## Background of my work

Homotopy type theory is a new branch of mathematics that combines aspects of several different fields in a surprising way. It is based on a recently discovered connection between homotopy theory and type theory. Homotopy theory is an outgrowth of algebraic topology and homological algebra, with relationships to higher category theory; while type theory is a branch of mathematical logic and theoretical computer science. Although the connections between the two are currently the focus of intense investigation, it is increasingly clear that they are just the beginning of a subject that will take more time and more hard work to fully understand. It touches on topics as seemingly distant as the homotopy groups of spheres, the algorithms for type checking,and the definition of weak -groupoids. Homotopy type theory also brings new ideas into the very foundation of mathematics. Ont he one hand, there is Voevodsky’s subtle and beautiful univalence axiom. The univalence axiom implies, in particular, that isomorphic structures can be identified, a principle that mathematicians have been happily using on workdays, despite its incompatibility with the “official” doctrines of conventional foundations. On the other hand, we have higher inductive types, which provide direct, logical descriptions of some of the basic spaces and constructions of homotopy theory: spheres, cylinders, truncations, localizations, etc. Both ideas are impossible to capture directly in classical set-theoretic foundations, but when combined in homotopy type theory, they permit an entirely new kind of “logic of homotopy types”.

## What I have done

My work aims to bring Homotopy Type Theory to the foundations of mathematical analysis, and implement the foundations via a programming language for further use. Now I us Agda language, but I'm considering to switch to Arend someday. HoTT suggests a new conception of foundations of mathematics, with intrinsic homotopical content, an “invariant” conception of the objects of mathematics — and convenient machine implementations. My work is just the the theory and new foundations of mathematical analysis.

On the first part of my paper, I develop a kind of type theory for the use of mathematical analysis. I represent the main ideas of set theory as the equivalent parts of type theory, providing basic tools for dealing with measure theory, infinite small and limit in mathematical analysis.

In set-theoretic foundations, at various points in homotopy theory and category theory one needs the axiom of choice to perform transfinite constructions. But with higher inductive types, we can encode these constructions directly and constructively.

In set theory, various circumlocutions are required to obtain notions of “cardinal number” and “ordinal number” which canonically represent isomorphism classes of sets and well-ordered sets, respectively — possibly involving the axiom of choice or the axiom of foundation. But with univalence and higher inductive types, we can obtain such representatives directly by truncating the universe. In univalent foundations, the basic objects are “homotopy types” rather than sets, but we can define a class of types which behave like sets. Homotopically, these can be thought of as spaces in which every connected component is contractible, that's to say, those which are homotopy equivalent to a discrete space. It is a theorem that the category of such “sets” satisfies Lawvere’s axioms (or related ones, depending on the details of the theory). Thus, any sort of mathematics that can be represented in an ETCS-like theory (which, experience suggests, is essentially all of mathematics) can equally well be represented in univalent foundations. I also category-theorize some theorems in set theory, for example, the Cantor-Schröder-Bernstein theorem, which states that for any pair of sets, if there is an injection of each one into the other, then the two sets are in bijection. I also provide a Agda proof to implement this result.

More specifically, consider on one hand the axiom of choice: “if for every x : A there exists a y : B such that R(x, y) , there is a function f : A → B such that for all x : A we have R ( x, f ( x)).” The pure propositions-as-types notion of “there exists” is strong enough to make this statement simply provable — yet it does not have all the consequences of the usual axiom of choice. However, in (−1)-truncated logic, this statement is not automatically true, but is a strong assumption with the same sorts of consequences as its counterpart in classical set theory. On the other hand, consider the law of excluded middle: “for all A, either A or not A.” Interpreting this in the pure propositions-as-types logic yields a statement that is inconsistent with the univalence axiom. For since proving “A” means exhibiting an element of it, this assumption would give a uniform way of selecting an element from every nonempty type — a sort of Hilbertian choice operator. Univalence implies that the element of A selected by such a choice operator must be invariant under all self-equivalences of A, since these are identified with self-identities and every operation must respect identity; but clearly some types have automorphisms with no fixed points, e.g. we can swap the elements of a two-element type. However, the “(−1)-truncated law of excluded middle”, though also not automatically true, may consistently be assumed with most of the same consequences as in classical mathematics.

Then, I seek to represent natural number, rational number and real number in terms of HoTT. There is another basic issue: the difficulty of working with types, such as the natural numbers, that are essentially sets (i.e., discrete spaces), containing only trivial paths. At present, homotopy type theory can really only characterize spaces up to homotopy equivalence, which means that these “discrete spaces” may only be homotopy equivalent to discrete spaces. Type-theoretically, this means there are many paths that are equal to reflexivity, but not judgmentally equal to it. While this homotopy-invariance has advantages, these “meaningless” identity terms do introduce needless complications into arguments and constructions, so it would be convenient to have a systematic way of eliminating or collapsing them. This problem is not solved yet, so I'm stuck here.

There many ways to deal with this problem. Think more generally, homotopy type theory should be the “internal language” of -toposes, just as intuitionistic higher-order logic is the internal language of ordinary 1-toposes. Despite this general consensus, however, details remain to be worked out.