

CHAPTER 4

Initial Cases of Transfinite Induction

The goal here is to study the in a sense “most complex” proofs in first-order arithmetic. The main tool for proving theorems in arithmetic is clearly the induction schema

$$A(0) \rightarrow \forall_x (A(x) \rightarrow A(Sx)) \rightarrow \forall_x A(x).$$

An equivalent form of this schema is “course-of-values” or cumulative induction

$$\forall_x (\forall_{y < x} A(y) \rightarrow A(x)) \rightarrow \forall_x A(x).$$

Both schemes refer to the standard ordering of $\mathbb{N}$. It is tempting to try to strengthen arithmetic by allowing more general induction schemes, e.g., w.r.t. the lexicographical ordering of $\mathbb{N} \times \mathbb{N}$. Even more generally, let $\prec$ be a well-ordering of $\mathbb{N}$ and use *transfinite induction*:

$$\forall_x (\forall_{y \prec x} A(y) \rightarrow A(x)) \rightarrow \forall_x A(x).$$

It can be understood as

Suppose the property $A(x)$ is “progressive”, i.e., from the validity of $A(y)$ for all $y \prec x$ we can conclude that $A(x)$ holds. Then $A(x)$ holds for all x .

For which well-orderings this schema is derivable in arithmetic? We will discuss a classic result of Gentzen (1943) which in a sense answers this question completely. However, to state the result we have to be more explicit about the well-orderings used.

4.1. Ordinals below ε_0

We need some knowledge and notations for ordinals. This can be done without relying on set theory: we introduce an initial segment of the ordinals (the ones $< \varepsilon_0$) in a formal, combinatorial way, i.e., via ordinal notations based on “Cantor normal form”. From now on “ordinal” means “ordinal notation”.

DEFINITION. We define

- (a) α is an ordinal,
- (b) $\alpha < \beta$ for ordinals α, β .

simultaneously by induction, as follows.

- (a) If $\alpha_m, \dots, \alpha_0$ are ordinals, $m \geq -1$ and $\alpha_m \geq \dots \geq \alpha_0$ (where $\alpha \geq \beta$ means $\alpha > \beta$ or $\alpha = \beta$), then

$$\omega^{\alpha_m} + \dots + \omega^{\alpha_0}$$

is an ordinal. The empty sum (denoted by 0) is allowed.

- (b) If $\omega^{\alpha_m} + \dots + \omega^{\alpha_0}$ and $\omega^{\beta_n} + \dots + \omega^{\beta_0}$ are ordinals, then

$$\omega^{\alpha_m} + \dots + \omega^{\alpha_0} < \omega^{\beta_n} + \dots + \omega^{\beta_0}$$

iff there is an $i \geq 0$ such that $\alpha_{m-i} < \beta_{n-i}$, $\alpha_{m-i+1} = \beta_{n-i+1}$, $\dots$, $\alpha_m = \beta_n$, or else $m < n$ and $\alpha_m = \beta_n, \dots, \alpha_0 = \beta_{n-m}$.

We shall use the notation:

$$\begin{aligned} 1 &:= \omega^0, \\ k &:= \omega^0 + \dots + \omega^0 \quad \text{with } k \text{ copies of } \omega^0, \\ \omega^\alpha k &:= \omega^\alpha + \dots + \omega^\alpha \quad \text{with } k \text{ copies of } \omega^\alpha. \end{aligned}$$

The *level* of an ordinal is defined by $\text{lev}(0) := 0$, $\text{lev}(\omega^{\alpha_m} + \dots + \omega^{\alpha_0}) := \text{lev}(\alpha_m) + 1$. For ordinals of level $k + 1$ we have $\omega_k \leq \alpha < \omega_{k+1}$, where $\omega_0 := 0$, $\omega_1 := \omega^1$, $\omega_{k+1} := \omega^{\omega_k}$.

LEMMA 4.1.1. $<$ is a linear order with 0 the least element.

PROOF. By induction on the levels. □

EXAMPLE.

$$\begin{aligned} 0 &< 1 < 2 \dots < \omega < \omega + 1 \dots < \omega 2 < \omega 2 + 1 \dots < \omega 3 \dots < \omega^2 \\ &< \omega^2 + 1 \dots < \omega^2 + \omega \dots < \omega^3 \dots < \omega^\omega = \omega_2 \dots < \omega_3 \dots \end{aligned}$$

DEFINITION (Addition of ordinals).

$$(\omega^{\alpha_m} + \dots + \omega^{\alpha_0}) + (\omega^{\beta_n} + \dots + \omega^{\beta_0}) := \omega^{\alpha_m} + \dots + \omega^{\alpha_i} + \omega^{\beta_n} + \dots + \omega^{\beta_0}$$

where i is minimal such that $\alpha_i \geq \beta_n$.

LEMMA 4.1.2. $+$ is an associative operation which is strictly monotonic in the second argument and weakly monotonic in the first argument.

PROOF. Exercise. □

REMARK. $+$ is *not* commutative:

$$1 + \omega = \omega \neq \omega + 1.$$

There is also a commutative version on addition, the *natural sum* (or Hessenberg sum). It is defined by

$$(\omega^{\alpha_m} + \dots + \omega^{\alpha_0}) \# (\omega^{\beta_n} + \dots + \omega^{\beta_0}) := \omega^{\gamma_{m+n}} + \dots + \omega^{\gamma_0},$$

where $\gamma_{m+n}, \dots, \gamma_0$ is a decreasing permutation of $\alpha_m, \dots, \alpha_0, \beta_n, \dots, \beta_0$. It is easy to see that $\#$ is associative, commutative and strictly monotonic in both arguments.

How ordinals of the form $\beta + \omega^\alpha$ can be approximated from below? First note that

$$\delta < \alpha \rightarrow \beta + \omega^\delta k < \beta + \omega^\alpha.$$

For any $\gamma < \beta + \omega^\alpha$ we can find a $\delta < \alpha$ and a k such that

$$\gamma < \beta + \omega^\delta k.$$

It is easy to code ordinals $< \varepsilon_0$ bijectively by natural numbers:

$$o(\ulcorner \alpha \urcorner) = \alpha \quad \text{and} \quad \ulcorner o(x) \urcorner = x$$

such that relations and operations on ordinals transfer to elementary relations and operations on $\mathbb{N}$. Abbreviations:

$$\begin{aligned} x \prec y &:= o(x) < o(y), \\ \omega^x &:= \ulcorner \omega^{o(x)} \urcorner, \\ x \oplus y &:= \ulcorner o(x) + o(y) \urcorner, \\ xk &:= \ulcorner o(x)k \urcorner, \\ \omega_k &:= \ulcorner \omega_k \urcorner. \end{aligned}$$

4.2. Provability of initial cases of transfinite induction

We will derive initial cases of transfinite induction in arithmetic:

$$\forall x (\forall_{y \prec x} Py \rightarrow Px) \rightarrow \forall_{x \prec a} Px$$

for some number a and a predicate symbol P , where $\prec$ is the standard order of order type ε_0 defined before.

REMARK. Gentzen (1943) proved that this result is optimal in the sense that for the full system of ordinals $< \varepsilon_0$ the principle

$$\forall x (\forall_{y \prec x} Py \rightarrow Px) \rightarrow \forall_x Px$$

of transfinite induction is underivable. However, we will not present a proof in these notes.

By an *arithmetical system* $\mathbf{Z}$ we mean a theory based on minimal logic in the $\forall \rightarrow$ -language (including equality axioms) such that

- (a) The language of $\mathbf{Z}$ consists of a fixed supply of function and relation constants assumed to denote computable functions and relations on the non-negative integers.
- (b) Among the function constants there is a constant S for the successor function and 0 for (the 0-place function) zero.
- (c) Among the relation constants we have $=$, P and also $\prec$ for the ordering of type ε_0 of $\mathbb{N}$, as introduced before.
- (d) *Terms* are built up from object variables x, y, z by $f(t_1, \dots, t_m)$, where f is a function constant.
- (e) We identify closed terms which have the same value; this expresses that each function constant is computable.
- (f) Terms of the form $S(S(\dots S0\dots))$ are called *numerals*. Notation: $S^n 0$ or $\underline{n}$ or just n .
- (g) *Formulas* are built up from atomic formulas $R(t_1, \dots, t_m)$, with R a relation constant, by $A \rightarrow B$ and $\forall_x A$.

The axioms of $\mathbf{Z}$ are

- Compatibility of equality

$$x = y \rightarrow A(x) \rightarrow A(y),$$

- the *Peano axioms*, i.e., the universal closures of

$$(42) \quad Sx = Sy \rightarrow x = y,$$

$$(43) \quad Sx = 0 \rightarrow A,$$

$$(44) \quad A(0) \rightarrow \forall_x (A(x) \rightarrow A(Sx)) \rightarrow \forall_x A(x),$$

with $A(x)$ an arbitrary formula.

- $R\vec{n}$ whenever $R\vec{n}$ is true (to express that R is computable).
- Irreflexivity and transitivity for $\prec$

$$x \prec x \rightarrow A,$$

$$x \prec y \rightarrow y \prec z \rightarrow x \prec z$$

Further axioms – following Schütte – are the universal closures of

$$(45) \quad x \prec 0 \rightarrow A,$$

$$(46) \quad z \prec y \oplus \omega^0 \rightarrow (z \prec y \rightarrow A) \rightarrow (z = y \rightarrow A) \rightarrow A,$$

$$(47) \quad x \oplus 0 = x,$$

$$(48) \quad x \oplus (y \oplus z) = (x \oplus y) \oplus z,$$

$$(49) \quad 0 \oplus x = x,$$

$$(50) \quad \omega^x 0 = 0,$$

$$(51) \quad \omega^x(Sy) = \omega^x y \oplus \omega^x,$$

$$(52) \quad z \prec y \oplus \omega^{Sx} \rightarrow z \prec y \oplus \omega^{e(x,y,z)} m(x, y, z),$$

$$(53) \quad z \prec y \oplus \omega^{Sx} \rightarrow e(x, y, z) \prec Sx,$$

where $\oplus$, $\lambda_{x,y}(\omega^x y)$, e and m denote function constants and A is any formula. These axioms are formal counterparts to the properties of the ordinal notations observed above.

THEOREM 4.2.1 (Provable initial cases of transfinite induction in $\mathbf{Z}$). *Transfinite induction up to ω_n , i.e., for arbitrary $A(x)$ the formula*

$$\forall x (\forall_{y \prec x} A(y) \rightarrow A(x)) \rightarrow \forall_{x \prec \omega_n} A(x),$$

is derivable in $\mathbf{Z}$.

PROOF. To every formula $A(x)$ we assign a formula $A^+(x)$ (with respect to a fixed variable x) by

$$A^+(x) := \forall_y (\forall_{z \prec y} A(z) \rightarrow \forall_{z \prec y \oplus \omega^x} A(z)).$$

We first show

If $A(x)$ is progressive, then $A^+(x)$ is progressive,

where “ $B(x)$ is progressive” means $\forall_x (\forall_{y \prec x} B(y) \rightarrow B(x))$. Assume that $A(x)$ is progressive and

$$(54) \quad \forall_{y \prec x} A^+(y).$$

Our goal is $A^+(x) := \forall_y (\forall_{z \prec y} A(z) \rightarrow \forall_{z \prec y \oplus \omega^x} A(z))$. Assume

$$(55) \quad \forall_{z \prec y} A(z)$$

and $z \prec y \oplus \omega^x$. We have to show $A(z)$.

Case $x = 0$. Then $z \prec y \oplus \omega^0$. By (46):

$$z \prec y \oplus \omega^0 \rightarrow (z \prec y \rightarrow A) \rightarrow (z = y \rightarrow A) \rightarrow A$$

it suffices to derive $A(z)$ from $z \prec y$ as well as from $z = y$. If $z \prec y$, then $A(z)$ follows from (55), and if $z = y$, then $A(z)$ follows from (55) and the progressiveness of $A(x)$.

Case Sx . From $z \prec y \oplus \omega^{Sx}$ we obtain $z \prec y \oplus \omega^{e(x,y,z)} m(x, y, z)$ by (52) and $e(x, y, z) \prec Sx$ by (53). By (54) we have $A^+(e(x, y, z))$, i.e.

$$\forall_{u \prec y \oplus \omega^{e(x,y,z)} v} A(u) \rightarrow \forall_{u \prec (y \oplus \omega^{e(x,y,z)} v) \oplus \omega^{e(x,y,z)}} A(u)$$

and hence, using (48) and (51)

$$\forall_{u \prec y \oplus \omega^{e(x,y,z)} v} A(u) \rightarrow \forall_{u \prec y \oplus \omega^{e(x,y,z)} (Sv)} A(u).$$

Also from (55) and (50), (47) we obtain

$$\forall_{u \prec y \oplus \omega^{e(x,y,z)} 0} A(u).$$

By induction:

$$\forall_{u \prec y \oplus \omega^{e(x,y,z)} m(x,y,z)} A(u)$$

and hence $A(z)$.

Next we show, by induction on n , how to derive

$$\forall_x (\forall_{y \prec x} A(y) \rightarrow A(x)) \rightarrow \forall_{x \prec \omega_n} A(x) \quad \text{for arbitrary } A(x).$$

Assume the left hand side, i.e., that $A(x)$ is progressive.

Case 0. Then $x \prec \omega^0$ and hence $x \prec 0 \oplus \omega^0$ by (49). By (46) it suffices to derive $A(x)$ from $x \prec 0$ as well as from $x = 0$. Now $x \prec 0 \rightarrow A(x)$ holds by (45), and $A(0)$ then follows from the progressiveness of $A(x)$.

Case $n+1$. Since $A(x)$ is progressive, also $A^+(x)$ is. By IH: $\forall_{x \prec \omega_n} A^+(x)$, hence $A^+(\omega_n)$ since $A^+(x)$ is progressive. By definition of $A^+(x)$ (with (45): $x \prec 0 \rightarrow A$ and (49): $0 \oplus x = x$) we obtain $\forall_{z \prec \omega^{\omega_n}} A(z)$. $\square$

REMARK. In the induction step we derived transfinite induction up to ω_{n+1} for $A(x)$ from transfinite induction up to ω_n for $A^+(x)$. Define the *level* of a formula by

$$\begin{aligned} \text{lev}(R\vec{t}) &:= 0, \\ \text{lev}(A \rightarrow B) &:= \max(\text{lev}(A) + 1, \text{lev}(B)), \\ \text{lev}(\forall_x A) &:= \max(1, \text{lev}(A)). \end{aligned}$$

Then $\text{lev}(A^+(x)) = \text{lev}(A(x)) + 1$. Hence to prove transfinite induction up to ω_n , the induction scheme in $\mathbf{Z}$ is used for formulas of level n .

4.3. Iteration operators of higher types

We have just seen that the strength of the induction scheme increases with the level of the formula proved by induction. A similar phenomenon occurs when one considers types instead of formulas, and iteration (a special case of recursion) instead of induction. Such operators have a similar relation to ordinals $< \varepsilon_0$.

DEFINITION. An ordinal $\omega^{\alpha_n} + \dots + \omega^{\alpha_0}$ is a *successor* if $\alpha_0 = 0$. It is a *limit* if α_0 is neither 0 nor a successor. For every limit $\lambda = \omega^{\alpha_n} + \dots + \omega^{\alpha_0}$ we define its *fundamental sequence* $\lambda[x]$ by

$$\lambda[x] := \begin{cases} \omega^{\alpha_n} + \dots + \omega^{\alpha_1} + \omega^{\alpha_0-1} \cdot x & \text{if } \alpha_0 \text{ is a successor} \\ \omega^{\alpha_n} + \dots + \omega^{\alpha_1} + \omega^{\alpha_0[x]} & \text{if } \alpha_0 \text{ is a limit.} \end{cases}$$

EXAMPLES.

$$\begin{aligned}\omega[x] &= x, \\ (\omega + \omega)[x] &= \omega + x, \\ \omega^2[x] &= \omega x, \\ \omega^3[x] &= \omega^2 x, \\ \omega^\omega[x] &= \omega^x.\end{aligned}$$

DEFINITION (Extended Grzegorzczuk hierarchy $(F_\alpha)_{\alpha < \varepsilon_0}$).

$$\begin{aligned}F_0(x) &:= 2^x, \\ F_{\alpha+1}(x) &:= F_\alpha^{(x)}(x) \quad (F_\alpha^{(x)} \text{ } x\text{-th iterate of } F_\alpha), \\ F_\lambda(x) &:= F_{\lambda[x]}(x).\end{aligned}$$

We also define $F_{\varepsilon_0}(x) := F_{\omega_x}(x)$.

REMARK. F_ω is a variant of the Ackermann function (1940), and the F_n for $n < \omega$ were (essentially) defined and studied by Grzegorzczuk (1953).

LEMMA 4.3.1. *The function F_1 is not an elementary function, but its graph is an elementary relation.*

PROOF. That F_1 is not elementary was essentially proved as a lemma in Section 2.2.1. To see that the graph of F_1 is elementary observe that

$$F_1(x) = y \leftrightarrow \exists z ((z)_0 = 0 \wedge \forall_{i < x} ((z)_{i+1} = 2^{(z)_i}) \wedge (z)_x = y).$$

Now it suffices to prove that z can be bounded by an elementary function in x and y . But since F_0 is increasing we can bound z by $\langle y, \dots, y \rangle$ with x occurrences of y , and by a lemma in Section 2.2.5 we have

$$\underbrace{\langle y, \dots, y \rangle}_x < (y + 1)^{2^x}. \quad \square$$

Using similar arguments one can prove that all functions F_α for $\alpha < \varepsilon_0$ have elementary graphs.

Let T be a theory in a language containing $0, S$ with the property that every elementary relation is representable in T . We call a function $f: \mathbb{N} \rightarrow \mathbb{N}$ *provably recursive* in T if we have a formula A_f representing the graph of f such that

$$T \vdash \forall x \exists y A_f(x, y).$$

In standard arithmetical systems like Z one can prove that all functions F_α for $\alpha < \varepsilon_0$ are provably recursive, with methods similar to what we used in Section 4.2. Again ε_0 is a sharp bound: F_{ε_0} is not provably recursive.

We can characterize $(F_\alpha)_{\alpha < \varepsilon_0}$ by higher type iteration. To this end we extend the definition of the functions F_α into higher types.

Types are generated from the base type $\mathbb{N}$ by the formation of *function types* $\tau \rightarrow \sigma$. The *level* of a type is defined similar to the level of a formula (in Section 4.2), by

$$\begin{aligned}\text{lev}(\mathbb{N}) &:= 0, \\ \text{lev}(\tau \rightarrow \sigma) &:= \max(\text{lev}(\tau) + 1, \text{lev}(\sigma)).\end{aligned}$$

It is convenient here to introduce *integer types* ρ_n :

$$\begin{aligned}\rho_0 &:= \mathbb{N}, \\ \rho_{n+1} &:= \rho_n \rightarrow \rho_n.\end{aligned}$$

If $x_0, \dots, x_{n+1}$ are of integer types $\rho_0, \dots, \rho_{n+1}$, then we can form $x_{n+1}(x_n)$ (of type ρ_n) and so on, finally $x_{n+1}(x_n) \dots (x_0)$, or shortly $x_{n+1}(x_n, \dots, x_0)$. Note that $\text{lev}(\rho_n) = n$.

We define F_α^{n+1} of type ρ_{n+1} for $\alpha < \varepsilon_0$:

$$\begin{aligned}F_0^{n+1}(x_n, \dots, x_0) &:= \begin{cases} 2^{x_0} & \text{if } n = 0 \\ x_n^{(x_0)}(x_{n-1}, \dots, x_0) & \text{otherwise.} \end{cases} \\ F_{\alpha+1}^{n+1}(x_n, \dots, x_0) &:= (F_\alpha^{n+1})^{(x_0)}(x_n, \dots, x_0), \\ F_\lambda^{n+1}(x_n, \dots, x_0) &:= F_{\lambda[x_0]}^{n+1}(x_n, \dots, x_0).\end{aligned}$$

Here $x_n^{(y)}(x_{n-1}, \dots, x_0)$ denotes $I(y, x_n, \dots, x_0)$ with an *iteration* functional I of type $\mathbb{N} \rightarrow \rho_n \rightarrow \rho_{n-1} \rightarrow \dots \rightarrow \rho_0 \rightarrow \rho_0$ defined by

$$\begin{aligned}I(0, y, z) &:= z, \\ I(x+1, y, z) &:= y(I(x, y, z)).\end{aligned}$$

THEOREM 4.3.2.

$$F_\alpha^{n+1}(F_\beta^n) = F_{\beta+\omega^\alpha}^n$$

provided $\beta + \omega^\alpha = \beta \# \omega^\alpha$, i.e., in the Cantor normal form of β the last summand ω^{β_0} (if it exists) has an exponent $\beta_0 \geq \alpha$.

PROOF. By induction on α . *Case* $\alpha = 0$.

$$\begin{aligned}F_0^{n+1}(F_\beta^n, x_{n-1}, \dots, x_0) &= (F_\beta^n)^{(x_0)}(x_{n-1}, \dots, x_0) \\ &= F_{\beta+1}^n(x_{n-1}, \dots, x_0).\end{aligned}$$

Case α successor.

$$\begin{aligned}F_\alpha^{n+1}(F_\beta^n, x_{n-1}, \dots, x_0) &= (F_{\alpha-1}^{n+1})^{(x_0)}(F_\beta^n, x_{n-1}, \dots, x_0) \\ &= F_{\beta+\omega^{\alpha-1} \cdot x_0}^n(x_{n-1}, \dots, x_0) \quad \text{by IH} \\ &:= F_{(\beta+\omega^\alpha)[x_0]}^n(x_{n-1}, \dots, x_0) \\ &:= F_{\beta+\omega^\alpha}^n(x_{n-1}, \dots, x_0).\end{aligned}$$

Recall the claim

$$F_\alpha^{n+1}(F_\beta^n) = F_{\beta+\omega^\alpha}^n \quad \text{if } \beta + \omega^\alpha = \beta \# \omega^\alpha.$$

Case α limit.

$$\begin{aligned} F_\alpha^{n+1}(F_\beta^n, x_{n-1}, \dots, x_0) &= F_{\alpha[x_0]}^{n+1}(F_\beta^n, x_{n-1}, \dots, x_0) \\ &= F_{\beta+\omega^{\alpha[x_0]}}^n(x_{n-1}, \dots, x_0) \quad \text{by IH} \\ &= F_{(\beta+\omega^\alpha)[x_0]}^n(x_{n-1}, \dots, x_0) \\ &= F_{\beta+\omega^\alpha}^n(x_{n-1}, \dots, x_0). \quad \square \end{aligned}$$

The result just proved indicates the computational complexity involved in the use of finite types. The functionals $(F_\alpha^{n+1})_{\alpha < \varepsilon_0}$ and in particular the functions $(F_\alpha^1)_{\alpha < \varepsilon_0}$ can be built from *iteration* functionals (and $F_0(x) = 2^x$) by application alone. In the resulting representation of the functions $(F_\alpha)_{\alpha < \varepsilon_0}$ we do not need the fundamental sequences $\lambda[x]$. The application pattern for F_α corresponds to the Cantor normal form of α .

To be done:

- Computable functionals
- A theory TCF of computable functionals
- Computational content of proofs
- Applications: algorithms on stream-represented real numbers