x_1{:}A_1, \ldots, x_n{:}A_n[x_1, \ldots, x_{n-1}] \vdash B[x_1, \ldots, x_n] \ \text{type}$$

- Observe that types in the context may depend on preceding types.
- Well-typed terms $x{:}A \vdash t[x] : B[x]$ can represent
  - either families of elements ('choice functions') of a family of sets, or
  - proofs of a predicate.

Type formers in dependent type theory

Besides ‘simple’ type formers

$$\frac{\Gamma \vdash A \ \text{type} \quad \Gamma \vdash B \ \text{type}}{\Gamma \vdash A \times B \ \text{type}} \qquad \frac{\Gamma \vdash A \ \text{type} \quad \Gamma \vdash B \ \text{type}}{\Gamma \vdash A + B \ \text{type}} \qquad \frac{\Gamma \vdash A \ \text{type} \quad \Gamma \vdash B \ \text{type}}{\Gamma \vdash A \to B \ \text{type}}$$

and constant types $0, 1, N$, we have dependent sum and dependent product type formers:

$$\frac{\Gamma, x{:}A \vdash B \ \text{type}}{\Gamma \vdash \Pi x{:}A \,.\, B \ \text{type}} \qquad \frac{\Gamma, x{:}A \vdash B \ \text{type}}{\Gamma \vdash \Sigma x{:}A \,.\, B \ \text{type}}$$

These type formers admit different readings, depending on whether we see types as sets or propositions

	types as sets	types as propositions
$A \times B$	cartesian product	conjunction $\wedge$
$A + B$	disjoint union	disjunction $\vee$
$A \to B$	set of functions	implication $\Rightarrow$
1	singleton	‘true’
0	empty set	‘false’
$\Pi x{:}A \,.\, B$	product of a family of sets	universal quantification $\forall$
$\Sigma x{:}A \,.\, B$	disjoint union of a family of sets	existential quantification $\exists$

Introduction and elimination rules

All types come with ‘introduction’ and ‘elimination’ rules. For example:

Type former	Intro	Elim
$(- \times -)$	$\dfrac{\Gamma \vdash t : A \quad \Gamma \vdash u : B}{\Gamma \vdash (t, u) : A \times B}$	$\dfrac{\Gamma \vdash p : A \times B}{\Gamma \vdash \pi_1(p) : A} \qquad \dfrac{\Gamma \vdash p : A \times B}{\Gamma \vdash \pi_2(p) : B}$
$(- + -)$	$\dfrac{\Gamma \vdash t : A}{\Gamma \vdash \sigma_1(t) : A + B} \qquad \dfrac{\Gamma \vdash u : B}{\Gamma \vdash \sigma_2(u) : A + B}$	$\dfrac{\Gamma \vdash t : A{+}B \quad \Gamma, x{:}A \vdash u : C \quad \Gamma, y{:}B \vdash v : C}{\Gamma \vdash \mathrm{match}(t, \sigma_1(x).u, \sigma_2(y).v) : C}$
Π	$\dfrac{\Gamma, x{:}A \vdash t : B}{\Gamma \vdash (\lambda x{:}A \,.\, t) : \Pi x{:}A \,.\, B}$	$\dfrac{\Gamma \vdash t : \Pi x{:}A \,.\, B \quad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B[u/x]}$
Σ	$\dfrac{\Gamma \vdash a : A \quad \Gamma \vdash b : B[a/x]}{\Gamma \vdash (a, b) : \Sigma x{:}A \,.\, B}$	$\dfrac{\Gamma \vdash p : \Sigma x{:}A \,.\, B}{\Gamma \vdash \pi_1(p) : A} \qquad \dfrac{\Gamma \vdash p : \Sigma x{:}A \,.\, B}{\Gamma \vdash \pi_2(p) : B}$

Again, these rules admit a set theoretic and a logical reading.

Definitional equality

- **Definitional equality** identifies terms of the same type.
- For each type former there's a 'β-rule' and possibly an η-rule, e.g.:
  - If $\Gamma, x{:}A \vdash t : B$ and $\Gamma \vdash u : A$ then $\Gamma \vdash (\lambda x{:}A\,.\,t)u \equiv t[u/x] : B$ (β-equality)
  - If $\Gamma \vdash f : \Pi x{:}A\,.\,B$ then $\Gamma \vdash f \equiv (\lambda x{:}A\,.\,f\,x) : \Pi x{:}A\,.\,b$ (η-equality)
  - If $\Gamma \vdash t : A$ and $\Gamma \vdash u : B$ then $\Gamma \vdash \pi_1(t, u) \equiv t : A$ and $\Gamma \vdash \pi_2(t, u) \equiv u : B$ (β-equality)
  - If $\Gamma \vdash p : A \times B$ then $\Gamma \vdash p \equiv (\pi_1(p), \pi_2(p)) : A \times B$ (η-equality)
  - ...
- Definitional equality is the **congruence relation** generated by these equations, i.e. we close up under reflexivity, transitivity, symmetry, and term formers.
- Because we can substitute types into terms, we have to extend definitional equality to types:

$$\frac{\Gamma, x{:}A \vdash B \ \text{type} \qquad \Gamma \vdash t \equiv u : A}{\Gamma \vdash B[t/x] \equiv B[u/x]}$$

- This forces us to introduce the dreaded **conversion rule**:

$$\frac{\Gamma \vdash t : A \qquad \Gamma \vdash A \equiv B}{\Gamma \vdash t : B}$$

Propositional equality: Identity types

- Definitional equality lives on the 'meta-level' of type theory, cannot be used to formalize mathematical 'hypothetical' reasoning in type theory where equalities are hypotheses.
- This is because definitional equality cannot occur 'on the left of the turnstile' in a judgment.
- An 'object-level' notion of equality is given by **identity types** which in accordance with the propositions-as-types principle represent the equality predicate on a type A as a type family in two arguments of type A:

Id-*types*

Formation
$$\frac{\Gamma \vdash t : A \quad \Gamma \vdash u : A}{\Gamma \vdash \mathsf{Id}_A(t, u) \ \text{type}}$$

Introduction
$$\frac{\Gamma \vdash a : A}{\Gamma \vdash \mathsf{refl}(a) : \mathsf{Id}_A(a, a)}$$

Elimination
$$\frac{\Gamma, x{:}A, z : \mathsf{Id}_A(a, x) \vdash C[x, z] \ \text{type} \qquad \Gamma \vdash t : C[a, \mathsf{refl}_a] \qquad \Gamma \vdash p : \mathsf{Id}_A(a, b)}{\Gamma \vdash t \triangleright_{C[-,-]} p : C[b, p]}$$

Def. equality
$$\Gamma \vdash (t \triangleright_{C[-,-]} \mathsf{refl}(a)) \equiv_\beta t : C[a, \mathsf{refl}_a]$$

Extensional vs intensional type theory

- We can also formulate an η-**rule for identity types**:

$$\frac{\Gamma, x{:}A, z : \mathsf{Id}_A(a, x) \vdash t[x, z] : C[x, z]}{\Gamma, x{:}A, z : \mathsf{Id}_A(a, x) \vdash t[x, z] \equiv (t[a, \mathsf{refl}(a)] \triangleright z) : C[x, z]}$$

- This rule implies the **equality reflection** rule

$$\frac{\Gamma \vdash p : \mathsf{Id}_A(a, b)}{\Gamma \vdash a \equiv b : A}$$

  and leads us to the system of **extensional type theory**, which Martin-Löf originally considered[4].
- Unfortunately, type checking in extensional type theory is not decidable, and thus the system is not suited for *computer implementations*.
- Therefore computer scientists turned their attention to *intensional type theory* (without η-for identity types).

[4] P. Martin-Löf and G. Sambin. *Intuitionistic type theory*. Bibliopolis Napoli, 1984.

(Non-)Uniqueness of identity proofs

- The absence of equality reflection in intensional type theory has a curious consequence, which was long seen as a weakness or 'incompleteness phenomenon':
- Although
  - the only 'canonical elements' (those given by intro rules) of identity types are reflexivity terms, and
  - it is possible to show *externally* that whenever we have $\vdash t : \mathsf{Id}_A(a, b)$ in *empty context*, then $\vdash a \equiv b : A$ and $\vdash t \equiv \mathsf{refl}(a) : \mathsf{Id}_A(a, b)$

  it is impossible to show *internally* that identity proofs are unique, i.e. the type

$$\Pi(a\, b : A)\Pi(p\, q : \mathsf{Id}_A(a, b))\, .\, \mathsf{Id}_{\mathsf{Id}_A(a,b)}(p, q)$$

  is not inhabited for a generic type A.
- The latter was shown by Hofmann and Streicher[5] by giving a countermodel in the category of **groupoids**, where
  - types are interpreted as groupoids,
  - terms are interpreted as objects of a groupoid, and
  - $\mathsf{Id}_A(a, b)$ is interpreted as the set of morphisms between a and b.

[5] M. Hofmann and T. Streicher. "The Groupoid Model Refutes Uniqueness of Identity Proofs". In: *Proceedings of the Ninth Annual Symposium on Logic in Computer Science (LICS '94), Paris, France, July 4-7, 1994*. IEEE Computer Society, 1994.

The groupoid model

In[6], Hofmann and Streicher write

> *«In this paper we have shown that the current formulation of intensional constructive type theory is incomplete w.r.t. the intuitive understanding of identity types. Namely, it is possible to construct a model where not all proof objects of an identity type are necessarily propositionally equal.»*

In[7] they write:

> *«... allows for the addition of axioms (inconsistent with UIP) expressing a view of propositional equality as a generalised notion of isomorphism. Intuitively, these axioms state that for a universe U and $A, B : U$ the identity set $\mathrm{Id}_U(A, B)$ corresponds to the set of isomorphisms between A and B.»*

---

[6] M. Hofmann and T. Streicher. "The Groupoid Model Refutes Uniqueness of Identity Proofs". In: *Proceedings of the Ninth Annual Symposium on Logic in Computer Science (LICS '94), Paris, France, July 4-7, 1994*. IEEE Computer Society, 1994.

[7] M. Hofmann and T. Streicher. "The groupoid interpretation of type theory". In: *Twenty-five years of constructive type theory (Venice, 1995)* (1998).

Part II – Homotopy Type Theory

Homotoy type theory

- **Homotopy type theory** extends Hofmann and Streicher's ideas radically: instead of groupoids, it proposes an interpretation of type theory in ∞-**groupoids**, which by **Grothendieck's homotopy hypothesis** are equivalent to **homotopy types**.

$$\{\text{Homotopy types}\} \simeq \{\infty\text{-groupoids}\}$$

- Under this interpretation, elements of an identity type $\mathsf{Id}_A(a, b)$ can be thought of as **paths** between points a and b in a topological space representing the homotopy type A.
- *Higher homotopies* between paths can be accessed through iterated identity types.
- This idea can be traced back to
    - the work of Awodey and Warren[8] (first presented at FMCS 2006) which makes the idea of the homotopical interpretation precise using **path objects** in **Quillen model categories** to interpret identity types, and
    - the work of Voevodsky, who proposed a system called *homotopy lambda calculus*[9] as a system for formalization of mathematics.

---

[8] S. Awodey and M. Warren. "Homotopy theoretic models of identity types". In: *Mathematical Proceedings of the Cambridge Philosophical Society*. Cambridge University Press. 2009.

[9] Vladimir Voevodsky. "A very short note on the homotopy λ-calculus". In: *Unpublished note* (2006).

h-levels and univalence

One of Voevodsky's early core insights was that the homotopy theoretic notion of *truncation level* can be expressed using identity types.

Truncation levels

Let U be a type theoretic universe. We define the predicates $\mathsf{isTrunc}_n$ on U by induction on $n \geq -2$:

- $\mathsf{isContr}(A) \equiv \mathsf{isTrunc}_{-2}(A) \equiv \Sigma a{:}A\, \Pi b{:}A \,.\, \mathsf{Id}_A(a, b)$
- $\mathsf{isTrunc}_{(n+1)}(A) \equiv \Pi(a\, b : A) \,.\, \mathsf{isTrunc}_n(\mathsf{Id}_A(a, b))$

- (-1)-types are also called **propositions**, and 0-types **sets**

Equivalences and univalence

- Given $f : A \to B$ we define $\mathsf{isEquiv}(f) \equiv \Pi b \,.\, \mathsf{isContr}(\Sigma a \,.\, \mathsf{Id}(f\ a, b))$
- Given types A, B we define $\mathsf{Equiv}(A, B) \equiv \Sigma(f : A \to B) \,.\, \mathsf{isEquiv}(f)$
- Given types A, B we define $\mathsf{IdtoEquiv}(A, B) : \mathsf{Id}_U(A, B) \to \mathsf{Equiv}(A, B)$ by $\mathsf{IdtoEquiv}\, p\, a \equiv (a \triangleright_{\mathsf{el}} p)$, where el is the **universal type family** over U.
- The **univalence axiom** asserts that the map $\mathsf{IdtoEquiv}(A, B)$ itself is an equivalence.
  In other words, the notions of equality and equivalence coincide for types in U, or '*equivalent types are equal*'.

Synthetic homotopy theory in homotopy type theory

- The term **homotopy type theory (HoTT)** refers to a system of Martin-Löf type theory with *universes*, certain *(higher) inductive types*, and a constant for the *univalence axiom*.
- This system is powerful enough to express the core concepts, and to prove central results in homotopy theory *synthetically*, including
  - the *Freudenthal suspension theorem*[10],
  - the *Blakers–Massey theorem*[11],
  - the notion of *Eilenberg-MacLane spaces*[12], and
  - the *Serre spectral sequence*[13].
- The majority of this work has been **computer formalized** in one or more of the proof assistants *Agda*, *Coq*, and *Lean* that implement variants of MLTT.

[10] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study: https://homotopytypetheory.org/book, 2013.

[11] K. Hou (Favonia), E. Finster, D. Licata, and P. Lumsdaine. "A mechanization of the Blakers-Massey connectivity theorem in homotopy type theory". In: *Proceedings of the 31st Annual ACM-IEEE Symposium on Logic in Computer Science (LICS 2016)*. ACM, New York, 2016.

[12] D. Licata and E. Finster. "Eilenberg-MacLane spaces in homotopy type theory". In: *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. ACM, New York, 2014.

[13] F. van Doorn. "On the formalization of higher inductive types and synthetic homotopy theory". In: *PhD thesis arXiv:1808.10690* (2018).

Models of homotopy type theory

- The 'standard' model of HoTT is the model in simplicial sets[14].
- Recently, Shulman showed[15] that HoTT admits models in arbitrary ∞-**toposes**[16].
- Being the ∞-categorical generalization of Grothendieck's toposes, and ∞-toposes are ∞-categories of **sheaves** of homotopy types.
- Shulman's result means that HoTT can be used as an **internal language** in ∞-toposes, in the same way that higher order logic can be used as internal language in toposes[17].
- Results in pure HoTT hold in arbitrary ∞-toposes, to talk about specific ∞-toposes we need additional axioms and type theoretic principles.

---

14 K. Kapulkin and P. Lumsdaine. "The simplicial model of univalent foundations (after Voevodsky)". In: *Journal of the European Mathematical Society (JEMS)* (2021).

15 M. Shulman. "All $(\infty, 1)$-toposes have strict univalent universes". In: *arXiv preprint arXiv:1904.07004* (2019).

16 J. Lurie. *Higher topos theory*. Princeton University Press, 2009.

17 J. Lambek and P.J. Scott. *Introduction to higher order categorical logic*. Cambridge: Cambridge University Press, 1986.

Part III – Cohesive Homotopy Type Theory

Lawvere's axiomatic cohesion

- Lawvere introduced **axiomatic cohesion**[18] as a category theoretic axiomatization of the "Dialectic between continuous and discrete"
- Paradigmatic example: chain of adjoint functors between **sets** and **locally connected topological spaces**:

$$\mathsf{lcTop} \quad \downarrow \Pi_0 \;\dashv\; \uparrow \Delta \;\dashv\; \downarrow \Gamma \;\dashv\; \uparrow \nabla \quad \mathbf{Set}$$

Π_0 : connected components

Δ : discrete space on set

Γ : points

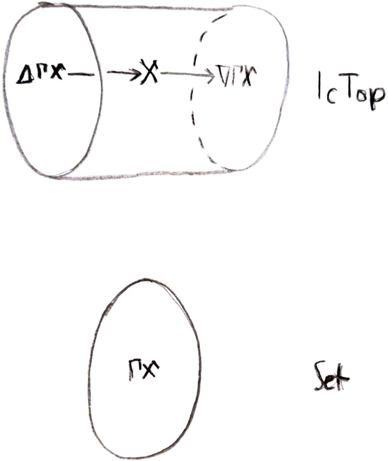
∇ : codiscrete space on set

- The following **axioms of cohesion** hold:
  1. Δ and ∇ are fully faithful
  2. Π_0 preserves finite products
  3. The **point-to-pieces** map $\Gamma X \overset{\cong}{\to} \Pi_z(\Delta(\Gamma X)) \to \Pi X$ is surjective for all $X \in \mathsf{lcTop}$ ('Nullstellensatz')
  4. The terminal projection $\Pi_0(\nabla S) \to 1$ is injective for all sets S ('connected codiscreteness')

[18] F.W. Lawvere. "Cohesive toposes and Cantor's "lauter Einsen"". In: Categories in the foundations of mathematics and language. 1994.

Adjoint cylinder

- The rightmost three adjoints $\Delta \dashv \Gamma \dashv \nabla$ constitute an **adjoint cylinder**
- The functors Δ and ∇ embed **Set** into lcTop in two extremal ways: discrete and codiscrete
- Every space X is 'suspended' between its 'discretization' and its 'codiscretization'



Cohesive toposes

Definition

A **cohesive topos** is a Grothendieck topos $\mathcal{E}$ whose global sections functor $\Gamma = \mathcal{E}(1, -) : \mathcal{E} \to \mathbf{Set}$ fits into a sequence $\Pi_0 \dashv \Delta \dashv \Gamma \dashv \nabla$ of adjoints satisfying the axioms of cohesion 1–4.

Examples

- The topos $\mathbf{SSet} = [\Delta^{\mathrm{op}}, \mathbf{Set}]$ of simplicial sets
- The topos $\mathbf{Sh}(\mathsf{Man})$ of sheaves on the category Man of second-countable topological manifolds with the open-cover Grothendieck topology

- The second example is very closely analogous to the example of locally connected topological spaces. In particular, $\mathbf{Sh}(\mathsf{Man})$ contains the category of Δ-generated spaces as a full subcategory.

Cohesive ∞-toposes

Around 2010 (??), Urs Schreiber and Michael Shulman realized that the cohesive framework works even better in the ∞-categorical setting:

Definition

A **cohesive ∞-topos** is an ∞-topos $\mathcal{E}$ whose global sections functor $\Gamma = \mathcal{E}(1, -) : \mathcal{E} \to \mathcal{S}$ fits into a sequence $\Pi \dashv \Delta \dashv \Gamma \dashv \nabla$ of adjoints satisfying the axioms of cohesion 1–4.

Remark

(Here $\mathcal{S}$ is the ∞-topos of homotopy types, a.k.a. 'spaces'.)

Examples

- The ∞-topos $[\Delta^{\mathrm{op}}, \mathcal{S}]$ of simplicial types
- The topos ∞-$\mathbf{Sh}(\mathsf{Man})$ of 'topological stacks', i.e. ∞-sheaves on Man

- Note that we dropped the subscript 0 on the left-most adjoint Π. This is because in the higher world, Π sends a cohesive/spatial objects $X \in \mathcal{E}$ not to their connected components, but to their **homotopy types**.

The cohesive ∞*-topos* $\infty\text{-}\mathbf{Sh}(\mathrm{Man})$ *of 'topological stacks'*

- Topological spaces embed into $\infty\text{-}\mathbf{Sh}(\mathrm{Man})$ in three ways:
  - Every space X gives a stack $\mathbf{Top}(J-, X)$ (its 'nerve'), where $J : \mathrm{Man} \to \mathbf{Top}$ is the inclusion
  - Every space X represents a homotopy type, i.e. an object of $\mathcal{S}$, which can be mapped into $\infty\text{-}\mathbf{Sh}(\mathrm{Man})$ via Δ and ∇
- We reserve the word 'space' for the 'cohesive' stacks obtained by the nerve construction, and therefore refer to the objects of $\mathcal{S}$ as *(homotopy) types*.
- There's also potential for confusion around the word *discrete*.
  For us, 'discrete' means 'cohesively discrete', i.e. in the image of Δ, rather than 0-truncated.

The modalities

- Schreiber and Shulman introduced names for the (co)monads induced by the adjunctions $\Pi \dashv \Delta \dashv \Gamma \dashv \nabla$:
    - $\int = \Delta \circ \Pi$ is the **shape modality**
    - $\flat = \Delta \circ \Gamma$ is the **flat modality**
    - $\sharp = \nabla \circ \Gamma$ is the **sharp modality**
    - $\int$ and $\sharp$ are idempotent *monads*, $\flat$ is an idempotent *comonad*
- Would like to use these modalities in type theoretic reasoning, i.e. use HoTT with modal operators as internal language of cohesive ∞-toposes.
- Ideally, one would like to introduce new type formers:

$$\frac{\Gamma \vdash A \text{ type}}{\Gamma \vdash \int A \text{ type}} \qquad \frac{\Gamma \vdash A \text{ type}}{\Gamma \vdash \flat A \text{ type}} \qquad \frac{\Gamma \vdash A \text{ type}}{\Gamma \vdash \sharp A \text{ type}}$$

- However, there's a problem: it's not clear what the meaning of $\flat$ should be in arbitrary context Γ: Given a map $A \to \Gamma$ in $\mathcal{E}$, there is no canonical way to 'vertically discretize' while keeping the same points.
- This means we can only allow to use $\flat$ in *discrete contexts*

Flat type theory

- To accommodate $\flat$ we introduce split contexts and consider judgments of the forms

$$\Delta \mid \Gamma \vdash A \ \mathsf{type} \qquad\qquad \Delta \mid \Gamma \vdash t : A,$$

  where Δ is the *discrete context* and Γ the *cohesive context*.
- Then the rules for the flat modality are then the following

$$\frac{\Delta \mid \cdot \vdash A \ \mathsf{type}}{\Delta \mid \cdot \vdash \flat A \ \mathsf{type}} \qquad\qquad \frac{\Delta \mid \cdot \vdash t : A}{\Delta \mid \cdot \vdash t^\flat : \flat A}$$

$$\frac{\Delta \mid x : \flat A, \Gamma \vdash C \ \mathsf{type} \qquad \Delta \mid \Gamma \vdash t : \flat A \qquad \Delta, y :: A \mid \Gamma \vdash u : C[y^\flat / x]}{\Delta \mid \Gamma \vdash (\mathsf{let}\ y := t \ \mathsf{in}\ u) : C[t/x]}$$

  i.e. we can only form $\flat A$ in a purely discrete context.
- Introduced by Shulman in[19], based on ideas in[20].

[19] M. Shulman. "Brouwer's fixed-point theorem in real-cohesive homotopy type theory". In: *Mathematical Structures in Computer Science* (2018).

[20] F. Pfenning and R. Davies. "A judgmental reconstruction of modal logic". English. In: *MSCS. Mathematical Structures in Computer Science* (2001).

Brouwer's fixed point theorem in cohesive HoTT

- Shulman introduced flat type theory (and more generally *spatial* type theory) in[21], to give an analysis of **Brouwer's fixed point theorem**.

Theorem (Brouwer)

Let $D^2 = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 \leq 1\}$. Every continuous $f : D^2 \to D^2$ has a fixed point.

- It's known that the fixed point can't be chosen 'effectively' in a way continuously depending on f.
- This non-effectivity can be made explicit in $\mathbf{Sh}(\mathsf{Man})$ as follows:

Theorem (Shulman)

The judgment

$$f :: D^2 \to D^2 \mid \cdot \vdash \exists x \,.\, f(x) = x$$

holds in $\mathbf{Sh}(\mathsf{Man})$, but the judgment

$$\cdot \mid f : D^2 \to D^2 \vdash \exists x \,.\, f(x) = x$$

does *not* hold in $\mathbf{Sh}(\mathsf{Man})$,

[21] M. Shulman. "Brouwer's fixed-point theorem in real-cohesive homotopy type theory". In: *Mathematical Structures in Computer Science* (2018).

Covering spaces in cohesive HoTT

- Given an object $A \in \infty\text{-}\mathbf{Sh}(\mathrm{Man})$, we can form its **fundamental groupoid**

$$\Pi_1(A) = \|\Pi(A)\|_1$$

  as 1-**truncation** of its homotopy type $\Pi(A) \in \mathcal{S}$.
- By the **fundamental theorem of covering spaces**, covering spaces on A correspond to maps

$$\Pi_1(A) \to \mathbf{Set} = \{S \in U_{\mathcal{S}} \mid S \text{ 0-truncated}\}.$$

- Since $\mathbf{Set}$ is 1-truncated we can drop the truncation and transpose to get an equivalent presentation as

$$A \to \Delta(\mathbf{Set}) = \flat\{B \in U \mid B \text{ discrete and 0-truncated}\}.$$

- Ideas along these lines are developed in[22], see also[23].

[22] F. Cherubini and E. Rijke. "Modal descent". English. In: *MSCS. Mathematical Structures in Computer Science* (2021).

[23] https://youtu.be/ACGjJDarEc4?t=2788

Outlook

- Differential cohesion introduces even more modalities
- Need for new type theories[24] and more modular/flexible proof assistants

[24] D. Gratzer, G.A. Kavvos, A. Nuyts, and L. Birkedal. "Multimodal dependent type theory". English. In: *Logical Methods in Computer Science* (2021). Id/No 11. URL: lmcs.episciences.org/7713.

Thanks for your attention!