Logical frameworks, reverse mathematics, and formal proofs translation

Gilles Dowek

The question

Formal proofs
developed in different systems: Coq, Matita, HOL Light...
expressed in different theories

Can a proof expressed in a theory $\mathcal{T}$ be translated to $\mathcal{T}'$?

Can a proof in ZF be translated to ZFC?
a proof in ZFC to ZF?
a proof in Euclidean geometry to Hyperbolic geometry?
a proof in Hyperbolic geometry to Euclidean geometry?

Proof transformation

There exists a basis of $\mathbb{R}^2$

- by the incomplete basis theorem (axiom of choice)
- $\langle 1, 0 \rangle$, $\langle 0, 1 \rangle$

automatically (for example: constructivization) or by hand

Reformulating the project of reverse mathematics

- Formal proofs, not pencil-paper-LaTeXones
- Expressive theories (Set theory, Type theory...) and not fragments of arithmetic
- Analyze proofs before (possibly) transforming them

How is this possible?

Because ZF and ZFC, Euclidean geometry, Hyperbolic geometry...
expressed in the same logical framework: predicate logic

In predicate logic, a theory: several axioms

Permits to raise the question: which axioms are used in a proof

Not if ZF and ZFC were defined by Turing machines terminating
on their theorems

The revolution of predicate logic

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

Hilbert and Ackermann (1928): predicate logic
A common framework for geometry, arithmetic, set theory...
A theory: symbols and axioms

But a short revolution: not Russell's Type theory, not Church's Type theory, not Martin-Löf's Type theory, not the Calculus of constructions...

Three attitudes

- Consider logical framework as a dead concept
- Express Russell's type theory, Church's, Martin-Löf's, the Calculus of constructions... in predicate logic by will of by force (Henkin, Davis, D...)
- Extend predicate logic to a better logical framework

The limits of predicate logic

- No bound variables ($\lambda x\ x$)
- No syntax for proofs
- No notion of computation
- No good notion of cut
- 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 cut: Deduction modulo theory
- Classical and not constructive: Ecumenical logic

$\lambda\Pi$-calculus modulo theory (DEDUKTI) generalizes them all

Simple type theory in DEDUKTI

$$
\begin{array}{rcl}
\mathit{type} & : & \mathit{Type} \\
\eta & : & \mathit{type} \to \mathit{Type} \\
o & : & \mathit{type} \\
\mathit{nat} & : & \mathit{type} \\
\mathit{arrow} & : & \mathit{type} \to \mathit{type} \to \mathit{type} \\
\varepsilon & : & (\eta\ o) \to \mathit{Type} \\
\Rightarrow & : & (\eta\ o) \to (\eta\ o) \to (\eta\ o) \\
\forall & : & \Pi x : \mathit{type}\ (((\eta\ x) \to (\eta\ o)) \to (\eta\ o))
\end{array}
$$

$$
\begin{array}{rcl}
(\eta\ (\mathit{arrow}\ x\ y)) & \longrightarrow & (\eta\ x) \to (\eta\ y) \\
(\varepsilon\ (\Rightarrow\ x\ y)) & \longrightarrow & (\varepsilon\ x) \to (\varepsilon\ y) \\
(\varepsilon\ (\forall\ x\ y)) & \longrightarrow & \Pi z : (\eta\ x)\ (\varepsilon\ (y\ z))
\end{array}
$$

The Calculus of constructions in DEDUKTI

$$
\begin{array}{rcl}
\mathit{type} & : & \mathit{Type} \\
\eta & : & \mathit{type} \to \mathit{Type} \\
o & : & \mathit{type} \\
\mathit{nat} & : & \mathit{type} \\
\mathit{arrow} & : & \Pi x : \mathit{type}\ (((\eta\ x) \to \mathit{type}) \to \mathit{type}) \\
\varepsilon & : & (\eta\ o) \to \mathit{Type} \\
\Rightarrow & : & \Pi x : (\eta\ o)\ (((\varepsilon\ x) \to (\eta\ o)) \to (\eta\ o)) \\
\forall & : & \Pi x : \mathit{type}\ (((\eta\ x) \to (\eta\ o)) \to (\eta\ o)) \\
\pi & : & \Pi x : (\eta\ o)\ (((\varepsilon\ x) \to \mathit{type}) \to \mathit{type})
\end{array}
$$

$$
\begin{array}{rcl}
(\eta\ (\mathit{arrow}\ x\ y)) & \longrightarrow & \Pi z : (\eta\ x)\ (\eta\ (y\ z)) \\
(\varepsilon\ (\Rightarrow\ x\ y)) & \longrightarrow & \Pi z : (\varepsilon\ x)\ (\varepsilon\ (y\ z)) \\
(\varepsilon\ (\forall\ x\ y)) & \longrightarrow & \Pi z : (\eta\ x)\ (\varepsilon\ (y\ z)) \\
(\eta\ (\pi\ x\ y)) & \longrightarrow & \Pi z : (\varepsilon\ x)\ (\eta\ (y\ z))
\end{array}
$$

A comparison

- *arrow* dependent in the Calculus of constructions but not in Simple type theory
- Same for $\Rightarrow$
- An extra symbol π in the Calculus of constructions: express functions mapping proofs to terms

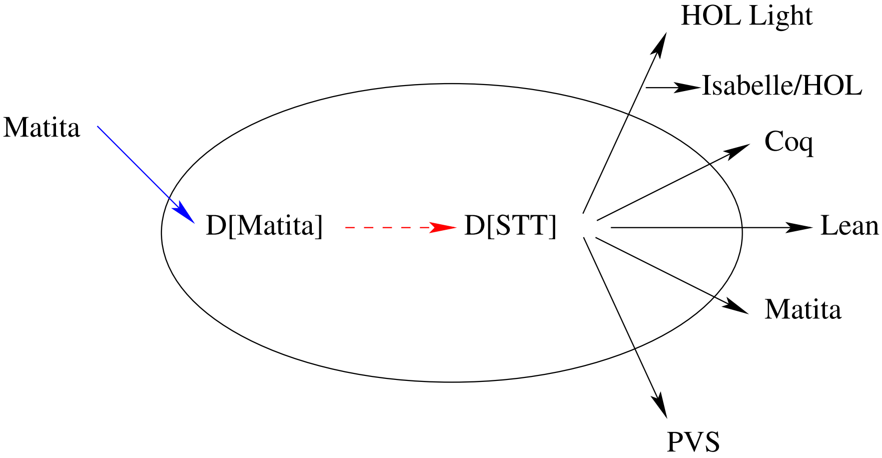
Reverse mathematics in DEDUKTI

- All proofs in Simple type theory can be translated to the Calculus of constructions
- The proofs in the Calculus of constructions that do not use these three features can be translated to Simple type theory (not the others: genuine Calculus of constructions proofs)

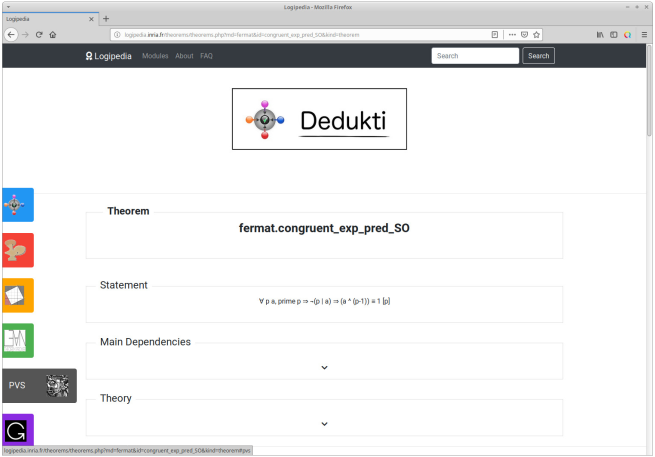
For example: all the proofs of the arithmetic library of MATITA

"First" proof of Fermat's little theorem in constructive Simple type theory (further: predicative, PA, fragments of PA...)

Reverse mathematics as the basis of interoperability



http://logipedia.science



Why does it work?

Because proof systems implement very expressive theories and use only a tiny part of it

Early empirical evidences

- Proof systems: very different theories, but very similar libraries
- Mathematicians are not very interested in the axioms used in their proofs: any theory seems to fit