

**Update procedures and the 1-consistency  
of arithmetic**

Jeremy Avigad

Carnegie Mellon University

avigad@cmu.edu

<http://www.andrew.cmu.edu/~avigad>

**The ordinal  $\varepsilon_0$**

$\varepsilon_0$  is defined to be the limit of the sequence

$$\omega, \omega^\omega, \omega^{\omega^\omega}, \dots$$

Every nonzero  $\alpha < \varepsilon_0$  can be written in *Cantor normal form*:

$$\alpha = \omega^{\alpha_1} \cdot n_1 + \dots + \omega^{\alpha_k} \cdot n_k$$

where  $\alpha > \alpha_1 > \dots > \alpha_k$  and  $n_i \in \mathbb{N}$ .

So there is an effective (primitive recursive, or even elementary recursive) set of notations, such that the associated ordering is also effective.

## Ordinal recursive functionals

### Ordinal recursive functions

An  $\alpha$ -recursive function is given by elementary functions  $start(\vec{x})$ ,  $next(q)$ ,  $norm(q)$ ,  $result(q)$ .

These data define a function  $F(\vec{x})$ :

```
clock  $\leftarrow$   $\alpha$ 
state  $\leftarrow$   $start(\vec{x})$ 
while  $norm(state) \prec clock$  do
  clock  $\leftarrow$   $norm(state)$ 
  state  $\leftarrow$   $next(state)$ 
return  $result(state)$ 
```

The previous definition relativizes well.

A  $\alpha$ -recursive functional  $F(\vec{x}, f_1, \dots, f_k)$  is given by elementary functions  $start(\vec{x})$ ,  $next(q, u_1, \dots, u_k)$ ,  $query_1(q)$ ,  $\dots$ ,  $query_k(q)$ ,  $norm(q)$ , and  $result(q)$ .

These define a functional  $F(\vec{x}, f_1, \dots, f_k)$ :

```
clock  $\leftarrow$   $\alpha$ 
state  $\leftarrow$   $start(\vec{x})$ 
while  $norm(state) \prec clock$  do
  clock  $\leftarrow$   $norm(state)$ 
  state  $\leftarrow$   $next(state, f_1(query_1(state)), \dots,$ 
     $f_k(query_k(state)))$ 
return  $result(state)$ 
```

## The ordinal analysis of PA

**Theorem.** Suppose Peano Arithmetic proves  $\forall x \exists y \varphi(x, y)$ , for a  $\Sigma_1$  formula  $\varphi$ . Then there is a  $< \varepsilon_0$ -recursive function  $F$  such that for every  $x$ ,  $\varphi(x, F(x))$ .

Notes:

- The statement can be relativized to a function parameter.
- A suitable formalization can be proved in primitive recursive arithmetic.
- This yields the usual results of the ordinal analysis of  $PA$ .

## Embedding PA in a quantifier-free calculus

Iteratively introduce Skolem functions for quantifier-free formulae:

$$\varphi(\vec{x}, y) \rightarrow \varphi(\vec{x}, \mu_\varphi(\vec{x})) \wedge \mu_\varphi(\vec{x}) \leq y$$

In other words,  $\mu_\varphi(\vec{x})$  returns a least  $y$  satisfying  $\varphi(\vec{x}, y)$ , if there is one.

**Proposition.** If  $PA$  proves  $\forall x \exists y \psi(x, y)$  with  $\psi$  quantifier-free, then for each  $n$  there is a *propositional* proof of  $\varphi(\vec{n}, \mu_\varphi(\vec{n}))$  from closed instances of  $\mu$  axioms, the definitions of the initial functions and relations  $(0, 1, +, \times, <, \dots)$ , and axioms of equality.

## The task

Given a finite set  $S$  of closed instances of  $\mu$  axioms, find a finite arithmetic interpretation of the  $\mu$  symbols.

Idea: assume everything returns 0 by default. Suppose the following instance of an axiom fails under this interpretation:

$$\varphi(\vec{s}, t) \rightarrow \varphi(\vec{s}, \mu_\varphi(\vec{s})) \wedge \mu_\varphi(\vec{s}) \leq t.$$

Correct it by mapping  $\mu_\varphi(\vec{s}) = t$ . Iterate.

The difficulty:

- Can order  $\mu_1, \mu_2, \dots, \mu_n$  so that the definition of  $\mu_i$  involves only  $\mu_j$  with  $j < i$ .
- But the terms  $\vec{s}$  and  $t$  above can involve any  $\mu_i$ .

## Overview

1. We will define the notion of a “nested system of update procedures.”
2. By general continuity considerations, these always have solutions.
3. The task on the previous slide amounts to finding solutions to elementary systems of equations.
4. Ordinals can be used in place of continuity.

This is essentially a repackaging of Ackermann’s proof, using the Hilbert substitution method. The emphasis on continuity dates back to Tait ’65.

## Background definitions

Let  $\rho, \sigma, \tau, \dots$  range over finite partial functions from  $\mathbb{N}$  to  $\mathbb{N}$ .

Let  $\hat{\sigma}$  to denote the extension to a total function:

$$\hat{\sigma}(x) = \begin{cases} \sigma(x) & x \in \text{dom}(\sigma) \\ 0 & \text{otherwise.} \end{cases}$$

Let  $\sigma \oplus \langle u, v \rangle$  to denote the modification of  $\sigma$  that maps  $u$  to  $v$ :

$$(\sigma \oplus \langle u, v \rangle)(x) = \begin{cases} \sigma(x) & \text{if } x \in \text{dom}(\sigma), x \neq u \\ v & \text{if } x = u \\ \text{undef.} & \text{otherwise} \end{cases}$$

Define  $\sigma \oplus \emptyset$  to be  $\sigma$ .

## Update procedures

A functional  $F(f_1, \dots, f_k)$  is *continuous* if its value depends on only finitely many values of  $f_1, \dots, f_k$ .

Suppose  $F(g, f_1, \dots, f_k)$  is continuous with range  $\mathbb{N} \times \mathbb{N} \cup \{\emptyset\}$ . Consider the map

$$\sigma \mapsto \sigma \oplus F(\hat{\sigma}, f_1, \dots, f_k).$$

$F$  is an *update procedure* in  $g$  if the following holds: whenever

- $F(\hat{\sigma}, f_1, \dots, f_k) = \langle a, b \rangle$ ,
- $\tau$  extends  $\sigma \oplus \langle a, b \rangle$ , and
- $F(\hat{\tau}, h_1, \dots, h_k) = \langle a, c \rangle$ ,

then  $b = c$ .

In other words, once  $F$  “sets”  $\sigma(a)$  to  $b$ , it does not change it, no matter how the other arguments vary.

## Fixed points

If  $F(g)$  is a unary update procedure, a *finite fixed point* of  $F$  is a  $\sigma$  such that

$$\sigma = \sigma \oplus F(\hat{\sigma}).$$

**Lemma.**  $F(g)$  has a finite fixed point.

*Proof.* Let  $\sigma^0 = \emptyset$ , and for each  $i$ , let  $\sigma^{i+1} = \sigma^i \oplus F(\hat{\sigma}^i)$ . Let  $g = \bigcup_{i \in \mathbb{N}} \sigma^i$ . By continuity, for some  $i$  we have  $F(\hat{g}) = F(\hat{\sigma}^i)$ .  $\square$

The proof shows that if  $F(g, \vec{h})$  is continuous, and for each  $\vec{h}$

$$g \mapsto F(g, \vec{h})$$

is an update procedure, there is a continuous functional  $G(\vec{h})$  returning fixed points.

## Nested update procedures

A *system of nested update procedures* is a sequence of continuous functionals

$F_1(f_1, \dots, f_n), \dots, F_n(f_1, \dots, f_n)$  such that for each  $i$  and fixed  $f_1, \dots, f_{i-1}$ , the functional

$$f_i, f_{i+1}, \dots, f_n \mapsto F_i(f_1, \dots, f_n)$$

is an update procedure for  $f_i$ .

A *finite fixed point* of such a system is a sequence of finite partial functions  $\sigma_1, \dots, \sigma_n$  such that the equations

$$\begin{aligned} \sigma_1 &= \sigma_1 \oplus F_1(\hat{\sigma}_1, \dots, \hat{\sigma}_n) \\ &\vdots \\ \sigma_n &= \sigma_n \oplus F_n(\hat{\sigma}_1, \dots, \hat{\sigma}_n) \end{aligned}$$

are all satisfied.

## Finding fixed points

**Theorem.** Every system of nested update procedures has a finite fixed point.

*Proof.* Use induction on  $n$ . We have already taken care of  $n = 1$ .

For the induction step, given  $F_1, \dots, F_{n+1}$ , let  $G(f_1, \dots, f_n)$  be a continuous functional returning finite fixed points of the functional

$$f_{n+1} \mapsto F_{n+1}(f_1, \dots, f_{n+1}).$$

Then

$$f_1, \dots, f_n \mapsto F_i(f_1, \dots, f_n, G(f_1, \dots, f_n))$$

for  $i = 1, \dots, n$  is a system of nested update procedures of size  $n$ .

By the IH, the smaller system has a finite fixed point,  $\sigma_1, \dots, \sigma_n$ . Let  $\sigma_{n+1} = G(\hat{\sigma}_1, \dots, \hat{\sigma}_n)$ .  $\square$

## The main theorem

Now restrict to systems of nested update procedures given by *elementary* functions  $F_1, \dots, F_n$ .

**Theorem.** The following are pairwise equivalent:

1. Every  $\prec_{\varepsilon_0}$ -recursive function is total.
2. Every system of nested elementary update procedures has a finite fixed point.
3. Every  $\Pi_2$  sentence provable in  $PA$  is true.

This theorem is provable in  $PRA$ .

## Finding finite partial models

### Fixed-point equations and arithmetic

Remember that Peano arithmetic can be embedded in a q.f. theory based on  $\mu$  axioms:

$$\varphi(\vec{x}, y) \rightarrow \varphi(\vec{x}, \mu_\varphi(\vec{x})) \wedge \mu_\varphi(\vec{x}) \leq y$$

**Lemma.** Suppose every system of nested elementary update procedures has a finite fixed point. Then every set of closed instances of  $\mu$ -axioms has a finite partial model.

The conclusion implies (and is in fact equivalent to) the  $\Pi_2$  soundness of arithmetic.

Let  $S$  be a finite set of closed instances of  $\mu$ -axioms. Let  $\mu_1, \dots, \mu_n$  be the  $\mu$ -symbols mentioned in  $S$ , such that if the definition of  $\mu_i$  involves  $\mu_j$ , then  $i > j$ .

For each  $i$ , let  $F_i(f_1, \dots, f_n)$  find an instance of the defining axiom for  $\mu_i$  in  $S$ ,

$$\theta(\vec{s}, t) \rightarrow \theta(\vec{s}, \mu_i(\vec{s})) \wedge \mu_i(\vec{s}) \leq t.$$

that is falsified under the assignment

$$\mu_1, \dots, \mu_n \mapsto f_1, \dots, f_n.$$

Update  $\mu_i(\vec{s}) \mapsto \mu m \leq t^{\vec{f}} (\theta(\vec{s}, m))^{\vec{f}}$ .

Then:

- This is a system of nested update procedures.
- A fixed point is a finite interpretation of  $\mu_1, \dots, \mu_n$  satisfying  $S$ .



## Finding a fixed-point

We only need to show that one can find fixed-points using ordinal recursion (instead of continuity).

**Lemma.** Suppose  $F(g, \vec{h})$  is  $\alpha$ -recursive, and for each  $\vec{h}$ ,  $g \mapsto F(g, \vec{h})$  is an update procedure. Then there is an  $\omega^\alpha$ -recursive functional  $G(\vec{h})$  that returns finite fixed points.

Idea: start with  $i = 0, \sigma^0 = \emptyset$ . Then

- Compute  $F(\hat{\sigma}^i, \vec{h})$ .
- Update:  $\sigma^{i+1} = \sigma^i \oplus F(\hat{\sigma}^i, \vec{h})$ .
- If the computation sequence for  $F(\hat{\sigma}^i, \vec{h})$  is no longer valid, revise it, and compute  $F(\hat{\sigma}^{i+1}, \vec{h})$ .
- Repeat.

To show this converges, it suffices to assign ordinals to steps.

## Assigning ordinals

Consider a partial computation sequence

$$s_0, s_1, \dots, s_m$$

of  $F(\hat{\sigma}^i, \vec{h})$ , with norms

$$\alpha_0, \alpha_1, \dots, \alpha_m.$$

Assign to this the ordinal

$$\omega^{\alpha_0} \cdot \delta_0 + \dots + \omega^{\alpha_{m-1}} \cdot \delta_{m-1} + \omega^{\alpha_m} \cdot (\delta_m + 1),$$

where

$$\delta_i = \begin{cases} 2 & \text{if } \text{query}_{H,1}(s_i) \text{ is not } \text{dom}(\sigma) \\ 0 & \text{otherwise.} \end{cases}$$

Two cases:

1. Computation of  $F$  is not done. Take the next step.
2. Computation of  $F$  is done. Update  $\sigma^i$  and revise computation.

### First case

If the computation of  $F$  is not done,

- let  $s_{m+1}$  the next state in the computation of  $F(\sigma^i, \vec{h})$ , and
- let  $\alpha_{m+1}$  be the corresponding norm.

Then the ordinal drops from

$$\omega^{\alpha_0} \cdot \delta_0 + \dots + \omega^{\alpha_{m-1}} \cdot \delta_{m-1} + \omega^{\alpha_m} \cdot (\delta_m + 1),$$

to

$$\omega^{\alpha_0} \cdot \delta_0 + \dots + \omega^{\alpha_{m-1}} \cdot \delta_{m-1} + \omega^{\alpha_m} \cdot \delta_m + \omega^{\alpha_{m+1}} \cdot (\delta_{m+1} + 1).$$

### Second case

Otherwise, let  $\sigma^{i+1} = \sigma^i \oplus F(\hat{\sigma}^i, \vec{h})$ .

If this invalidates the computation sequence for  $F(\hat{\sigma}^{i+1}, \vec{h})$ , let  $j$  be the first point at which the new value is queried.

In other words, the computation sequence is cut back from

$$s_0, \dots, s_j, \dots, s_m$$

to

$$s_0, \dots, s_j.$$

The norm drops from

$$\begin{aligned} \omega^{\alpha_0} \cdot \delta_0 + \dots + \omega^{\alpha_{j-1}} \cdot \delta_{j-1} + \omega^{\alpha_j} \cdot 2 + \dots \\ \omega^{\alpha_{m-1}} \cdot \delta_{m-1} + \omega^{\alpha_m} \cdot (\delta_m + 1), \end{aligned}$$

to

$$\omega^{\alpha_0} \cdot \delta_0 + \dots + \omega^{\alpha_{j-1}} \cdot \delta_{j-1} + \omega^{\alpha_j} \cdot 1$$

since  $\delta_j$  has dropped from 2 to 0.

## Finishing it off

**Theorem.** Suppose every  $\prec_{\varepsilon_0}$ -recursive function is total. Then every system of nested elementary update procedures has a finite fixed point.

*Proof.* Let  $F_1, \dots, F_n$  be a system of nested update procedures.

As before, use induction (recursion) up to  $n$ ; the previous lemma handles the induction step.  $\square$

## Final remarks

The approach

- Yields sharp bounds for fragments of arithmetic.
- Works for transfinite induction.

Related approaches to the OA of

- Systems with transfinite jump hierarchies (predicative analysis)
- Admissible set theory

can be put in this framework.

Questions:

- Can one eliminate the nesting?
- Will this lead to interesting computational or combinatorial principles?