МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт математики, механики и компьютерных наук
имени И. И. Воровича
Кафедра информатики и вычислительного эксперимента

СТЕПАНЕНКО СЕРГЕЙ ДЕНИСОВИЧ

ИЗОМОРФИЗМ ТРИВИАЛЬНОГО РАССЛОЕНИЯ И СЕМЕЙСТВА ТИПОВ

КУРСОВАЯ РАБОТА
по направлению 02.03.02 — Фундаментальная информатика
и информационные технологии

Научные руководители:
старший преподаватель В. Н. Брагилевский
М. Сохацкий

Ростов-на-Дону — 2019

Содержание

Introduction

Результаты, полученные в современной математике, становятся всё сложнее. Постепенно подступает момент, когда математики, работающие в разных областях, окончательно перестанут понимать друг друга, а их результаты станет невозможно проверить. Одним из наиболее активно использующихся решений является проверка математических результатов с помощью средств доказательства теорем. Этот класс инструментов сам по себе заслуживает особого внимания, так как было создано множество непохожих подходов, в основе которых лежат интересные теории.

Гомотопическая теория типов (Homotopy type theory, HoTT) [1] — относительно новая область в информатике, которая базируется на неожиданной связи между теорией типов и теорией гомотопий. Эта область лежит в основе *Унивалентных оснований математики* (Univalent Foundation, UF) [2] — попытки формализовать математику, используя в качестве основы не множества, а гомотопические типы или ∞-группоиды, а также *высшие индуктивные типы* (Higher Inductive Types, HIT). Этот подход в первую очередь интересен тем, что позволяет работать с гомотопической теорией, используя *синтетический метод*, то есть не опираясь на более базовые примитивы, как, например, множества, также очень важным преимуществом HoTT являются хорошие вычислительные свойства, унаследованные от зависимой теории типов Мартин-Лёфа [3]. Первоначальная версия гомотопической теории типов включала в себя *аксиому унивалентности*, что открыло возможность для формализации крупного пласта математики на теоретико-типовом языке: экстенсиональности функций и равенства структур, между которыми доказано существование изоморфизма. Но у этой версии было одно очень важное ограничение — аксиома не имела вычислительной интерпретации. В данный момент наиболее многообещающим решением является использование *кубической теории типов* [4], в которой эта аксиома является теоремой, имеющей конструктивное доказательство.

Для обоснования гомотопической интерпретации следует доказать эквивалентность конструкций, которые есть в теории типов, с гомотопическими понятиями. И если гомотопическая интерпретация функций, типов-равенств интуитивно понятна, то зависимые типы требуют большего внимания. В этой работе исследуется доказывательство изоморфизма между семействами типов и расслоением в системе верификации доказательств Arend. [5]

Код, которому посвящена данная работа, можно найти на GitHub [6] в репозитории организации Groupoid Infinity.

1. Dependent types

Dependent type theory is an powerful language, which allows to express complex assertions, write sophisticated software specifications, and reason about this in a natural way. A fast majority of modern proof assistants work with some king of dependently typed language.

The following work was performed with one of the flavors of intensional type theory — *homotopy type theory*, which takes seriously the natural interpretation of identity types as formalizing path space objects in homotopy theory. This flavor comes with a few benefits: higher inductive types, which can be used to obtain quotients objects or free structures, univalence that simplify the process of moving back and forth between isomorphic structures and synthetic approach to homotopy theory, which is much easier to formalize in HoTT than classical set-based homotopy theory.

1.1. Brief preliminaries

Dependent types can be briefly summarized with a few postulates:

1. A hierarchy of type universes is given: $U_0 : U_1 : U_2 : \ldots$.
2. If the universe level is unambiguous, then the level may be dropped.
3. The statement "A is a type"means that $A : U_i$ for some $i \in \mathbb{N}$.
4. The statement "$a : A$"is given a meaning, that a is an term of type A.
5. For any types A and B, we may form a type $A \to B$ that represents a function. And a term $f : A \to B$ of this type may be constructed by demonstrating that $f(a) : B$ under the assumption $a : A$.
6. A dependent type is a morphism to the universe $P : A \to U$, so a dependent type $P : A \to U$ provides us with a type $P(a)$ for any $a : A$.

   It can be easily formalized using Arend:

```
\func family (B : \Type) : \Type => B -> \Type
```

The basic operations on dependent types are the dependent sum $\Sigma_{(x:A)}B(x)$, which is the type of pairs $(x : A, y : B(x))$, and the dependent product $\Pi_{(x:A)}B(x)$, which is the type of functions f such that $\forall x : A, f(x) : B(x)$. When B is constant, the dependent sum reduces to the binary product $A \times B$, and the dependent product reduces to the function type B^A.

1.2. Task

For the purpose of working with homotopy theory *synthetically*, an isomophism between basic type theory constructions and its homotopy equivalents ought to be proved. The dependent type definition reminds of the notion of family of objects indexed over a different object, which is ubiquitous in mathematics; this hint can lead to one of the most basic homotopy theory concept — *fiber bundle*.

2. Fiber bundles

2.1. Idea

A fiber bundle over an object B is simply a an object E paired with a morphism from E to B in which every fiber is isomorphic to a standard fiber F and a fiber of a morphism over a point of B is the collection of elements of E that are mapped by given morphism to this point. It can be viewed as a parameterized family of objects, each "isomorphic"in some way to F, where the family is parameterized by points in B.

Fiber bundles arise in several diverse fields such as physics (this concept embodies the two central principles of modern physics: the gauge principle and the principle of locality), Lie theory (principal G-bundles) and homotopy theory (they generalize notion of a covering space in homotopy theory, because a covering space is an fiber bundle where the fibers are discrete sets). Also fundamental objects in algebraic topology (fibrations) and algebraic geometry (sheaves) are simply generalized fiber bundles.

2.2. Definition

Some definitions are equipped with a corresponding Arend functions or data types. [7]

Definition (Bundle). *A bundle over an object B is an object E equipped with a morphism ϕ from E to B:*

$$\begin{tikzcd} E \arrow[r, "\phi"] & B \end{tikzcd}$$

◊

```
\func bundle (B : \Type) : \Type
        => \Sigma (E : \Type) (phi : E -> B)
```

Definition (Fiber). *The fiber of a morphism $f : E \to B$ over a point of B is the collection of elements of E that are mapped by f to this point, hence it is the following pullback (or fiber product) of pt and f:*

$$\begin{tikzcd}
E \times \star \arrow[r] \arrow[d] & \star \arrow[d, "pt"'] \\
E \arrow[r, "f"] & B
\end{tikzcd}$$

$\diamondsuit$

```
\func fiber (E B : \Type) (f : E -> B) (base : B)
        => \Sigma (x : E) (f x = base)
```

Definition (Fiber bundle). *A fiber bundle over B with standard fiber F is a bundle π over B such that, given any global element $x : \star \to B$, the pullback of E along x is isomorphic to F.* $\diamondsuit$

Definition (Trivial fiber bundle). *The fiber bundle is trivial if E is simply a product $B \times F$ and π is the projection map of the first element of E.*

```
\func total (B : \Type) (F : family B) : \Type
        => \Sigma (x : B) (F x)

\func trivial (B : \Type) (F : family B) : total B F -> B
        => \lam (x : total B F) => x.1
```

2.3. Example

An idea behind the fiber bundle is that it encapsulates the "global twist". Two fiber bundles over a S_1 with $[0 \ldots 1]$ fibers are visualized below and formalized (Arend code for Moebius strip can be found in Appendix A) for clarification. Locally cylinder and Moebius strip are identical, but globally they differ. As an example, if we start with a circle as the base space, and map every point of the circle to a copy of the interval $[0 \dots 1]$, such that as we go once around the circle, the interval comes back with a "twist then we obtain the Moebius strip of Figure 1.

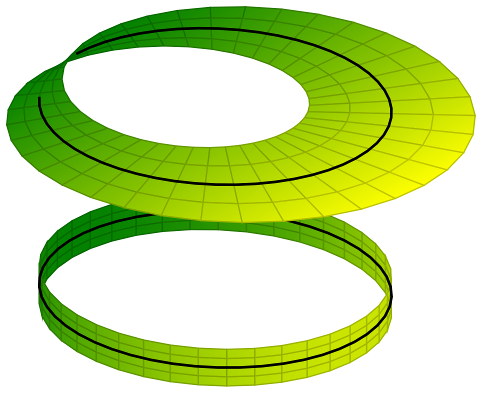

Figure 1 — Fiber bundles over a circle

3. Proof

The proof is straightforward — as we already have both formalizations, we need to construct an equivalence between them and obtain a path between type, which represents fiber bundles, and dependent type using univalence.

Definition (Equivalences). *$f : A \to B$ is an equivalence if there is a map $g : B \to A$ and homotopies $p : \Pi_{a:A}(g(f(a)) = a)$ and $q : \Pi_{b:B}(f(g(b)) = b)$* $\diamond$

Definition (Univalence). *For any two types X, Y, this map $(X = Y) \to (X \simeq Y)$ is an equivalence.* $\diamond$

Conclusion

In this paper I have demonstrated that homotopy type theory is a practical language to work with homotopy theory synthetically by showing the correspondence between fiber bundles and dependent types. We can benefit from synthetic approach, because computer formalizing sophisticated homotopy constructs, such as Hopf-fibrations and homotopy fiber sequences, becomes much easier if language of formalization naturally express basic homotopy theory structures. Because paper proofs in synthetic homotopy theory are often proven with many details, giving a fully formal proof is not much more work, but it could drastically reduce chances of making errors. But sometimes formalization takes more work than was expected, because proofs should be encoded in the corresponding logic, intensional type theory. [8] This result is not problematic to proof in HoTT-supportive proof assistant, such as Arend, but plays a role as an example of using univalence in formal proof. The proof can be simplified by using proof assistant that supports heterogeneous composition.

Bibliography

1. *Univalent Foundations Program T.* Homotopy Type Theory: Univalent Foundations of Mathematics. — Institute for Advanced Study : `https://homotopytypetheory.org/book`, 2013.
2. *Voevodsky V.* Univalent Foundations Project. — 2010.
3. *Martin-Löf P.* An intuitionistic theory of types. — 1972.
4. Cubical Type Theory: a constructive interpretation of the univalence axiom / C. Cohen [и др.] // CoRR. — 2016. — T. abs/1611.02108. — arXiv: `1611.02108`. — URL: `http://arxiv.org/abs/1611.02108`.

5. JetBrains/arend. — URL: `https://github.com/JetBrains/arend`.
6. groupoid/arend. — URL: `https://github.com/groupoid/arend`.
7. *nLab authors*. ncatlab. — 05.2019. — Revision 276.
8. *Doorn F. van*. On the Formalization of Higher Inductive Types and Synthetic Homotopy Theory. — 2018.

Приложение А. Moebius strip

```
\func squeezePath {A : \Type} {a a’ : A}
        (p : a = a’) (i : I) : a = p @ i
        => path (\lam j => p @ meet i j)

\func neg_neg :
        \Pi (x : I) -> ((inv seg) @ (inv seg @ x)) = x
        => \lam x
        => ((\lam x
        => (\lam i
                => inv (squeezePath (inv seg) i)) (inv seg @ x))
                  x)
    # (inv ((\lam x
        => ((\lam x
                => (\lam i
                        => inv (squeezePath seg i)) (seg @ x)) x)
                            # seg) x))

\func twist : I = I
        => Iso=>Path \new Iso I I neg neg neg_neg neg_neg

\func M : \Pi (x : S1) -> \Type => \lam x => \case x \with {
  | base => I
  | loop i => twist @ i
}

\record Moebius (p1 : S1) {
  \field f1 : M(p1)
}
```