Conservativity of

Type Theory

over

Higher-order Arithmetic

Daniël Otten

Joint work with Benno van den Berg

Overview

We investigate the relation between arithmetic and type theory.

In dependent type theory, the number of universes influence how much can prove about the natural numbers:

- Classical Result: ML0 is conservative over HA (Beeson 1979).
- Our Result: type theories with a single level of universes are conservative over Higher-order Heyting Arithmetic (HAH).

The amount of conservativity depends on our interpretation of logic:

- for proof-irrelevant versions: they prove exactly the same arithmetical theorems as HAH (of any order).
- for proof-relevant versions: they prove more second-order, but the same first-order arithmetical theorems as HAH.

Contents

$$\boxed{\text{arithmetic}} \;\rightleftarrows\; \boxed{\text{type theory}}$$

Higher-order Heyting Arithmetic

In higher-order logic we can quantify over powersets of the domain.
If we write $\exists x^n$ or $\forall x^n$ then x is an element of the n-th powerset:

- x^0 is an element of the domain,
- x^1 is a set,
- x^2 is a set of sets,
- and so on.

For x^n and Y^{n+1} we have a new atomic formula $x \in Y$.
We have two additional logical axiom schemes:

$$\forall X^{n+1}, Y^{n+1}\,(\forall z^n\,(z \in X \leftrightarrow z \in Y) \to X = Y), \quad \text{(extensionality)}$$

$$\exists X^{n+1}\,\forall z^n\,(z \in X \leftrightarrow \phi[z]). \quad \text{(comprehension)}$$

HAH has the axioms of PA but in intuitionistic higher-order logic.

Contents

$$\begin{tikzcd} \boxed{\text{arithmetic}} \arrow[r, shift left] & \boxed{\text{type theory}} \arrow[l, shift left] \end{tikzcd}$$

Interpreting Logic in Type Theory

There are two interpretations of logic in type theory.

A proof-irrelevant way, using propositional truncation:

$$(A \vee B)^\bullet := \|A^\bullet + B^\bullet\|, \quad (\exists x^n\, B[x^n])^\bullet := \|\Sigma(x : \mathcal{P}^n\,\mathbb{N})\, B[x^n]^\bullet\|,$$

$$(A \wedge B)^\bullet := A^\bullet \times B^\bullet, \quad (\forall x^n\, B[x^n])^\bullet := \Pi(x : \mathcal{P}^n\,\mathbb{N})\, B[x^n]^\bullet,$$

$$(A \to B)^\bullet := A^\bullet \to B^\bullet.$$

And a proof-relevant way:

$$(A \vee B)^\circ := A^\circ + B^\circ, \quad (\exists x^n\, B[x^n])^\circ := \Sigma(x : \mathcal{P}^n\,\mathbb{N})\, B[x^n]^\circ,$$

$$(A \wedge B)^\circ := A^\circ \times B^\circ, \quad (\forall x^n\, B[x^n])^\circ := \Pi(x : \mathcal{P}^n\,\mathbb{N})\, B[x^n]^\circ,$$

$$(A \to B)^\circ := A^\circ \to B^\circ.$$

Interpreting Powersets in Type Theory

Using a universe $\mathcal{U}$, we can define powertypes:

$$\mathcal{P}\,A := A \to \mathcal{U}.$$

We use this to interpret higher-order logic:

- To satisfy extensionality, we need $\mathcal{U}$ to satisfy:

$$\text{funext} : \Pi(f, f' : \mathcal{P}\,A)\,(\Pi(x : A)\,(f\,x = f'\,x) \to (f = f')),$$

$$\text{propext} : \Pi(P, P' : \mathcal{U})\,((P \leftrightarrow P') \to (P = P')).$$

  Alternatively, we can use setoids or quotients.

- To satisfy comprehension, we need $\mathcal{U}$ to be impredicative.

  Alternatively, we can use propositional-resizing.

All alternatives give exactly the same conservativity results.

Contents

$$\boxed{\text{arithmetic}} \;\rightleftarrows\; \boxed{\text{type theory}}$$

Type Theory

We work with a restriction λC$+$ of the Calculus of Inductive Constructions (the theory behind Coq and Lean).

We have two impredicative universes: Prop, Set : Type.

- We use Prop to interpret propositions and powertypes.
  For this universe we assume propext.
- We use Set for data types, and assume $\mathbb{0}, \mathbb{1}, +, \Sigma, \Pi, =, \|A\|$.
  We can also assume W-types, uip, or even an extensional theory.

Lastly, we assume funext in general.

Contents

$$\boxed{\text{arithmetic}} \rightleftarrows \boxed{\text{type theory}}$$

Main Result (proof-irrelevant)

Theorem (for a proof-irrelevant interpretation)

λC$+$ proves the same arithmetical formulas as HAH (of any order).

Proof Sketch. We can show that λC$+$ proves the axioms of HAH. The difficult part is showing that it does not prove more.

We first give a conservative extension of HAH, named $\underbrace{\text{HAHP}}_{\lambda x\, b[x],\, \langle a,b\rangle}$.
Then we construct an arrow:

$$\begin{tikzcd} \text{HAH} \arrow[r, "\bullet"] \arrow[d, hook] & \lambda\text{C}+ \arrow[dl] \\ \text{HAHP} & \end{tikzcd}$$

And show that the diagram commutes up to logical equivalence. $\square$

Realizability

We use realizability where we interpret:

$$\begin{aligned} \text{propositions} &\rightsquigarrow \text{subsingletons}, \\ \text{sets} &\rightsquigarrow \text{partial equivalence relations (PERs)}, \\ \text{types} &\rightsquigarrow \text{assemblies or } \omega\text{-sets}. \end{aligned}$$

This takes inspiration from a well-known model for the calculus of constructions (Hyland1988, Reus1999), modified in two ways:

- we restrict assemblies to those which live in some $\mathcal{P}^n\,\mathbb{N}$,
- we extend the interpretation to our extended theory.

Main Result (proof-relevant)

Theorem (for a proof-relevant interpretation)

λC$+$ proves more second-order, but the same first-order arithmetical formulas as HAH.

Proof Sketch. An example of a second-order formula that is provable in type theory but not in HAH is the axiom of choice.

For first-order formulas we modify our earlier proof.
We extend HAHP to HAHPϵ by adding a computable choice principle.
This theory is conservative over HAH, and we can construct an arrow:

$$\begin{tikzcd} \mathsf{HAH} \arrow[r, "\circ"] \arrow[d, hook] & \lambda\mathsf{C}+ \arrow[dl] \\ \mathsf{HAHP}\epsilon & \end{tikzcd}$$

s.t. the diagram commutes for first-order formulas up to equiv. $\square$

Generalisations and Summary

We can restrict our methods to systems in the lambda cube to show:

λC$+$ is conservative over HAH,

λP2$+$ is conservative over HA2,

λP$+$ is conservative over HA,

where the interpretation of logic determines the conservativity:

- for proof-irrelevant versions: conservativity for all formulas,
- for proof-relevant versions: conservativity for first-order, but not for second- or higher-order formulas.