From the universality of mathematical truth to the interoperability of proof systems

Gilles Dowek

I. Yet another crisis of the universality of mathematical truth

The universality of mathematical truth

The truth conditions of a mathematical statement must be the object of unanimous agreement

- Constitutive of the notion of mathematical truth itself
- Yet, constantly jeopardized
- When mathematicians disagree on the truth of some statements: a crisis of the universality of mathematical truth

In the past

- The incommensurability of the diagonal and side of a square

$$\exists x \, (x \textit{ is a number} \wedge x^2 = 2)$$

- The introduction of infinite series

$$\sum_n \frac{1}{2^n} = 2 \qquad\qquad \sum_n (-1)^n = 0$$

- The non-Euclidean geometries

The sum of the angles in a triangle equals the straight angle

- The independence of the axiom of choice

Every vector space has a basis

- Constructivity

If $A \cup B$ infinite, then A infinite or B infinite

All these crises have been resolved

The incommensurability of the diagonal and side of a square: rational numbers and real numbers

Infinite series: limit

Non-Euclidean geometries: several solutions

- Different spaces: truth of

  On a space of zero curvature, the sum of the angles in a triangle equals the straight angle

  but not of

  On a space of negative curvature, the sum of the angles in a triangle equals the straight angle

- Axiomatic theories: E and H, truth of

  $$E \vdash \textit{the sum of the angles in a triangle equals the straight angle}$$

  but not of

  $$H \vdash \textit{the sum of the angles in a triangle equals the straight angle}$$

Equivalent (soundness and completeness)

The second solution

- A true $\longrightarrow \Gamma \vdash A$ true
- Truth conditions: for the statements of geometry $\longrightarrow$ for arbitrary sequents
- Separation between the definition of the truth conditions of a sequent: the logical framework and the definition of the various geometries as theories
- A logical framework: Predicate logic
- The various geometries defined in this logical framework

The axiom of choice

First solution: truth of

In a model of ZFC, every vector space has a basis

but not of

In a model of ZF, every vector space has a basis

Second: *Every vector space has a basis* consequence of the axiom of choice

First solution does not work:
- Too far from the original formulation
- Problem of the “absolute” theory in which this should be proved

Thus, second chosen, paving the way to Reverse mathematics

Constructivity

First solution: truth of

In a model valued in a Boolean algebra, if $A \cup B$ *infinite, then* A *infinite or* B *infinite*

but not of

In a model valued in a Heyting algebra, if $A \cup B$ *infinite, then* A *infinite or* B *infinite*

Again, too far from the original formulation and question of the “absolute” theory

Second: *if* $A \cup B$ *infinite, then* A *infinite or* B *infinite* consequence of the excluded middle

A third solution: Ecumenism

Changing the axioms while keeping the same symbols?

Axioms express the meaning of the symbols:
different axioms $\longrightarrow$ different meanings $\longrightarrow$ different symbols (just like $\vee$ and $\oplus$)

The only "mistake" is not to accept or to reject the excluded middle, but to use the same symbol for $\vee$ and $\vee_c$

Nothing prevents from using them both

Truth of

$$Infinite(A \cup B) \Rightarrow_c Infinite(A) \vee_c Infinite(B)$$

but not of

$$Infinite(A \cup B) \Rightarrow Infinite(A) \vee Infinite(B)$$

$\sqrt{2}$: $\mathbb{Q}$ vs. $\mathbb{R}$ already Ecumenical (mass vs. weight...)

Past crises ($\sqrt{2}$, $\sum_n$, non-Euclidean geometries, AC, Constructivism) have been resolved

But... yet another crisis: computerized proof systems

Computerized proof systems

Coq, Isabelle/HOL, PVS, HOL Light, Lean...
A major step forward in the quest of mathematical rigor

But jeopardizes, once again, the universality of mathematical truth

A proof of Fermat's little theorem $\longrightarrow$ a Coq proof of Fermat's little theorem, a PVS proofs of Fermat's little theorem...

Each proof system: its own language and its own truth conditions

Yet another crisis to be resolved

II. Logical frameworks

A solution that (already) worked for several crises

Express the theories implemented in Coq, Isabelle/HOL, PVS, HOL Light, Lean... in Predicate logic

- (if we are lucky) many common axioms and few differentiating the theories
- (if we are lucky) mixing the axioms differentiating the symbols (Ecumenism)
- analyze which proof uses which axiom (just like for the axiom of choice)
- try to find better proofs using less axioms (just like constructivization, Reverse mathematics...)

A solution that (already) worked for several crises

Express the theories implemented in Coq, Isabelle/HOL, PVS, HOL Light, Lean... in a logical framework

- (if we are lucky) many common axioms and few differentiating the theories
- (if we are lucky) mixing the axioms differentiating the symbols (Ecumenism)
- analyze which proof uses which axiom (just like for the axiom of choice)
- try to find better proofs using less axioms (just like constructivization, Reverse mathematics...)

Beyond Predicate logic

In a century: some limitations of Predicate logic

Other logical frameworks: λ-Prolog, Isabelle, the Edinburgh logical framework, Pure type systems, Deduction modulo theory, Ecumenical logic, DEDUKTI

In DEDUKTI

- Function symbols can bind variables (like in λ-Prolog, Isabelle, the Edinburgh logical framework)
- Proofs are terms (like in the Edinburgh logical framework)
- Deduction and computation are mixed (like in Deduction modulo theory)
- Both constructive and classical proofs can be expressed (like in Ecumenical logic)

The two features of Dedukti

Dedukti is a typed λ-calculus with

- Dependent types
- Computation rules

Several implementations: Dkcheck, Lambdapi, Kocheck...

No typing rules today, but illustration of these features with examples

In a logical framework, you can

- Define your theory
- Check proofs expressed in this theory

A theory in Predicate logic: a language (sorts, function symbols, and predicate symbols) and a set of axioms

A theory in Dedukti: a set of symbols (replace sorts, function symbols, predicate symbols, and axioms) and a set of computation rules

III. Examples of axioms in Dedukti

Catching up with Predicate logic

Predicate logic is a sophisticated framework with notions of sort, function symbol, predicate symbol, arity, variable, term, proposition, proof...

A typed λ-calculus is much more primitive

These notions must be constructed

A good exercise to start with, but also an interest in itself: the first book of Euclid's elements (originally formalized in Coq) can be expressed in Predicate logic + the axioms of geometry and exported to many systems (Géran)

Terms and propositions: a first attempt

I : TYPE
$Prop$: TYPE

function symbols: $I \to \ldots \to I \to I$
predicate symbols: $I \to \ldots \to I \to Prop$
connectives: $Prop \to \ldots \to Prop \to Prop$
$\forall$: $(I \to Prop) \to Prop$

- $\forall$ binds (higher-order abstract syntax: $\forall x\ A$ expressed as $\forall\ \lambda x\ A$)
- Symbol declarations only (no computation rules yet)
- Simply typed λ-calculus (no dependent types yet)
- Types are terms of type TYPE

Works if we want one sort

But if we want several (like in geometry: points, lines, circles...)

I_1 : TYPE

I_2 : TYPE

I_3 : TYPE

Several (an infinite number of?) symbols and several (an infinite number of?) quantifiers

$\forall_1 : (I_1 \to Prop) \to Prop$

$\forall_2 : (I_2 \to Prop) \to Prop$

$\forall_3 : (I_3 \to Prop) \to Prop$

Making the universal quantifier generic

Something like
$\forall : \Pi X : \mathsf{TYPE}, ((X \to \mathit{Prop}) \to \mathit{Prop})$

But does not work for two reasons

- (a minor one) no dependent products on TYPE in DEDUKTI
- (a major one) many things in TYPE beyond I_1, I_2, and I_3 (for example *Prop*)

Making the universal quantifier generic

I : TYPE
I_1 : TYPE, I_2 : TYPE, I_3 : TYPE

Set : TYPE

ι : Set
ι_1 : Set, ι_2 : Set, ι_3 : Set

El : $Set \to$ TYPE

$El\ \iota \longrightarrow I$
$El\ \iota_1 \longrightarrow I_1$, $El\ \iota_2 \longrightarrow I_2$, $El\ \iota_3 \longrightarrow I_3$

$Prop$: TYPE

$\forall : \Pi x : Set, (El\ x \to Prop) \to Prop$

Uses dependent types and computation rules

Reminiscent of expression of Simple type theory in Predicate logic, universes *à la* Tarski...

Proofs

So far: terms and propositions. Now: proofs

Proofs are trees, they can be expressed in Dedukti

Curry-de Bruijn-Howard: $P \Rightarrow P$ should be the type of its proofs
But not possible here $P \Rightarrow P : \mathit{Prop} : \mathrm{TYPE}$ is not itself a type

$\mathit{Prf} : \mathit{Prop} \to \mathrm{TYPE}$
mapping each proposition to the type of its proofs: $\mathit{Prf}\,(P \Rightarrow P) : \mathrm{TYPE}$

Not all types are types of proofs (for example I, $\mathit{El}\ \iota$, Prop...)

Proofs

Brouwer-Heyting-Kolmogorov: $\lambda x : (\mathit{Prf}\ P), x$ should be a proof of $P \Rightarrow P$
But has type $(\mathit{Prf}\ P) \to (\mathit{Prf}\ P)$ and not $\mathit{Prf}\ (P \Rightarrow P)$
$\mathit{Prf}\ (P \Rightarrow P)$ and $(\mathit{Prf}\ P) \to (\mathit{Prf}\ P)$ must be identified

A computation rule

$$\mathit{Prf}\ (x \Rightarrow y) \longrightarrow (\mathit{Prf}\ x) \to (\mathit{Prf}\ y)$$

In the same way

$$\mathit{Prf}\ (\forall\ x\ p) \longrightarrow \Pi z : (\mathit{El}\ x), (\mathit{Prf}\ (p\ z))$$

The function Prf is an injective morphism from propositions to types: it is the Curry-de Bruijn-Howard isomorphism

If you want to express Predicate logic proofs, you know enough

Simple type theory (HOL4, HOL Light, Isabelle/HOL...): two features

- Propositions as objects

  $o : Set$

  $El\ o \longrightarrow Prop$

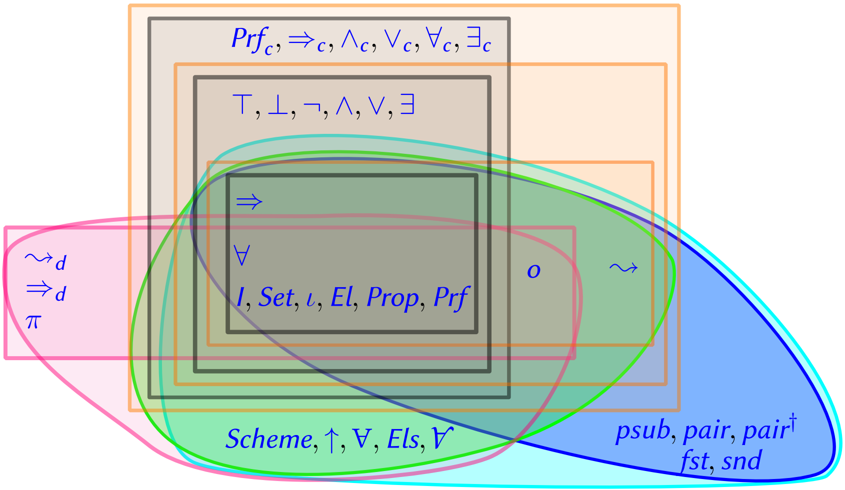
- Functions

  $\leadsto\ : Set \to Set \to Set$

  $El\ (x \leadsto y) \longrightarrow (El\ x) \to (El\ y)$

More symbols

Pick cherries according to your taste



Enough to express Predicate logic, Simple type theory, Simple type theory with predicate subtyping, The Calculus of constructions...

More symbols: universes, universe polymorphism, predicativity, inductive types, cubical type theory (Barras), set theory (Traversié)

IV. The benefits of universality

- Reverse engineering proofs

  First book of Euclide's Elements in Coq $\longrightarrow$ in Predicate logic

  Fermat's little theorem in Matita $\longrightarrow$ in constructive Simple type theory (Thiré)

  Bertrand's postulate in Matita $\longrightarrow$ in Predicative type theory (Felicíssimo)

- Interoperability

  The first book of Euclide's element in Isabelle/HOL, TSTP...

  Fermat's little theorem in Isabelle/HOL, HOL Light, Coq, Lean, PVS...

  Bertrand's postulate in Agda

- Cross-verification

A social motivation: mathematicians and industrials more likely to develop proofs in mathematics (possibly with some axioms they can debate) than in an exotic system

And a philosophical one: Universality has survived many crises: we ought not to give up on it (and we do not need to)

Mathematics is necessarily always in crisis, and always in the process of resolving it.

Michel Serres