PROOF THEORY OF ARITHMETIC

Lecture 4 – Provably total recursive functions

Anupam Das

University of Birmingham

34TH EUROPEAN SUMMER SCHOOL
IN LOGIC, LANGUAGE AND INFORMATION

Ljubljana, Slovenia

10 ~~9~~ August 2023

These slides are available at http://www.anupamdas.com/esslli23.

Some perspectives

We reduced PA to judgements $\vdash_{\overline{d}}$, for $d < \omega$, finitistically (i.e. in PRA).

We proved a *cut-elimination* theorem, i.e. $\vdash_{\overline{d}} \subseteq \vdash_{0}$, by induction on ordinals $< \varepsilon_0$, by consideration of the height of derivations. Thus:

Theorem

PRA $+$ *"induction up to ε_0" proves the consistency of* PA.

(We will make this more formal today.)

NB. We have finitistically reduced the entire consistency of PA to a 'simple' computational principle: induction (or *recursion*) up to ε_0.

NB. It is immediate that PA cannot prove well-foundedness of (any appropriate representation of) ε_0. *What about smaller ordinals?*

Outline

Let us make more structural observation:

Proposition (A 'Herbrand' theorem)

If $\vdash_0 \exists x \varphi_0(x)$, *for* $\varphi_0(x)$ *quantifier-free (or even* Δ_0*), then,*

$$\vdash_0 \varphi_0(n_0), \ldots, \varphi_0(n_{k-1})$$

for some $\vec{n} \in \mathbb{N}$.

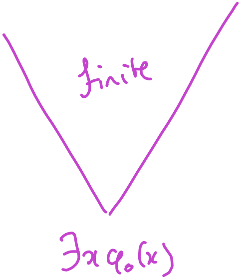
- even Σ_1

cut-free

no (unbounded) $\forall$

formally by ind. on $\vdash_0$.

$$\frac{\Gamma, \exists x \varphi_0(x), \varphi_0(\underline{n})}{\Gamma, \exists x \varphi_0(x)}$$



so there are only finitely many $\exists$-steps, say, introducing $n_0, \ldots, n_{k-1}$

Provably total recursive functions

Let $f : \mathbb{N} \to \mathbb{N}$ be a recursive function. We say that PA **(recursively) (well-)defines** f if there is a Σ_1 formula $\varphi_f(x, y)$ with:

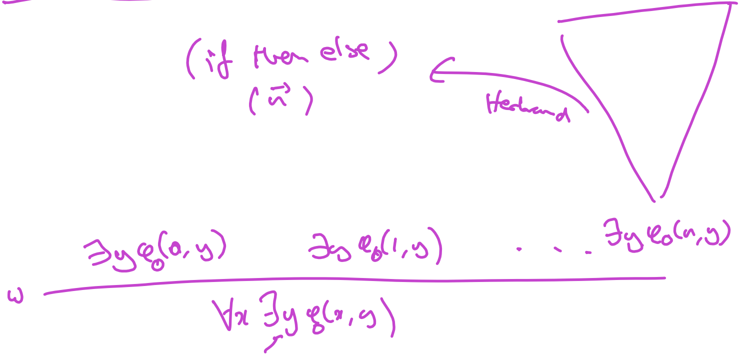
- $\mathfrak{N} \models \varphi_f(m, n) \iff f(m) = n$. : φ_f computes graph of f
- PA $\vdash \forall x \, \exists y \, \varphi_f(x, y)$.

[opt: PA $\vdash \forall x, y, y' \, (\varphi(x,y) \to \varphi(x,y') \to y = y')$]

Corollary

The definable functions of PA *are definable by* recursion on ordinals $< \varepsilon_0$.

Again, *what about smaller ordinals?*



Example: Exponentiation

2^x /

$2^0 = 1$

$2^{x+1} = 2 \cdot 2^x$

Exp(x, y) for "∃z codes a sequence ⟨1, 2, 4, … y⟩
and |z| = x+1"

Outline

A puzzle: Hydra!

Let $\prec \subseteq \mathbb{N} \times \mathbb{N}$ be given by a formula of $\mathcal{L}_A$.
The **$\prec$-induction** principle is:

$$\prec\text{-ind} \quad : \quad \forall x\, (\forall y \prec x\, \varphi(y) \rightarrow \varphi(x)) \rightarrow \forall x\, \varphi(x)$$

Proposition

If $\prec$ is well-founded, then $\mathfrak{N} \models \prec$-ind.

In fact, we can arithmetise this result. The **$\prec$-well-foundedness** principle is:

$$\prec\text{-wf} \quad : \quad \exists x\, \varphi(x) \rightarrow \exists x\, (\varphi(x) \wedge \forall y \prec x\, \neg\varphi(y))$$

Exercise

Show that $\mathsf{PA} + \prec$-wf is equivalent to $\mathsf{PA} + \prec$-ind.

Recursion on a well-founded relation

Induction has a natural computational counterpart: recursion.
To define this properly, let us first define:

$$(\lambda y \prec x f(y))(z) \ := \begin{cases} f(z) & z \prec x \\ 0 & \text{otherwise} \end{cases}$$

Now, we define $\prec$**-recursion** ($\prec$-rec) as:

- From a term $t[g]$, for g a fresh function symbol, define a function f satisfying:

$$f(x) = t[\lambda y \prec x f(y)]$$

Definition ($\prec$-recursive functions)

The $\prec$**-recursive functions** are the smallest extension of the primitive recursive functions closed under $\prec$-rec.

Exercise (Course of Values)

Show that the primitive recursive functions are already closed under $\prec$-rec.

Example: Ackermann-Péter

(Blank slide)

Proposition

PA *defines all primitive recursive functions*.

Proof idea.

By straightforward induction on the definition of a primitive recursive function.
Only induction on Σ_1 (or Π_1) formulas is needed. □

Let $\prec \subseteq \mathbb{N} \times \mathbb{N}$ be a primitive recursive relation (and so is provably Δ_1 in PA).

Theorem

If $\mathsf{PA} \vdash \prec\text{-wf}$, *then the definable functions of* PA *are closed under* $\prec$-rec.

In fact, a more complex result (beyond the scope of this course) is:

Theorem (Gentzen)

If $\prec$ *is a well-order s.t.* $\mathsf{PA}(X) \vdash \prec\text{-wf}$, *for* X *a new predicate symbol, then* $|\prec| < \varepsilon_0$.

Outline

Proposition (Cantor Normal Form)

For each ordinal α, there are unique ordinals $\alpha_0 \geqslant \cdots \geqslant \alpha_{n-1}$ such that:

$$\alpha = \omega^{\alpha_0} + \cdots + \omega^{\alpha_{n-1}}$$

Notice that, for $\alpha < \varepsilon_0$, necessarily $\alpha_0 < \alpha$. Let us call this decomposition the **Cantor Normal Form (CNF)** of α.

Comparison of ordinals $< \varepsilon_0$ can now be reduced to a lexicographical comparison of their CNFs:

Proposition (Comparison of CNFs)

For $\beta < \varepsilon_0$ with CNF $\omega^{\beta_0} + \cdots + \omega^{\beta_{m-1}}$, and α as above, we have

$$\alpha < \beta \iff \alpha_0 = \beta_0, \ldots, \alpha_{i-1} = \beta_{i-1}, \alpha_i < \beta_i$$

for some $i < n$.

(Blank slide)

This allows us to easily code ordinals $< \varepsilon_0$ by a rudimentary *notation* recursively:

$$\ulcorner \alpha \urcorner = \langle \ulcorner \alpha_0 \urcorner, \dots, \ulcorner \alpha_{n-1} \urcorner \rangle$$

For simplicity we shall construe *metavariables* α, β etc. as ranging over such codes.

For α as above, let us write $l(\alpha) := n$ (which is bounded above by $\ulcorner \alpha \urcorner$).

Definition

The **rank** $\mathrm{rk}(\alpha)$ of ordinals $\alpha < \varepsilon_0$ is defined by induction on CNFs:

- $\mathrm{rk}(\omega^{\alpha_0} + \cdots + \omega^{\alpha_{n-1}}) := \max\limits_{i<n}(\mathrm{rk}(\alpha_i) + 1)$.

Intuitively, the rank of an ordinal is the maximal height of ω-towers in its CNF.
Moreover, we have a recursive definition of comparison at rank $\leqslant r$ by:

$$\alpha <_{r+1} \beta := \exists i < l(\alpha)\, (\alpha_i <_r \beta_i \wedge \forall j < i\, \alpha_i = \beta_i)$$

(Blank slide)

Provable well-foundedness of ordinals $< \varepsilon_0$

Our main Lemma is:

Lemma
If $\mathsf{PA} \vdash <_r\text{-wf}$ *then* $\mathsf{PA} \vdash <_{r+1}\text{-wf}$.

NB: this crucially requires induction of logical complexity increasing in r.

Theorem
$\mathsf{PA} \vdash <_r\text{-wf}$, *for each* $r < \omega$.

Since each ordinal $< \varepsilon_0$ has a finite rank, we have:

Corollary
$\mathsf{PA} \vdash \alpha\text{-wf}$ *for (an appropriate representation of) each* $\alpha < \varepsilon_0$.

Corollary
The definable functions of PA *are closed under recursion up to any ordinal* $< \varepsilon_0$.

(Blank slide)

Outline

1. Prove over PA that $\prec$-well-foundedness is equivalent to $\prec$-induction.
2. Extend Herbrand's theorem to Σ_1 formulas $\varphi_0(x)$.
3. Show that PA well-defines all primitive recursive functions.
   Hint: proceed by induction on the definition of a primitive recursive function. You should only have to use Σ_1 (or Π_1) induction.
4. Prove in PA that the (graph of) the Ackermann-Péter function is total.
   Hint: you must use induction on at least Π_2 (or Σ_2) formulas, assuming that the graph of Ackermann-Péter is Σ_1.
5. An ε ordinal is a fixed point $\varepsilon = \omega^\varepsilon$. What are the CNFs of ε numbers?
6. **(Long.)** Show that the primitive recursive functions are closed under $<$-rec, where $<$ is the usual order on $\mathbb{N}$.
7. **(Hard.)** Use the Cantor Normal Form of ordinals $< \varepsilon_0$ to induce a well-ordering of finite ordered trees. Use this to show that Hercules always defeats Hydra.

(Blank slide)

Outline

Summary

We have shown that PA can prove the well-foundedness of (appropriate representations of) all ordinals $< \varepsilon_0$.

We thus arrive at two classifications:

Theorem (Consistency strength)

$\mathsf{PRA} + {<}\varepsilon_0\text{-}\mathsf{ind}$ *proves the consistency of* PA. *Moreover,* ε_0 *is the least such ordinal.*

Theorem (Computational strength)

The definable functions of PA *are precisely the* ${<}\varepsilon_0$*-recursive functions.*

Many other characterisations of PA's computational strength are known, notably via higher type recursion, but that is a story for another day!