Sharing geometry proofs across logics and systems

Gilles Dowek

Yet another crisis of the universality of mathematical truth

A moment in time when mathematicians disagree of the truth of some statement

- The irrationality of $\sqrt{2}$: does there exist a number r such that $r^2 = 2$?
- Infinite sums: is $\sum_{n=0}^{\infty}(-1)^n$ equal to 0?
- Non Euclidean geometries: is the sum of the angles of a triangle equal to π?
- Constructivism: if $0 \in E$ and $2 \notin E$, does there exists a number n such that $n \in E$ and $n+1 \notin E$?
- Choice: does every vector space have a basis?

Etc.
Not a satisfactory situation: the crisis has to be resolved
And these five crises have

Proof processing systems

HOL 4, Isabelle/HOL, HOL Light, Coq, Matita, Lean, PVS, etc.
A huge step forward in the quest of mathematical rigor
New proofs that could not be built by hand

But a new crisis of the universality of mathematical truth

A proof of the four color theorem $\longrightarrow$ A Coq proof of the four color theorem
A HOL Light proof of Hales' theorem
A PVS proof of the correctness of the 3R3D algorithm
etc.

Major obstacle to teach formal proof
Major obstacle for formal proof to be used in industry

One way (among others) to solve a crisis

- Express the axioms of the various set theories (Euclidean, Hyperbolic, Elliptic... geometry) in the same logical framework (Predicate logic)
- Note that they have a lot of axioms in common and differ on a few
- Analyze which axiom is used in which proof

A method used to solve (at least) the crises of non Euclidean geometries and of the axiom of choice

Towards a solution of the crisis of proof systems

- Express the theories implemented in HOL Light, Coq, PVS, etc. in a common logical framework (for instance Predicate logic)
- Analyze which "axiom" is used in which proof, regardless the system it has been developed in

Predicate logic?

In 1928, Predicate logic (Hilbert and Ackermann): a revolution

Since Euclid: geometry, arithmetic, set theory, etc. each system its syntax, its notion of proof, etc.

A common framework for geometry (with or without the parallel axiom), arithmetic, set theory (with or without the axiom of choice), etc.

But a short revolution

At that time, another theory: Type theory (*Principia Mathematica*)
No expression in Predicate logic

Soon (1940) Church: a new formulation of Type theory (based on λ-calculus) impossible to express in Predicate logic (λ binds)

After 1970: Martin-Löf's type theory, the Calculus of constructions, Type theory with predicate subtyping, etc. not in Predicate logic

The limits of Predicate logic

- No bound variables ($\lambda x\ x$)
- No syntax for proofs
- No notion of computation
- No good notion of proof reduction
- Classical and not constructive

New logical frameworks

- No bound variables ($\lambda x\ x$): λ-Prolog, Isabelle, $\lambda\Pi$-calculus
- No syntax for proofs: $\lambda\Pi$-calculus
- No notion of computation: Deduction modulo theory
- No good notion of proof reduction: Deduction modulo theory
- Classical and not constructive: Ecumenical logic

The $\lambda\Pi$-calculus modulo theory that generalizes them all
Dedukti: an implementation of it

Examples of axioms in DEDUKTI: Terms and propositions

Weak framework: terms and propositions are not primitive but need to be built

$I : TYPE$
$Prop : TYPE$
$\Rightarrow : Prop \to Prop \to Prop$
$\forall : (I \to Prop) \to Prop$

Many-sorted?

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 $P \Rightarrow P : \mathit{Prop} : \mathit{TYPE}$ is not itself a type

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

Proofs

Brouwer-Heyting-Kolmogorov: $\lambda x : (Prf\, P)\ x$ should be a proof of $P \Rightarrow P$ but it has type $(Prf\, P) \rightarrow (Prf\, P)$

$Prf\,(P \Rightarrow P)$ and $(Prf\, P) \rightarrow (Prf\, P)$ must be identified

A reduction rule

$$Prf\,(P \Rightarrow P) \longrightarrow (Prf\, P) \rightarrow (Prf\, P)$$

This reduction rule is the Curry-de Bruijn-Howard correspondence

More axioms

(Constructive and classical) Connectives and quantifiers

Propositions as objects, functions (as in HOL Light...)

Dependency (as in Coq...)

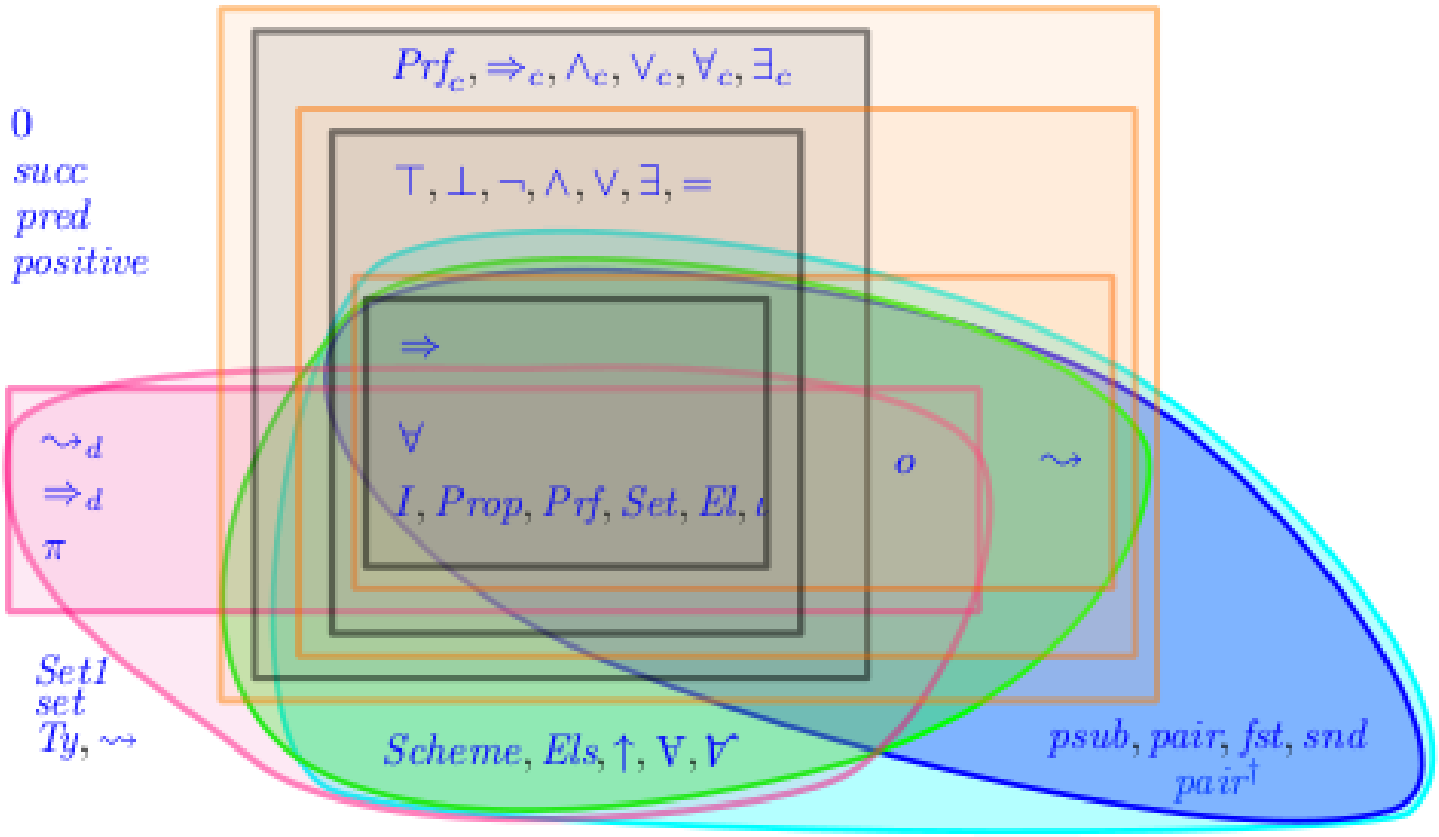
Object-level predicative polymorphism (as in HOL Light...)

Object-level dependent types (as in Coq...)

Predicate subtyping (as in PVS)

Infinity

Some axioms and some theories



The theory $\mathcal{U}$ (with Blanqui, Grienenberger, Hondet, Thiré)
And more: universes, universe polymorphism (Assaf, Férey, Genestier), inductive types (Boespflug, Burel), coinductive types (Felicissimo), etc.

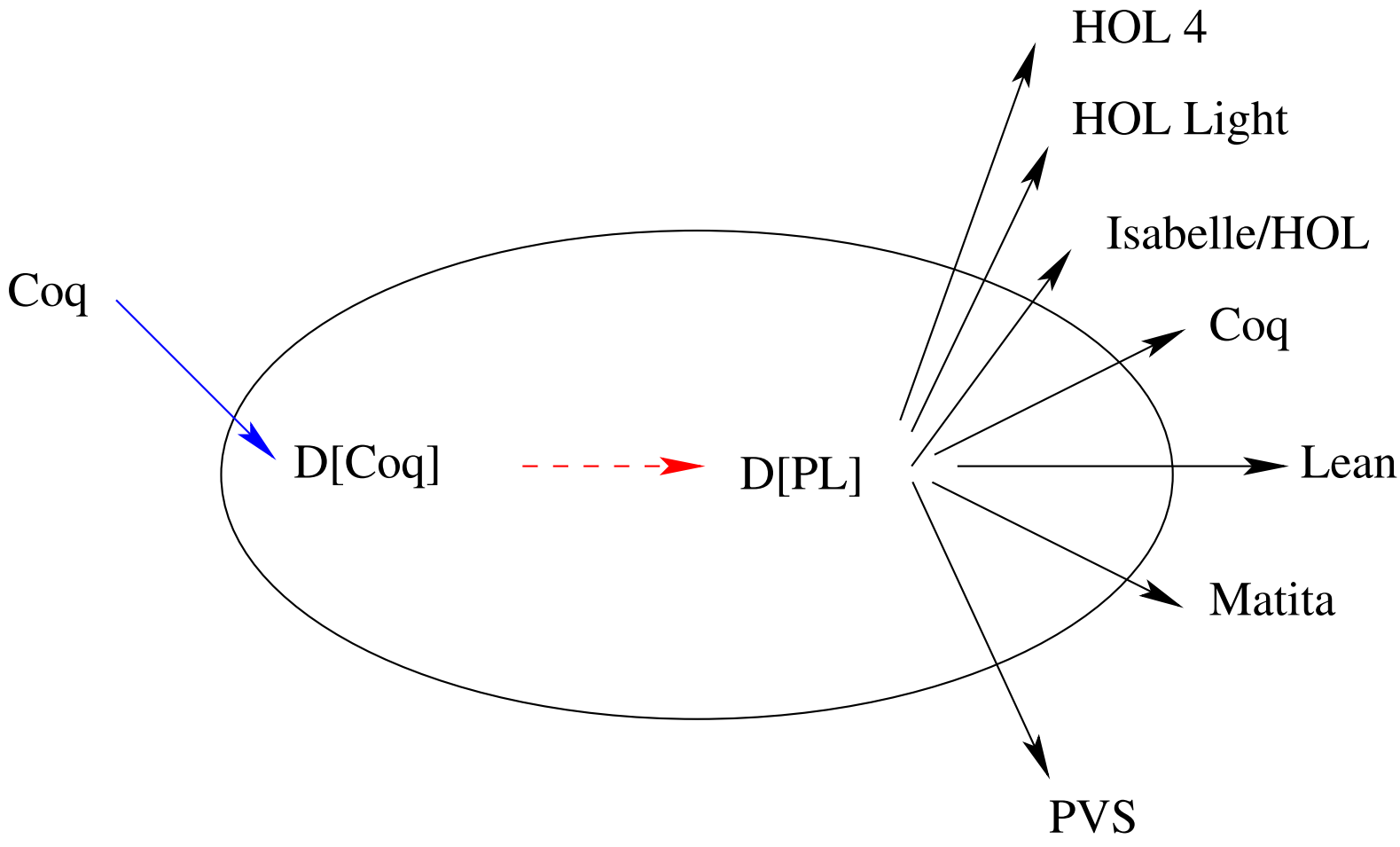
Reverse mathematics in Dedukti

The Calculus of constructions: 12 axioms
Minimal predicate logic: a subset formed with 8 axioms

- All proofs in Minimal predicate logic can be translated to the Calculus of constructions
- The proofs in the Calculus of constructions that do not use these four axioms can be translated to Minimal predicate logic (not the others: genuine Calculus of constructions proofs)

Sharing geometry proofs across logics and systems

- (Boutry): the first book of Euclid's elements in Coq + EA
- (Boutry and Férey): the first book of Euclid's elements in D[Coq+EA]
- (Géran): the first book of Euclid's Elements in D[PL+EA]
- and in seven systems: HOL 4, Isabelle/HOL, HOL Light, Coq, Matita, Lean, PVS (+ EA)



The first book of Euclid's elements cross-checked in seven systems

Towards a shared proof library

Thiré: the same picture for the arithmetic library of Matita
Two first steps in the constitution of a shared library of proofs (Logipedia) where

- proofs are independent of the system used t build them
- but not of the theory that is required to express them

One objective of the COST project Euro proof net (Blanqui)

Everyone can contribute: develop a (small or large) library, translate the library to Dedukti (use an existing translator or develop your own), eliminate the superfluous axioms, add it to Logipedia