Formalization of constructive mathematics

Vienna, April 28

Formalization of Constructive Mathematics

Main activities

(1) development of constructive mathematics

(2) actual formalization of constructive mathematics in dependent type theory

I select two main examples for (1) and (2)

Part I: Constructive homological algebra

Homological algebra: originates from Hilbert *On the theory of algebraic forms* Math. Annalen, vol. 36, 473-534, 1890

From a logical point of view, one would expect most of homological algebra to be directly expressed in first-order logic

This is *not* the case: most text books use Noetherian hypotheses

Exception: Northcott's book on *Finite Free Resolutions*

Constructive homological algebra

In Northcott's book, the *statements* are first-order schemas

Most *proofs* however use existence of prime ideals and minimal prime ideals

According to the Skolem-Gödel completness Theorem, there should be direct first-order proofs

What are they?

Constructive Finite Free Resolution

Hilbert-Burch Theorem

Theorem: *If we have an exact sequence*

$$0 \to R^n \xrightarrow{A} R^{n+1} \to \langle a_0, \ldots, a_n \rangle \to 0$$

then the elements $a_0, \ldots, a_n$ *have a GCD, which is a nonzero divisor*

For a fixed size, this is a first-order statement

Hilbert-Burch Theorem

For $n = 2$ with $A = \begin{pmatrix} u_0 & v_0 \\ u_1 & v_1 \\ u_2 & v_2 \end{pmatrix}$

Hypotheses: $a_0u_0 + a_1u_1 + a_2u_2 = a_0v_0 + a_1v_1 + a_2v_2 = 0$

If $a_0x_0 + a_1x_1 + a_2x_2 = 0$ then x_0, x_1, x_2 is a linear combination of u_0, u_1, u_2 and v_0, v_1, v_2

Conclusion: $\exists g.\ g|a_0 \wedge g|a_1 \wedge g|a_2 \wedge (\forall x.\ x|a_0 \wedge x|a_1 \wedge x|a_2 \wedge \rightarrow x|g)$

Question: can we/how do we compute the gcd of a_0, a_1, a_2 from the given data? Notice that the statement is not a Glivenko statement, so that we cannot be sure for the direct first-order proof to be intuitionistic

Part II: Propositions-as-Types

How to formalize (constructive) mathematics?

Represent and check mathematical proofs on a computer

de Bruijn, Tait, Curry, Howard, Martin-Löf: explicit proof objects

Proofs as programs/propositions as types

Satisfactory in practice, but so far only for discrete structures (work of G. Gonthier on the 4 color theorem and finite group theory)

Equality types?

Propositions-as-Types

Identification of Propositions and Types

$A \wedge B \;=\; A \times B \qquad A \vee B \;=\; A + B \qquad A \Rightarrow B \;=\; A \rightarrow B$

E.g. $\emptyset \rightarrow X$ is a singleton

Kolmogorov 1933 explanation of intuitionistic logic

Dependent products $\prod_{x:A} B(x)$ correspond to universal quantifications

Dependent sums $\sum_{x:A} B(x)$ correspond to existential quantifications

Equality Proofs as Paths

What should be the rules for equality?

V. Voevodsky (2006)

The equality proofs of two elements $a_0, a_1 : A$ should form a *new* type

The elements of this type should be thought of as *paths* between a_0 and a_1

$\mathsf{Id}\ a_0\ a_1 \ = \ \mathsf{Path}\ a_0\ a_1$

The Path Space is contractible

Serre's use of the path space to compute homotopy groups

(J.P.Serre) *when I was working on homotopy groups (around 1950), I convinced myself that, for a space X, there should exist a fibre space E, with base X, which is contractible; such a space would allow me (using Leray's methods) to do lots of computations on homotopy groups... But how to find it? It took me several weeks (a very long time, at the age I was then) to realize that the space of "paths" on X had all the necessary properties-if only I dared call it a "fiber space". This was the starting point of the loop space method in algebraic topology.*

(Interview in the Matematical Intelligencer, 1986)

The Path Space is contractible

That the path space is contractible happens to be exactly one key property of equality as formulated in type theory, which states that

$$\sum_{x:A} \mathsf{Path}\ a\ x$$

has exactly one element for any $a : A$

New instance of the propositions-as-types principle

Propositions as Types as Spaces

Voevodsky Stratification

Stratification of types following the complexity of their equality types

contractible (level 0), proposition (level 1),

set (level 2), groupoid (level 3), $\ldots$

This stratification seems to be important for the representation of mathematics in type theory. For instance, to formalize Yoneda Lemma, we should consider only types of level 2.

What seems to matter is the complexity of the equality, as much as the "size" of the type: the collection of all rings (object of level 2) is of level 3

Mathematical model using Kan simplicial sets

Set theory/type theory

Tarski 1966 *What are logical notions*

logical statement = a statement invariant under isomorphisms

Is mathematics part of logic?

Lindenbaum and Tarski *On the limitations of the means of expression of deductive theories* 1936 formulate and claim the invariance under isomorphism principle for *type theory*

Set theory/type theory

All operations should preserve isomorphisms

This is *not* the case for set theory: $X = \{0, 1\}$ and $Y = \{1, 2\}$ are isomorphic but $X \cup X$ and $X \cup Y$ are not

This is the case for type theory

Voevodsky Univalence Axiom

This axiom states that equality proofs of two types should be isomorphisms

A consequence is that two *isomorphic mathematical* structures are *equal*, which should be important for modularity of formal proof development

Main question: can we justify the univalence axiom?

Analogy with Takeuti-Gandy explanation of the axiom of extensionality

Computational rules for extensionality for simple type theory, computational rules for the ι symbol