Formalization of Cubical Type Theory in Nuprl

Mark Bickford, Cornell Computer Science

January 15, 2020

Abstract

We report on the results of a two year project to formalize the semantics of *cubical type theory* in Nuprl. We were able to interpret cubical type theory (without higher inductive types) in Nuprl's extensional constructive type theory, thus verifying that a univalent higher dimensional type theory can be given a purely constructive interpretation. The *univalence axiom* is provable in cubical type theory from the rules for the *Glue type*. Verifying the rules for the Glue type led to the largest proofs ever done in Nuprl. We also began an investigation into how to relate synthetic higher dimensional type theory with constructive analysis and discovered that the intuitionistic theory of analysis (using the FAN theorem and the continuity principle) makes building a connection between the two theories much easier.

1 Univalence

There is great interest in Vladimir Voevodsky's *Univalent Foundations* because it could provide a sufficiently abstract formal foundation that mathematicians can actually use, while, at the same time being sufficiently concrete so that proofs written using it can be verified by a computer. Voevodsky's major insight was that a *type theory* that satisfied the *univalence axiom* could be defined. Essentially, the univalence axiom says that when we can prove that there is a bijection between types A and B then we can also prove that A and B are *equal* in the universe of types. This kind of type equality is closer to normal mathematical practice than the more restrictive type equality used in the theories implemented in Nuprl or Coq.

Voevodsky gave a semantics for a univalent type theory using *Kan simplicial sets* but his proofs made use of a strong form of the axiom of choice. It was an open question whether a univalent type theory could be given a purely constructive interpretation. Such an interpretation is desirable because then proofs in the theory would generate programs. In particular, once an integer bound or invariant is proved to exist, it could be computed.

2 Goals of the project

The *cubical type theory* defined by Coquand and colleagues is a theory that has many of the features desired for Univalent Foundations and that has a constructive interpretation. The interpretation was given using an informal constructive set theory.

The first major goal of this project was to build a formal, constructive model of *cubical type theory* in the constructive type theory of the Nuprl proof system and verify that all the rules of cubical type theory hold in this model.

The second major goal of the project was to investigate how cubical type theory and Nuprl's type theory can coexist and collaborate within a new proof assistant – a *next generation* proof assistant.

Since Nuprl has an extensive formalization of *constructive analysis*, a third goal of the project was to make some connections between the concepts of topological spaces, homotopies, homotopy groups, etc. as defined synthetically in cubical type theory and the corresponding concepts defined analytically in constructive analysis.

3 Results on the first goal: Formalization

Most of the technical work needed to build the formalization of cubical type theory in Nuprl has been accomplished and is published on the Nuprl website at `www.nuprl.org/wip/Mathematics/cubical!type!theory/index.html`

To do this, several "background" theories had to be formalized first, in particular, category theory and theories of free distributive lattices. Some of this had been done earlier for a formalization of a precursor to cubical type theory, but we have since added more to these background theories.

The crucial concept, and technical hurdle for building a constructive model of Voevodsky's *Univalence Axiom*, is the notion of a *fibrant* type that is a type with a *composition structure*. Types are analogous with (topological) spaces, and the composition structure allows, among other things, paths in the space to be composed, and this is the basis for the properties of equality in the theory. To formalize cubical type theory we must give the constructive realizer (i.e. the program) for composition operations in each type. Cubical type theory has basic types like numbers, and the Pi and Sigma types for functions and pairs. For each type, T, there is also a type Path(T) corresponding to equality in T. In addition to these, cubical type theory adds "System" types and, crucially, "Glue" types. Members of the System type are constructed from lower-dimensional "faces" that are compatible where they overlap. The Glue type is a major innovation of cubical type theory. Members of the Glue type carry the data needed to "glue" equivalent things together, and its composition is used to constructively validate the univalence axiom.

When we wrote our proposal for this work we had proved one major technical lemma about composition operators. Since then we have proved that all the composition operators needed for cubical type theory exist. The composition for the Glue type is, in fact, the most technically difficult part of the whole development. The formal proof that the Glue composition operation has the needed properties is the single largest proof in the entire Nuprl library and took many weeks to construct. Even to automatically recheck that proof takes Nuprl two hours.

The realizer for the Glue composition operation is a Nuprl program that can be written out on one slide using the realizers of other theorems. We can make a self-contained program that includes all the other needed code by asking Nuprl to translate the realizer into LISP. When we do that, the code for the Glue composition operator is 907 lines of LISP.

It is important that we actually synthesize the programs that realize the constructive model of cubical type theory. Here is an interesting example of what could be done with such programs. In his 2016 thesis Guillaume Brunerie used homotopy type theory (as in the HoTT book) to prove that $\pi_4(S^3) \simeq \mathbb{Z}/2\mathbb{Z}$. He had earlier proved that there exists a natural number n such that $\pi_4(S^3) \simeq \mathbb{Z}/n\mathbb{Z}$ but because the implementation of homotopy type theory he used did not have a fully constructive interpretation, he could not simply use the proof of that theorem to compute n and find that $n = 2$. Instead, he had to do many more formal proofs to establish that fact. Once we have a full implementation of cubical type theory that includes the *higher inductive types* needed for Brunerie's first proof, we will be able to compute such numbers and verify $n = 2$ by computation.

Cubical type theory also has a *universe* $\mathbb{U}$ of (fibrant) types. Two types A and B are equal if there is a path between them in $\mathbb{U}$. Using the Pi and Sigma types, Voevodsky defines $Equiv(A, B)$ the type of equivalences between A and B (essentially bijections between A and B). *Univalence* asserts that an equivalence between A and B corresponds to a path in $\mathbb{U}$ between A and B, and, in fact, this correspondence is itself an equivalence

$$Equiv(Path_{\mathbb{U}}\ A\ B, Equiv(A, B))$$

In cubical type theory, univalence is not an axiom but a theorem, derivable from the other rules, in particular, the rules for the Glue type. In order to verify univalence and other rules about the universe $\mathbb{U}$ we must give the formal definition of a universe of fibrant types and define the composition structure for the universe. In the Nuprl formalization we were able to define an infinite hierarchy of fibrant universes, $\mathbb{U},\ i = 0, 1, 2, \ldots$. Each $\mathbb{U}_i$ has a composition structure, making it fibrant, so we have $\mathbb{U}_i \in \mathbb{U}_{i+1}$.

Proving the existence of a composition operation for $\mathbb{U}_i$ is another challenging part of the theory. The composition operator for Glue can be used to build a composition operator for the universe and we have completed that construction and proof. The program for that composition operator adds an additional 300 lines of LISP code.

3.1 Two discoveries

The formalization of cubicalTT in Nuprl presented many technical challenges requiring us to develop new tactics and prove over one thousand lemmas. We highlight here two discoveries that we made during the formalization, that were unexpected before we did the work.

The first discovery concerns the use of *nominal logic*. Nominal logic introduces a primitive concept of *names* and has new rules about formulas that mention names. Set theory with urelements is often used to give a semantics for nominal logic. In Nuprl, we had already introduced a type of *unguessable atoms* and given a *super-valuation* semantics for these atoms. The rules for atoms make Nuprl a nominal logic.

To construct a path between two terms a and b in cubical type theory, one can introduce a new dimension i and construct a term p where $p = a$, when $i = 0$ and and $p = b$ when $i = 1$. The *path abstraction* operator $\langle i \rangle p$ binds i and is a path between a and p that is independent of the dimension i. The first work on a constructive interpretation for cubical type theory used names to represent formal dimensions, and this use of nominal logic seemed to be essential to the formal semantics.

We initially used Nuprl's atoms in the formalization of path types and path abstraction. However, as we proceeded with the formalization we realized that the type $Path_A\ a\ b$ could be defined as a subtype of the *free path space*, $\mathbb{I} \to A$ where $\mathbb{I}$ is the formal interval type in cubicalTT. Using this definition we did not have to use either atoms or quotient types, and this gave us a simpler semantics. In the end, all use of nominal logic disappeared, showing that nominal logic is not, in fact, needed to formalize cubical type theory.

The second discovery concerns the formal definition of fibrant types. Bezem, Coquand, and Huber, discovered that a constructive model of univalence needs a stronger invariant than Voevodsky's model based on Kan simplicial sets. Fibrant cubical types must have *uniform* composition operations. The uniformity condition is expressed by quantifying over all cubical sets, and this is problematic. In type theory, we can not quantify over *all* sets (i.e types), we can only quantify over all types is a given universe, say the n^{th} universe. The resulting proposition is then a member of the $(n+1)^{th}$ universe. But then, the definition of a universe of fibrant types becomes impredicative since if we put into the universe all the fibrant cubical types in Nuprl universe n, the uniformity condition that defines which types are fibrant also quantifies over universe n (and such impredicative definitions can not be shown to be well formed.)

To solve this problem, Coquand was able to express the needed uniformity condition by quantifying over only *representable* cubical sets, not all cubical sets (the representable cubical sets come from the Yoneda embedding of the base category of formal cubes.) The drawback of this approach was that the reasoning about composition operators becomes more difficult. As we proceeded with the formalization, we discovered that we could prove that the two definitions of uniform composition operators were, in fact, equivalent. This means that even though the "nice" definition is in a higher universe, it has a realizer if and only if the "representable" version has a realizer. Therefore we can use the "representable" definition to define the universe of fibrant types, but then reason about them using the "nice" definition.

This pleasant phenomenon was unexpected, but welcome. Coquand has since found a more categorical explanation for this phenomenon and has written a paper on the topic and lectured about it.

4 Results on second goal: Combining cubicalTT and Nuprl

Since we have a formal semantics of cubicalTT in Nuprl, the meaning of a theorem proved in cubcallTT is a theorem of Nuprl. But we want a closer connection between the two type theories. It would be especially nice if cubicalTT could be seen as a conservative extension of Nuprl, but that can not be exactly true because Univalence is not true for the Nuprl types.

However, we made some progress relating the two type theories. For any Nuprl type A there is a cubical type $discrete(A)$ in which all paths are constant. The cubical type of natural numbers is just $discrete(\mathbb{N})$ where $\mathbb{N}$ is the

Nuprl natural numbers. For any Nuprl type family $B \in A \rightarrow Type$ there is a cubicalTT family, $discrete(B)$ over $discrete(A)$, and we can form the cubical Pi and Sigma types $\Pi(discrete(A)\ discrete(B)$ and $\Sigma(discrete(A)\ discrete(B)$. We have proved that there are bijections between these types and the discrete versions of the corresponding Nuprl dependent function and dependent pair types. So, for example,

$$\Pi(discrete(A)\ discrete(B) \sim discrete(a : A \rightarrow B(a))$$

As we mentioned, any path p from a to b in $discrete(A)$ is equal to the constant path refl(a), and therefore $discrete(A)$ satisfies the *uniqueness of identity proofs* (UIP). We would like to show that the Nuprl types can be embedded in the cubical types as exactly those types for which every path is refl. Since the cubical universe $\mathbb{U}$ does not have this property (a consequence of Univalence), it is not the discrete version of the Nuprl universe. So in a combined Nuprl and cubicalTT theory there will be two kinds of universes. The cubical universe of fibrant types and the universe of discrete types.

5 Results on third goal: relating cubicalTT and constructive analysis

This third goal was not one of the initial goals of the project but there is interest in this topic because Univalent Foundations has been proposed as a theory in which such seemingly disparate disciplines as quantum physics and computer science can share a common foundation. To understand how results proved in a formal *synthetic* theory of higher-dimensional types can be relevant to the real world of physics, we need a better understanding of how some of the abstract concepts of homotopy type theory relate to the continuous mathematics of the real and complex numbers.

To this end, we studied the work of Mike Shulman on *real-cohesive homotopy type theory* that attempts to connect homotopy type theory with real analysis. He introduced some new modalities to control the topology of the space associated with a type and introduce the notion of a *crisp* variable. Using this rather complex infrastructure he derives a weak version of Brouwer's fixedpoint theorem from results proved in homotopy type theory.

One of Shulman's axioms about the real numbers stated that if sets A and B cover $\mathbb{R}$ and are *crisp* then their intersection $A \cap B$ is non-empty. We wondered how the concept of *crisp* could be interpreted to make this axiom true in Nuprl. We made the somewhat startling discovery that in *intuitionistic* mathematics, as implemented in Nuprl, this axiom is true with no restrictions on A and B (that is, *crisp* can be interpreted, trivially, as true). This discovery, formally proved in Nuprl, follows from the continuity principle for numbers that is a fundamental fact of intuitionistic mathematics and is true of Nuprl's type theory. This result was published in Mathematical Logic Quarterly in the article *Connectedness of the continuum in intuitionistic mathematics.*

Based on this result, we think that the relation between homotopy type theory and analysis will work out much more easily in intuitionistic mathematics than in classical analysis or in Bishop-style constructive analysis. A fundamental theorem of Brouwer's, proved in Nuprl using intuitionistic mathematics, is that all functions from a compact metric space X to a metric space Y are uniformly continuous. This fact makes some of the machinery used by Shulman superfluous so that a simpler connection between homotopy type theory and intuitionistic analysis should be possible.

We think that such a connection will be based on some other fundamental results of Brouwer, namely his *simplicial approximation theorem*. In the semantics of cubical type theory, however, *simplicial* sets have been replaced by *cubical* sets in order to overcome some problems that would otherwise require the axiom of choice. So it is likely that a *cubical* version of Brouwer's approximation theorem will be more useful.

To begin building such a constructive theory we formalized the theory of *cubical complexes* in Nuprl and used this theory to formalize a new, cubical, version of a proof by Karol Sieklucki (1983) of the *no retraction theorem*. It is a little-known fact that the no retraction theorem constructively implies the approximate version of Brouwer's fixedpoint theorem (for every function f from a unit ball to itself, and for every $\epsilon > 0$ there is a point x for which the distance from x to $f(x)$ is less than ϵ).

We thereby gave a complete formal proof of (the approximate) Brouwer's fixedpoint theorem in Nuprl. This theorem is weaker than the classical version because we can find only an approximate fixedpoint for any ϵ (in classical analysis, using the Bolzano-Weirstrauss theorem, we could then get the existence of an exact fixedpoint, but this is not true constructively). But our fixedpoint theorem is stronger than the classical or Bishop-style theorem because it holds for all functions on the unit ball (in any finite dimension n) rather than just the continuous functions.

6 Further work

There are a few parts of a full interpretation of Univalent Foundations that remain to be formalized. In particular, the theory of *higher inductive types*.

The paper by Cohen, Coquand, and Mortberg on cubicalTT does not cover a general form of higher inductive types, but it does sketch how higher inductive types for n-spheres and also for *truncation* types can be done.

Eventually, we want to complete of higher inductive types in Nuprl, if not the general case then at least the constructions of spheres and truncation types. When those are done we can say that the current cubicalTT is fully formalized in Nuprl.

A complete semantics for cubicalTT does not immediately give us a version of cubicalTT that we can use, inside Nuprl, to easily construct proofs. The reason is that the semantics is expressed in a "name-free" form. A sequent like x:Nat, i:I, y:A ⊢ B(x,i,y) mentions the names x,i, and y. In the semantics the context x:Nat, i:I, y:A is represented by a product Nat.I.A with no names, so the expression B(x,i,y) must be represented by B(p(p(q)), p(q), q) where q returns the last element of a context and p returns the part of the context with the last element removed. Thus, we can think of q as the number 0 and p(n) = n+1. Then expressions like p(p(q)) are *DeBruijn indices* that refer to parts of the name-free context. This name-free representation is convenient for proving the rules of cubicalTT from the semantics, but it is inconvenient for carrying out proofs in the theory.

Therefore we will build a syntactical version of cubicalTT that uses names. Defining the meaning of the syntactic version and proving its rules will require reasoning about substitution and alpha-equality of syntactic terms. Once that work has been done we will have a version we can use to carry out proofs in cubicalTT in the intended way. That version will be easier to integrate with Nuprl.

7 Dissemination of Results

Once we completed the full formal model of cubicalTT (without the higher inductive types) we wrote an arXiv paper on the formalization of presheaf models of type theory in Nuprl. We also lectured on the formalization of cubicallTT in Nuprl is a course in Germany. In the audience were most of the experts in cubical type theore including Thierry Coquand and Simon Huber. So, it is now well-known that a fully constructive interpretation of cubical type theory has been constructed in Nuprl. Also, Coquand and Mortberg have been giving talks about cubicalTT in conferences and workshops, and these talks include discussion of the Nuprl formal model and the realizers we have constructed for, e.g. composition for the Glue type. As a result, other authors are citing our work, using the formal content published on the Nuprl website as the reference.