

PROOF THEORY OF ARITHMETIC

Lecture 5 – Perspectives and further directions

Anupam Das

University of Birmingham

34TH EUROPEAN SUMMER SCHOOL
IN LOGIC, LANGUAGE AND INFORMATION

Ljubljana, Slovenia

11 August 2023



These slides are available at <http://www.anupamdas.com/ess11i23>.

- 1 Recursion complexity vs induction complexity
- 2 Higher type recursion
- 3 Beyond PA and ε_0
- 4 Conclusions
- 5 Break: questions and exercises
- 6 References

Example: Ackermann-Péter, logically

Rank vs logical complexity

It turns out that we can **refine** our results to account for logical complexity.

Writing $I\Sigma_n$ for the fragment of PA with induction on only Σ_n -formulas:

Theorem

$I\Sigma_n$ has 'prooftheoretic ordinal' ω_{n+1} .

Proof idea.

- The restriction of \vdash_{PA} to $I\Sigma_n$ requires only Σ_n and Π_n formulas, by a **partial cut-elimination** in PA.
- We only need induction on Σ_r -formulas to prove $<_{r+1}$ -wf. □

An excellent reference:

- [Takeuti, 1975]

(Blank slide)

- 1 Recursion complexity vs induction complexity
- 2 Higher type recursion**
- 3 Beyond PA and ε_0
- 4 Conclusions
- 5 Break: questions and exercises
- 6 References

The road not taken: higher-type computation

There is another notable successful realisation of Hilbert's Program: the **Dialectica functional interpretation**.

Here, instead of extending PRA by recursion on higher ordinals, we allow the recursive definition of **higher order** functionals.

An excellent reference:

- [Avigad and Feferman, 1998]

In recent years there have been astonishing applications via **proof mining**.

An excellent reference:

- [Kohlenbach, 2008]

Definition (Finite types)

The **finite types** (or **simple types**), written σ, τ etc., are generated by:

$$\sigma, \tau, \dots ::= N \mid \sigma \rightarrow \tau$$

Think of types as extra **sorts** in our logic. We may extend the standard model \mathfrak{N} to a higher **type structure** by setting:

- $N^{\mathfrak{N}} := \mathbb{N}$
- $(\sigma \rightarrow \tau)^{\mathfrak{N}} := \{f : \sigma^{\mathfrak{N}} \rightarrow \tau^{\mathfrak{N}}\}$

Definition (Informal)

System T extends PRA by appropriate constants and **primitive recursion at all finite types**.

NB: We are being imprecise about higher-type **equality** here!

Example: Ackermann-Péter, again!



Gödel, 1958

For each formula φ of \mathcal{L}_A , there is a **quantifier-free** T-formula $\varphi_D^N(x, y)$ and a T-term $t(x)$ s.t.:

$$\text{PA} \vdash \varphi \quad \Longrightarrow \quad \text{T} \vdash \varphi_D^N(x, t(x))$$

The Dialectica functional interpretation



Gödel, 1958

For each formula φ of \mathcal{L}_A , there is a **quantifier-free** T-formula $\varphi_D^N(x, y)$ and a T-term $t(x)$ s.t.:

$$\text{PA} \vdash \varphi \quad \Longrightarrow \quad \text{T} \vdash \varphi_D^N(x, t(x))$$

This **finitistically reduces** the consistency of PA to the **termination** of higher-typed programming language.

Level-by-level refinement

- 1 Recursion complexity vs induction complexity
- 2 Higher type recursion
- 3 Beyond PA and ε_0**
- 4 Conclusions
- 5 Break: questions and exercises
- 6 References



*“Indeed, it seems not entirely unreasonable to me to suppose that **contradictions** might possibly be concealed even in **classical analysis**. . . . the most important [consistency] proof of all in practice, that for analysis, is still **outstanding**.”*



*“Indeed, it seems not entirely unreasonable to me to suppose that **contradictions** might possibly be concealed even in **classical analysis**. . . . the most important [consistency] proof of all in practice, that for analysis, is still **outstanding**.”*

85 YEARS LATER: this is still the *biggest open problem* in proof theory!

Analysis: second-order Peano Arithmetic

The language \mathcal{L}_2 of PA2 admits a sort not only for $\mathbb{N}(x, y, \dots)$, but also type I objects (sets, functions, etc.) (X, Y, \dots) .

Crucially we have a **comprehension** axiom,

$$\exists X \forall x (X(x) \leftrightarrow \varphi(x))$$

for each formula $\varphi(x)$.

We can think of X as:

- A defined **predicate**.
- A **set** of natural numbers, with $X(x)$ meaning $x \in X$.
- A **real number**, with $X(x)$ being the x^{th} bit of X .

Reduction to pure logic

Unlike PA, we can reduce PA2 to **pure second-order logic**:

$$N(x) \iff \forall X(X(0) \wedge \forall y(X(y) \rightarrow X(sy)) \rightarrow X(x))$$

This gives us a definition of \mathbb{N} whence we **recover** the induction principle.

This **reduces** Gentzen's problem, consistency of PA2 to cut-elimination for second-order logic...

Reduction to pure logic

Unlike PA, we can reduce PA2 to **pure second-order logic**:

$$\mathbb{N}(x) \iff \forall X(X(0) \wedge \forall y(X(y) \rightarrow X(sy)) \rightarrow X(x))$$

This gives us a definition of \mathbb{N} whence we **recover** the induction principle.

This **reduces** Gentzen's problem, consistency of PA2 to cut-elimination for second-order logic...

...but there is **no free lunch**:

$$\frac{\frac{\frac{\text{P}}{\vdash \Gamma, \varphi(\psi)}}{\vdash \Gamma, \exists X \varphi(X)} \quad \frac{\frac{\text{Q}(Y)}{\vdash \Gamma', \bar{\varphi}(Y)}}{\vdash \Gamma', \forall X \bar{\varphi}(X)}}{\vdash \Gamma, \Gamma'} \quad \rightsquigarrow \quad \frac{\frac{\text{P}}{\vdash \Gamma, \varphi(\psi)} \quad \frac{\text{Q}(\bar{\psi})}{\vdash \Gamma', \bar{\varphi}(\bar{\psi})}}{\vdash \Gamma, \Gamma'} \text{cut}$$

Takeuti's conjecture, 1953



Does cut-elimination hold for second-order logic?

Building on foundational work of Schütte:



Theorem (Tait '66)

Cut is admissible for second-order logic.



Theorem (Takahashi '67, Prawitz '68)

Cut is admissible in Church's simple type theory, i.e. at all finite types.

Building on foundational work of Schütte:



Theorem (Tait '66)

Cut is admissible for second-order logic.



Theorem (Takahashi '67, Prawitz '68)

Cut is admissible in Church's simple type theory, i.e. at all finite types.

However, these arguments are somewhat **unsatisfactory**:

- They are *non-explicit*: we have **no proof-theoretic ordinal** for PA2.
- They are *admissibility* results: there is **no computational process**.

The success of proof interpretations



Theorem (Spector '62)

PA2 is ND-interpreted into an extension of T by **bar recursion**.

Theorem (Girard '71)

PA2 is ND-interpreted into a **second-order** extension of T.



The story since Gentzen: the first steps into impredicativity

There has nonetheless been significant progress since the days of Gentzen:



Takeuti '67

Ordinal analysis of theories of Π_1^1 -comprehension.

Rathjen '90s-'00s

Ordinal analysis of theories of Π_2^1 -comprehension.



An excellent reference:

- [Rathjen and Sieg, 2022]

“He once confided in me that he was really quite content since now he at last had time to think about a consistency proof for analysis. He was in fact fully convinced that he would succeed in carrying out such a proof.”

- 1 Recursion complexity vs induction complexity
- 2 Higher type recursion
- 3 Beyond PA and ε_0
- 4 Conclusions**
- 5 Break: questions and exercises
- 6 References

Conclusions

There are *many* other legacies of proof theory we have not discussed:¹

- The application of proof theory to **complexity theory**.
- The application of proof theory to **set theory**.
- The application of proof theory to **type theory** and constructive mathematics.

These all constitute **highly active** areas of research, that are certainly beyond the scope of this course!²

¹ ...and I did not have time to write slides about!

² ...but you can ask me for **references**.

Conclusions

There are *many* other legacies of proof theory we have not discussed:¹

- The application of proof theory to **complexity theory**.
- The application of proof theory to **set theory**.
- The application of proof theory to **type theory** and constructive mathematics.

These all constitute **highly active** areas of research, that are certainly beyond the scope of this course!²

Thank you.

¹ ...and I did not have time to write slides about!

² ...but you can ask me for **references**.

- 1 Recursion complexity vs induction complexity
- 2 Higher type recursion
- 3 Beyond PA and ε_0
- 4 Conclusions
- 5 Break: questions and exercises**
- 6 References

Exercises

- 1 We can think of functions $\mathbb{N} \rightarrow \mathbb{N}$ as *streams* (i.e. infinite sequences) of natural numbers. Use (higher-order) primitive recursion to define functionals:
 - $\text{hd} : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ returning the first element of a stream input.
 - $\text{tl} : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow (\mathbb{N} \rightarrow \mathbb{N})$ by $(n_i)_{i \geq 0} \mapsto (n_i)_{i \geq 1}$.
- 2 Show that the two recursive definitions of the Ackermann-Péter function, A (by higher-ordinal recursion) and A' (by higher-type recursion) coincide.
- 3 Show that second-order logic with equality can be interpreted in second-order logic *without* equality.
- 4 A function f on the ordinals is **normal** if:
 - (Strict monotonicity.) $\alpha < \beta \implies f(\alpha) < f(\beta)$.
 - (Continuity.) λ is a limit ordinal $\implies f(\lambda) = \sup_{\alpha < \lambda} (f(\alpha))$.

Show that:

- a f commutes with suprema, i.e. $f(\sup A) = \sup_{\alpha \in A} f(\alpha)$.
 - b $f(\alpha) \geq \alpha$ for all ordinals α .
 - c f has arbitrarily high fixed points, i.e. for all α there is $\gamma \geq \alpha$ with $f(\gamma) = \gamma$.
- 5 (**Long.**) Show that each primitive recursive function $f(\vec{x})$ is dominated by $A(m, \max(\vec{x}))$, for some $m \in \mathbb{N}$. Conclude that A is not primitive recursive.
 - 6 Write $I\Sigma_1$ and III_1 for the fragments of PA with induction on only Σ_1 -formulas and Π_1 -formulas, respectively. Show that $I\Sigma_1 = III_1$.
(Hard.) Show that $I\Sigma_1$ is even equal to the fragment of PA with induction only on *Boolean combinations* of Σ_1 -formulas.

Outline

- 1 Recursion complexity vs induction complexity
- 2 Higher type recursion
- 3 Beyond PA and ε_0
- 4 Conclusions
- 5 Break: questions and exercises
- 6 **References**

References I

Avigad, J. and Feferman, S. (1998).

Gödel's functional ("Dialectica") interpretation.

In Buss, S. R., editor, *Handbook of Proof Theory*, volume 137, pages 337–405. Elsevier.

Kohlenbach, U. (2008).

Applied Proof Theory - Proof Interpretations and their Use in Mathematics.

Springer Monographs in Mathematics. Springer.

Rathjen, M. and Sieg, W. (2022).

Proof Theory.

In Zalta, E. N. and Nodelman, U., editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2022 edition.

Takeuti, G. (1975).

Proof Theory.

New York, N.Y., U.S.A.: Sole distributors for the U.S.A. and Canada, Elsevier Science Pub. Co.