Constructive mathematics in univalent type theory

Martín Hötzel Escardó

University of Birmingham, UK

Summer School Types, Sets and Constructions,
Hausdorff Research Institute for Mathematics, Bonn, May 2018

Plan

1. Peter Dybjer is explaining what Martin-Löf type theory is.
2. Simon Huber is explaining what certain univalent type theories are.
3. I will explain how to use them for doing (constructive) mathematics. (User manual.)
   - I take examples from my recent work. (Published and unpublished.)
   - I will re-introduce the univalent concepts and ingredients as they are needed (verbally).
   - I will also tell you why I like univalent mathematics and think univalent type theory is a viable language for (concrete and) abstract (constructive and non-constructive) mathematics.
   - I will show how to use natural, non-formal language to rigorously formulate theorems and proofs in type theory, just as one does in ZF-set based mathematics.
   - Additionally, I will compare the Bishop–Martin-Löf approach, the topos approach, and Voevodsky's univalent approach to mathematics, to put the latter in perspective.

Bishop considered a type theory, explained how to compile it to Algol, and said (early 1970's, unpublished):

> *"The possibility of such a compilation demonstrates the existence of a new type of programming language, one that contains theorems, proofs, quantifications, and implications, in addition to the more conventional facilities for specifying algorithms."*

1. As I said, I take examples of constructive mathematics from my recent work.
2. All definitions, theorems, constructions and proofs are implemented in Agda, without the use of third-party mathematical libraries, currently in a 15 500 - line development, which includes lots of accompanying English prose, just as any mathematical journal paper would.
3. I also discuss a separate Agda development by my former PhD student Chuangjie Xu.
4. Because Agda is such a new type of programming language, we can run the proofs and constructions, which is why I write them in Agda. I will illustrate this in practice.
5. As a bonus, the proofs are rigorously refereed by Agda for mathematical correctness.
6. Additionally, when I eventually use this development to write papers (or give these lectures), the results are nicely recorded in a precise, organized way, with clear dependencies, making my life much more manageable.

Preview - the ladder of univalent concepts and ingredients

1. Some results work in pure MLTT, but use univalent concepts, such as univalent propositions, univalent sets, the notion of equivalence ... (in the constructions and proofs).

2. Proceeding further, we often need functional extensionality as an ingredient, which is a consequence of the univalence axiom.

3. Further still, we sometimes need propositional extensionality as an ingredient, which also follows from univalence.

4. We also often need propositional truncation as an ingredient.

5. For doing set-level mathematics, functional and propositional extensionality, and propositional truncations, are enough ingredients.

   - A first example of a type which is not a univalent set is that of groups. It is a 1-groupoid.
   - The type of categories is a 2-groupoid.
   - It is in these kinds of examples that the univalence axiom plays a role.

Preview - some constructive theorems: constructing truncations

For any given function $f : \mathbb{N} \to \mathbb{N}$ and element $y : \mathbb{N}$, the fiber $\Sigma(x : \mathbb{N}), fx = y$ has a propositional truncation in pure MLTT.

1. Attempt: Just take the type of minimal solutions to $fx = y$.
   - Any two elements of this type are equal.
   - The proof uses the fact that $\mathbb{N}$ is a univalent set (Hedberg's Theorem).
   - But it also uses function extensionality and hence is not in pure MLTT.
2. Another construction builds a constant endomap on the fiber, that given any solution finds the minimal solution.
   - Then the set of fixed points of this map is the required proposition.
   - These subtle construction and proof, due to Nicolai Kraus, use many concepts from univalement mathematics, but no axiom of univalent type theory, so it works in pure MLTT.

In particular, we can construct the image $\Sigma(y : \mathbb{N}), \|\Sigma(x : \mathbb{N}), fx = y\|$ of the function f.

Preview - some constructive theorems: an omniscient set

Let $\mathbb{N}_\infty$ be the type of decreasing binary sequences.

- Then for any $p : \mathbb{N}_\infty \to \mathbb{N}$, it is decidable whether p has a root.
- Symbolically,

$$\Pi(p : \mathbb{N}_\infty \to \mathbb{N}), (\Sigma(x : \mathbb{N}_\infty), p(x) = 0) + \neg\Sigma(x : \mathbb{N}_\infty), p(n) = 0$$

  or, equivalently,

$$\Pi(p : \mathbb{N}_\infty \to \mathbb{N}), (\Sigma(x : \mathbb{N}_\infty), p(x) = 0) + \Pi(x : \mathbb{N}_\infty), p(n) \neq 0.$$

- This looks like LPO, which is undecided in MLTT (and is a taboo).
- This has many surprising corollaries.

  (For example, it is decidable whether a given function $f : \mathbb{N}_\infty \to \mathbb{N}$ is not continuous.)
- Function extensionality is a crucial ingredient here.

Preview - some constructive examples: totally separated reflection

Say that a type S separates the points of a type X if whenever two points have the same values for all functions $X \to S$, then they are equal:

$$\Pi(x, y : X), (\Pi(f : X \to S), fx = fy) \to x = y.$$

- Example: if S is the type Ω of univalent propositions, then the S-separated types are precisely the univalent sets.
- I will consider $S = 2$, the two-point type. Terminology: 2-Separated $=$ totally separated.
- Examples. Discrete types, Baire and Cantor types $\mathbb{N} \to \mathbb{N}$ and $\mathbb{N} \to 2$, all simple types e.g. $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$. Counter-examples. The reals (of any known kind), the type of categories.
- Any totally separated type is a univalent set.
- Theorem. Any type has a totally separated reflection.
- This uses functional extensionality and propositional truncation.

Preview - some constructive examples: total separation via apartness

Two points x and y of a type X are said to be 2-apart if they can be separated by some function into the two-point type:

$$(x \sharp_2 y) = \|\Sigma(f : X \to 2), fx \neq fy\|.$$

1. This is an apartness relation $X \times X \to \Omega$:
   - $\neg(x \sharp_2 x)$,
   - $x \sharp_2 y \to y \sharp_2 x$,
   - $x \sharp_2 y \to \|(x \sharp_2 z) + (y \sharp_2 z)\|$.
2. It is tight,
   - $\neg(x \sharp_2 y) \to x = y$,

   if and only if X is totally separated.

Preview - some constructive examples: tight reflection

Any apartness type $(X, \sharp)$ has a tight reflection, where the morphisms of apartness spaces are the strongly extensional maps (those that reflect apartness):

1. We can construct
   - a tight apartness type $(TX, \sharp)$ and a strongly extensional map $\eta : X \to TX$,

   such that for any given
   - tight apartness type $(Y, \sharp)$ and strongly extensional map $f : X \to Y$,

   we can find a unique $\overline{f} : TX \to Y$ extending f along η.
2. Unique existence has to be expressed in the univalent way, meaning that not only f is unique, but also the data that explains the extension:
$$\mathrm{isSingleton}(\Sigma(\overline{f} : TX \to Y), \overline{f} \circ \eta = f).$$
3. If you recall Simon's lectures, this amounts to saying we have an equivalence
$$(g \mapsto g \circ \eta) : (TX \to Y) \to (X \to Y)$$
4. This needs functional and propositional extensionality, and propositional truncations.

Preview - some constructive examples: Injective types

Are there more infinite types like $\mathbb{N}_\infty$ that allow "exhaustive search"? This led me to consider injective types as a tool to build them.

1. A type A is injective if for every embedding $e : X \to Y$, any map $f : X \to A$ extends to a map $f' : Y \to A$ along e.
   - To be an embedding means that the fibers $\Sigma(x : X), ex = y$ are all subsingletons.
   - Should we say that the extension exists (truncated Σ) or that we can find one (just Σ)?

     We choose "find" (which corresponds to "internally injective" in a cartesian closed category).
2. Theorem ∞. The injective types are precisely the retracts of the exponential powers $\mathcal{U}^X = (X \to \mathcal{U})$ of universes.

   The construction and proof use full univalence (twice).

   Theorem 0. The injective sets are precisely the retracts of exponential powers of Ω.

   Theorem 1. The injective 1-groupoids are precisely the retracts of exponential powers of universes $\Sigma(X : \mathcal{U}), \mathrm{isSet}(X)$ of sets.

   Theorem $n+1$. The injective $n+1$-groupoids are precisely the retracts of the exponential powers of the universes of n-groupoids.

Preview - some constructive examples: All functions are continuous

Theorem of (intensional) spartan Martin-Löf type theory without universes

If all functions $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ are continuous then $0 = 1$.

$$\big(\Pi(f\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}),\ \Pi(\alpha : \mathbb{N}^{\mathbb{N}}),\ \Sigma(n : \mathbb{N}),\ \Pi(\beta : \mathbb{N}^{\mathbb{N}}),\ \alpha =_n \beta \to f\alpha = f\beta\big) \to 0 = 1.$$

However, it is consistent that all functions are continuous in the following modified sense:

$$\Pi(f\colon \mathbb{N}^{\mathbb{N}} \to \mathbb{N}),\ \Pi(\alpha : \mathbb{N}^{\mathbb{N}}),\ \|\Sigma(n : \mathbb{N}),\ \Pi(\beta : \mathbb{N}^{\mathbb{N}}),\ \alpha =_n \beta \to f\alpha = f\beta\|.$$

A model is Johnstone's topological topos.

Preview - some constructive examples: Uniform continuity question

The topological topos validates

$$\Pi(f\colon 2^{\mathbb{N}} \to \mathbb{N}), \|\Sigma(n : \mathbb{N}),\ \Pi(\alpha, \beta : 2^{\mathbb{N}}),\ \alpha =_n \beta \implies f\alpha = f\beta\|.$$

But what about the untruncated version?

$$\Pi(f\colon 2^{\mathbb{N}} \to \mathbb{N}), \Sigma(n : \mathbb{N}),\ \Pi(\alpha, \beta : 2^{\mathbb{N}}),\ \alpha =_n \beta \implies f\alpha = f\beta.$$

What motivates this question is that the Cantor space is compact.

We need some preparation regarding "exiting truncations".

Preview - some constructive examples: Exiting truncations I

The elimination rule is $(X \to P) \to (\|X\| \to P)$ for subsingleton P.

We can disclose a secret $\|X\|$ to P provided we have a map $X \to P$.

Example. If $A(n)$ is decidable and subsingleton valued, then

$$\|\Sigma(n : \mathbb{N}),\ A(n)\| \to \Sigma(n : \mathbb{N}),\ A(n).$$

If there exists some $n : \mathbb{N}_\infty$ with $A(n)$, then we can exhibit one.

Proof sketch. If we have any n with $A(n)$, we can find the minimal n, using the decidability of $A(n)$, but "having a minimal n such that $A(n)$" is a subsingleton.

Preview - some constructive examples: Exiting truncations II

Assume that $A(n)$ is a subsingleton for every $n : \mathbb{N}$.

If for any given n we have that $A(n)$ implies that $A(m)$ is decidable for all $m < n$, then

$$\|\Sigma(n : \mathbb{N}),\ A(n)\| \to \Sigma(n : \mathbb{N}),\ A(n).$$

Preview - some constructive examples: Uniform continuity answer

Theorem

$$\Pi(f\colon 2^{\mathbb{N}} \to \mathbb{N}),\ \big\|\Sigma(n:\mathbb{N}),\ \Pi(\alpha,\beta:2^{\mathbb{N}}),\ \alpha =_n \beta \implies f\alpha = f\beta\big\| \\ \to \Sigma(n:\mathbb{N}),\ \Pi(\alpha,\beta:2^{\mathbb{N}}),\ \alpha =_n \beta \implies f\alpha = f\beta.$$

Proof. Set $A(n) = \big(\Pi(\alpha,\beta:2^{\mathbb{N}}),\ \alpha =_n \beta \implies f\alpha = f\beta\big)$ in the lemma.

Corollary. The topological topos also validates the uniform-continuity principle

$$\Pi(f\colon 2^{\mathbb{N}} \to \mathbb{N}),\ \Sigma(n:\mathbb{N}),\ \Pi(\alpha,\beta:2^{\mathbb{N}}),\ \alpha =_n \beta \implies f\alpha = f\beta.$$

Because the premise of the theorem is validated.

(In the topological topos, the theorem can be seen as getting global existence from local existence by compactness.)

Preview - some constructive examples: Chuangjie Xu's work

1. A constructively defined sheaf model that validates uniform continuity.
2. Also implemented in Agda with univalent concepts and ingredients.
3. Ported to cubical Agda.
4. Can compute moduli of uniform continuity using this implementation of the model.
5. We could not handle universes.
6. But Thierry Coquand and his collaborators have shown how to extend this to universes, by replacing sheafs by stacks (a concept developed in the 1960's by the Grothedieck School).

Next

1. Construct and prove some of the above in order to show how univalent type theory works in practice.
2. But first pause to compare this to other languages for constructive mathematics.
   - Elementary-topos type theory.
   - Spartan MLTT.
   - Spartan univalent type theory.

Elementary-topos type theory (Lambek and Scott 1986)

Internal language of the free elementary topos with NNO.

They also call it "intuitionistic type theory".

1. Simply typed λ-calculus with finite product types with type $\mathbb{N}$ of natural numbers.
2. Type Ω of truth values (corresponding to subsingletons of 1).
3. Functions $(=_X) : X \times X \to \Omega$ for each type/object X.
4. Functions $(\wedge), (\vee), (\implies) : \Omega \times \Omega \to \Omega$.
5. Functions $\forall_X, \exists_X : (X \to \Omega) \to \Omega$ for each X.
6. Axioms (for equality, function and propositional extensionality, induction, ...).
7. Intuitionistic deductions rules.
8. Has the existence and disjunction properties.

   In the *empty context*, $\vee$ and $\exists$ behave like $+$ and Σ in MLTT.

- Although toposes have Π and Σ, this type theory doesn't include them.
- Can state property directly, but need to give structure indirectly.

Intensional Martin-Löf type theory (a spartan one here)

1. Dependently typed λ-calculus with Π, Σ, Id, $+$, 0, 1, $\mathbb{N}$, $\mathcal{U}$.
2. No axioms. Rules to derive types, contexts, terms.
3. Need to add definitional (or judgmental) equality.

   Doesn't occur in terms.

   Can only be written when it holds.

   Plays a role only in derivation rules.
4. All types, not just subsingletons, are considered to be propositions.
5. And is $\times$, or is $+$, implies is $\to$, for all is Π, and exists is Σ.
6. Lacks function extensionality (and propositional extensionality doesn't make sense).

- Property and structure (or data) are conflated.
- Use setoids to collapse structure to property.
- The universe allows to define types of mathematical structures (e.g. the type of groups).
- Its identity type is underspecified (compatible with both K and UA).

Example: Dedekind reals in MLTT

A Dedekind real is a pair of functions $l, u : \mathbb{Q} \to \mathcal{U}$ together with some data, including

1. A function $\Pi(q : \mathbb{Q}), l(q) \to \Sigma(r : \mathbb{Q}), (q < r) \times l(r)$.

   This function, given any rational in the lower section, picks a bigger rational in the lower section.

   (Which rational number is picked?)

2. A function $\Pi(p, q), p < q \to l(p) + u(q)$.

   This function, given rational numbers $p < q$, decides which of $l(p)$ or $u(q)$ holds.

   What does it answer when both hold?

- E.g. the number π is represented by (infinitely) many different Dedekind sections.
- Need to work with an equivalence relation.
- No quotients, hence work with setoids (type $\&$ equivalence relation).

(Like Bishop proposed, although he worked with Cantor reals.)

Example: image

For $f : X \to Y$,

1. The candidate for the image is $\Sigma(y : Y), \Sigma(x : X), \mathrm{Id}(f(x), y)$.

   This is the type of $y : Y$ for which there is some x mapped to a point identified with y.

2. But this is in bijection with X.

   Again need an equivalence relation on this type, identifying (y, x, p) with (y, x', p')

3. Moreover, we don't actually work with $\mathrm{Id}(f(x), y)$.

   Instead we have that X and Y already are setoids, $f : X \to Y$ preserves the equivalence relation, and the image has underlying type $\Sigma(y : Y), \Sigma(x : X), f(x) \sim y$ with equivalence relation defined by

   $$((y, x, p) \sim (y', x', p')) = (y \sim y').$$

The identity type is hardly used when working with setoids.

Problems with setoids

1. Practical one (nicknamed "setoid hell").
   Incredible amount of bookkeeping is needed in rigorous proofs.
   (Bishop simply ignores the bookkeeping in his book.)

2. It doesn't seem to be possible to handle the universe as a setoid.

Univalent type theory (again a spartan one)

1. Spartan MLTT + propositional truncation + univalence.
2. Univalent propositions taken to be subsingletons.

   $\mathrm{isProp}(X) = \Pi(x, y : X), \mathrm{Id}(x, y)$.

   $\Omega = \Sigma(P : \mathcal{U}), \mathrm{isProp}(P)$.
3. Get functional and propositional extensionality.
4. The propositional truncation of X is the universal solution $|-| : X \to \|X\|$ to the problem of mapping X to a subsingleton:

   If $X \to P$ then $\|X\| \to P$.
5. $P \vee Q = \|P + Q\|$.

   $\exists(x : X), P(x) = \|\Sigma(x : X), P(x)\|$.

cubicaltt is an example of constructive univalent type theory:

1. Univalence is a theorem.
2. Has the canocity property. (Which gives the disjunction and existence properties.)

Examples revisited

1. The image of $f : X \to Y$ is $\Sigma(y : Y), \|\Sigma(x : X), \mathrm{Id}(f(x), y)\|$.

   We get the correct equality without the need of taking setoid-quotients.

2. A Dedekind real is a pair of functions $l, u : \mathbb{Q} \to \Omega$ together with properties, including

   2.1 A function $\Pi(q : \mathbb{Q}), l(q) \to \|\Sigma(r : \mathbb{Q}), (q < r) \times l(r)\|$.

   2.2 A function $\Pi(p, q), p < q \to \|l(p) + u(q)\|$.

   - Such functions, as well as the ones we have omitted, are unique if they exist.
   - Moreover, the identity type gives the correct notion of equality without quotienting.
   - This relies on functional and propositional extensionality (which follow from univalence).

In univalent mathematics, the identity types "absorb" the setoid machinery and functions automatically preserve identifications.

Univalent TT $\approx$ MLTT + Topos TT + ∞

1. MLTT favours data (Σ).
2. Topos TT favours property ($\exists$).
3. UTT incorporates both (Σ and $\exists$)

   More meaningful distinctions become possible.

In his "The formulae-as-types notion of contruction", Howard actually discusses two notions of existence.

Perhaps univalent logic is the true Curry–Howard logic.

Next

I will prove some of the above claims to illustrate how (univalent) type theory feels like in practice for a working (constructive) mathematician.

1. Searchability of $\mathbb{N}_\infty$.
2. Totally separated reflection or maybe tight reflection.
3. Injective types.