Wellordering Two Sorts: a Slow-Growing Proof Theory for Variable Separation

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von
Marc Wirz
von Zollikon ZH

Leiter der Arbeit:
Prof. Dr. G. Jäger
Institut für Informatik und angewandte Mathematik

Wellordering Two Sorts: a Slow-Growing Proof Theory for Variable Separation

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von
Marc Wirz
von Zollikon ZH

Leiter der Arbeit:
Prof. Dr. G. Jäger
Institut für Informatik und angewandte Mathematik

Von der Philosophisch-naturwissenschaftlichen Fakultät angenommen.

Bern, 13. Juni 2005

Der Dekan
Prof. Dr. P. Messerli

Contents

Introduction

The concept of variable separation forms the nucleus and is still a major branch of the research program called "implicit computational complexity", which tries to investigate complexity theory under a foundational viewpoint, independant of machine models and explicit resource bounds. It has been used in the study of low complexity classes in numerous contexts for more than 15 years now. The general idea is always to separate variables into two kinds. The first kind is thought of as the (potential) output of a computation, which is not yet completed. Having incomplete knowlege on these values only implies that we should restrict the operations allowed to be performed on them to operations which are still safe to apply. Thus this kind of variable has variously been named as safe, incomplete, or output variable. The second kind of variable is meant to represent the input of a computation, which is given as a complete value (written on the input tape of a Turing machine, for example), and allows normal access. These variables are called normal, complete, or input variables.

One particular operation which is considered unsafe, and therefore disallowed for output values, is to determine the depth of a recursion. For such a value, being the output of some computation, could itself involve a recursion. But this leads to an uncontrolled nesting of recursions and therefore to a potential blow-up of the computational complexity. This explains why, taking a one-sorted formalism, the effect of imposing variable separation is always a reduction of its power from computable to feasible. By this it allows to transfer techniques developed in computability theories to the study of feasability.

This idea, present in Nelson [15] and Simmons [21] already, has become widely known due to the work of Bellantoni and Cook [3], and Leivant. Since then, it has been adapted to various fields. Following [3], many of the most common complexity classes have been characterized by "safe" recursion schemes, such as NC^1 and polylog space by Bloch [6], Pspace by Oitavem [16] and others. Leivant characterized the polytime functions as being the functions provably terminating in systems of second order [10] (refined in [14]) and first order [12] arithmetic, and the elementary functions by higher type recursion [11, 13]. To obtain also a characterization of polynomial time through higher type recursion, Bellantoni, Niggl and Schwichtenberg [5] added an additional "linearity" constraint. Aehlig, Berger, Hofmann and Schwichtenberg [1, 20] then used such term systems to construct an equivalent (under the Curry-Howard isomorphism) arithmetic for non-size-increasing poly-

nomial time. Bellantoni and Hofmann [4] gave a comparable arithmetic but used modal oparators to realize variable separation instead.

Ostrin and Wainer [17] proposed another version of a tiered arithmetic called $EA(;)$ which realizes variable separation through the distinction between two syntactically different kinds of variables, one for induction and the other for quantification (as opposed to explicit predicates as in Leivant's [12]). Then they reworked the results of [12] and extended them to a characterization of the whole exponential hierarchy between the Grzegorczyk classes $\mathcal{E}^2$ and $\mathcal{E}^3$, but using techniques much nearer to the ones used in the classical proof theory of Peano Arithmetic. They argue (in [18, Section 5]) that introducing variable separation basically amounts to replacing the fast growing hierarchy with the slow-growing hierarchy in the ordinal analysis. Indeed the fast growing hierarchy up to ε_0 corresponds to the set of the provably total recursive functions of Peano Arithmetic, whereas the slow-growing hierarchy up to ε_0 corresponds to $\mathcal{E}^3$ which is exactly the set of the provably total functions of $EA(;)$. This idea became the starting point and guideline for this thesis.

Chapter 1 reviews shortly the formal system $EA(;)$ and its "must-know" properties, following closely Ostrin and Wainer ([18], [17]). Only a few remarks are original, namely the call-by-name conditional and the remarks on the use of characteristic functions and on the arithmetizations of propositional logic.

Chapters 2 and 3 deal with a major drawback of the concept of variable separation. While this has widely been rewarded for its conceptual purity, imposing no *a priori* bounds like the smash function of bounded arithmetic, theories obtained from this approach often are blamed for being awkward to work within. A typical example is the bubble sort algorithm which indeed is not straightforward to treat. This algorithm nests two recursions, which is forbidden by variable separation, even though they are safe in this particular example because they don't increase the size of the output. Chapter 2 exhibits, as lemma 3, a principle of bounded induction which is able to deal with this kind of harmless nesting of recursions. The techniques we use have been used before implicitely, but they have never been exploited systematically or even made the basis of a concept on its own.

Chapter 3 deals with a restriction which is more specific to $EA(;)$, the problem of substituting into input positions. $EA(;)$ is very strict in this respect, but we show that more liberal substitution principles are derivable.

The first section of chapter 4 exploits these new techniques to develop a wide range of (elementary time) arithmetical principles. The thesis up to this point can be regarded as an analogue to the bootstrapping of theories of bounded arithmetic, see Buss [7, sections 2.4.–2.6.]. This goal is achieved in a way which entirely stays in a simple and pure setting. This contrasts with earlier steps in this direction, such as the introduction of Irwin, Royer and Kapron's [9] "coercion rule", and also the above mentioned approaches using the Curry-Howard isomorphism and higher types.

The remainder of chapter 4 is dedicated to the arithmetizations of ternary sequences and,

relying on this, of ordinal arithmetic. This part, apart from setting the groundwork for the following well-ordering proof, can also serve as a case study for the use of the enhanced arithmetical capabilities, as it heavily depends on these. Chapter 5 then presents the well-ordering proof.

The very short chapter 6 shows that no provable well-ordering of the type considered in the previous chapters can be any more complex. This is performed by reducing the problem to the problem of finding upper bounds for the complexity of $EA(;)$'s provably total functions, which has already been solved by Ostrin and Wainer [18]. More interesting than the result itself is the fact that it opens the way to a natural definition of a proof-theoretic ordinal for two-sorted theories. This definition is based on structured tree ordinals rather than set theoretic ordinals. There are several reasons why this choice seems to be more adequate in our context. Following Wainer's program (as in [18]), introducing variable separation gives rise to a proof theory based on the slow-growing hierarchy, and we can see tree ordinals as a slow-growing counterpart to set theoretical ordinals. Secondly, set theoretical ordinals are simply too coarse. Sommer ([22], [23]) shows that the proof-theoretic ordinal in the classical sense must be ω^2 for all theories between $I\Delta_0$ (or T^2_1 respectively) and $I\Sigma_1$, see Beckmann [2, p. 4] for more details. This in particular means that classical ordinal analysis can't separate any of these theories. But $EA(;)$, as well as it's fragments and extensions which are also of interest, lie all in that range, at least when comparing the respective provably total functions: Linspace for $I\Delta_0$, elementary time for $EA(;)$, and all primitive recursive functions for $I\Sigma_1$. A more technical reason finally is that the lower bound we will give in theorem 35 does not well-order any set-theoretical ordinal at all, but rather a family of (increasing) suborderings $\prec_n$. This perfectly fits the correspondance between tree ordinals and their sets of n-predecessors $\alpha[n]$, see Fairtlough and Wainer [8].

Acknowledgements

The first person to thank here is Professor Gerhard Jäger. He gave me the opportunity to work in proof theory for many years, and his supervision was full of responsability and understanding.

I'm also grateful to Thomas Strahm, not only for much valuable advice when I was new to the field of proof theory, but also for the uncountable thankless tasks for the benefit of the research group he takes care of without being asked for.

I owe a lot to Geoff Ostrin. His expertise on $EA(;)$ influenced me to turn my research interest into this more successful direction, and he took a great effort to thoroughly proof-read the mathematics as well as the English. This thesis profited a lot from his detailed feedback, and the responsability for any remaining faults is clearly mine.

All the other former and present members of our research group deserve my gratitude for the friendly atmosphere, but in particular my office mates: Luca Alberucci, who was and

still is a great partner in many deep discussions on proof theory and any other aspect of life, and his worthy successor Thomas Studer.

Stan Wainer gave some useful feedback to a preliminary draft of this thesis.

The Swiss National Science Foundation and the University of Bern supported my work on variable separation.

Last but not least I would like to thank my friends from music, the mountains, and elsewhere for making life a joy even during the periods when mathematical research is frustrating.

Chapter 1

The Theory $EA(;)$

1.1 Basic Definitions

The theory $EA(;)$ is formulated in a language with two sorts of first order variables, *output variables* $a, b, c, \ldots$, and *input variables* $x, y, z, \ldots$. *Basic terms* are terms built up from variables of either sort, the constant $\mathsf{0}$, the successor function s and the predecessor function p. *General terms* are additionally closed under application of arbitrary function symbols $f, g, h, \ldots$. The *atomic formulas* are the equations $t = t'$ and inequalities $t \neq t'$ between general terms t and t', and *formulas* $A, B, \ldots$ are built up from these closing under the propositional connectives $\wedge$ and $\vee$, and under quantification $\forall a$, $\exists a$ over output variables a. The negation $\neg A$ of a formula A is defined through the De Morgan laws, implication $A \to B$ and double implication $A \leftrightarrow B$ are defined as usual to be abbreviations for $\neg A \vee B$ and $(A \to B) \wedge (B \to A)$ respectively. Parentheses are used when necessary, we follow the usual rule that conjunction and disjunction bind stronger than implication. Furthermore we adopt the notational convention of writing $f(\vec{a})$ for $f(a_0, \ldots, a_{n-1})$ where f is a function symbol of arity n. Similarly, for a unary predicate P we may abbreviate a conjunction $P(a) \wedge P(b) \wedge P(c)$ with $P(a, b, c)$ in some places. This in particular applies to the bounding relation $a \leq x$ (when viewed as a unary predicate with parameter x), and to the predicate *Ord* for ordinals.

In some places we will use the notion of Σ_1-formulas. They are classically defined as the closure of the atomic formulas under conjuntion, disjunction, existential quantification and *bounded* universial and existential quantification $\forall a \leq t$ and $\exists a \leq t$, the latter being abbreviations for constructs of the form $\forall a.a \leq t \to \ldots$ and $\exists a.a \leq t \wedge \ldots$, where t is an arbitrary term.

We will present the axioms and rules of $EA(;)$ as a Tait-style calculus, writing sets of formulas as $A_0, A_1, \ldots, A_n$ and using capital greek letters $\Gamma, \Lambda, \ldots$ to denote them. The logical axioms are the identity axioms $\Gamma, \neg A, A$ and the equality axioms $\Gamma, t = t$ and

$\Gamma, t \neq t', \neg A[t], A[t']$, for arbitrary formulas A and terms t and t'. There's one non-logical axiom, $\Gamma, \mathsf{s}b \neq \mathsf{0}$. This is merely for convenience, to allow case distinctions of the form $b = \mathsf{0} \vee b \neq \mathsf{0}$. We could actually do without the axiom by replacing all instances of such a case distinction with the weaker $b = \mathsf{0} \vee \exists a.b = \mathsf{s}a$ and applying the cases rule, but then we would have to be extremely cautious in writing down our formulas. The propositional rules are the usual ones,

$$\frac{\Gamma, A \qquad \Gamma, B}{\Gamma, A \wedge B} \qquad \text{and} \qquad \frac{\Gamma, A, B}{\Gamma, A \vee B.}$$

The Quantifier rules are

$$\frac{\Gamma, A[a]}{\Gamma, \forall a.A[a]}$$

where a is not free in Γ, and

$$\frac{\Gamma, A[t]}{\Gamma, \exists a.A[a]}$$

where t is a basic term. The "computational content" of $EA(;)$ is introduced via the following two rules: The Induction rule,

$$\frac{\Gamma, A[\mathsf{0}] \qquad \Gamma, \neg A[a], A[\mathsf{s}a]}{\Gamma, A[t]}$$

where a does not appear in $\Gamma, A[\mathsf{0}]$ and t is a basic term without output variables, and the cases rule

$$\frac{\Gamma, A[\mathsf{0}] \qquad \Gamma, A[\mathsf{s}a]}{\Gamma, A[b]}$$

with the same restriction on a as in the induction rule, and for any output variable b – or even a basic term, as will be immediate from the substitution lemma to come soon. The free variable a in the premises of the universal, the induction and the cases rule is called *eigenvariable*, and the side condition on it the eigenvariable restriction.

Finally we have to mention the cut rule. It takes the usual form,

$$\frac{\Gamma, A \qquad \Gamma, \neg A}{\Gamma.}$$

We don't need to include contraction and weakening because the former is implicitly built-in by the use of sets Γ, and the latter can trivially be derived. Similarly we will make free use of the equality rule

$$\frac{\Gamma, A[t]}{\Gamma, t \neq t', A[t'].}$$

We define, as an abbreviation,

$$t{\downarrow} :\equiv \exists a.t = a$$

and say that the term t is *defined*. Thus, a term is defined if and only if it can be reduced (i.e. is provably equal) to a basic term. In particular, any basic term is defined. From this definition we immediately get the following generalized existential rule:

$$\frac{\Gamma, A[t]}{\Gamma, \neg t\!\downarrow, \exists a.A[a]}$$

for any term t, as from the premise we get $\Gamma, t \neq a, A[a]$ by the equality rule, from which the conclusion follows by quantifying. In a very similar way we also get a substitution lemma,

$$\frac{\Gamma, A[a]}{\Gamma, \neg t\!\downarrow, A[t]} \qquad (a \text{ not free in } \Gamma, A[\mathsf{0}]),$$

and ($\forall$)-inversion,

$$\frac{\Gamma, \forall a.A[a]}{\Gamma, \neg t\!\downarrow, A[t]} \qquad (a \text{ not free in } \Gamma, A[\mathsf{0}]).$$

Substitution for input variables is much more restricted. At a first glance it seems that input variables can only be substituted by basic terms without output variables. However, we will show that a much more general substitution rule is derivable, see lemma 10.

All functions are introduced via (consistent) Herbrand-Gödel equational programs, like in Leivant's theories. However, the behaviour of equality is covered by the equality axioms already, and we are left with just a set of arbitrary equations. A function f is *provably total* from an equational program P, if we can prove

$$\neg\forall P, f(\vec{x})\!\downarrow$$

where $\forall P$ is the set of the universal closures of the equations in P. When $EA(;)$ proves the stronger statement $\neg\forall P, f(\vec{a})\!\downarrow$ for output variables $\vec{a}$ we say that f is *provably total on outputs*. We won't be too specific about what equations P consists of but rather assume it contains all function definitions used in the sequel, and we will usually omit mentioning $\forall P$.

1.2 First Steps

In the next step we develop, within $EA(;)$, some of the most basic arithmetic. This also serves for giving a few examples of function definitions in our theory. Let P contain the following equations:

$$\begin{array}{ll} a + \mathsf{0} = a, & a + \mathsf{s}b = \mathsf{s}(a + b), \\ \mathsf{p}(\mathsf{0}) = \mathsf{0}, & \mathsf{p}(\mathsf{s}a) = a, \\ a \dot{-} \mathsf{0} = a, & a \dot{-} \mathsf{s}b = \mathsf{p}(a) \dot{-} b. \end{array}$$

Then we can prove $\neg\forall P, a + x\!\downarrow$ and $\neg\forall P, a \dot{-} x\!\downarrow$ by using induction, and $\neg\forall P, \mathsf{p}(a)\!\downarrow$ by using the cases rule, but we can't prove $\neg\forall P, a + b\!\downarrow$. Note that we also get

$$\neg\forall P, a = \mathsf{0} \vee \exists b.a = \mathsf{s}b$$

(or even $\neg\forall P, a = 0 \vee a = \mathsf{sp}a$) by cases again, and

$$\neg\forall P, \mathsf{s}a = \mathsf{s}b \to a = b$$

by computing $a = \mathsf{ps}a = \mathsf{ps}b = b$.

Another application of cases is the choice function. Add

$$C(0, a, b) = a \quad C(\mathsf{s}c, a, b) = b$$

to the equational program P to obtain

$$\begin{aligned}\neg\forall P, c = 0 &\to C(c, a, b) = a,\\ \neg\forall P, c \neq 0 &\to C(c, a, b) = b.\end{aligned}$$

Note that this is a call-by-value conditional, since we can't conclude $c \neq 0 \to C(c, t, b) = b$ unless t is defined. However, we can also handle call-by-name conditionals, as follows.

Lemma 1. *Given an n-ary function p, thought of as the characteristic function of some predicate, and two functions g and h of the same arity n. Then there is a function*

$$f(\vec{a}) = \begin{cases} g(\vec{a}) & \text{if } p(\vec{a}) = 0\\ h(\vec{a}) & \text{else}\end{cases}$$

such that we can prove

$$\begin{aligned}p(\vec{a}) \neq 0, f(\vec{a}) = g(\vec{a}) \quad &\text{and}\\ \neg p(\vec{a})\!\downarrow, p(\vec{a}) = 0, f(\vec{a}) = h(\vec{a})\end{aligned}$$

(where we have omitted mentioning the premise $\forall P$).

Furthermore, if p is provably total and $p(\vec{a}) \neq 0, g(\vec{a})\!\downarrow$ and $\neg p(\vec{a})\!\downarrow, p(\vec{a}) = 0, h(\vec{a})\!\downarrow$ are provable in $EA(;)$, then f is provably total as well.

Proof. The idea is to introduce a "dummy" pairing function $(\cdot,\cdot)$ which isn't given any numerical extension, and a new conditional C_2 which is defined on such pairs. In detail, let f be defined by the following equational program:

$$\begin{aligned}C_2(0, (0, \vec{a}), (1, \vec{a})) &= (0, \vec{a}),\\ C_2(\mathsf{s}b, (0, \vec{a}), (1, \vec{a})) &= (1, \vec{a}),\\ f_0(\vec{a}) &= C_2(p(\vec{a}), (0, \vec{a}), (1, \vec{a})),\\ f_1((0, \vec{a})) &= g(\vec{a}),\\ f_1((1, \vec{a})) &= h(\vec{a}),\\ f(\vec{a}) &= f_1(f_0(\vec{a})).\end{aligned}$$

Then we have $p(\vec{a}) \neq 0, f_0(\vec{a}) = (0, \vec{a})$ and $f_0(\vec{a}) \neq (0, \vec{a}), f(\vec{a}) = g(\vec{a})$, so the first claim follows by a cut. The second claim is similar. For the totality of f, from the additional assumptions we immediately can show $p(\vec{a}) = b \to f(\vec{a})\downarrow$ by using the cases rule on b. As p is provably total by assumption we can cut the premise after existentially quantifying over b. □

This conditional in its general form, choosing cases according to a predicate rather than a simple variable, requires the predicate to be given by its characteristic function. For the time being, the less-than relations will suffice:

$$\begin{aligned} a \leq b &:\equiv a \dot{-} b = 0, \\ a < b &:\equiv \mathsf{s}a \dot{-} b = 0. \end{aligned}$$

These definitions are tailored to trivially satisfy $a < b \leftrightarrow \mathsf{s}a \leq b$. We also use the complements $a \not\leq b :\equiv \neg a \leq b$ and $a \not< b :\equiv \neg a < b$.

Characteristic functions are closed under the (arithmetizations of the) boolean operators if we include the following equations into P:

$$\begin{gathered} \top = 1, \quad \bot = 0, \quad \sim 0 = 1, \quad \sim \mathsf{s}a = 0, \\ 0 \,\&\, 0 = 0, \quad 0 \,\&\, \mathsf{s}a = 0, \quad \mathsf{s}a \,\&\, 0 = 0, \quad \mathsf{s}a \,\&\, \mathsf{s}b = 1. \end{gathered}$$

Here "$\sim$" is a unary function which acts as a negation, and the binary function "&" corresponds to conjunction. Both functions are provably total on outputs, and their correspondence with the logic of $EA(;)$ can be proved, e.g.

$$\neg\forall P, a \neq 0 \wedge b \neq 0 \leftrightarrow a \,\&\, b \neq 0.$$

We will sometimes refer to this observation as the "Adequacy of the arithmetization of propositional logic".

When defining predicates, the use of characteristic functions makes things even easier because there is no problem in using recursive function definitions. Examples for such definitions are the characteristic functions for equality on page 25, for codes of ordinals on page 32, and the ordinal less-than relation on page 35.

Chapter 2

Induction principles

2.1 Input Bounded Outputs

This section is centered around lemma 3. Call an output variable a appearing in a set Γ *input bounded*, if Γ contains a formula $\neg a \leq x$. Roughly speaking lemma 3 modifies the induction rule such that the conclusion is given for input bounded outputs rather than for (basic terms on) inputs. Notice that this new rule incorporates the original induction rule, as we always can substitute back x for a, once we have established $x \leq x$. The converse direction, the substitution of input bounded outputs for input variables, is possible as well and could be shown by a straightforward induction on proofs (just replacing every induction rule with lemma 3). We don't give the details but rather wait for lemma 10 whereof this result is just a special case.

We start with a handful of simple technical observations needed for the proof of lemma 3.

Lemma 2. *If P contains the defining equations for all functions introduced in chapter 1, we have (where we omit mentioning the premise $\forall P$):*

1. $\vdash \mathsf{0} \leq x$,
2. $\vdash a \leq 0 \to a = 0$,
3. $\vdash a \leq b \to \mathsf{s}a \leq \mathsf{s}b \wedge \mathsf{p}a \leq \mathsf{p}b$,
4. $\vdash a \leq x \to \mathsf{p}a \leq x$.

Proof. The first item is proved by induction for the formula $\mathsf{0} \leq a$, where the induction step follows from $\mathsf{0} \dot{-} \mathsf{s}a = \mathsf{p0} \dot{-} a = \mathsf{0} \dot{-} a$. The second item says $a \dot{-} \mathsf{0} = \mathsf{0} \to a = \mathsf{0}$ which is immediate from $a \dot{-} \mathsf{0} = a$. The first part of the third item is immediate from the equational program, as we have $\mathsf{s}a \dot{-} \mathsf{s}b = \mathsf{ps}a \dot{-} b = a \dot{-} b$, whereas the second part uses the

cases rule: The case $b = 0$ then is immediate from item 2, and the successor case follows from $\mathsf{p}a \dot{-} \mathsf{ps}b = a \dot{-} \mathsf{sps}b = a \dot{-} \mathsf{s}b$. The final item follows directly from $\forall a.\mathsf{p}a \dot{-} x = \mathsf{p}(a \dot{-} x)$ which is proved by induction. Here the base case is obtained from $\mathsf{p}a \dot{-} \mathsf{0} = \mathsf{p}a = \mathsf{p}(a \dot{-} \mathsf{0})$, and the induction step comes from

$$\mathsf{pp}a \dot{-} b \neq \mathsf{p}(\mathsf{p}a \dot{-} b), \mathsf{p}a \dot{-} \mathsf{s}b = \mathsf{pp}a \dot{-} b = \mathsf{p}(\mathsf{p}a \dot{-} b) = \mathsf{p}(a \dot{-} \mathsf{s}b)$$

by quantification (note that $\mathsf{p}a$ is a basic term). □

Now we are ready to state and prove the main result of this chapter. It is independant from Ostrin and Wainer [17, p. 381], which used a similar technique with a specific induction formula A when proving the totality of the factorial.

Lemma 3 (Induction for input bounded outputs). *$EA(;)$ proves, for all formulas A, the rule*

$$\frac{\Gamma, A[\mathsf{0}] \qquad \Gamma, \neg a \leq x, \neg A[a], A[\mathsf{s}a]}{\Gamma, \neg a \leq x, A[a]}$$

(where a does not appear free in $\Gamma, A[\mathsf{0}]$).

Proof. By induction on b in the formula $\forall a \leq b.a \leq x \to A[a]$. The base case follows from lemma 2.2 and the first premise. For the induction step we use lemmas 2.3 and 2.4 and the second premise in the form of $\Gamma, \neg \mathsf{p}a \leq x, \neg A[\mathsf{p}a], A[a]$ to find

$$\neg(\mathsf{p}a \leq b \wedge \mathsf{p}a \leq x \to A[\mathsf{p}a]), a \leq \mathsf{s}b \wedge a \leq x \to A[a].$$

Quantification and the induction rule then lead to $\forall a \leq x.a \leq x \to A[a]$. □

This lemma plays a fundamental role throughout this thesis. For illustrating its application, recall the equational programs for addition and subtraction. Instead of proving $a + x\!\downarrow$ and $a \dot{-} x\!\downarrow$ we now get $b \leq x \to a + b\!\downarrow$ and $b \leq x \to a \dot{-} b\!\downarrow$ which in contrast to the former allows us to perform another induction on the second arguments of both functions – as long as we take into account the bound x. This will soon pay off, for the first time in lemma 5.3 where we could neither prove the base case nor the induction step otherwise.

Another few useful applications of the lemma are collected into the following lemma. Notice that they also hold for input variables in place of the input bounded outputs, as observed above. The notation $a \leq b \leq x$ means $a \leq b \wedge b \leq x$, of course. Recall also the abbreviated notation $a, b \leq x$ as introduced earlier. Then the third item, for example, would read as $\vdash (a \leq x \wedge b \leq x) \to a \dot{-} b \leq a \wedge a \leq a + b$, correctly spelled out.

Lemma 4. *If P contains the defining equations for all functions introduced in chapter 1, we have (where we omit mentioning the premise $\forall P$):*

1. $\vdash b \leq x \to a + b\!\downarrow \wedge\, a \dot{-} b\!\downarrow \wedge\, \mathsf{0} + b = b \wedge \mathsf{s}a + b = \mathsf{s}(a + b) \wedge a \dot{-} \mathsf{s}b = \mathsf{p}(a \dot{-} b)$,

2. $\vdash b \leq x \to b \leq b \wedge \mathsf{0} \leq b \leq \mathsf{s}b$,

3. $\vdash a, b \leq x \to a \dot{-} b \leq a \leq a + b$,

4. $\vdash (a \leq b \leq x \to a \leq x) \wedge (a \leq b \leq c \leq x \to a \leq c)$,

5. $\vdash b \leq x \to (a \leq b \to a < b \vee a = b)$,

6. $\vdash a, b \leq x \to (a \leq b \vee b \leq a) \wedge (a < b \to a \leq b \wedge b \not\leq a)$.

Proof. 1. For the first and the second conjunct see above. The third and fourth are proven by using lemma 3, where both the base cases and the induction steps are simple. For the final one, notice that we already have shown $\mathsf{p}a \dot{-} x = \mathsf{p}(a \dot{-} x)$ in the proof of lemma 2.4. Replacing the induction rules with applications of lemma 3 turns its proof into a proof of the fifth conjunct, because $a \dot{-} \mathsf{s}b = \mathsf{p}a \dot{-} b$.

2. The first conjunct is immediate from lemma 3 where $A[a] :\equiv a \leq a$. If we had used the induction rule instead of lemma 3 we had got $x \leq x$. As observed above, this allows us to replace input bounded outputs with the bounding input variables.

   As $x \leq x$, the first, third and fourth items of lemma 2 combine into $\mathsf{0} \leq x \leq \mathsf{s}x$. Replacing the induction rules with applications of lemma 3 in these proofs yields a proof of the second conjunct.

3. Using the previous items of this lemma and lemma 2.3 we prove the first inequality by input bounded induction on b, with base case $a \dot{-} \mathsf{0} = a \leq a$ and induction step $a \dot{-} \mathsf{s}b = \mathsf{p}(a \dot{-} b) \leq \mathsf{p}a \leq a$, and the second inequality by input bounded induction on a, where the base case is $\mathsf{0} \leq b = \mathsf{0} + b$ and induction step $\mathsf{s}a \leq \mathsf{s}(a + b) = \mathsf{s}a + b$.

4. The first conjunct is proven by (ordinary) induction on

$$A[c] :\equiv \forall a \forall b. a \leq b \leq c \to a \leq c.$$

   The base case follows from two applications of lemma 2.2. For the induction step we assume $a \leq b \leq \mathsf{s}c$ and use the third item of the same lemma to obtain $\mathsf{p}a \leq \mathsf{p}b \leq c$. Then $\mathsf{p}a \leq c$ by induction hypothesis. If $a \neq \mathsf{0}$ we get $a = \mathsf{sp}a \leq \mathsf{s}c$ by the same lemma again. Otherwise, $a = \mathsf{0}$ means $\mathsf{p}a = a$, so $a \leq c$ which implies $a \leq \mathsf{s}c$ as shown in the proof of lemma 2.1, and the induction step is complete.

   Using lemma 3 in place of the induction rule turns this proof into a proof of the second conjunct.

5. We first prove $\mathsf{p}a = \mathsf{0} \to (a = \mathsf{0} \vee a = \mathsf{1})$ by using the cases rule. Substituting $\mathsf{s}a \dot{-} b$ (which is defined if $b \leq x$) for a, and using the fact that $a \leq b$ implies $\mathsf{s}a \leq \mathsf{s}b$ by lemma 2.3, and thus $\mathsf{p}(\mathsf{s}a \dot{-} b) = \mathsf{s}a \dot{-} \mathsf{s}b = \mathsf{0}$ by lemma 4.1 and the definition of $\leq$,

we obtain $\mathsf{s}a \dot{-} b = \mathsf{0} \vee \mathsf{s}a \dot{-} b = \mathsf{1}$. Back to the main proof, if the first disjunct holds we have $a < b$ by definition. If the second holds, we are done if we can show

$$\neg b \leq x, \forall a.\mathsf{s}a \dot{-} b = \mathsf{1} \to a = b$$

which we can by using lemma 3: The base case $b = \mathsf{0}$ is immediate. For the induction step we assume $\mathsf{s}a \dot{-} \mathsf{s}b = \mathsf{1}$ and show that $a = \mathsf{s}b$. The case $a = \mathsf{0}$ is impossible as the premise by lemma 4.2 would imply that $\mathsf{0} = \mathsf{0} \dot{-} b = \mathsf{1} \dot{-} \mathsf{s}b = \mathsf{1}$ which isn't the case. So a must be a successor. This implies that $\mathsf{ps}a = \mathsf{sp}a$, and we compute $\mathsf{sp}a \dot{-} b = \mathsf{ps}a \dot{-} b = \mathsf{s}a \dot{-} \mathsf{s}b = \mathsf{1}$, so the induction hypothesis gives $\mathsf{p}a = b$, thus $a = \mathsf{ps}a = \mathsf{sp}a = \mathsf{s}b$ as required.

6. The first conjunct is shown by input bounded induction on a. In the base case we have $a = \mathsf{0} \leq b$ from the second item of this lemma. In the induction step we distinguish the two cases given by the induction hypothesis. If $b \leq a$, as $a \leq \mathsf{s}a$ by the second item again, we are done by applying the fourth item. (To be precise, we have to substitute $\mathsf{s}x$ for x in 4. beforehand, then the premise $\mathsf{s}a \leq \mathsf{s}x$ is met by lemma 2.3.) The other case, $a \leq b$, has two subcases according to item 5. But the subcase $b < a$ means $\mathsf{s}a \leq b$ by definition, otherwise $b = a \leq \mathsf{s}a$ by 2. once more.

   In the second conjunct, $a \leq b$ is immediate from $a \leq \mathsf{s}a \leq b$ and the fourth item of this lemma. For the last claim we show $\mathsf{s}a \dot{-} b \neq \mathsf{0} \vee b \dot{-} a \neq \mathsf{0}$, which is just another way to express the implication $a < b \to b \not\leq a$, by input bounded induction on b. In the base case we have trivially $\mathsf{s}a \dot{-} \mathsf{0} = \mathsf{s}a \neq \mathsf{0}$. In the induction step we distinguish the two cases given by the induction hypothesis. If $b \dot{-} a \neq \mathsf{0}$, then $\mathsf{s}b \dot{-} a$ can't equal $\mathsf{0}$ as this would imply $b \leq \mathsf{s}b \leq a$ which contradicts the premise by item 4. Otherwise $\mathsf{s}a \dot{-} b$ doesn't equal $\mathsf{0}$, therefore it is the successor of some c. Now if $c \neq \mathsf{0}$ we simply compute $\mathsf{s}a \dot{-} \mathsf{s}b = \mathsf{p}(\mathsf{s}a \dot{-} b) = \mathsf{ps}c = c \neq \mathsf{0}$ using the first item of this lemma, and if not, $\mathsf{s}a \dot{-} b = \mathsf{s0}$. But we have shown in the proof of item 4 that this implies $a = b$. Then $\mathsf{s}b \dot{-} a \neq \mathsf{0}$ because we can show $\mathsf{s}a \dot{-} a = \mathsf{s0}$ by an easy (input bounded) induction. □

Lemma 4.4 will be used throughout, usually in the form $(t{\downarrow} \wedge t'{\downarrow} \wedge t \leq t' \leq x) \to t \leq x$, and we will refer to it as the "Transitivity of $\leq$".

2.2 Variations

We conclude this chapter by deriving two variants of lemma 3, that is a course-of-value induction and "induction for $a \geq b$". Again we have to set up a few technical results beforehand.

Lemma 5. *If P contains the defining equations for all functions introduced in chapter 1, we have (where we omit mentioning the premise $\forall P$):*

1. $\vdash a \not< 0$,
2. $\vdash c, a + \mathsf{s}c \leq x \to a + c \leq x$,
3. $\vdash a \leq b \leq x \to a + (b \dot{-} a) = b$.

Proof. 1. This is immediate from lemma 2.2, as $\mathsf{s}a \neq \mathsf{0}$.

2. This is just a variant of lemma 2.4 where we substitute the term $\mathsf{s}(a + c)$ (which is defined if $c \leq x$) for a, and apply the defining equations of $+$ and p.

3. By input bounded induction for $A[a] :\equiv \forall b. a \leq b \leq x \to a + (b \dot{-} a) = b$. The base case is immediate from lemma 4.1. For the induction step, if $b = \mathsf{0}$, then by lemma 2.2, $a = \mathsf{0}$ as well, and the result is trivial. Otherwise $\mathsf{sp}b = b$. Assuming $\mathsf{s}a \leq b \leq x$, lemmas 2.3 and 2.4 give us $a \leq \mathsf{p}b \leq x$, so we want to substitute $\mathsf{p}b$ for the universal in the induction hypothesis to compute

$$\mathsf{s}a + (b \dot{-} \mathsf{s}a) = \mathsf{s}a + (\mathsf{p}b \dot{-} a) = \mathsf{s}(a + (\mathsf{p}b \dot{-} a)) = \mathsf{sp}b = b.$$

The second step is justified by lemma 4.1 where we substitute $\mathsf{p}b \dot{-} a$ for b. This is allowed since we have $a \leq x$ by transitivity, thus $\mathsf{p}b \dot{-} a\!\downarrow$ and $\mathsf{p}b \dot{-} a \leq x$ by lemmas 4.1 and 4.3 (and transitivity again).

This completes the induction step. Lemma 3 gives $\neg a \leq x, a \leq b \leq x \to a + (b \dot{-} a) = b$, but the premise $a \leq x$ is superfluous due to transitivity. □

Corollary 6. *$EA(;)$ proves, for all formulas A, the rule*

$$\frac{\Gamma, \neg b \leq x, (\forall a < b. A[a]) \to A[b]}{\Gamma, \neg b \leq x, A[b]}$$

(where b does not appear free in $\Gamma, A[\mathsf{0}]$).

Proof. Apply lemma 3 to the formula $B[b] :\equiv \forall a < b. A[a]$. Because of lemma 5.1 the base case $\Gamma, B[\mathsf{0}]$ is trivial. For the induction step we assume $b \leq x$, $B[b]$, and $a < \mathsf{s}b$, we have to show $A[a]$. The final assumption says $\mathsf{s}a \leq \mathsf{s}b$ by definition, so by lemmas 2.3 and 4.5 we obtain $a < b \vee a = b$. In case of the first disjoint, $A[a]$ holds by $B[b]$. Therefore $A[b]$ follows from the premise of the corollary, which settles the case of the second disjoint. Now lemma 3 gives $\Gamma, \neg b \leq x, \forall a < b. A[a]$ which by the premise leads to $\Gamma, \neg b \leq x, A[b]$. □

Corollary 7. *$EA(;)$ proves, for all formulas A, the rule*

$$\frac{\Gamma, \neg a \leq x, A[a] \qquad \Gamma, \neg a \leq b \leq x, \neg A[b], A[\mathsf{s}b]}{\Gamma, \neg a \leq b \leq x, A[b]}$$

(where b does not appear free in $\Gamma, A[\mathsf{0}]$).

Proof. Apply lemma 3 to the formula $B[c] :\equiv a+c \leq x \to A[a+c]$, for a new variable c that does not appear free in $\Gamma, A[\mathsf{0}]$. Then $\Gamma, B[\mathsf{0}]$ follows immediately from the first premise. For the induction step assume $c \leq x$ and $a + \mathsf{s}c \leq x$. Then lemma 5.2 allows us to obtain $A[a + c]$ from the induction hypothesis. Furthermore we have $a \leq a + c$ from lemma 5.2. Now we can substitute the defined term $a+c$ for b in the second premise to obtain $A[a+\mathsf{s}c]$, which concludes the induction step. Now by lemma 3 we have $\Gamma, \neg a, c, a + c \leq x, A[a + c]$. Substitute back the term $b \dot{-} a$, which is defined under the assumption $a \leq x$, for c and use lemma 5.3 to obtain $\Gamma, \neg a \leq b \leq x, \neg a, b \dot{-} a \leq x, A[b]$, and finally lemma 4.3 and transitivity to remove the premises $b \dot{-} a \leq x$ and $a \leq x$. □

Chapter 3

Input Substitution

It is a common criticism that $EA(;)$ doesn't provide a direct mechanism for subsitute terms for input variables. That is, its provably total functions aren't intensionally closed under (predicative) composition. The fact that they are indeed traditionally is established via an extensional characterization like being exactly the functions computable in elementary time. However, we can do better. Ostrin and Wainer [18, lemma 2.2] show that $EA(;)$ proves $A[2_k(p(\vec{x}))]$ for every progressive formula A (where $2_k(x)$ is the k-time iterated exponentiation to the base 2, and p is any polynomial). A straightforward induction on proofs would show that therefore free input variables always can be substituted with $2_k(p(\vec{x}))$. We are going to show that this argument can be generalized to any provably total function of $EA(;)$.

The proof idea is to transform, with respect to a fixed progressive formula A, the given proof of $\exists d.f(x) = d$ into a proof of $\exists d.f(x) = d \wedge A[d]$, roughly speaking, by relativizing all sequents to A. There remains one technical obstacle: The potential presence of the predecessor function in the definition of f requires that the relativizing formula be closed under predecessors, which need not be the case for A. This is the reason why we relativize the proof to A^* instead:

$$\begin{aligned}
A^*[a] &:\equiv a \leq a \wedge \forall d \leq a.\mathsf{p}d \leq a \wedge A[d],\\
(t = s)^A &:\equiv t = s,\\
(B \wedge C)^A &:\equiv B^A \wedge C^A,\\
(B \vee C)^A &:\equiv B^A \vee C^A,\\
(\neg B)^A &:\equiv \neg B^A,\\
(\forall a.B[a])^A &:\equiv \forall a.A^*[a] \to B^A[a],\\
(\exists a.B[a])^A &:\equiv \exists a.A^*[a] \wedge B^A[a],\\
(B_0, \ldots, B_n)^A &:\equiv B_0^A, \ldots, B_n^A.
\end{aligned}$$

It is worth noticing that $A^*[t]$ implies $A[t]$ for any defined term t, this is the reason for

which we include the clause $a \leq a$ into the definition of A^*. For better reading we are going to use a Gentzen-style notation in the following two lemmas, writing $A, B \vdash \Gamma$ when we actually mean $\vdash \neg A, \neg B, \Gamma$.

Lemma 8. *If $EA(;)$ proves $\vdash \Gamma(\vec{a})$, where all free output variables are displayed, then $EA(;)$ proves*

$$\mathrm{Prog}(A), A^*[\vec{a}] \vdash \Gamma^A(\vec{a}),$$

where $\mathrm{Prog}(A)$ *denotes the formula* $A[\mathsf{0}] \wedge (\forall a.A[a] \to A[\mathsf{s}a])$*, and the abbreviation* $A^*[\vec{a}]$ *stands for* $\bigwedge_{a\in\vec{a}} A^*[a]$.

Proof. Let us first observe that $\mathrm{Prog}(A)$ implies $A^*[\mathsf{0}]$ as well as $A^*[c] \to A^*[t(c)]$ for any basic term t. The first claim is immediate, as $d \leq \mathsf{0}$ implies $d = \mathsf{0}$ by lemma 2.2. The second is shown inductively along the construction of t, where $t(c) \leq t(c)$ is immediate from the assumption $c \leq c$ using lemma 2.3 repeatedly.

As to the the second conjunct of A^*, the base case $t(c) \equiv c$ is trivial. In the case of a successor term $\mathsf{s}t$ we assume $d \leq \mathsf{s}t$, and we have to show $\mathsf{p}d \leq \mathsf{s}t \wedge A[d]$, using the induction hypothesis $A^*[t]$. By lemma 2.3 we first deduce that $\mathsf{p}d \leq \mathsf{ps}t = t$. From $A^*[t]$ it follows that $\mathsf{pp}d \leq t$ and $A[\mathsf{p}d]$, as $\mathsf{p}d$ is defined. As A is progressive we conclude $A[d]$, the second conjunct. For the first conjunct we apply lemma 2.3 again to obtain $\mathsf{spp}d \leq \mathsf{s}t$, but $\mathsf{spp}d$ equals $\mathsf{p}d$, unless $\mathsf{p}d = \mathsf{0}$, in which case $\mathsf{p}d = \mathsf{0} \leq \mathsf{s}t$ follows from $\mathsf{p}d \leq t$ by $\mathsf{0} \dot{-} \mathsf{s}t = \mathsf{p0} \dot{-} t = \mathsf{0} \dot{-} t$.

In the predecessor case we assume $d \leq \mathsf{p}t$ and $A^*[t]$ and show $\mathsf{p}d \leq \mathsf{p}t \wedge A[d]$. The case $t = \mathsf{0}$ is trivial because then $\mathsf{p}t = t$. Otherwise $\mathsf{sp}t = t$. Thus the first assumption implies $\mathsf{s}d \leq t$ by lemma 2.3 once again. Instantiating the universal quantifier in $A^*[t]$ with $\mathsf{s}d$ yields $d \leq t$ (as $\mathsf{ps}d$ always equals d). Now $\mathsf{p}d \leq \mathsf{p}t$ is immediate from lemma 2.3, and $A[d]$ follows from $A^*[t]$, this time with the universal quantifier instantiated with d.

To prove the lemma we proceed by induction on the length of the derivation of $\vdash \Gamma(\vec{a})$. If $\Gamma(\vec{a})$ is an axiom, then so is $\Gamma(\vec{a})^A$. If the last rule applied was a conjunction or a disjunction rule, we can apply the same rule to the induction hypothesis. The universal rule is straightforward as well.

If $\Gamma, B[b]$ was derived by using the cases rule from the premises $\Gamma, B[\mathsf{0}]$ and $\Gamma, B[\mathsf{s}c]$, applying the same rule to the induction hypothesis gives

$$\mathrm{Prog}(A), A^*[\vec{a}] \wedge A^*[c] \vdash \Gamma^A(\vec{a}), B^A[\vec{a}, t(b)].$$

Without loss of generality we may assume that $b \in \vec{a}$, otherwise we simply add the premise $A^*[b]$ by weakening. Now, if $c \in \vec{a}$, we are done. If not, we can substitute $\mathsf{0}$ for c and cut the premise $A^*[\mathsf{0}]$ which is provable as seen above.

If Γ was derived by a cut from the premises Γ, B and $\Gamma, \neg B$, we proceed in a similar way, applying the cut rule to the induction hypothesis possibly followed by removing premises of the form $A^*[b]$ for output variables b present in B but not in Γ.

If $\Gamma, \exists b.B[b]$ was derived from $\Gamma, B[t(c)]$, we assume by induction hypothesis that

$$\mathrm{Prog}(A), A^*[\vec{a}] \wedge A^*[c] \vdash \Gamma^A(\vec{a}), B^A[\vec{a}, t(c)]$$

is derivable. From this and the introductory observation we conclude

$$\mathrm{Prog}(A), A^*[\vec{a}] \wedge A^*[c] \vdash \Gamma^A(\vec{a}), \exists b.A^*[b] \wedge B^A[\vec{a}, b]$$

by the conjunction rule followed by the existential rule. Now, as in the cases rule, we are done, if $c \in \vec{a}$. If not, we substitute $\mathsf{0}$ for c and cut the premise $A^*[\mathsf{0}]$.

All that remains is the induction rule. Here applying the induction hypothesis to the premises gives

$$\mathrm{Prog}(A), A^*[\vec{a}] \vdash \Gamma^A(\vec{a}), B^A[\vec{a}, \mathsf{0}]$$

and

$$\mathrm{Prog}(A), A^*[\vec{a}] \wedge A^*[b] \vdash \Gamma^A(\vec{a}), \neg B^A[\vec{a}, b], B^A[\vec{a}, \mathsf{s}b].$$

From this and the observation above it is easy to see that we can derive the premises of the induction rule for the formula $A^*[b] \wedge B^A[\vec{a}, b]$. We end up with

$$\mathrm{Prog}(A), A^*[\vec{a}] \vdash \Gamma^A(\vec{a}), A^*[x] \wedge B^A[\vec{a}, x],$$

so we are done once we drop the first conjunct $A^*[x]$. □

Corollary 9. *Let Δ be a finite set of Σ_1 formulas, where all free output variables are among $\vec{a}$. If $EA(;)$ proves $\vdash \Delta, f(\vec{a}, \vec{x})\!\downarrow$, then $EA(;)$ proves*

$$\mathrm{Prog}(A), A^*[\vec{a}] \vdash \Delta, f(\vec{a}, \vec{x})\!\downarrow \wedge f(\vec{a}, \vec{x}) \leq f(\vec{a}, \vec{x}) \wedge A[f(\vec{a}, \vec{x})].$$

This corollary in particular generalizes lemma 2.2 of Ostrin and Wainer [18] to all provably terminating functions f, because the (usually hidden) premise $\neg\forall P$ consists of Σ_1 formulas only. However, we can apply it even in presence of any side formulas of complexity Σ_1.

Proof. From the lemma we get $\Delta^A, \exists b.A^*[b] \wedge f(\vec{a}, \vec{x}) = b$ (under the assumptions $\mathrm{Prog}(A)$ and $A^*[\vec{a}]$). This logically implies our claim, as $B^A \to B$ is a tautology for every Σ_1 formula B, and $A^*[b] \wedge t = b$ first implies $b \leq b \wedge A[b]$ (as b is always defined), which in turn implies $t \leq t \wedge A[t]$ by the equality rule. □

Theorem 10 (Input Substition). *Let Δ be a finite set of Σ_1 formulas, and assume that z is not free in $\Gamma, A[\mathsf{0}]$. Then $EA(;)$ proves, for all formulas A, terms t and function symbols f, the rule*

$$\frac{\Gamma, A[z] \qquad \Delta, f(\vec{x})\!\downarrow}{\Gamma, \Delta, \neg t\!\downarrow, \neg t \leq f(\vec{x}), A[t].}$$

This theorem has two special cases which are especially interesting. First, if t is $f(\vec{x})$, the premises $t\!\downarrow$ and $t \leq f(\vec{x})$ become provable (the latter by the previous corollary) and we are left with

$$\frac{\Gamma, A[z] \qquad \Delta, f(\vec{x})\!\downarrow}{\Gamma, \Delta, A[f(\vec{x})].}$$

Notice however that, in contrast to the similar-looking substitution for outputs, this holds only when $f(\vec{x})\!\downarrow$ can actually be proved. In addition, we can't substitute an f that is applied to output variables. Although this matches exactly the safe composition scheme of Bellantoni and Cook [3], one might ask whether we can we generalize further. It seems plausible for instance that one could even derive $A[a] \to A[f(a, \vec{x})]$ when $f(a, \vec{x})$ is provably total.

Secondly, when t is an input variable d and f is the output variable x, then f is provably total with an empty Δ, and t is trivially defined, so we have

$$\frac{\Gamma, A[z]}{\Gamma, \neg d \leq x, A[d].}$$

This together with earlier remarks shows that inputs and input bounded outputs are freely interchangeable. In particular, "provably total" will often be interpreted as "defined on input bounded outputs"in the sequel.

As a last remark before we are going to prove the theorem we have to mention that it generalizes to sets $\Gamma, A[z]$ that are proved in any theory that extends $EA(;)$ by additional axioms, as long as these axioms are closed under substitution of defined terms for input variables. An example of such an extension is the variant of $EA(;)$ used in Wainer and Williams [24] where the condition on the additional axioms is met because they are formulated exclusively over output variables. However, the proof of $\Delta, f(\vec{x})\!\downarrow$ must be a proof of $EA(;)$ itself.

Proof. Given the previous corollary, the proof idea is quite simple and has already been used a few times in proving some items of lemma 4: Replace each induction rule in the given proof of $\Gamma, A[z]$ with an application of lemma 3, you end up proving the second special case. The same procedure, with the corollary playing the role of lemma 3, would yield the first special case. A slight generalization of this argument proves the theorem itself, still proceeding by induction on the length of the derivation of $\Gamma, A[z]$.

If $\Gamma, A[z]$ is an axiom, so is $\Gamma, \neg t\!\downarrow, A[t]$, and the claim follows from weakening. The conjunction and disjunction rules and the cut rule are immediate from the induction hypothesis. So are the cases and the universal rule, as we can safely assume that the eigenvariable does not occur in Δ and t.

If $\Gamma, \exists a.A[a, z]$ was derived from $\Gamma, A[s, z]$, and s doesn't contain z, the claim is immediate from the induction hypothesis again. If s contains z we have to notice that $s(t)$ is defined under the assumption $t\!\downarrow$, thus it can serve as a witness for the (generalized) existential rule.

The last case is when $\Gamma, A[s(y), z]$ is derived by an induction. Again we can assume that the eigenvariable doesn't occur in Δ and t. Now if y is different from z we simply apply the induction rule to the induction hypothesis. If not, what we can get from the induction hypothesis is that $A[\cdot, t]$ is provably progressive (under the assumptions $t\!\downarrow$ and $t \leq f(\vec{x})$).

Once we have concluded that the formula $B[b] :\equiv \forall d \leq b.A[s(d), t]$ is progressive in b, we can apply the previous corollary to obtain $f(\vec{x})\!\downarrow \wedge \forall d \leq f(\vec{x}).A[s(d), t]$. The claim then follows by instantiating d with t.

For the progressiveness of B we notice that $s(\mathsf{0})$ provably equals $\mathsf{s}^n(\mathsf{0})$ for some natural number n, so $B[\mathsf{0}]$ is immediate from lemma 2.2, applying the progressiveness of $A[\cdot, t]$ n times. For the induction step we need that $EA(;)$ proves

$$(*) \qquad t(a) = \mathsf{s}t(\mathsf{p}a) \vee t(a) = t(\mathsf{p}a)$$

for any basic term t. We show this by induction on the construction of t. It is trivial (using the cases rule) if $t(a) \equiv a$. If $t(a) \equiv \mathsf{s}t'(a)$ we assume that $(*)$ holds for t' and compute

$$t(a) = \mathsf{s}t'(a) = \begin{cases} \mathsf{ss}t'(\mathsf{p}a) = \mathsf{s}t(\mathsf{p}a) & \text{if } t'(a) = \mathsf{s}t'(\mathsf{p}a), \\ \mathsf{s}t'(\mathsf{p}a) = t(\mathsf{p}a) & \text{if } t'(a) = t'(\mathsf{p}a). \end{cases}$$

If $t(a) \equiv \mathsf{p}t'(a)$ on the other hand we have

$$t(a) = \mathsf{p}t'(a) = \begin{cases} \mathsf{ps}t'(\mathsf{p}a) = t'(\mathsf{p}a) = \begin{cases} \mathsf{p}t'(\mathsf{p}a) = t(\mathsf{p}a) & \text{if } t'(\mathsf{p}a) = \mathsf{0}, \\ \mathsf{sp}t'(\mathsf{p}a) = \mathsf{s}t(\mathsf{p}a) & \text{if } t'(\mathsf{p}a) = \mathsf{s}b, \end{cases} \\ \mathsf{p}t'(\mathsf{p}a) = t(\mathsf{p}a). \end{cases}$$

Returning back to the proof of the induction step for B, we assume $B[b]$ and $d \leq \mathsf{s}b$ in order to show $A[s(d), t]$. From lemma 2.3 we get $\mathsf{p}d \leq b$, so instantiating the universial with $\mathsf{s}b$ in the induction hypothesis yields $A[s(\mathsf{p}d), t]$. Since A is progressive, the claim follows from $(*)$. □

We have seen that inputs and input bounded outputs are interchangeable. In view of this it is no surprise that the same substitution principle applies to input bounded outputs as well, as the following rephrasing shows. It will play no role in the sequel, however.

Corollary 11 (Input bounded Output Substition). *Let Δ be a finite set of Σ_1 formulas, and assume that a and x are not free in $\Gamma, \Delta, A[\mathsf{0}]$. Then $EA(;)$ proves, for all formulas A and all function terms f, the rule*

$$\frac{\Gamma, \neg a \leq x.A[a] \qquad \Delta, f(x)\!\downarrow}{\Gamma, \Delta, \neg a \leq x, A[f(a)].}$$

Proof. As $x \leq x$ is provable, and a is not free in $\Gamma, A[\mathsf{0}]$, we immediately get $\Gamma, A[x]$. The first special case of theorem 10 then gives $\Gamma, \Delta, A[f(x)]$, and the claim follows from the second special case of the same theorem. □

A quite surprising, but extremely useful consequence is the following corollary. Comparing the first premise to the conclusion you find that it simply allows us to drop a premise of the form $f(\vec{a}) \leq z$ for a provably total f in favour of $\vec{a} \leq z$. Even if looking implausible at first sight, when thinking of the free input variable z as being universially quantified rather than a fixed parameter, it turns into a commonly used principle of arithmetic, because we then just have to instantiate z with any provable common upper bound for $\vec{a}$ and $f(\vec{a})$ in the premise. Indeed, inspecting the proof reveals that this is exactly what is going on here, that the bound z in the conclusion is different from the z in the premise.

Corollary 12. *Let Δ be a finite set of Σ_1 formulas, and assume that z and $\vec{x}$ are not free in $\Gamma, \Delta(\vec{a})$. Then $EA(;)$ proves, for all function terms f, the rule*

$$\frac{\neg\vec{a}, f(\vec{a}) \leq z, \Gamma(\vec{a}) \qquad \Delta, f(\vec{x})\downarrow}{\neg\vec{a} \leq z, \Delta, \Gamma(\vec{a}).}$$

Proof. For notational simplicity we only prove the case where $\vec{a} \equiv a, b$. The general case is similar.

Define $f'(a,b) = \max(a, b, f(a,b))$ such that $EA(;)$ proves

$$(*) \qquad \max(x,y,f(x,y))\downarrow \wedge\, x, y, f(x,y) \leq \max(x,y,f(x,y)).$$

This can be achieved by defining

$$\max(a,b) = a + (b \dot{-} a), \qquad \max(a,b,c) = \max(\max(a,b),c).$$

From lemma 4.1 and 4.3 we know that $x + y\downarrow$, $y \dot{-} x\downarrow$ and $x \leq x + y$. By the first special case of theorem 10 we get $\max(x,y)\downarrow$ and $x \leq \max(x,y)$. Once we have shown $y \leq \max(x,y)$, which we will do below, we can use the same special case to push this up to $\max(x,y,z)\downarrow \wedge\, x, y, z \leq \max(x,y,z)$, and then, using the second premise of this corollary, also to $(*)$.

In the main proof we continue using theorem 10. Apply the first special case to the first premise of this lemma to substitute $f'(x,y)$ for z, and substitute x and y for a and b respectively. This leaves us with $\neg x, y, f(x,y) \leq f'(x,y), \Delta, \Gamma(x,y)$. Now we can cut the three bounds, and use the second special case of theorem 10 twice to get $\neg a, b \leq x, \Delta, \Gamma(a,b)$ which proves the corollary.

As to the proof of $y \leq \max(x,y)$, it arises by replacing each input bounded induction with the original induction rule of $EA(;)$ in the following proof of $a, b \leq x \to b \leq \max(a,b)$. This proceeds by case distinction according to the first conjunct of lemma 4.6. If $a \leq b$, we simply compute $\max(a,b) = b$ by lemma 5.3, and the claim is immediate from lemma 4.2. Otherwise $b \dot{-} a = \mathsf{0}$ by definition, so $b \leq a = a + (b \dot{-} a) = \max(a,b)$. □

Of course, this corollary can also be applied to remove several functions simultaneously. For, if we have the premises $\neg\vec{a}, f(\vec{a}), g(\vec{a}) \leq z, \Gamma(\vec{a})$ and $\Delta, f(\vec{x})\downarrow \wedge\, g(\vec{x})\downarrow$, we can define $f'(\vec{a}) = \max(\vec{a}, f(\vec{a}), g(\vec{a}))$ like in the proof above and continue in the same way.

Even though the corollary removes a premise, its application, by an abuse of terminology, may rather look like adding one. For, when trying to prove $\Delta, \Gamma(\vec{a})$ from the premise $\vec{a} \leq z$, the corollary allows us to use the additional premise $f(\vec{a}) \leq z$ as well.

Chapter 4

Computing within $EA(;)$

4.1 Arithmetic

This section uses the results from the previous sections to develop a sufficient amount of arithmetic needed for the encoding of sequence numbers.

We first define a characteristic function for equality by

$$\chi_=(0,0) = 1, \qquad \chi_=(0,\mathsf{s}b) = 0, \qquad \chi_=(\mathsf{s}a,0) = 0, \qquad \chi_=(\mathsf{s}a,\mathsf{s}b) = \chi_=(a,b).$$

This definition satisfies

$$\vdash b \leq x \to \chi_=(a,b)\!\downarrow \wedge (\chi_=(a,b) = 1 \leftrightarrow a = b)$$

which we often will refer to as the "Adequacy of $\chi_=$". Notice that we don't need to assume $a \leq x$, this motivates our choice for the somewhat unusual defining equations and will be exploited in the sequel. The proof would use input bounded induction for $A[b] :\equiv \forall a.\chi_=(a,b)\!\downarrow \wedge (\chi_=(a,b) = 1 \leftrightarrow a = b)$, where the induction step follows from $\chi_=(\mathsf{p}a,b)\!\downarrow \to \chi_=(a,\mathsf{s}b)\!\downarrow$ and $(\chi_=(\mathsf{p}a,b) = 1 \leftrightarrow \mathsf{p}a = b) \to (\chi_=(a,\mathsf{s}b) = 1 \leftrightarrow a = \mathsf{s}b)$. Both are proven by cases on a.

Next, in addition to the already defined addition and modified subtraction we define multiplication and exponentiation in the usual way,

$$a \cdot 0 = 0, \qquad a \cdot \mathsf{s}b = (a \cdot b) + a, \qquad a^0 = 1, \qquad a^{\mathsf{s}b} = a^b \cdot a.$$

Under these definitions it is quite straightforward to prove the following results which we are going to use without always referring to explicitly:

Lemma 13. *$EA(;)$ proves*

1. $b \leq x \to (a + b) \dot{-} b = a$,

2. $a, b \leq x \to a + b = b + a$,
3. $b, c \leq x \to a + (b + c) = (a + b) + c$,
4. $a, b \leq x \to a \cdot b\!\downarrow$,
5. $a, b \leq x \to a \cdot 1 = a \wedge 0 \cdot a = 0 \wedge \mathsf{s}b \cdot a = b \cdot a + a \wedge a \cdot b = b \cdot a$,
6. $a, b, c \leq x \to (a + b) \cdot c = (a \cdot c) + (b \cdot c) \wedge (a \cdot b) \cdot c = a \cdot (b \cdot c)$,
7. $a, b \leq x \wedge c < b \to a + c < a + b \wedge a \cdot c < a \cdot b \wedge c \leq c^2 < b^2$,
8. $a, b \leq x \to a^b\!\downarrow$,
9. $a, b, c \leq x \to a^{(b+c)} = a^b \cdot a^c \wedge (a^b)^c = a^{b \cdot c} \wedge (c \neq 0 \to a^c + b^c \leq (a + b)^c)$,
10. $a, b \leq x \wedge c \leq b \wedge a \neq 0 \to a^c \leq a^b$.

Proof. The first item is immediate by (input bounded output) induction on b, the second by induction on a, where the base case and the induction step both use lemma 4.1 and the third by induction on c where the induction step is

$$(a + b) + \mathsf{s}c = \mathsf{s}((a + b) + c) = \mathsf{s}(a + (b + c)) = a + \mathsf{s}(b + c) = a + (b + \mathsf{s}c)$$

and the premise $b, c \leq x$ is needed to ensure $a + b\!\downarrow$ and $b + c\!\downarrow$.

In item 4, $a \cdot b\!\downarrow$ is proven by using input bounded output induction on b, where in the induction step the induction hypothesis gives $a \cdot b\!\downarrow$, so we can substitute it for a in lemma 4.1, and we are done.

The first conjunct of item 5 is immediate from lemma 4.1, and the second and the third conjunct are both proven by induction on a, where the induction step of the third uses previous items of this lemma and lemma 4.1 to ensure $b \cdot a\!\downarrow$ and $b \cdot a + a\!\downarrow$, and to compute

$$\mathsf{s}b \cdot \mathsf{s}a = \mathsf{s}b \cdot a + \mathsf{s}b = \mathsf{s}((b \cdot a + a) + b) = \mathsf{s}(b \cdot a + (a + b)) = \mathsf{s}(b \cdot a + (b + a)) = \mathsf{s}((b \cdot a + b) + a = b \cdot \mathsf{s}a + \mathsf{s}a.$$

Now we are ready to show the fourth conjunct by induction on b. The base case is immediate from the second conjunct, and the induction step from the first and the third by computing $a \cdot \mathsf{s}b = a \cdot b + a = b \cdot a + a = \mathsf{s}b \cdot a$.

The proof of item 6 is interesting because it shows how to apply corollary 12: When proving the first conjunct by induction on c, for the induction step we want to compute

$$(a+b) \cdot \mathsf{s}c = (a+b) \cdot c + (a+b) = (a \cdot c + b \cdot c) + (a+b) = ((a \cdot c + b \cdot c) + a) + b = (a \cdot c + (b \cdot c + a)) + b$$
$$= (a \cdot c + (a + b \cdot c)) + b = ((a \cdot c + a) + b \cdot c) + b = (a \cdot c + a) + (b \cdot c + b) = (a \cdot \mathsf{s}c) + (b \cdot \mathsf{s}c).$$

This computation is justified by the second and third items of this lemma upon the assumptions that the terms $a + b$, $a \cdot c$, $b \cdot c$, $a \cdot c + a$ and $a \cdot c + b \cdot c$ are all defined and

that $a, b, b \cdot c \leq x$. Now definedness follows from previous items and the latter bounds (augmented with $c \leq x$), and finally corollary 12 allows us to remove the premise $b \cdot c \leq x$.

All other items are verified in the usual way (after computing $b \leq x \to b^2 = b^{\mathsf{0}} \cdot b \cdot b = b \cdot b$ in item 7), applying corollary 12 whenever we need output bounds for terms that already have been shown to be defined, except for item 8 which needs some more work.

Here we can't use induction outright, as for the induction step we would need $a^b \cdot a{\downarrow}$ for which we would have to be able to assume $a^b \leq x$. But we can't use corollary 12 here because $a^b{\downarrow}$ is only the induction hypothesis, but no theorem of $EA(;)$ yet. We need a principle of function bounded induction instead, with a variant of $f(a, b) = a + 2^b$ as bounding function. This function f has defining equations

$$f(a, \mathsf{0}) = \mathsf{s}a, \qquad f(a, \mathsf{s}b) = f(f(a, b), b),$$

and has been shown to be provably total even on outputs in its first argument in Ostrin and Wainer [17, p. 379]. We first show

$$b, c \leq x \;\to\; f(a + b, c) = f(a, c) + b, \tag{1}$$

$$b \leq x \;\to\; f(\mathsf{0}, \mathsf{s}b) = f(\mathsf{0}, b) + f(\mathsf{0}, b), \tag{2}$$

$$a, b \leq x \;\to\; f(\mathsf{0}, a + b) = f(\mathsf{0}, a) \cdot f(\mathsf{0}, b). \tag{3}$$

The first claim is proven by induction for $A[c] :\equiv \forall a. f(a+b, c) = f(a, c)+b$. The base case is immediate as we have seen $b \leq x \to a + b{\downarrow} \wedge s(a + b) = \mathsf{s}a + b$ above. For the induction step we instantiate the universal in the induction hypothesis to a and to $f(a, c)$ (which is defined) to compute

$$f(a + b, \mathsf{s}c) = f(f(a + b, c), c) = f(f(a, c) + b, c) = f(f(a, c), c) + b = f(a, \mathsf{s}c) + b.$$

The second claim then uses the additional assumption $f(\mathsf{0}, b) \leq x$ allowed by corollary 12, first to deduce $f(\mathsf{0}, b) = \mathsf{0} + f(\mathsf{0}, b)$ from lemma 4.1, then to use (1) in computing

$$f(\mathsf{0}, \mathsf{s}b) = f(f(\mathsf{0}, b), b) = f(\mathsf{0} + f(\mathsf{0}, b), b) = f(\mathsf{0}, b) + f(\mathsf{0}, b).$$

The third claim is by induction on a. The base case under the additional assumption $f(\mathsf{0}, b) \leq x$ is immediate from lemma 4.1 and the fifth item of this lemma, the step uses (2) and previous items of this lemma for

$$\begin{aligned} f(\mathsf{0}, a + \mathsf{s}b) &= f(\mathsf{0}, \mathsf{s}(a + b)) = f(\mathsf{0}, a + b) + f(\mathsf{0}, a + b) \\ &= f(\mathsf{0}, a) \cdot f(\mathsf{0}, b) + f(\mathsf{0}, a) \cdot f(\mathsf{0}, b) = f(\mathsf{0}, a) \cdot (f(\mathsf{0}, b) + f(\mathsf{0}, b)) = f(\mathsf{0}, a) \cdot f(\mathsf{0}, \mathsf{s}b) \end{aligned}$$

where the necessary bounds are provided by corollary 12.

Now we are ready to return to the main proof. This uses lemma 3, applied to the formula $A[b] :\equiv a^b{\downarrow} \wedge a^b \leq f(\mathsf{0}, a \cdot b)$. The base case is immediate. For the induction step we first use

(2) to show $a \leq f(\mathsf{0}, a)$ by induction on a. Then, assuming $f(\mathsf{0}, a \cdot b) \leq x$ by corollary 12, the induction hypothesis and transitivity give $a^b{\downarrow}$ and $a^b \leq x$ and we compute

$$a^{\mathsf{s}b} = a^b \cdot a \leq f(\mathsf{0}, a \cdot b) \cdot f(\mathsf{0}, a) = f(\mathsf{0}, a \cdot b + b) = f(\mathsf{0}, a \cdot \mathsf{s}b),$$

using corollary 12 to provide bounds. □

Corollary 12 also helps a lot in defining the bounded maximum of a given function, and a bounded μ-operator that searches for its least null.

Lemma 14. *If $f(a, b)$ is a provably total function of $EA(;)$, then there is a function $f_0(a, b) = \max_{d \leq a}(f(d, b))$ such that $EA(;)$ proves*

$$a, b \leq x \rightarrow f_0(a, b){\downarrow} \wedge f(a, b) \leq f_0(a, b) \wedge \exists d \leq a. f_0(a, b) = f(d, b) \\ \wedge \forall d \leq a. f_0(d, b) \leq f_0(a, b).$$

Proof. Let f_0 be defined by

$$\begin{aligned} f_0(\mathsf{0}, b) &= f(\mathsf{0}, b), \\ f_0(\mathsf{s}a, b) &= \begin{cases} f_0(a, b) & \text{if } f(\mathsf{s}a, b) \dot{-} f_0(a, b) = \mathsf{0}, \\ f(\mathsf{s}a, b) & \text{else.} \end{cases} \end{aligned}$$

The conditional construction in the defining equation is meant to stand for the call-by-name construction of lemma 1, although this is not crucial here. The definitional clause for the successor case defines $f_0(\mathsf{s}a, b)$ to be a kind of $\max(f(\mathsf{s}a, b), f_0(a, b))$. We could have used the max function defined in the proof of corollary 12, but our choice makes the proof of the lemma easier.

We show the first three conjuncts by input bounded output induction (lemma 3) on a. In the base case $f_0(\mathsf{0}, b)$ is defined because $f(\mathsf{0}, b)$ is. Using corollary 12 to assume $f(\mathsf{0}, b) \leq x$ enables us to get $f(\mathsf{0}, b) \leq f(\mathsf{0}, b) = f_0(\mathsf{0}, b)$ from lemma 4.2. The third conjunct is trivial.

For the induction step we assume $\mathsf{s}a \leq x$ by corollary 12. By the induction hypothesis, $f_0(a, b)$ is defined and equals $f(d, b)$ for some $d \leq a$, where $d \leq \mathsf{s}a$ by lemma 4.2 and transitivity. This d by transitivity also satisfies $d \leq x$, so we have $f(d, b){\downarrow}$, and we can use corollary 12 to add the premise $f(d, b) \leq x$. Then the term $f(\mathsf{s}a, b) \dot{-} f_0(a, b)$ is defined and we distinguish cases according to lemma 1. If it equals $\mathsf{0}$, then $f_0(\mathsf{s}a, b) = f_0(a, b)$ which we already have shown to be defined and to equal $f(d, b)$ for some $d \leq \mathsf{s}a$, and we have $f(\mathsf{s}a, b) \leq f_0(a, b) = f_0(\mathsf{s}a, b)$ by definition of $\leq$. Otherwise $f_0(\mathsf{s}a, b) = f(\mathsf{s}a, b)$ which is defined by assumption on f, and $\mathsf{s}a$ can serve as witness for the existential quantifier. Finally, under the additional premise $f(\mathsf{s}a, b) \leq x$ which is allowed by corollary 12 we immediately get $f(\mathsf{s}a, b) \leq f(\mathsf{s}a, b) = f_0(\mathsf{s}a, b)$ from lemma 4.2.

The fourth conjunct is proven by applying corollary 7 to $A[a] :\equiv f_0(d,b) \leq f_0(a,b)$. The base case $a = d$ is immediate from lemma 4.2 as we can assume $f_0(a,b) \leq x$ by corollary 12. The induction step uses the same case distinction as above. Either we have $f(\mathsf{s}a,b) \leq f_0(a,b)$, where $f_0(d,b) \leq f_0(a,b) = f_0(\mathsf{s}a,b)$ by induction hypothesis and the defining equation, otherwise $f_0(a,b) \leq f(\mathsf{s}a,b) = f_0(\mathsf{s}a,b)$ by the first conjunct of lemma 4.6 (assuming $f_0(a,b), f(\mathsf{s}a,b) \leq x$ by corollary 12) and the claim then follows from the induction hypothesis and transitivity. □

Lemma 15. *If $f(a,b)$ is a provably total function of $EA(;)$, then there is a function $f_1(a,b) = \mu d \leq a.(f(d,b) = \mathsf{0})$ such that $EA(;)$ proves*

$$a,b \leq x \to f_1(a,b)\!\downarrow \wedge\, (f(a,b) = \mathsf{0} \to f_1(a,b) \leq a) \\ \wedge\, (f(a,b) = \mathsf{0} \wedge \forall d < a.f(d,b) \neq \mathsf{0} \to \forall d \leq x.d \geq a \to f_1(d,b) = a).$$

Proof. Let f_1 be defined by

$$f_1(\mathsf{0},b) = \begin{cases} \mathsf{0} & \text{if } f(\mathsf{0},b) = \mathsf{0}, \\ \mathsf{1} & \text{else,} \end{cases}$$

$$f_1(\mathsf{s}a,b) = \begin{cases} \begin{cases} \mathsf{s}a & \text{if } f(\mathsf{s}a,b) = \mathsf{0} \\ \mathsf{ss}a & \text{else} \end{cases} & \text{if } \chi_{=}(f_1(a,b),\mathsf{s}a) = \mathsf{1}, \\ f_1(a,b) & \text{else.} \end{cases}$$

Assuming $a,b \leq x$, we first show

$$f_1(a,b)\!\downarrow \wedge\, f_1(a,b) \leq \mathsf{s}a \wedge (\forall d \leq a.f(d,b) \neq \mathsf{0}) \to f_1(a,b) = \mathsf{s}a$$

by input bounded induction on a, assuming $b \leq x$. The base case is immediate from lemma 1 (and lemma 2.2 for the third claim), as $f(\mathsf{0},b)$ is defined by assumption.

For the induction step we can additionally assume $\mathsf{s}a \leq x$ by corollary 12, thus $f(\mathsf{s}a,b)\!\downarrow$, and the induction hypothesis ensures $\chi_{=}(f_1(a,b),\mathsf{s}a)\!\downarrow$. By lemma 1 and adequacy of $\chi_{=}$ we get

$$f_1(a,b) = \mathsf{s}a \wedge f(\mathsf{s}a,b) = \mathsf{0} \;\to\; f_1(\mathsf{s}a,b) = \mathsf{s}a, \tag{4}$$

$$f_1(a,b) = \mathsf{s}a \wedge f(\mathsf{s}a,b) \neq \mathsf{0} \;\to\; f_1(\mathsf{s}a,b) = \mathsf{ss}a, \tag{5}$$

$$f_1(a,b) \neq \mathsf{s}a \;\to\; f_1(\mathsf{s}a,b) = f_1(a,b). \tag{6}$$

One of the three cases must hold, and each satisfies $f_1(\mathsf{s}a,b)\!\downarrow$ and $f_1(\mathsf{s}a,b) \leq \mathsf{ss}a$ (in case of (6), use the induction hypothesis and transitivity). For the third conjunct we assume $\forall d \leq \mathsf{s}a.f(d,b) \neq \mathsf{0}$. This by lemma 4.2 implies $\forall d \leq a.f(d,b) \neq \mathsf{0}$ as well as $f(\mathsf{s}a,b) \neq \mathsf{0}$. From the former we get $f_1(a,b) = \mathsf{s}a$ by induction hypothesis, so by (5) we get $f_1(\mathsf{s}a,b) = \mathsf{ss}a$ which completes the induction step.

We now can show $f(a,b) = \mathsf{0} \to f_1(a,b) \leq a$ by case distinction on a. The case $a = \mathsf{0}$ is immediate from the defining equations, in the successor case the premise excludes implication (5), whereas implication (4) implies $f_1(\mathsf{s}a, b) = \mathsf{s}a \leq \mathsf{s}a$ with help of lemma 4.2, and in case of (6) we have $f_1(\mathsf{s}a, b) = f_1(a,b) \leq \mathsf{s}a$ by the above.

Finally we show

$$(f(a,b) = \mathsf{0} \wedge \forall d < a.f(d,b) \neq \mathsf{0} \wedge a \leq d \leq x) \to f_1(d,b) = a$$

by using corollary 7 for $A[d] :\equiv f_1(d,b) = a$, still under the assumption $b \leq x$. This will be enough to prove the third conjunct of the lemma.

The base case $d = a$ uses cases on a. If $a = \mathsf{0}$ we immediately have $f_1(\mathsf{0}, b) = \mathsf{0} = a$. In the successor case the premise $\forall d < \mathsf{s}a.f(d,b) \neq \mathsf{0}$ by definition of $<$ and lemma 2.3 is equivalent to $\forall d \leq a.f(d,b) \neq \mathsf{0}$, so from the above we have $f_1(a,b) = \mathsf{s}a$, and by (4) we conclude $f_1(\mathsf{s}a, b) = \mathsf{s}a$.

In the induction step the induction hypothesis says $f_1(d,b) = a$. Notice that $a \leq d$ implies $a \neq \mathsf{s}d$ (for example by observing that $a = \mathsf{s}d$ would imply $\mathsf{s}d \leq a$, thus contradicting $a < \mathsf{s}d$ by lemma 4.6). Then implication (6) yields $f_1(\mathsf{s}d, b) = f_1(d,b) = a$. □

4.2 Sequence Numbers

To continue towards the encoding of ordinal arithmetic this section introduces an arithmetization of ternary sequences $\langle \cdot, \cdot, \cdot \rangle$ including the corresponding projection functions and its characteristic function. It is given by the following equations which, as always, we assume to be part of the (suppressed) equational program P:

$$\begin{gathered}
\langle a,b \rangle = (a+b)^2 + \mathsf{s}b, \qquad \langle a,b,c \rangle = \langle \langle a,b \rangle, c \rangle, \\
f'(a,b) = \mu c \leq a.(b \dot{-} (\mathsf{s}c)^2 = \mathsf{0}), \qquad h(a) = f'(a,a), \\
(a)_1 = \mathsf{p}(a \dot{-} h(a)^2), \qquad (a)_0 = h(a) \dot{-} (a)_1, \\
\mathit{3Seq}(a) = \chi_{=}(\langle \langle (a)_{0,0}, (a)_{0,1} \rangle, (a)_1 \rangle, a).
\end{gathered}$$

In the last equations $(a)_{0,0}$ and $(a)_{0,1}$ stand for $((a)_0)_0$ and $((a)_0)_1$ respectively and we will usually write $\mathit{3Seq}(a)$ instead of $\mathit{3Seq}(a) = \mathsf{1}$. These definitions satisfy

Lemma 16. *$EA(;)$ proves*

1. $a,b,c \leq x \to \langle a,b \rangle\!\downarrow \wedge \langle a,b,c \rangle\!\downarrow \wedge f'(a,b)\!\downarrow \wedge h(a)\!\downarrow \wedge (a)_0\!\downarrow \wedge (a)_1\!\downarrow \wedge \mathit{3Seq}(a)\!\downarrow$.
2. $a,b,c \leq x \to h(\langle a,b \rangle) = a + b \wedge (\langle a,b \rangle)_0 = a \wedge (\langle a,b \rangle)_1 = b \wedge \mathit{3Seq}(\langle a,b,c \rangle)$.
3. $a,b,c \leq x \to \langle a,b,c \rangle \neq \mathsf{0} \wedge (a' \leq a \wedge b' \leq b \wedge c' \leq c \to a',b',c' < \langle a',b',c' \rangle \leq \langle a,b,c \rangle)$.

4. $a, b, c, c' \leq x \to \mathsf{s}(a+b)^2 \leq \langle a, b\rangle \wedge \langle a, b, \mathsf{s}c\rangle \leq (a+b+\mathsf{ss}c)^4 \wedge \langle a, b, c+c'\rangle \leq \langle\langle a, b, c\rangle, b, c'\rangle$.

5. $d \leq x \wedge \mathit{3Seq}(d) \to \exists a, b, c < d.d = \langle a, b, c\rangle$.

Proof. 1. This is fairly immediate, making extensive use of corollary 12. For example, as $a + b\downarrow$ is provable, we can use the additional assumption $a + b \leq x$. This gives $(a + b)^2\downarrow$ and $\langle a, b\rangle\downarrow$, which in turn allows us to add the premise $\langle a, b\rangle \leq x$, and we can conclude $\langle a, b, c\rangle\downarrow$. To continue, as $f'(a, b)$ is defined by lemma 15, so is $h(a)$, we can assume $h(a) \leq x$ again, so $h(a)^2\downarrow$, thus $(a)_1\downarrow$ by assuming $h(a)^2 \leq x$ and so on.

2. Let $f(a, b) = b \dot{-} (\mathsf{s}a)^2$. Then $f'(a, b) = \mu c \leq a.(f(c, b) = \mathsf{0})$. Using the abbreviations $c := a + b$ and $d := \langle a, b\rangle = c^2 + \mathsf{s}b$ we are going to show

$$f(c, d) = \mathsf{0} \qquad \text{and} \qquad \forall c' < c.f(c', d) \neq \mathsf{0},$$

always assuming $a, b \leq x$. As we can add the premise $\langle a, b\rangle \leq x$ by corollary 12, and $c \leq c^2 + \mathsf{s}b$ is provable, we then get $f'(d, d) = c$ from lemma 15 which will prove the first conjunct.

But $f(c, d) = \mathsf{0}$ follows immediately from

$$d = c^2 + \mathsf{s}b \leq c^2 + \mathsf{s}c \leq \mathsf{s}c \cdot c + \mathsf{s}c = (\mathsf{s}c) \cdot (\mathsf{s}c) = (\mathsf{s}c)^2,$$

the definition of $\leq$, and transitivity (assuming $(\mathsf{s}c)^2 \leq x$ by corollary 12). On the other hand, for $c' < c$ we have $\mathsf{s}c' \leq c$ and $(\mathsf{s}c')^2 \leq c^2 \leq c^2 + b < \mathsf{s}(c^2 + b) = d$, so $d \not\leq (\mathsf{s}c')^2$ by lemma 4.6.

The remaining parts are now immediate: Compute first, under the additional premise $(a + b)^2 \leq x$ justified by corollary 12,

$$(\langle a, b\rangle)_1 = \mathsf{p}((a+b)^2+\mathsf{s}b) \dot{-} (a+b)^2 = ((a+b)^2+b) \dot{-} (a+b)^2 = (b+(a+b)^2) \dot{-} (a+b)^2 = b,$$

then $(\langle a, b\rangle)_0 = (a+b) \dot{-} b = a$, and finally $\mathit{3Seq}(\langle a, b, c\rangle) = \chi_{=}(\langle\langle a, b\rangle, c\rangle, \langle a, b, c\rangle) = \mathsf{1}$.

3. By simple computations, still using corollary 12 many times. In the first conjunct we observe $(\langle a, b\rangle + \mathsf{s}c)^2\downarrow$, this allows us to compute

$$\langle a, b, c\rangle = (\langle a, b\rangle + c)^2 + \mathsf{s}c = \mathsf{s}((\langle a, b\rangle + c)^2 + c) \neq \mathsf{0}.$$

For the second we first establish $a', b' \leq (a' + b')^2 + b' < \mathsf{s}((a' + b')^2 + b') = \langle a', b'\rangle$ and $a'+b' \leq a+b$, so $(a'+b')^2 \leq (a+b)^2$ and $\langle a', b'\rangle = (a'+b')^2+\mathsf{s}b' \leq (a+b)^2+\mathsf{s}b = \langle a, b\rangle$, and the result immediately lifts up to the ternary sequences.

4. Some more easy computations, using corollary 12 extensively. The first conjunct is given by $\mathsf{s}(a + b)^2 \leq \mathsf{s}(a + b)^2 + b = (a + b)^2 + \mathsf{s}b = \langle a, b\rangle$, and the second by

$$\begin{aligned}\langle a, b, \mathsf{s}c\rangle = ((a + b)^2 + \mathsf{s}b + \mathsf{s}c)^2 + \mathsf{ss}c &\leq ((a + b)^2 + b + \mathsf{ss}c + \mathsf{ss}c)^2 \\ &\leq ((a + b)^2 + 2 \cdot (a + b) \cdot \mathsf{ss}c + (\mathsf{ss}c)^2)^2 = (a + b + \mathsf{ss}c)^4.\end{aligned}$$

As a preparation for the third conjunct we compute

$$\langle a, b+c\rangle = (a+(b+c))^2 + \mathsf{s}(b+c) = ((a+b)+c)^2 + b + \mathsf{s}c$$
$$\leq ((a+b)+c+b)^2 + \mathsf{s}c \leq ((a+b)^2 + c + \mathsf{s}b)^2 + \mathsf{s}c = ((a+b)^2 + \mathsf{s}b + c)^2 + \mathsf{s}c = \langle\langle a, b\rangle, c\rangle.$$

Then

$$\langle a, b, c+c'\rangle = \langle\langle a, b\rangle, c+c'\rangle \leq \langle\langle a, b\rangle, c+(b+c')\rangle \leq \langle\langle\langle\langle a, b\rangle, c\rangle, b\rangle, c'\rangle = \langle\langle a, b, c\rangle, b, c'\rangle.$$

5. This is trivial for $d = \mathsf{0}$ because $\langle\langle(\mathsf{0})_{0,0}, (\mathsf{0})_{0,1}\rangle, (\mathsf{0})_1\rangle = \langle\langle\mathsf{0}, \mathsf{0}\rangle, \mathsf{0}\rangle = \langle\mathsf{1}, \mathsf{0}\rangle = 2 \neq \mathsf{0}$ shows that *3Seq*($\mathsf{0}$) doesn't hold. For the successor case we first show the general principle $\mathsf{0} \neq a \leq x \to (a)_0, (a)_1 < a$. This is easy for the right projection, as $(a)_1 = \mathsf{p}(a \dot{-} h(a)^2) \leq \mathsf{p}a < a$ for $a \neq \mathsf{0}$ (uses corollary 12 in order to add the premise $h(a)^2 \leq x$). For the left projection we observe that $a \dot{-} (\mathsf{sp}a)^2 = a \dot{-} a^2 = \mathsf{0}$ for $a \neq \mathsf{0}$, so $f'(\mathsf{p}a, a) \leq \mathsf{p}a < a$ by lemma 15, and $(a)_1 = h(a) \dot{-} (a)_0 \leq h(a) = f'(a, a) = f'(a, \mathsf{p}a)$.

   To prove the lemma choose $a := (d)_{0,0}, b := (d)_{0,1}, c := (d)_1$ as witnesses for the existential. By the above we get $a, b, c < d$. This implies $a, b, c \leq x$ by transitivity, so $\langle\langle a, b\rangle, c\rangle\!\downarrow$, and the assumption *3Seq*(a) by adequacy of $\chi_=$ implies $\langle\langle a, b\rangle, c\rangle = d$. □

4.3 Induction on Term Structure

In this section we introduce the coding of ordinals as terms and develop a principle of induction over term structure, stated as lemma 18.

1. $\mathit{left}(\langle c, a, b\rangle) = c$,
   $\mathit{exp}(\langle c, a, b\rangle) = a$,
   $\mathit{coeff}(\langle c, a, b\rangle) = b$.

2. $$\mathit{Ord}(a) = \begin{cases} \mathsf{1} & \text{if } a = 0, \\ \mathit{Ord}'(a) & \text{if } \mathit{3Seq}(a), \\ \mathsf{0} & \text{if } \sim\mathit{3Seq}(a). \end{cases}$$
   $\mathit{Ord}'(\langle c, a, \mathsf{0}\rangle) = \mathsf{0}$,
   $$\mathit{Ord}'(\langle c, a, \mathsf{s}b\rangle) = \begin{cases} \mathit{Ord}(a) & \text{if } c = 0, \\ \mathit{Ord}(c)\ \&\ \mathit{Ord}(a)\ \&\ {\sim}(\chi_=(\mathit{exp}(c), a)) & \text{if } \mathit{3Seq}(c), \\ \mathsf{0} & \text{if } \sim\mathit{3Seq}(c). \end{cases}$$
   $\mathit{Ord}(a) :\equiv \mathit{Ord}(a) = \mathsf{1}$.

It is important to notice that the conditionals used to define *Ord* and *Ord*′ are call-by-name as in lemma 1. We further observe that $a \leq x \wedge \mathit{3Seq}(a) \to \mathit{exp}(a)\!\downarrow \wedge\, \mathit{exp}(a) \leq x$ is immediate from lemma 16.5. While *Ord*($\mathsf{0}$) holds trivially, $\neg\mathit{Ord}(\mathsf{1})$ follows from $\neg\mathit{3Seq}(\mathsf{1})$

which we simply compute by the equational program. Since $3Seq(a)\downarrow$ for $a \leq x$ according to the previous lemma, it's legal to use the case distinction $a = 0 \vee 3Seq(a) \vee \sim 3Seq(a)$ when proving properties of Ord and Ord'.

$Ord(a)$ is meant to be true if and only if a is (the code of) an ordinal (in some weak normal form), with a triple $\langle c, a, b\rangle$ coding the ordinal $c + \omega^a \cdot b$ when c and a are codes of ordinals again and b is an ordinary natural number. This makes clear the meaning of the functions $left$, exp and $coeff$, as $left(c + \omega^a \cdot b)$ then evaluates to c, $exp(c + \omega^a \cdot b)$ to a and $coeff(c + \omega^a \cdot b)$ to b, but it's true only when $\langle c, a, b\rangle$ really encodes an ordinal. This restriction is necessary if we want equality of ordinals to be expressed by equality on their codes. Otherwise $\langle c, a, 0\rangle$ (which doesn't code an ordinal) and c would denote the same ordinal, but they are definitively different numbers. We will make all this more explicit in the next section.

For the time being we content ourselves with exposing the following observations which are needed in the proof of lemma 18. The first says that Ord is a (total) characteristic function, and the other two reflect the inductive structure of ordinal terms.

Lemma 17. *$EA(;)$ proves*

1. $a \leq x \rightarrow Ord(a) = 0 \vee Ord(a) = 1$.
2. $a \leq x \wedge Ord(a) \rightarrow a = 0 \vee (3Seq(a) \wedge Ord(left(a)) \wedge Ord(exp(a)))$.
3. $c, a, b \leq x \wedge Ord(\langle c, a, b\rangle) \rightarrow b \neq 0 \wedge (c = 0 \vee (3Seq(c) \wedge \sim \chi_{=}(rgt(c), a)))$.

Proof. 1. By course-of-value induction (lemma 6). Under the assumptions $a \leq x$ and $\forall b < a.Ord(b) = 1 \vee Ord(b) = 0$, we are going to show $Ord(a) = 1 \vee Ord(a) = 0$ by using the case distinction mentioned after the defining equations for Ord. If $a = 0$ we have $Ord(a) = 1$, if $\sim 3Seq(a)$ then $Ord(a) = 0$.

For the remaining case, $3Seq(a)$, we can assume $a = \langle c, a', b\rangle$ for some $c, a', b < a$ by lemma 16.5, and we have $Ord(a) = Ord'(a)$. If $b = 0$ then $Ord(a) = 0$, so we are left with the successor case, which uses another case distinction, now on $c \leq x$. If $c = 0$ we are done by the induction hypothesis. If $3Seq(c)$ then $exp(c)\downarrow$, so $\chi_{=}(exp(c), a'))\downarrow$ by adequacy of $\chi_{=}$, and therefore $(Ord(c)\ \&\ Ord(a')\ \&\ \chi_{=}(exp(c), a'))$ is defined and evaluates to 1 or to 0, provided that $Ord(c)$ and $Ord(a')$ are defined, which is the case due to the induction hypothesis. If $\sim 3Seq(b)$ finally we trivially have $Ord(b) = 0$.

Now we can apply lemma 6 which concludes the proof.

2. and 3. are proven together. We will prove

   i) $a \leq x \wedge Ord(a) \rightarrow a = 0 \vee 3Seq(a)$,

   ii) $c, a, b \leq x \wedge Ord(\langle c, a, b\rangle) \rightarrow b \neq 0 \wedge Ord(c, a) \wedge (c = 0 \vee (3Seq(c) \wedge exp(c) \neq a))$,

which together are equivalent to the conjunction of both items (recall that $Ord(c,a)$ abbreviates $Ord(c) \wedge Ord(a)$). For i), assume $a \leq x$ which allows us to carry out the following case distinction: If $a = \mathsf{0}$ or $\mathit{3Seq}(a)$, we are done, and if $\sim\mathit{3Seq}(a)$ we have $\neg Ord(a)$ from the defining equations.

For ii) assume $c, a, b \leq x$ and $Ord(\langle c,a,b\rangle)$, so we have $\langle c,a,b\rangle\!\downarrow$ and $\mathit{3Seq}(\langle c,a,b\rangle)$ by lemma 16. Thus lemma 1 gives $Ord(\langle c,a,b\rangle) = Ord'(\langle c,a,b\rangle)$. Now $b = \mathsf{0}$ would contradict the assumption $Ord(\langle c,a,b\rangle)$, so by the cases rule we can assume that b is a successor, and by lemma 1 again we have

$$\begin{aligned} c = \mathsf{0} &\rightarrow Ord(\langle c,a,b\rangle) = Ord(a), \\ \mathit{3Seq}(c) &\rightarrow Ord(\langle c,a,b\rangle) = (Ord(c)\ \&\ Ord(a)\ \&\ \sim\chi_{=}(exp(c),a)), \\ \sim\mathit{3Seq}(b) &\rightarrow \neg Ord(\langle b,c,d\rangle). \end{aligned}$$

Furthermore we have $Ord(b,c)\!\downarrow$, $\mathit{3Seq}(c)\!\downarrow$ and $\chi_{=}(exp(c),a)\!\downarrow$ by the first item of this lemma, lemma 16.1 and adequacy of $\chi_{=}$.

Now we distinguish the three cases on c mentioned just prior to this lemma. If $c = \mathsf{0}$ then $Ord(c) \wedge Ord(a) \wedge c = \mathsf{0}$ follows immediately from the first line. The case $\mathit{3Seq}(c)$ implies $Ord(c) \wedge Ord(a) \wedge \mathit{3Seq}(c) \wedge exp(c) \neq a$ by the second line and adequacy of both the encoding of propositional logic and of $\chi_{=}$. Finally, the case $\sim\mathit{3Seq}(c)$ is excluded by the third line. □

Lemma 18 (induction on ordinal terms). *$EA(;)$ proves, for all formulas A, the rule*

$$\frac{\Gamma, A[0] \qquad \Gamma, \neg\alpha, \gamma, a \leq x, \neg Ord(\langle\gamma,\alpha,\mathsf{s}a\rangle), \neg A[\gamma], \neg A[\alpha], A[\langle\gamma,\alpha,\mathsf{s}a\rangle]}{\Gamma, \neg\alpha \leq x, \neg Ord(\alpha), A[\alpha].}$$

In this lemma and in the sequel, greek minuscules $\alpha, \beta, \gamma, \ldots$ stand for output variables that are intended to range over (codes of) ordinals.

Proof. We show the conclusion by course-of-value induction (lemma 6), applied to the formula $B[a] :\equiv Ord(a) \rightarrow A[a]$. To do so we assume $a \leq x$, $\forall b < a.Ord(b) \rightarrow A[b]$ and $Ord(a)$ and we have to show $A[a]$.

We distinguish cases according to lemma 17.2. If $a = 0$ we have $A[a]$ from the first premise. Otherwise, $a = \langle\gamma,\alpha,b\rangle$ for some $\gamma, \alpha, b < a$ satisfying $Ord(\gamma,\alpha)$ and $b \neq \mathsf{0}$ by lemmas 16.5, 17.2 and 17.3, so by the induction hypothesis $A[\gamma]$ and $A[\alpha]$, and $A[a]$ follows from the second premise. □

Lemma 18 is mainly used to show that certain functions are total on codes of ordinals. Corollary 12 applies to such functions as well. For, if f is a function such that $EA(;)$ proves $(\alpha \leq x \wedge Ord(\alpha)) \rightarrow f(\alpha)\!\downarrow$, simply define $f'(a) = \begin{cases} f(a) & \text{if } Ord(a) \\ \mathsf{0} & \text{else} \end{cases}$. Then a premise $Ord(\alpha) \wedge f(\alpha) \leq x$ can be replaced by $Ord(\alpha) \wedge f'(\alpha) \leq x$, and f' is provably total by lemma 1. So corollary 12 applies and we are done.

4.4 Ordinal Arithmetic

In this section we will develop the amount of formalized ordinal arithmetic we need for the well-ordering proof. We start with introducing some operations and relations on ordinals, the intuitions behind them being explained below.

Definition 19.

1. $0 = \mathsf{0}$,
   $\omega^a = 0 + \omega^a \cdot \mathsf{1}$,
   $c + \omega^a \cdot \mathsf{0} = c$,
   $0 + \omega^a \cdot \mathsf{s}b = \langle 0, a, \mathsf{s}b\rangle$,
   $\langle c, a', b'\rangle + \omega^a \cdot \mathsf{s}b = \begin{cases} \langle c, a', \mathsf{s}(b' + b)\rangle & \text{if } \chi_=(a', a) = \mathsf{1}, \\ \langle\langle c, a', b'\rangle, a, \mathsf{s}b\rangle & \text{else.} \end{cases}$

2. $Lim(\alpha) :\equiv Ord(\alpha) \wedge \exists c, a, b.\alpha = \langle c, \mathsf{s}a, b\rangle$,
   $Succ(\alpha) :\equiv Ord(\alpha) \wedge \exists c, b.\alpha = \langle c, 0, b\rangle$.

3. $pred(0, e) = 0$,
   $pred(\langle\gamma, 0, \mathsf{1}\rangle, e) = \gamma$,
   $pred(\langle\gamma, 0, \mathsf{ss}a\rangle, e) = \langle\gamma, 0, \mathsf{s}a\rangle$,
   $pred(\langle\gamma, \langle a, \mathsf{0}, b\rangle, \mathsf{1}\rangle, e) = \gamma + \omega^{pred(\langle a,\mathsf{0},b\rangle,e)} \cdot e$,
   $pred(\langle\gamma, \langle a, \mathsf{0}, b\rangle, \mathsf{ss}a\rangle, e) = \langle\gamma, \langle a, 0, b\rangle, \mathsf{s}a\rangle + \omega^{pred(\langle a,\mathsf{0},b\rangle,e)} \cdot e$,
   $pred(\langle\gamma, \langle a, \mathsf{s}c, b\rangle, \mathsf{1}\rangle, e) = \gamma + \omega^{pred(\langle a,\mathsf{s}c,b\rangle,e)} \cdot \mathsf{1}$,
   $pred(\langle\gamma, \langle a, \mathsf{s}c, b\rangle, \mathsf{ss}a\rangle, e) = \langle\gamma, \langle a, \mathsf{s}c, b\rangle, \mathsf{s}a\rangle + \omega^{pred(\langle a,\mathsf{s}c,b\rangle,e)} \cdot \mathsf{1}$.

4. $\prec_e(\alpha, \beta) = \begin{cases} \mathsf{0} & \text{if } \neg Ord(\beta), \\ \mathsf{0} & \text{if } \beta = 0, \\ \mathsf{1} & \text{if } Ord(\beta)\ \&\ \beta \neq 0\ \&\ \chi_=(\alpha, pred(\beta, e)) = \mathsf{1}, \\ \prec_e(\alpha, pred(\beta, e)) & \text{else.} \end{cases}$
   $\alpha \prec_e \beta :\equiv \prec_e(\alpha, \beta) = \mathsf{1}$,
   $\alpha \preceq_e \beta :\equiv \alpha \prec_e \beta \vee \alpha = \beta$.

The first group ensures that the ordinal 0 is encoded by the constant $\mathsf{0}$, and defines the ternary function $(\gamma, \alpha, b) \mapsto \gamma + \omega^\alpha \cdot b$. This function, in which ω is no variable, but thought of as part of the function symbol, combines ordinal addition, multiplication and exponentiation, and is strong enough to reach all ordinals up to ε_0. This encoding is a little bit more efficient with respect to the sizes of codes than the Cantor Normal Form because it collects ω-powers with the same exponent, i.e. $\gamma + \omega^\alpha + \ldots + \omega^\alpha$ is coded into the compact form $\langle\gamma, \alpha, n\rangle$ of constant length in contrast to the long sequence $\langle\gamma, \alpha, \ldots, \alpha\rangle$ which corresponds to the Cantor Normal Form and whose length depends on n. As another difference to the Cantor Normal Form we don't require the exponents to be ordered.

In a few places we want to use a shorthand notation for ordinal exponentiation alone. The second equation defines it as a unary function in terms of the ternary one. The three defining equations of the latter indicate how to prove its properties: By a case distinction on its third argument, where the successor case is divided into the two subcases on the first argument given by lemma 17.2, provided this is (a code of) an ordinal. Here a last sub-subcase distinction has to deal with the conditional construction in the third equation.

$Lim(\alpha)$ and $Succ(\alpha)$ are true if and only if α is (code of) a limit or successor ordinal respectively. We won't use them in any function definition (e.g. to select branches in a choice function) thus there's no need to define their characteristic functions. $pred(\alpha, e)$ computes the immediate e-predecessor of α with respect to the standard fundamental sequence for α, and $\prec_e$ is its transitive closure, i.e. $\alpha \prec_e \beta$ holds if and only if α belongs to the set of all e-predecessors of β (i.e. if $\alpha \in \beta[e]$).

The inductive structure of the equational program for $pred(\alpha, e)$ reveals that its properties will be proven by applying lemma 18 where the induction step only uses the induction hypothesis for $exp(\alpha)$, but distinguishes three cases depending on the structure of $exp(\alpha)$, each of which being divided into subcases on $coeff(\alpha)$. Note also that $\alpha \leq x$ means that the code of the ordinal α is less than the number x, whereas $\alpha \preceq_e \beta$ says that the ordinal α is less than the ordinal β.

Most of the remainder of this thesis is dedicated to the proof of the wellfoundedness of $\prec_x$ (or of $\prec_d$, for any input bounded parameter d). We conclude this section with setting up the technical groundwork.

Lemma 20. $\vdash \beta \leq x \wedge Ord(\beta) \to \beta = 0 \vee Succ(\beta) \vee Lim(\beta)$.

Proof. Immediate from lemmas 17.2 and 16.5, the definitions of *Succ* and *Lim* and the cases rule. □

Lemma 21. $\vdash \gamma, \alpha, a, b \leq x \to (Ord(\langle \gamma, \alpha, \mathsf{s}a \rangle) \leftrightarrow Ord(\langle \gamma, \alpha, \mathsf{s}b \rangle))$.

Proof. The premises (augmented with $\mathsf{s}a, \mathsf{s}b \leq x$ with the help of corollary 12) according to lemma 16.2 ensure that both $3Seq(\langle \gamma, \alpha, \mathsf{s}a \rangle)$ and $3Seq(\langle \gamma, \alpha, \mathsf{s}b \rangle)$ hold. The claim then is immediate from the defining equations for *Ord* and *Ord*′, observing that the defining term for $Ord'(\langle c, a, \mathsf{s}b \rangle)$ only depends on c and a, but not on b. □

Lemma 22. *The following formulas are provable in* $EA(;)$.

1. $\gamma, \alpha, a \leq x \wedge Ord(\gamma) \to$
   $\gamma + \omega^\alpha \cdot a\downarrow \wedge \gamma \leq \gamma + \omega^\alpha \cdot a \leq \langle \gamma, \alpha, a \rangle \wedge \gamma + \omega^\alpha \cdot \mathsf{s}a \neq 0 \wedge exp(\gamma + \omega^\alpha \cdot \mathsf{s}a) = \alpha$.
2. $\gamma, \alpha, a \leq x \wedge Ord(\gamma, \alpha) \to Ord(\gamma + \omega^\alpha \cdot a)$.
3. $\gamma, \beta \leq x \wedge Ord(\gamma) \to$
   $\exists \delta . \exists b . \gamma + \omega^\beta \cdot 1 = \langle \delta, \beta, \mathsf{s}b \rangle \wedge (b = 0 \to \gamma = \delta) \wedge (b \neq 0 \to \gamma = \langle \delta, \beta, b \rangle)$.

4. $\gamma + \omega^\alpha \cdot 0 = \gamma \wedge (\gamma, \alpha, a \leq x \wedge Ord(\gamma) \rightarrow (\gamma + \omega^\alpha \cdot a) + \omega^\alpha \cdot 1 = \gamma + \omega^\alpha \cdot \mathsf{s}a)$.

Proof. 1. We actually show $a \neq 0 \rightarrow \gamma + \omega^\alpha \cdot a \neq 0 \wedge exp(\gamma + \omega^\alpha \cdot a) = \alpha$ in place of the third and fourth conjuncts, whereas the first and the second remain unchanged. The proof uses the above mentioned case distinctions. The case $a = 0$ is trivial, and the case $\gamma = 0$ immediate from lemma 16.1 and 16.3 (and lemma 4.2 which guarantees $\alpha \leq \alpha$ and $a \leq a$). If $\gamma \neq 0$ we can assume $\gamma = \langle \delta, \beta, b \rangle$ for some $\delta, \beta, b \leq x$ by lemmas 17.2 and 16.5, where $b \neq 0$ by lemma 17.3 and $b \leq b + a$ by lemma 4.3. Then, assuming $b + a \leq x$, which is allowed by corollary 12, the claims hold in both cases, $\chi_=(\beta, \alpha) = 1$ and $\chi_=(\beta, \alpha) = 0$, To verify the inequalities observe that the former case implies $\gamma + \omega^\alpha \cdot a = \langle \delta, \beta, b + a \rangle$, so by lemmas 16.3 and 16.4 we can compute

$$\gamma = \langle \delta, \beta, b \rangle \leq \langle \delta, \beta, b + a \rangle \leq \langle \langle \delta, \beta, b \rangle, \beta, a \rangle = \langle \gamma, \alpha, a \rangle,$$

whereas in the latter $\gamma + \omega^\alpha \cdot a = \langle \gamma, \alpha, a \rangle$, so $\gamma \leq \langle \gamma, \alpha, a \rangle \leq \langle \gamma, \alpha, a \rangle$ holds trivially.

2. By a case distinction similar to the previous item. As $Ord(\gamma + \omega^\alpha \cdot 0) = Ord(\gamma) = 1$, by the cases rule it suffices to show $Ord(\gamma + \omega^\alpha \cdot \mathsf{s}a) = 1$, where $\mathsf{s}a \leq x$. We proceed by case distinction according to lemmas 17.2 and 17.3. If $\gamma = 0$ we have

$$Ord(\gamma + \omega^\alpha \cdot \mathsf{s}a) = Ord(\langle 0, \alpha, \mathsf{s}a \rangle) = Ord(\alpha) = 1.$$

If $\gamma = \langle \delta, \beta, b \rangle$ we have $b \neq 0$ and $\delta, \beta, b \leq x$ as in the previous item, and therefore $\chi_=(\beta, \alpha)\downarrow$. Now if $\chi_=(\beta, \alpha) = 1$ then $\gamma + \omega^\alpha \cdot \mathsf{s}a = \langle \delta, \beta, \mathsf{s}(b + a) \rangle$ which is an ordinal because of lemma 21 (assuming $(b + a) \leq x$ by corollary 12). If $\chi_=(\alpha, \beta) = 0$ finally, $\gamma + \omega^\alpha \cdot \mathsf{s}a = \langle \gamma, \alpha, \mathsf{s}a \rangle$ where $3Seq(\gamma)$ and $(Ord(\gamma)\ \&\ Ord(\alpha)\ \&\ {\sim}\chi_=(exp(\gamma), \alpha)) = 1$ by adequacy of the arithmetization of propositional logic, thus $Ord(\langle \gamma, \alpha, \mathsf{s}a \rangle)$.

3. By case distinction according to lemma 17.2. If $\gamma = 0$ we have $\gamma + \omega^\beta \cdot 1 = \langle 0, \beta, 1 \rangle$, so we choose $\delta = \gamma$ and $b = 0$. If $\gamma = \langle \delta, \alpha, \mathsf{s}a \rangle$ where $\alpha \leq x$ we distinguish two cases. Either $\chi_=(\alpha, \beta) = 1$, which means that $\gamma + \omega^\beta \cdot 1 = \langle \delta, \alpha, \mathsf{ss}a \rangle$, otherwise $\gamma + \omega^\beta \cdot 1 = \langle \gamma, \beta, 1 \rangle$. In both cases the claim holds.

4. The first conjunct is trivial, and the second is proved by cases on a. The case $a = 0$ is immediate from the defining equations. For the successor case we proceed by case distinction on γ according to lemma 17.2, and we notice that $\chi_=(\alpha, \alpha) = 1$. If $\gamma = 0$, then we are done by using the defining equations.

If $\gamma = \langle \delta, \beta, b \rangle$, then $\chi_=(a, \beta)\downarrow$. In the case $\chi_=(\beta, \alpha) = 0$ we can compute

$$(\gamma + \omega^\alpha \cdot \mathsf{s}a) + \omega^\alpha \cdot 1 = \langle \gamma, \alpha, \mathsf{s}a \rangle + \omega^\alpha \cdot 1 = \langle \gamma, \alpha, \mathsf{ss}a \rangle = \gamma + \omega^\alpha \cdot \mathsf{ss}a.$$

On the other hand if $\chi_=(\beta, \alpha) = 1$ we compute

$$(\gamma + \omega^\alpha \cdot \mathsf{s}a) + \omega^\alpha \cdot 1 = \langle \delta, \beta, \mathsf{s}(b + a) \rangle + \omega^\alpha \cdot 1 = \langle \delta, \beta, \mathsf{ss}(b + a) \rangle = \langle \delta, \beta, b \rangle + \omega^\alpha \cdot \mathsf{ss}a.$$

The computations make use of lemma 1, and further require that $\mathsf{s}(b + a)\downarrow$, which is a consequence of the premise $a \leq x$. □

Lemma 23. $\vdash \beta, e \leq x \wedge \mathit{Ord}(\beta) \rightarrow \mathit{pred}(\beta, e)\downarrow \wedge \mathit{Ord}(\mathit{pred}(\beta, e))$.

Proof. Define

$$\mathit{dpt}(0) = \mathsf{1}, \qquad \mathit{dpt}(\langle a, b, c\rangle) = \mathsf{s}\mathit{dpt}(b), \qquad f(a, d) = (a + d)^{4^{\mathit{dpt}(a)}}.$$

We easily get $(a \leq x \wedge \mathit{Ord}(a)) \rightarrow \mathit{dpt}(a)\downarrow$ by term induction (lemma 18). With the help of corollary 12, which applies to *dpt* as we remarked after lemma 18, this entails $(a, d \leq x \wedge \mathit{Ord}(a)) \rightarrow f(a, d)\downarrow$.

Now we are going to prove the lemma by term induction again, this time applied to the formula

$$A[\beta] :\equiv \mathit{pred}(\beta, e)\downarrow \wedge \mathit{Ord}(\mathit{pred}(\beta, e)) \wedge \mathit{pred}(\beta, e) \leq f(\beta, e).$$

The last conjunct of $A[\beta]$ makes the proof a kind of bounded induction, using a technique comparable to the termination proof for the exponential in lemma 13.

If $\beta = 0$ we are done. For the induction step, we assume $\gamma, \alpha, a \leq x$, $\mathit{Ord}(\langle\gamma, \alpha, \mathsf{s}a\rangle)$ and $A[\alpha]$ to show $A[\langle\gamma, \alpha, \mathsf{s}a\rangle]$. Assuming $f(\alpha, e) \leq x$ by corollary 12 the induction hypothesis implies $\alpha \neq 0 \rightarrow \mathit{pred}(\alpha, e) \leq x$, and we have $\mathit{Ord}(\gamma, \alpha)$, and also $a \neq \mathsf{0} \rightarrow \mathit{Ord}(\langle\gamma, \alpha, a\rangle)$, by lemmas 17.2 and 21 respectively.

Using the abbreviation $\beta := \langle\gamma, \alpha, \mathsf{s}a\rangle$ and the additional premise $\langle\gamma, \alpha, \mathsf{s}a\rangle \leq x$ we aim at proving $\mathit{pred}(\beta, e)\downarrow \wedge \mathit{Ord}(\mathit{pred}(\beta, e)) \wedge \mathit{pred}(\beta, e) \leq \langle\beta, \mathit{pred}(\alpha, e), e\rangle$. To do so we distinguish cases on α according to lemma 20, where each case is divided into the subcases $a = \mathsf{0}$ and $a \neq \mathsf{0}$. If $\alpha = 0$ the claim is a trivial consequence of lemma 16.3, and the cases $\mathit{Succ}(\alpha)$ and $\mathit{Lim}(\alpha)$ follow immediately from the induction hypothesis for α and lemmas 22.1 and 22.2.

As to the bound, we need to show $\langle\beta, \mathit{pred}(\alpha, e), e\rangle \leq f(\beta, e)$ which is done, using the abbreviation $c := 4^{\mathit{dpt}(\alpha)}$ (notice that $c \geq 4$), by computing

$$\begin{aligned}\langle\beta, \mathit{pred}(\alpha, e), e\rangle &\leq (\beta + \mathit{pred}(\alpha, e) + \mathsf{s}e)^4 \leq ((\gamma + \alpha + \mathsf{ss}a)^4 + f(\alpha, e) + \mathsf{s}e)^4 \\ &= ((\gamma + \alpha + \mathsf{ss}a)^4 + (\alpha + e)^c + \mathsf{s}e)^4 \leq ((\gamma + \alpha + \mathsf{ss}a)^c + (\alpha + \mathsf{s}e)^c)^4 \\ &\leq (\gamma + \alpha + \mathsf{ss}a + \alpha + \mathsf{s}e)^c)^4 \leq (\langle\gamma, \alpha, \mathsf{s}a\rangle + e)^c)^4 = (\langle\gamma, \alpha, \mathsf{s}a\rangle + e)^{4^{\mathsf{s}\mathit{dpt}(\alpha)}} = f(\beta, e),\end{aligned}$$

where the first and the second inequalities are justified by lemma 16.4 and the induction hypothesis $A[\alpha]$, the third by $(a+b)^c + \mathsf{s}b \leq (a+b)^c + (a+b) + \mathsf{1} \leq ((a+b)+\mathsf{1})^c = (a+\mathsf{s}b)^c$, the fourth by lemma 13 and the fifth by the general principle

$$\begin{aligned}c + a + \mathsf{s}b + a + \mathsf{1} &= c + \mathsf{2} \cdot \mathsf{s}a + b \leq c + \mathsf{2} \cdot \mathsf{s}(c + a) \cdot b + b^2 \\ &\leq (\mathsf{s}(c + a)^2)^2 + \mathsf{2} \cdot \mathsf{s}(c + a)^2 \cdot b + b^2 = (\mathsf{s}(c + a)^2 + b)^2 \leq \langle\langle c, a\rangle, b\rangle = \langle c, a, b\rangle\end{aligned}$$

for $b \neq \mathsf{0}$, here applied to $c := \gamma$, $a := \alpha$ and $b := \mathsf{s}a$. The bound on $\mathit{pred}(\beta, e)$ now follows by transitivity, after adding the premise $f(\beta, e) \leq x$ by using corollary 12, which is also used to provide other bounds needed in the computation. □

Lemma 24.

$$\vdash \beta, e \leq x \wedge \mathit{Ord}(\beta) \wedge \beta \neq 0 \rightarrow \mathit{pred}(\beta, e) \prec_e \beta \wedge (\forall \delta. \delta \prec_e \beta \leftrightarrow \delta \preceq_e \mathit{pred}(\beta, e)).$$

Proof. We have $\mathit{pred}(\beta, e)\downarrow$ by lemma 23, so after adding the premise $\mathit{pred}(\beta, e) \leq x$ by corollary 12 we get $\chi_=(\mathit{pred}(\beta, e), \mathit{pred}(\beta, e)) = \mathsf{1}$. The first conjunct then is immediate from the defining equations for $\prec_e$ and lemma 1.

For the second conjunct, observe $\chi_=(\delta, \mathit{pred}(\beta, e))\downarrow$, so by adequacy of the arithmetization of propositional logic and lemma 1 we can distinguish the four cases $\neg \mathit{Ord}(\beta)$, $\beta = 0$, $\mathit{Ord}(\beta) \wedge \beta \neq 0 \wedge \chi_=(\delta, \mathit{pred}(\beta, e)) = \mathsf{1}$ and $\mathit{Ord}(\beta) \wedge \beta \neq 0 \wedge \chi_=(\delta, \mathit{pred}(\beta, e)) = \mathsf{0}$. Now the first two are excluded by the premises. In the third we have $\delta = \mathit{pred}(\beta, e)$ by adequacy of $\chi_=$, thus $\delta \prec_e \beta$ and $\delta \preceq_e \mathit{pred}(\beta, e)$ are both true. In the last case we have $\delta \neq \mathit{pred}(\beta, e)$ by adequacy of $\chi_=$ again, thus

$$\delta \prec_e \beta \leftrightarrow \delta \prec_e \mathit{pred}(\beta, e) \leftrightarrow \delta \preceq_e \mathit{pred}(\beta, e). \qquad \square$$

Lemma 25.

1. $\vdash \gamma \leq x \wedge \mathit{Ord}(\gamma) \rightarrow \mathit{pred}(\gamma + \omega^0 \cdot \mathsf{1}, e) = \gamma$.
2. $\vdash \mathsf{1} \leq e \leq x \wedge \beta, \gamma \leq x \wedge \mathit{Ord}(\beta, \gamma) \wedge \beta \neq 0 \rightarrow$
   $\exists \alpha \prec_e \beta. \exists a \leq e. \mathit{Ord}(\alpha) \wedge \alpha = \mathit{pred}(\beta, e) \wedge \mathit{pred}(\gamma + \omega^\beta \cdot 1, e) = \gamma + \omega^\alpha \cdot a$.

Proof. 1. This is immediate from lemma 22.3 and the defining equations for *pred*.

2. By lemma 22.3 again we have that $\gamma + \omega^\beta \cdot 1 = \langle \delta, \beta, \mathsf{s}b \rangle$ for some δ and b, where γ equals δ or $\langle \delta, \beta, b \rangle$ depending on whether b is $\mathsf{0}$ or a successor respectively. We distinguish cases according to lemma 20. In the first case $\beta = 0$ there is nothing to do. In the second case, when $\mathit{Succ}(\beta)$, we distinguish cases on b. If $b = \mathsf{0}$, then $\mathit{pred}(\gamma + \omega^\beta \cdot 1, e) = \delta + \omega^{\mathit{pred}(\beta, e)} \cdot e$, otherwise $\mathit{pred}(\gamma + \omega^\beta \cdot 1, e) = \langle \delta, \beta, b \rangle + \omega^{\mathit{pred}(\beta, e)} \cdot e$. Hence in both cases $\mathit{pred}(\gamma + \omega^\beta \cdot 1, e) = \gamma + \omega^{\mathit{pred}(\beta, e)} \cdot e$. Now lemmas 23 and 24 say that $\mathit{pred}(\beta, e)\downarrow \wedge \mathit{pred}(\beta, e) \prec_e \beta \wedge \mathit{Ord}(\mathit{pred}(\beta, e))$, and $e \leq e$ follows from lemma 4.2, so the claim follows by existential quantification. The last case where β is a limit is very similar, but using $\mathsf{1} \leq e$ given in the premise in place of $e \leq e$. $\square$

Chapter 5

Transfinite Induction, Lower Bounds

5.1 Bounding Functions

In this purely technical section we define the function $h(\alpha, d)$ and an auxiliary formula G which both will be used for providing the bounds on outputs needed in the main proof and establish their relevant properties. They are explained more in detail below.

Definition 26.

1. $f_0(0) = \langle 0, 0, \mathsf{1}\rangle = 6$,
   $f_0(\langle\gamma, \alpha, a\rangle) = \langle\langle\gamma, \alpha, a\rangle, f_0(\alpha), \mathsf{1}\rangle$,
   $f(a) = \max\{f_0(\beta) : Ord(\beta)\ \&\ \beta \leq a\}$,
   $g(0, e) = \mathsf{0}$,
   $g(\langle\gamma, \alpha, a\rangle, e) = g(\gamma, e) + (\mathsf{s}e)^{g(\alpha,e)} \cdot a$,
   $\tilde{h}(\mathsf{0}) = \mathsf{0}$,
   $\tilde{h}(\mathsf{s}a) = \mathsf{s}f(\tilde{h}(a))$,
   $h_0(a, b, c, d) = \tilde{h}(a + (\mathsf{s}b)^c \cdot d)$,
   $h(\alpha, e) = \tilde{h}(g(\alpha, e))$.
2. $\mathrm{DptBd}(\alpha, e, x) :\equiv \forall\delta \preceq_e \alpha.Ord(\delta) \wedge h(\delta, e)\!\downarrow \wedge\, \delta, g(\delta, e) \leq h(\delta, e) \leq x$.
3. $\mathrm{Mon}(\alpha, e) :\equiv \forall\delta.\forall\varepsilon.\varepsilon \prec_e \delta \preceq_e \alpha \to g(\varepsilon, e)\!\downarrow \wedge\, g(\delta, e)\!\downarrow \wedge\, g(\varepsilon, e) < g(\delta, e) \leq g(\alpha, e)$.
4. $\mathrm{Tran}(\alpha, e) :\equiv \forall\gamma.\forall\delta.\forall\varepsilon.\varepsilon \prec_e \delta \prec_e \gamma \preceq_e \alpha \to \varepsilon \prec_e \gamma$.
5. $G[\alpha, e, x] :\equiv \mathrm{Tran}(\alpha, e) \wedge \mathrm{Mon}(\alpha, e) \wedge \mathrm{DptBd}(\alpha, e, x)$.

The main purpose of $h(\alpha, d)$ is to provide an upper bound for the values of (the codes of) all ordinals below α with respect to $\prec_d$. This is achieved by brute force: f_0 is constructed such that $\beta \leq f_0(pred(\beta, d))$ as shown in lemma 27.4 below, $g(\alpha, d)$ extensionally equals

the slow-growing hierarchy at level α, and $h(\alpha, d)$ basically iterates f_0 $g(\alpha, d)$-many times. h_0 provides an alternative characterization for h which is easier to handle in some places. As g and h are non-elementary functions, we won't be able to prove their totality for arbitrary input, but we can show definedness for all relevant ordinals simultaneously with the wellordering proof.

$G[\alpha, e, x]$ in particular entails a monotonicity property for g and transitivity of $\prec_e$. As $\prec_e$ is a transitive closure we can't prove the latter outright either, so we have to establish it simultaneously to the wellfoundedness property as well.

Lemma 27. *The following are theorems of EA(;).*

1. $a, b, c, d \leq x \to h_0(a, b, c, d)\downarrow \wedge\ a + (\mathsf{s}b)^c \cdot d \leq h_0(a, b, c, d) \leq h_0(a, b, c, \mathsf{s}d)$
   $\wedge (c' < c \wedge d \leq b \to h_0(a, b, c', d) \leq h_0(a, b, c, 1) \wedge f(h_0(a, b, c', d)) \leq h_0(a, b, c, 1))$.
2. $a, e, \gamma, \alpha, g(\gamma, e), g(\alpha, e) \leq x \wedge g(\gamma, e)\downarrow \wedge g(\alpha, e)\downarrow$
   $\to g(\gamma + \omega^\alpha \cdot a, e)\downarrow \wedge\ h(\gamma + \omega^\alpha \cdot a, e)\downarrow \wedge\ h(\gamma + \omega^\alpha \cdot a, e) = h_0(g(\gamma, e), e, g(\alpha, e), a)$.
3. $b \leq x \to f(b)\downarrow \wedge\ b \leq f(b) \wedge\ (a \leq b \to f(a) \leq f(b))$.
4. $\gamma, \beta \leq x \wedge \mathit{Ord}(\gamma, \beta) \to \gamma + \omega^\beta \cdot 1 \leq f(\mathit{pred}(\gamma + \omega^\beta \cdot 1, e))$.

Proof. We are first going to show

(1) $a, b, c, d \leq x \to a + (\mathsf{s}b)^c \cdot d\downarrow \wedge\ a + (\mathsf{s}b)^c \cdot d \leq a + (\mathsf{s}b)^c \cdot \mathsf{s}d$
$\wedge (c' < c \wedge d \leq b \to a + (\mathsf{s}b)^{c'} \cdot d < a + (\mathsf{s}b)^c \cdot 1)$,

(2) $a \leq x \to f(a)\downarrow \wedge\ (\mathit{Ord}(a) \to f_0(a)\downarrow \wedge\ f_0(a) \leq f(a)) \wedge \forall b \leq a.b \leq f(b) \leq f(a)$,

(3) $a \leq x \to \tilde{h}(a)\downarrow \wedge\ a \leq \tilde{h}(a) \wedge \forall b \leq a.\tilde{h}(b) \leq \tilde{h}(a)$,

(4) $a < b \leq x \to f(\tilde{h}(a)) < \tilde{h}(b)$,

(5) $\gamma, \alpha, a, e \leq x \wedge g(\gamma, e), g(\alpha, a) \leq x \wedge g(\gamma, e)\downarrow \wedge g(\alpha, a)\downarrow$
$\to g(\gamma + \omega^\alpha \cdot a, e) = g(\gamma, e) + (\mathsf{s}e)^{g(\alpha, e)} \cdot a$.

In the first claim, $a + (\mathsf{s}b)^c \cdot d\downarrow$ and $a + (\mathsf{s}b)^c \cdot d \leq a + (\mathsf{s}b)^c \cdot \mathsf{s}d$ are immediate from lemma 13, and the last conjunct is easily verified by

$$a + (\mathsf{s}b)^{c'} \cdot d < a + (\mathsf{s}b)^{c'} \cdot \mathsf{s}b = a + (\mathsf{s}b)^{\mathsf{s}c'} \cdot 1 \leq a + (\mathsf{s}b)^c \cdot 1.$$

For the second claim we first show, by combining the technique of function bounded induction with induction on ordinal terms (lemma 18),

$(*)$ $\qquad a \leq x \wedge \mathit{Ord}(a) \to f_0(a)\downarrow \wedge\ a \leq f_0(a) \wedge\ (a \neq 0 \to f_0(a) \leq a^8)$.

More explicitly, we apply lemma 18 to the formula

$$A[a] :\equiv f_0(a)\downarrow \wedge\ a \leq f_0(a) \wedge\ (a \neq 0 \to f_0(a) \leq a^8).$$

$A[\mathsf{0}]$ is immediate. For the induction step we show $A[\langle c,a,b\rangle]$ for $b \neq \mathsf{0}$, assuming $c,a,b \leq x$ and $A[a]$, and additionally $\langle c,a,b\rangle \leq x$ and $a^8 \leq x$ by using corollary 12. Then $f_0(a) \leq x$ by the induction hypothesis and transitivity (except when $a = 0$ in which case adding the premise $6 \leq x$ suffices, as $f_0(0) = \langle 0,0,\mathsf{1}\rangle = 6$). First, $f_0(\langle c,a,b\rangle)$ equals $\langle\langle c,a,b\rangle, f_0(a), \mathsf{1}\rangle$ which is defined under the assumed bounds, and satisfies $\langle c,a,b\rangle \leq \langle\langle c,a,b\rangle, f_0(a), \mathsf{1}\rangle$ by lemma 16.3.

The upper bound on $f_0(\langle c,a,b\rangle)$ is simple for the case $a = 0$ because $6 = \langle 0,0,\mathsf{1}\rangle \leq \langle c,0,b\rangle$ (recall that $b \neq \mathsf{0}$) implies

$$f_0(\langle c,0,b\rangle) = \langle\langle c,0,b\rangle, 6, \mathsf{1}\rangle \leq (\langle c,0,b\rangle + 6 + 2)^4 \leq \langle c,0,b\rangle^8,$$

where the first inequality is given by lemma 16.4, and the second by the computation $(d+8)^4 \leq (3 \cdot d^4 \leq (d \cdot d)^4 = d^8$ which is valid for all $d \geq 6$ (in our case $d = \langle c,0,b\rangle$).

If $a \neq 0$ we can use the induction hypothesis and compute

$$\begin{aligned} f_0(\langle c,a,b\rangle) &= \langle\langle c,a,b\rangle, f_0(a), \mathsf{1}\rangle \leq \langle\langle c,a,b\rangle, a^8, \mathsf{1}\rangle \leq ((c+a+\mathsf{s}b)^4 + a^8 + 2)^4 \\ &= (\mathsf{ss}(c+a+\mathsf{s}b)^4 + a^8)^4 \leq (\mathsf{s}(\mathsf{s}(c+a)^2 + b)^4)^4 \leq (\mathsf{s}(\mathsf{s}(c+a)^2 + b)^2)^8 \leq (\langle\langle c,a\rangle, b\rangle)^8. \end{aligned}$$

The steps are justified as follows. The first inequality is due to the induction hypothesis and lemma 16.3, the second to lemma 16.4. The third is proven below. In the fourth we recall that for all $d \leq x$ (in our case $d = \mathsf{s}(c+a)^2 + 2$, with the bound brought in by corollary 12) $d < \mathsf{s}d$, so $d^2 < (\mathsf{s}d)^2$, i.e. $\mathsf{s}d^2 \leq (\mathsf{s}d)^2$ and $(\mathsf{s}d^4)^4 \leq ((\mathsf{s}d^2)^2)^4 = (\mathsf{s}d^2)^8$. For the fifth we use lemma 16.4 twice.

Back to the third inequality, if $c \neq 0$ notice that $\mathsf{s}a \leq 2 \cdot c \cdot a$ (as we still are in the case $a \neq 0$), so we can use lemma 13 to compute

$$\mathsf{s}(c+a+\mathsf{s}b)^4 + a^8 \leq (c + \mathsf{s}a + \mathsf{s}b + a^2)^4 \leq (c^2 + 2 \cdot c \cdot a + a^2 + \mathsf{s}b)^4 = (\mathsf{s}(c+a)^2 + b)^4.$$

If $c = 0$ on the other hand we simply compute that $\langle\langle 0,a,b\rangle, a^8, \mathsf{1}\rangle - (\langle\langle 0,a\rangle, b\rangle)^8$ expands to a (tremendously long) positive polynomial. This is checked preferably by using a computer algebra system, but doesn't involve any high-level principles.

This completes the induction step and therefore the proof of $(*)$. Returning back to the proof of (2) we show $a \leq x \to f(a)\!\downarrow \wedge\, a \leq f(a) \wedge (a \geq 2 \to f(a) \leq a^8)$ as follows. Let

$$f_1(a) = \begin{cases} f_0(a) & \text{if } \mathit{Ord}(a), \\ a & \text{else,} \end{cases} \qquad \text{and} \qquad f_2(a) = \max_{b \leq a}(f_1(b)).$$

Then $f_2(a) = f(a)$, $a \leq f_1(a)$, and by lemma 14 $f(a)$ is defined, $f_1(a) \leq f(a)$, and $f(a) = f_1(c)$ for some $c \leq a$. Now, if $\mathit{Ord}(c)$ we have $f_1(c) = f_0(c) \leq c^8 \leq a^8$ (unless $c = \mathsf{0}$, in which case $f_1(c) = f_0(0) = 6 \leq a^8$ for $a \geq 2$), otherwise $f_1(c) = c \leq a \leq a^8$.

Then $(a \leq x \wedge \mathit{Ord}(a)) \to f_0(a) \leq f(a)$ is immediate from $\mathit{Ord}(a) \to f_1(a) = f_0(a)$, and $a \leq x \to \forall b \leq a.f(b) \leq f(a)$ from lemma 14.

The first two conjuncts of the third claim are proven by applying lemma 3 to the formula $A[a] :\equiv \tilde{h}(a)\!\downarrow \wedge\, a \leq \tilde{h}(a) \leq 2^{9^a}$. $A[\mathsf{0}]$ is immediate, and for the induction step we may assume $2^{9^a} \leq x$ by corollary 12. The induction hypothesis and transitivity then give $\tilde{h}(a) \leq x$ and therefore $\tilde{h}(\mathsf{s}a) = \mathsf{s}f(\tilde{h}(a))$ which is defined, and the lower bound follows from (3) and the last conjunct of (2) due to $\mathsf{s}a \leq \mathsf{s}\tilde{h}(a) \leq \mathsf{s}f(\tilde{h}(a)) = \tilde{h}(\mathsf{s}a)$. For the upper bound, if $\tilde{h}(a) \leq \mathsf{1}$, then $\tilde{h}(\mathsf{s}a) = \mathsf{s}f(\tilde{h}(a)) \leq \mathsf{s}f(\mathsf{1}) = \mathsf{s}f_0(0) = \mathsf{s}\mathsf{6} \leq 2^9 \leq 2^{9^{\mathsf{s}a}}$ (where the second equality holds by definition of f, because $\mathsf{0}$ is code of an ordinal whereas $\mathsf{1}$ isn't), otherwise

$$\tilde{h}(\mathsf{s}a) = \mathsf{s}f(\tilde{h}(a)) \leq \mathsf{s}f(2^{9^a}) \leq \mathsf{s}(2^{9^a})^8 = \mathsf{s}(2^{8\cdot 9^a}) \leq 2 \cdot 2^{8\cdot 9^a} = 2^{\mathsf{s}(8\cdot 9^a)} \leq 2^{8\cdot 9^a + 9^a} = 2^{9^{\mathsf{s}a}}.$$

Here the first inequality holds by (2) and the induction hypothesis, and the second because we have seen above $a \geq 2 \to f(a) \leq a^8$.

For the third conjunct apply corollary 7 to $A[a] :\equiv \tilde{h}(b) \leq \tilde{h}(a)$. Under $b \leq x$ and the additional assumption $\tilde{h}(b) \leq x$ which is allowed by corollary 12 we immediately get $A[b]$, this settles the base case. For the induction step we add the assumption $\tilde{h}(a) \leq x$. This entails $\tilde{h}(a) \leq f(\tilde{h}(a)) < \mathsf{s}f(\tilde{h}(a)) = \tilde{h}(\mathsf{s}a)$, so we are done by using transitivity and the induction hypothesis.

To prove (4) we observe that $a < b$ implies $\mathsf{s}a \leq b$ by definition, so the previous claim immediately gives $\mathsf{s}f(\tilde{h}(a)) = \tilde{h}(\mathsf{s}a) \leq \tilde{h}(b)$, and the last claim can finally be verified by easy computations, distinguishing cases according to the several defining equations for $\gamma + \omega^\alpha \cdot a$. Notice that we can't remove the premises $g(\gamma, e) \leq x$ and $g(\alpha, e) \leq x$ by using corollary 12 because definedness of both terms is not provable outright.

Now we are ready to prove the lemma.

1. The first item is immediate from (1), (3), (4), and the definition of h_0, using corollary 12 to add the premise $a + (\mathsf{s}b)^c \cdot d \leq x$.

2. $g(\gamma + \omega^\alpha \cdot a, e)\!\downarrow$ follows directly from (5) and (1). We verify the third conjunct by computing

   $$h(\gamma+\omega^\alpha\cdot a, e) = \tilde{h}(g(\gamma+\omega^\alpha\cdot a, e)) = \tilde{h}(g(\gamma,e)+(\mathsf{s}e)^{g(\alpha,e)}\cdot a) = h_0(g(\gamma,e), e, g(\alpha,e), a)$$

   using (5) and the fact that $\gamma + \omega^\alpha \cdot a$ is defined by lemma 22.1, and definedness of $h(\gamma + \omega^\alpha \cdot a, e)$ now follows directly from the first item of this lemma.

3. This has already been done in (2).

4. We first prove

   $$\beta \leq x \wedge \mathit{Ord}(\beta) \to (\beta \neq 0 \to \beta \leq f_0(\mathit{pred}(\beta, e))) \tag{6}$$

   by induction on the ordinal term β (lemma 18). The base case is trivial. For the induction step we have to prove the claim for $\beta = \langle \gamma, \alpha, \mathsf{s}a\rangle$, where $\gamma, \alpha, a \leq x$ and $\mathit{Ord}(\gamma, \alpha)$ by lemma 17.2, assuming that it holds for α.

If $\alpha = 0$ then we use cases on a, dividing the case $a = \mathsf{0}$ further into two subcases $\gamma = 0$ and $\mathit{3Seq}(\gamma)$ according to lemma 17.2. The first subcase is trivial as $f_0(\mathit{pred}(\langle 0,0,\mathsf{1}\rangle, e) = f_0(0) = \langle 0,0,\mathsf{1}\rangle = \beta$. In the other we notice that $f_0(\mathit{exp}(\gamma))$ by (2) is defined, so we may assume $0 \leq f_0(\mathit{exp}(\gamma)) \leq x$ by using lemma 4.2 and corollary 12, and by lemma 16 we can compute

$$\beta = \langle \gamma, 0, \mathsf{1}\rangle \leq \langle \gamma, f_0(\mathit{exp}(\gamma)), \mathsf{1}\rangle = f_0(\gamma) = f_0(\mathit{pred}(\beta, e)).$$

The case of a successor $\mathsf{s}a$ (while still $\alpha = 0$) follows, using lemma 16 again and transitivity, from

$$\beta = \langle \gamma, 0, \mathsf{ss}a\rangle \leq \langle\langle \gamma, 0, \mathsf{s}a\rangle, 0, \mathsf{1}\rangle \leq \langle\langle \gamma, 0, \mathsf{s}a\rangle, f_0(0), \mathsf{1}\rangle = f_0(\langle \gamma, 0, \mathsf{s}a\rangle) = f_0(\mathit{pred}(\beta, e)).$$

As to the case $\alpha \neq 0$ of the induction step, we have that $\mathit{pred}(\alpha, e)$ is defined and (a code of) an ordinal by lemma 23. So we can apply corollary 12 first to assume $\mathit{pred}(\alpha, e) \leq x$, which allows us to obtain $f_0(\mathit{pred}(\alpha, e))\downarrow$ from (2), then once more to add the premise $f_0(\mathit{pred}(\alpha, e)) \leq x$.

Then we check that $\beta \leq \langle \mathit{pred}(\beta, e), \alpha, \mathsf{1}\rangle$ holds in all cases left by lemma 20. Namely, if $\mathit{Succ}(\alpha)$ then $a = \mathsf{0}$ implies $\beta = \langle \gamma, \alpha, \mathsf{1}\rangle \leq \langle \gamma + \omega^{\mathit{pred}(\alpha,e)} \cdot e, \alpha, \mathsf{1}\rangle = \langle \mathit{pred}(\beta, e), \alpha, \mathsf{1}\rangle$ by lemmas 22.1 and 16.3, and in the successor case

$$\beta = \langle \gamma, \alpha, \mathsf{ss}a\rangle \leq \langle\langle \gamma, \alpha, \mathsf{s}a\rangle, \alpha, \mathsf{1}\rangle \leq \langle\langle \gamma, \alpha, \mathsf{s}a\rangle + \omega^{\mathit{pred}(\alpha,e)} \cdot e, \alpha, \mathsf{1}\rangle = \langle \mathit{pred}(\beta, e), \alpha, \mathsf{1}\rangle$$

(where the first inequality follows from lemma 16.4 when substituting $\mathsf{1}$ for c'). The case when $\mathit{Lim}(\alpha)$ holds is analogous. Furthermore, all of these cases satisfy $\mathit{exp}(\mathit{pred}(\beta, e)) = \mathit{pred}(\alpha, e)$ by lemma 22.1. All this together with the induction hypothesis justifies the computation

$$\beta \leq \langle \mathit{pred}(\beta, e), \alpha, \mathsf{1}\rangle \leq \langle \mathit{pred}(\beta, e), f_0(\mathit{pred}(\alpha, e)), \mathsf{1}\rangle = f_0(\mathit{pred}(\beta, e)).$$

After adding the premise $\mathit{pred}(\beta, e) \leq x$ by corollary 12 we can use transitivity to complete the induction step and the proof of (6).

The next step is to show $\beta \leq x \wedge \mathit{Ord}(\beta) \rightarrow (\beta \neq 0 \rightarrow \beta \leq f(\mathit{pred}(\beta, e)))$. Since $\mathit{pred}(\beta, e)$ is defined and an ordinal by lemma 23 we can additionally assume that it is bounded by x by using corollary 12. We then apply (2) to obtain $f_0(\mathit{pred}(\beta, e)) \leq f(\mathit{pred}(\beta, e))$ and $f(\mathit{pred}(\beta, e))\downarrow$. Now the claim is immediate from (6) and transitivity, as we can add the premise $f(\mathit{pred}(\beta, e)) \leq x$ by corollary 12 again.

Item 4 of the lemma itself now follows by another application of corollary 12: The premises according to lemma 22.1 guarantee $\gamma + \omega^\beta \cdot 1\downarrow$ and $\mathit{Ord}(\gamma + \omega^\beta \cdot 1)$, and by the above we are done when adding the premise $\gamma + \omega^\beta \cdot 1 \leq x$. □

Lemma 28.

1. $\vdash G[\alpha, e, x] \to \mathit{Ord}(\alpha) \wedge g(\alpha, e)\downarrow \wedge h(\alpha, e)\downarrow \wedge \alpha, g(\alpha, e) \leq x \wedge \alpha \leq h(\alpha, e) \leq x$.
2. $\vdash G[0, e, x]$.
3. $\vdash \alpha \prec_e \beta \to (G[\beta, e, x] \to G[\alpha, e, x])$.
4. $\vdash \mathsf{1} \leq e \leq x \wedge h(\gamma + \omega^\beta \cdot 1, e) \leq x \wedge \mathit{Ord}(\gamma, \beta) \to$
   $(G[\gamma, e, x] \wedge G[\beta, e, x] \wedge G[\mathit{pred}(\gamma + \omega^\beta \cdot 1, e), e, x] \to G[\gamma + \omega^\beta \cdot 1, e, x])$.

Proof. 1. This is immediate from the definition of $G[\alpha, e, x]$.

2. This item is trivial as well (notice that $\delta \preceq_e 0$ implies $\delta = 0$, as $\prec_e(\delta, 0) = \mathsf{0}$, and that $h(0, e) = \mathsf{0} \leq x$).

3. In order to prove $\mathrm{Tran}(\alpha, e)$ assume $\varepsilon \prec_e \delta \prec_e \gamma \preceq_e \alpha$ and we have to show $\varepsilon \prec_e \gamma$. If $\gamma = \alpha$ we trivially have $\varepsilon \prec_e \delta \prec_e \alpha \preceq_e \beta$ and the claim follows from $\mathrm{Tran}(\beta, e)$. Otherwise we apply $\mathrm{Tran}(\beta, e)$ first to $\gamma \prec_e \alpha \prec_e \beta \preceq_e \beta$ to obtain $\gamma \prec_e \beta$, then to $\varepsilon \prec_e \delta \prec_e \gamma \preceq_e \beta$ which yields $\varepsilon \prec_e \gamma$ as desired.

   For $\mathrm{DptBd}(\alpha, e, x)$ assume $\delta \preceq_e \alpha$. In order to exploit the premise $G[\beta, e, x]$ we first show $\delta \prec_e \beta$. This holds trivially if $\delta = \alpha$. If not, we have $\delta \prec_e \alpha \prec_e \beta \preceq_e \beta$, so the claim follows from $\mathrm{Tran}(\beta, e)$. Now $\mathit{Ord}(\delta)$, $h(\delta, e)\downarrow$ and $\delta, g(\delta, e) \leq h(\delta, e) \leq x$ are immediate from $\mathrm{DptBd}(\beta, e, x)$.

   All that remains is to show $\mathrm{Mon}(\alpha, e)$. To do so, assume $\varepsilon \prec_e \delta \preceq_e \alpha$. If $\delta = \alpha$ we have $\varepsilon \prec_e \delta = \alpha \preceq_e \beta$, so the claim follows from $\mathrm{Mon}(\beta, e)$, as soon as we have established $g(\delta, e) \leq g(\delta, e)$ which we do as follows: Since $\delta \preceq_e \beta$, from $\mathrm{Mon}(\beta, e)$ we get $g(\delta, e)\downarrow$, so by $\mathrm{DptBd}(\beta, e, x)$ and transitivity $g(\delta, e) \leq x$, and we are done by using lemma 4.2.

   If $\delta \prec_e \alpha$ we have $\delta \prec_e \alpha \prec_e \beta \preceq_e \beta$, so from $\mathrm{Tran}(\beta, e)$ we get $\varepsilon \prec_e \delta \prec_e \beta$, and we get $g(\varepsilon, e)\downarrow \wedge g(\delta, e)\downarrow \wedge g(\varepsilon, e) < g(\delta, e)$ by instantiating the universal quantifiers in $\mathrm{Mon}(\beta, e)$ to ε and δ. On the other hand, instantiating the universal quantifiers in $\mathrm{Mon}(\beta, e)$ to δ and α gives $g(\delta, e) < g(\alpha, e)$, so $g(\delta, e) \leq g(\alpha, e)$ by lemma 4.2, transitivity, and lemma 2.3.

4. The premises $G[\gamma, e, x]$ and $G[\beta, e, x]$ by the first item of this lemma entail $g(\gamma, e)\downarrow$, $g(\beta, e)\downarrow$ and $\gamma, \beta, g(\gamma, e), g(\beta, e) \leq x$, so we have $\gamma + \omega^\beta \cdot 1\downarrow$, $\gamma + \omega^\beta \cdot 1 \neq 0$, and $\mathit{Ord}(\gamma + \omega^\beta \cdot 1)$ by lemma 22, and $g(\gamma + \omega^\beta \cdot 1, e)\downarrow$ and $h(\gamma + \omega^\beta \cdot 1, e)\downarrow$ by lemma 27.2.

   Furthermore, if $\beta \neq 0$, then by lemma 25

   $$\mathit{pred}(\gamma + \omega^\beta \cdot 1, e) = \gamma + \omega^\alpha \cdot a$$

   for some $\alpha \prec_e \beta$ satisfying $\mathit{Ord}(\alpha)$ and some $a \leq e$, where $G[\alpha, e, x]$, $g(\alpha, e)\downarrow$ and $\alpha, g(\alpha, e) \leq x$ by items 3 and 1 of this lemma. This implies $\mathit{pred}(\gamma + \omega^\beta \cdot 1, e)\downarrow$ and

$Ord(pred(\gamma + \omega^\beta \cdot 1, e))$ by lemma 22 again, and also $g(pred(\gamma + \omega^\beta \cdot 1, e), e)\downarrow$ and $h(pred(\gamma + \omega^\beta \cdot 1, e), e)\downarrow$ by lemma 27.2. Now the premise $G[pred(\gamma + \omega^\beta \cdot 1, e), e, x]$ by lemma 28.1 entails $pred(\gamma + \omega^\beta \cdot 1, e) \leq h(pred(\gamma + \omega^\beta \cdot 1, e), e) \leq x$, and from $\mathrm{Mon}(\beta, e)$ we get $g(\alpha, e) < g(\beta, e)$. This allows us, using all items of lemma 27, to compute

$$\begin{aligned}\gamma + \omega^\beta \cdot 1 &\leq f(pred(\gamma + \omega^\beta \cdot 1, e)) \leq f(h(pred(\gamma + \omega^\beta \cdot 1, e), e)) = f(h(\gamma + \omega^\alpha \cdot a, e)) \\ &= f(h_0(g(\gamma, e), e, g(\alpha, e), a)) \leq h_0(g(\gamma, e), e, g(\beta, e), 1) = h(\gamma + \omega^\beta \cdot 1, e).\end{aligned}$$

This computation, together with the premise of the lemma, by transitivity also yields $\gamma + \omega^\beta \cdot 1 \leq x$, and, since $h(pred(\gamma + \omega^\beta \cdot 1, e), e) \leq f(h(pred(\gamma + \omega^\beta \cdot 1, e), e))$, also

$$h(pred(\gamma + \omega^\beta \cdot 1, e), e) \leq h(\gamma + \omega^\beta \cdot 1, e).$$

To complete our preparatory computations, first

$$g(pred(\gamma + \omega^\beta \cdot 1, e), e) = g(\gamma, e) + (\mathsf{s}e)^{g(\alpha,e)} \cdot a < g(\gamma, e) + (\mathsf{s}e)^{g(\beta,e)} \cdot 1 = g(\gamma + \omega^\beta \cdot 1, e)$$

and second, using (5), the second conjunct of lemma 27.1, and the final conjunct of lemma 27.2,

$$g(\gamma + \omega^\beta \cdot 1, e) = g(\gamma, e) + (\mathsf{s}e)^{g(\beta,e)} \cdot 1 \leq h_0(g(\gamma, e), e, g(\beta, e), 1) = h(\gamma + \omega^\beta \cdot 1, e),$$

which also entail $g(\gamma + \omega^\beta \cdot 1, e) \leq x$ and $g(pred(\gamma + \omega^\beta \cdot 1, e), e) \leq x$.

If $\beta = 0$ on the other hand, then $pred(\gamma + \omega^\beta \cdot 1, e) = \gamma$, and the same results hold: $\gamma\downarrow \wedge Ord(\gamma)$ are trivial, $g(\gamma, e)\downarrow$, $h(\gamma, e)\downarrow$ and $\gamma \leq h(\gamma, e) \leq x$ are immediate from the first item of this lemma, and we compute

$$\begin{aligned}\gamma + \omega^\beta \cdot 1 &\leq f(pred(\gamma + \omega^\beta \cdot 1, e)) = f(\gamma) \leq f(h(\gamma, e)) \leq \mathsf{s}f(h(\gamma, e)) \\ &= \mathsf{s}f(\tilde{h}(g(\gamma, e))) = \tilde{h}(\mathsf{s}g(\gamma, e)) = \tilde{h}(g(\gamma, e) + (\mathsf{s}e)^0 \cdot 1) = h(\gamma + \omega^\beta \cdot 1, e)\end{aligned}$$

and

$$g(\gamma, e) < \mathsf{s}g(\gamma, e) = g(\gamma, e) + (\mathsf{s}e)^0 \cdot 1 = g(\gamma + \omega^\beta \cdot 1, e).$$

This completes our preparations, we are going to proof item 4. We have to show $G[\gamma + \omega^\beta \cdot 1, e, x]$ which we do separately for each of its three conjuncts.

In order to prove $\mathrm{Tran}(\gamma + \omega^\beta \cdot 1, e)$, assume $\varepsilon \prec_e \delta \prec_e \gamma' \preceq_e \gamma + \omega^\beta \cdot 1$. In the case $\gamma' \prec_e \gamma + \omega^\beta \cdot 1$ we have $\gamma' \preceq_e pred(\gamma + \omega^\beta \cdot 1, e)$ by lemma 24 (this needs $\gamma + \omega^\beta \cdot 1 \leq x$, but not $\gamma' \leq x$ which we are unable to show at this stage), thus $\varepsilon \prec_e \gamma'$ by $\mathrm{Tran}(pred(\gamma + \omega^\beta \cdot 1, e), e)$. In the other case, when $\gamma' = \gamma + \omega^\beta \cdot 1$, we have $\delta \preceq_e pred(\gamma + \omega^\beta \cdot 1, e)$ by lemma 24 again. Now in the subcase $\delta \prec_e pred(\gamma + \omega^\beta \cdot 1, e)$ we have $\varepsilon \prec_e \delta \prec_e pred(\gamma + \omega^\beta \cdot 1, e) \preceq_e pred(\gamma + \omega^\beta \cdot 1, e)$, thus $\varepsilon \prec_e pred(\gamma + \omega^\beta \cdot 1, e)$ by $\mathrm{Tran}(pred(\gamma + \omega^\beta \cdot 1, e), e)$ and $\varepsilon \prec_e \gamma + \omega^\beta \cdot 1 = \gamma'$ by lemma 24. Finally, if

$\delta = \mathit{pred}(\gamma + \omega^\beta \cdot 1, e)$ we trivially have $\varepsilon \prec_e \mathit{pred}(\gamma + \omega^\beta \cdot 1, e)$, and we continue as in the previous subcase.

For $\mathrm{DptBd}(\gamma + \omega^\beta \cdot 1, e, x)$ we assume $\delta \preceq_e \gamma + \omega^\beta \cdot 1$. If $\delta = \gamma + \omega^\beta \cdot 1$ we have already shown everything we have to. In the case $\delta \prec_e \gamma + \omega^\beta \cdot 1$ we have $\delta \preceq_e \mathit{pred}(\gamma + \omega^\beta \cdot 1, e)$ by lemma 24, so we immediately get $\mathit{Ord}(\delta)$, $h(\delta, e)\!\downarrow$ and $\delta, g(\delta, e) \leq h(\delta, e) \leq x$ from $\mathrm{DptBd}(\mathit{pred}(\gamma + \omega^\beta \cdot 1, e), e, x)$.

For $\mathrm{Mon}(\gamma + \omega^\beta \cdot 1, e)$ we assume $\varepsilon \prec_e \delta \preceq_e \gamma + \omega^\beta \cdot 1$. We first settle the case $\delta \prec_e \gamma + \omega^\beta \cdot 1$. As we have $\delta \preceq_e \mathit{pred}(\gamma + \omega^\beta \cdot 1, e)$ by lemma 24, $g(\varepsilon, e)\!\downarrow$, $g(\delta, e)\!\downarrow$ and $g(\varepsilon, e) < g(\delta, e)$ follow directly from $\mathrm{Mon}(\mathit{pred}(\gamma + \omega^\beta \cdot 1, e), e)$. So does $g(\delta, e) \leq g(\mathit{pred}(\gamma + \omega^\beta \cdot 1, e), e)$ which together with the above implies $g(\delta, e) \leq g(\gamma + \omega^\beta \cdot 1, e)$ by transitivity. In the other case, when $\delta = \gamma + \omega^\beta \cdot 1$, we have $\varepsilon \preceq_e \mathit{pred}(\gamma + \omega^\beta \cdot 1, e)$ by lemma 24 once more. We have already shown everything we need for the case $\varepsilon = \mathit{pred}(\gamma + \omega^\beta \cdot 1, e)$, so assume $\varepsilon \prec_e \mathit{pred}(\gamma + \omega^\beta \cdot 1, e)$. Then we have $\varepsilon \prec_e \mathit{pred}(\gamma + \omega^\beta \cdot 1, e) \preceq_e \mathit{pred}(\gamma + \omega^\beta \cdot 1, e)$ and we are done by using $\mathrm{Mon}(\mathit{pred}(\gamma + \omega^\beta \cdot 1, e), e)$ and transitivity. □

5.2 The WO-Proof

We are now ready to carry out the well-ordering proof for $EA(;)$. Our proof is strongly inspired from the traditional well-ordering proof for Peano Arithmetic as in Pohlers [19] and also its adaptation to bounded arithmetic by Beckmann [2]. Two modifications are worth being explained. First, as $\prec_e$ is defined as a transitive closure, proving many of its properties like transitivity already needs transfinite induction. Therefore we collect them into the formula $G[\alpha, e, x]$ which is proved simultaneously with the wellfoundedness property. Second, working with input bounded outputs requires a bound of input type which is large enough to bound all input bounded outputs appearing in the proof. We achieve this by choosing a fresh input variable x and adding the additional premise $h(\alpha, e) \leq x$ to each sequent containing the input variable α. This property is then passed on to all $\beta \preceq_e \alpha$ via $G[\alpha, e, x]$ and in particular implies $\mathrm{DptBd}[\alpha, e, x]$.

This partly motivates our choice for the jump A^* below. The premise $\mathrm{OBd}(\alpha, \xi, e)$ present in $\tilde{A}$ is included for technical reasons. It becomes weaker and weaker when α runs up the ordinals and vanishes completely as soon as α completes the jump and reaches ω^ξ. $\mathrm{Prog}[A, e, \alpha]$ and $\mathrm{TI}[A, e, \alpha]$ formalise progressiveness of and transfinite induction for A with respect to the order $\prec_e$ and are quite standard. The additional premise on x can finally be removed with the help of corollary 12.

Definition 29.

1. $\mathrm{OBd}(\alpha, \xi, e) :\equiv \forall \delta \preceq_e \alpha . \delta \preceq_e \omega^\xi$.
2. $\tilde{A}(\gamma, \alpha, a, \xi, e) :\equiv \mathrm{OBd}(\gamma + \omega^\alpha \cdot a, \xi, e) \wedge \forall \delta \prec_e \gamma . A[\delta] \to \forall \delta \prec_e \gamma + \omega^\alpha \cdot a . A[\delta]$.

3. $A^*[\alpha,\xi,e,x] :\equiv \forall\gamma.h(\gamma+\omega^\alpha\cdot 1,e)\le x\wedge G[\gamma,e,x]$
$\to G[\gamma+\omega^\alpha\cdot 1,e,x]\ \wedge\ \gamma\prec_e\gamma+\omega^\alpha\cdot 1\ \wedge\ \tilde{A}(\gamma,\alpha,1,\xi,e).$

4. $\mathrm{Prog}(A,e,\beta) :\equiv (\forall\alpha\prec_e\beta.A[\alpha])\to A[\beta].$

5. $\mathrm{TI}(A,e,\alpha) :\equiv (\forall\delta\preceq_e\alpha.\mathrm{Prog}(A,e,\delta)\to\forall\delta\prec_e\alpha.A[\delta].$

Lemma 30.

1. *Assume* $h(\gamma+\omega^\alpha\cdot a,e)$, $a\le x$, $\alpha\prec_e\beta$, *and* $G[\alpha,e,x]$. *Then*

$$\vdash (\forall\alpha\prec_e\beta.A^*[\alpha,\xi,e,x])\wedge G[\gamma,e,x]$$
$$\to G[\gamma+\omega^\alpha\cdot a,e,x]\ \wedge\ \gamma\preceq_e\gamma+\omega^\alpha\cdot a\ \wedge\ \tilde{A}(\gamma,\alpha,a,\xi,e).$$

2. *Assume* $1\le e\le x$, $h(\gamma+\omega^\beta\cdot 1,e)\le x$, $\beta\neq 0$, $\beta\preceq_e\xi$, *and* $G[\xi,e,x]$. *Then*

$$\vdash (\forall\gamma\preceq_e\omega^\xi.\mathrm{Prog}(A,e,\gamma))\wedge(\forall\alpha\prec_e\beta.A^*[\alpha,\xi,e,x])\wedge G[\gamma,e,x]\to$$
$$G[\gamma+\omega^\beta\cdot 1,e,x]\ \wedge\ \gamma\prec_e\gamma+\omega^\beta\cdot 1\ \wedge\ \tilde{A}(\gamma,\beta,1,\xi,e).$$

3. *Assume* $1\le e\le x$ *and* $h(\gamma+\omega^0\cdot 1,e)\le x$. *Then*

$$\vdash \forall\gamma\preceq_e\omega^\xi.\mathrm{Prog}(A,e,\gamma)\wedge G[\gamma,e,x]\to G[\gamma+\omega^0\cdot 1,e,x]\wedge\gamma\prec_e\gamma+\omega^0\cdot 1\wedge\tilde{A}(\gamma,0,1,\xi,e).$$

Proof. 1. By input bounded induction on a for

$$B[a] :\equiv h(\gamma+\omega^\alpha\cdot a,e)\le x\to G[\gamma+\omega^\alpha\cdot a,e,x]\wedge\gamma\preceq_e\gamma+\omega^\alpha\cdot a\wedge\tilde{A}(\gamma,\alpha,a,\xi,e).$$

In the base case we have $\gamma+\omega^\alpha\cdot 0=\gamma$ by lemma 22.4, so there is nothing to do. For the induction step we may assume $a\le x$, and the premises $G[\gamma,e,x]$ and $G[\alpha,e,x]$ by lemma 28.1 imply $Ord(\gamma,\alpha)$, $g(\gamma,e)\downarrow$, $g(\alpha,e)\downarrow$ and $\gamma,\alpha,g(\gamma,e),g(\alpha,e)\le x$. So $\gamma+\omega^\alpha\cdot a\downarrow$, $Ord(\gamma+\omega^\alpha\cdot a)$, and $(\gamma+\omega^\alpha\cdot a)+\omega^\alpha\cdot 1=\gamma+\omega^\alpha\cdot\mathsf{s}a$ by lemma 22, and

$$h(\gamma+\omega^\alpha\cdot a,e)=h_0(g(\gamma,e),e,g(\alpha,e),a)\le h_0(g(\gamma,e),e,g(\alpha,e),\mathsf{s}a)=h(\gamma+\omega^\alpha\cdot\mathsf{s}a,e)\le x$$

by lemmas 27.1 and 27.2 (and one of the premises for the final inequality).

To prove $B[\mathsf{s}a]$ we assume $h(\gamma+\omega^\alpha\cdot\mathsf{s}a,e)\le x$. Then $h(\gamma+\omega^\alpha\cdot a,e)\le x$ by transitivity, and the induction hypothesis gives

$$G[\gamma+\omega^\alpha\cdot a,e,x]\ \wedge\ \gamma\preceq_e\gamma+\omega^\alpha\cdot a\ \wedge\ \tilde{A}(\gamma,\alpha,a,\xi,e).$$

From $A^*[\alpha,\xi,e,x]$ we obtain (instantiating γ with $\gamma+\omega^\alpha\cdot a$)

$$G[\gamma+\omega^\alpha\cdot\mathsf{s}a,e,x]\ \wedge\ \gamma+\omega^\alpha\cdot a\preceq_e\gamma+\omega^\alpha\cdot\mathsf{s}a\ \wedge\ \tilde{A}(\gamma+\omega^\alpha\cdot a,\alpha,1,\xi,e).$$

This in particular implies $\mathrm{Tran}(\gamma + \omega^\alpha \cdot \mathsf{s}a)$, and therefore $\gamma \preceq_e \gamma + \omega^\alpha \cdot \mathsf{s}a$. Finally, to show $\tilde{A}(\gamma, \alpha, \mathsf{s}a, \xi, e)$, we assume $\mathrm{OBd}(\gamma + \omega^\alpha \cdot \mathsf{s}a, \xi, e)$ and $\forall \delta \prec_e \gamma . A[\delta]$. This implies $\mathrm{OBd}(\gamma + \omega^\alpha \cdot a, \xi, e)$ by $\mathrm{Tran}(\gamma + \omega^\alpha \cdot \mathsf{s}a)$ again. Now from $\tilde{A}(\gamma, \alpha, a, \xi, e)$ we can deduce $\forall \delta \prec_e \gamma + \omega^\alpha \cdot a . A[\delta]$, which together with $\tilde{A}(\gamma + \omega^\alpha \cdot a, \alpha, \mathsf{1}, \xi, e)$ leads to $\forall \delta \prec_e \gamma + \omega^\alpha \cdot \mathsf{s}a . A[\delta]$. This completes the induction step, and an application of lemma 3 completes the proof.

2. By lemma 28.3 we also can assume $G[\beta, e, x]$. This and the premise $G[\gamma, e, x]$ ensure $\mathit{Ord}(\gamma, \beta)$ and $\gamma, \beta \leq x$, so we also have $\gamma + \omega^\beta \cdot 1\!\downarrow$, $\mathit{Ord}(\gamma + \omega^\beta \cdot 1)$, and $\gamma + \omega^\beta \cdot 1 \neq 0$ by lemma 22. From lemma 25 we get $\mathit{pred}(\gamma + \omega^\beta \cdot 1, e) = \gamma + \omega^\alpha \cdot a$ for some $a \leq e$ and some $\alpha \prec_e \beta$ that satisfies $\mathit{Ord}(\alpha)$. This α by lemmas 28.1 and 28.3 and by $\mathrm{Mon}(\beta, e)$ also satisfies $G[\alpha, e, x]$, $g(\alpha, e) < g(\beta, e)$ and $\alpha \leq x$. Furthermore the premise $h(\gamma + \omega^\beta \cdot 1, e) \leq x$ by lemmas 27.1 and 27.2 and by transitivity implies $h(\gamma + \omega^\alpha \cdot a, e) \leq x$. This means that we have can apply item 1 of this lemma to obtain $G[\gamma + \omega^\alpha \cdot a, e, x]$, $\gamma \preceq_e \gamma + \omega^\alpha \cdot a$ and $\tilde{A}(\gamma, \alpha, a, \xi, e)$. From this, by lemmas 28.4 and 24, we conclude $G[\gamma + \omega^\beta \cdot 1, e, x]$ and $\gamma \prec_e \gamma + \omega^\beta \cdot 1$, the first and the second conjunct.

   For the third assume $\mathrm{OBd}(\gamma + \omega^\beta \cdot 1, \xi, e)$, $\forall \delta \prec_e \gamma . A[\delta]$ and $\delta \prec_e \gamma + \omega^\beta \cdot 1$, and show that $A[\delta]$ holds. $G[\gamma + \omega^\beta \cdot 1, e, x]$ entails $\mathrm{Tran}(\gamma + \omega^\beta \cdot 1, e)$ and also $\gamma + \omega^\beta \cdot 1 \leq x$. But by the former we get $\mathrm{OBd}(\gamma + \omega^\alpha \cdot a, e)$, which in turn entails $\gamma + \omega^\alpha \cdot a \preceq_e \omega^\xi$ and, using the premises, $\mathrm{Prog}(A, e, \gamma + \omega^\alpha \cdot a)$. The latter together with lemma 24 $\delta \preceq_e \gamma + \omega^\alpha \cdot a$. Now if $\delta \prec_e \gamma + \omega^\alpha \cdot a$ we get $A[\delta]$ from $\tilde{A}(\gamma, \alpha, a, \xi, e)$, therefore $A[\gamma + \omega^\alpha \cdot a]$ holds by $\mathrm{Prog}(A, e, \gamma + \omega^\alpha \cdot a)$.

3. Somewhat similar to the previous item: $G[\gamma, e, x]$ ensures $\mathit{Ord}(\gamma)$ and $\gamma \leq x$, so $\gamma + \omega^0 \cdot 1\!\downarrow$, $\mathit{Ord}(\gamma + \omega^0 \cdot 1)$ and $(\gamma + \omega^0 \cdot 1) \neq 0$ by lemma 22, and $\mathit{pred}(\gamma + \omega^0 \cdot \mathsf{1}, e) = \gamma$ by lemma 25. Notice that $G[0, e, x]$ holds by lemma 28.2, so by lemmas 28.4 and 24 we can conclude $G[\gamma + \omega^0 \cdot \mathsf{1}, e, x]$ and $\gamma \prec_e \gamma + \omega^0 \cdot \mathsf{1}$.

   For the third conjunct assume $\mathrm{OBd}(\gamma + \omega^0 \cdot \mathsf{1}, \xi, e)$, $\forall \delta \prec_e \gamma . A[\delta]$ and $\delta \prec_e \gamma + \omega^0 \cdot \mathsf{1}$, and we want to show that $A[\delta]$ holds. $G[\gamma + \omega^0 \cdot \mathsf{1}, e, x]$ entails $\mathrm{Tran}(\gamma + \omega^0 \cdot \mathsf{1}, e)$ and $\gamma + \omega^0 \cdot \mathsf{1} \leq x$ again, and the former via $\mathrm{OBd}(\gamma, e)$ entails $\gamma \preceq_e \omega^\xi$ and $\mathrm{Prog}(A, e, \gamma)$. The latter by lemma 24 again implies $\delta \preceq_e \gamma$. Now if $\delta \prec_e \gamma$ we get $A[\delta]$ from $\tilde{A}(\gamma, \alpha, a, \xi, e)$, therefore $A[\gamma]$ holds by $\mathrm{Prog}(A, e, \gamma)$. □

Lemma 31. *Assume* $1 \leq e \leq x$ *and* $G[\xi, e, x]$. *Then*

$$\vdash (\forall \gamma \preceq_e \omega^\xi . \mathrm{Prog}(A, e, \gamma)) \to \forall \beta \preceq_e \xi . \mathrm{Prog}(A^*[\cdot, \xi, e, x], e, \beta),$$

where the notation $\mathrm{Prog}(A^*[\cdot, \xi, e, x], e, \beta)$ *means that the formula* $A^*[\alpha, \xi, e, x]$ *satisfies* Prog *with respect to its free variable* α.

Proof. Assume $\beta \preceq_e \xi$, $\forall \alpha \prec_e \beta . A^*[\alpha, \xi, e, x]$, $h(\gamma + \omega^\beta \cdot 1, e) \leq x$ and $G[\gamma, e, x]$. Then we have to show $G[\gamma + \omega^\beta \cdot 1, e, x]$, $\gamma \prec_e \gamma + \omega^\beta \cdot 1$ and $\tilde{A}(\gamma, \beta, \mathsf{1}, \xi, e)$, which is immediate from lemma 30. □

Lemma 32. *Assume* $1 \leq e \leq x$ *and* $h(\omega^\alpha, e) \leq x$. *Then*

$$\vdash G[\alpha, e, x] \wedge \mathrm{TI}(A^*[\cdot, \alpha, e, x], e, \alpha) \to \mathrm{TI}(A, e, \omega^\alpha).$$

Proof. Fairly standard: Assume $G[\alpha, e, x]$, $\mathrm{TI}(A^*[\cdot, \alpha, e, x], e, \alpha)$ and $\forall \delta \preceq_e \omega^\alpha.\mathrm{Prog}(A, e, \delta)$, we have to show $\forall \delta \prec_e \omega^\alpha.A[\delta]$.

Lemma 31 tells us that $\forall \beta \preceq_e \alpha.\mathrm{Prog}(A^*[\cdot, \alpha, e, x], e, \beta)$. From this we get, when using $\mathrm{TI}(A^*[\cdot, \alpha, e, x], e, \alpha)$, that $\forall \delta \prec_e \alpha.A^*[\delta, \alpha, e, x]$ holds, and also the particular instance $\mathrm{Prog}(A^*[\cdot, \alpha, e, x], e, \alpha)$. These together immediately lead to $A^*[\alpha, \alpha, e, x]$. Expanding definitions and instantiating γ by 0 (which is defined) we are done, as soon as we have shown $G[0, e, x]$, $\mathrm{OBd}(\omega^\alpha, \alpha, e)$ and $\forall \delta \prec_e 0.A[\delta]$. But all three are immediate, the first from lemma 28, the second from the definition of OBd, and the third from the fact that $\neg \delta \prec_e 0$ follows from the equational program. □

Lemma 33. *Let* $\top(\delta) :\equiv \delta = \delta$, *and assume* $1 \leq e \leq x$ *and* $h(\omega^\alpha, e) \leq x$. *Then*

$$\vdash G[\alpha, e, x] \wedge \mathrm{TI}(\top^*[\cdot, \alpha, e, x], e, \alpha) \to G[\omega^\alpha, e, x].$$

Proof. Almost the same as for the previous lemma. Notice that $\forall \delta \preceq_e \omega^\alpha.\mathrm{Prog}(\top, e, \delta)$ is trivially provable, so applying lemma 31 and using the premise $\mathrm{TI}(\top^*[\cdot, \alpha, e, x], e, \alpha)$ yields $\top^*[\alpha, \alpha, e, x]$, and the claim follows by expanding definitions as above. □

The n-fold ω-towers are defined as usual by

$$\omega_0(\alpha) = \alpha \qquad \omega_{n+1}(\alpha) = \omega^{\omega_n(\alpha)}.$$

Then $\omega_n(0)$, for all n, is a closed term, and thus trivially defined. Almost as trivial, $\omega_n(0)$ is a code of an ordinal, formally to be shown by a straightforward (meta-)induction on n.

Lemma 34. *For all formulas* A *and for any fixed* n,

$$\vdash 1 \leq e \leq x \wedge \bigwedge_{m \leq n} h(\omega_m(\mathsf{0}), e) \leq x \to G[\omega_n(0), e, x] \wedge \mathrm{TI}(A, e, \omega_n(0)).$$

Proof. As usual, by (meta-)induction on n. $G[0, e, x] \wedge \mathrm{TI}(A, e, 0)$ is immediate from lemma 28 and the definition of TI, as $\neg \delta \prec_e 0$. Now the induction hypothesis gives $G[\omega_n(0), e, x]$, $\mathrm{TI}(\top^*[\cdot, \omega_n(0), e, x], e, \omega_n(\mathsf{0}))$ and $\mathrm{TI}(A^*[\cdot, \omega_n(0), e, x], e, \omega_n(\mathsf{0}))$, so from the two previous lemmas we get $G[\omega_{n+1}(0), e, x] \wedge \mathrm{TI}(A, e, \omega_{n+1}(0))$, which concludes the induction step. □

Theorem 35. *For all formulas* A *and for any fixed* n, *if* x *is a fresh input variable, then*

$$\vdash \mathrm{TI}(A, x, \omega_n(0)).$$

This is the main theorem. It states that for all formulas, $EA(;)$ can uniformly prove transfinite induction up to $\omega_n(0)$ with respect to the orderings $\prec_x$.

Proof. The first step is to show $e \leq x \to g(\omega_m(0), e)\!\downarrow \wedge\, h(\omega_m(0), e)\!\downarrow$ by (meta-)induction on m. The base case is trivial. For the induction step we use corollary 12 and the induction hypothesis to assume $g(\omega_m(0), e), h(\omega_m(0), e) \leq x$. Then the claim is immediate from lemma 27.2.

To prove the theorem, the previous lemma gives

$$1 \leq e \leq x \wedge \bigwedge_{m \leq n} h(\omega_m(\mathsf{0}), e) \leq x \to \mathrm{TI}(A, e, \omega_n(0)).$$

Drop the premise $\bigwedge_{m\leq n} h(\omega_m(\mathsf{0}), e) \leq x$ by using corollary 12, and substitute x for e. We are left with $1 \leq x \to \mathrm{TI}(A, x, \omega_n(0))$. So all that remains to show is $\mathrm{TI}(A, \mathsf{0}, \omega_n(0))$.

The proof of $\mathrm{TI}(A, \mathsf{0}, \omega_n(0))$ relies on the fact that $\delta \prec_{\mathsf{0}} \omega_{n+2}(0)$ is equivalent to $\delta \preceq_{\mathsf{0}} \omega_n(0)$. This will be clear from lemma 24, once we have seen that $\mathit{pred}(\omega_{n+2}(0), \mathsf{0}) = \omega_n(0)$, since we can assume $\omega_{n+2}(0) \leq x$ by using corollary 12, and $\omega_{n+2}(0) \neq \mathsf{0}$ holds by lemma 22.1.

But $\mathit{pred}(\omega_{n+2}(0), \mathsf{0}) = \omega_n(0)$ is easily verified by (meta-)induction on n. Before doing this, observe that $\omega_{n+2}(0) = \langle 0, \langle 0, \omega_n(0), 1\rangle, 1\rangle$ is immediate from the equational program. Then so is the base case, because $\omega_0(0) = 0$ implies $\mathit{pred}(\omega_2(0), \mathsf{0}) = 0 + \omega^{\mathit{pred}(\langle 0,0,1\rangle, \mathsf{0})} \cdot \mathsf{0} = 0$, and the induction step as well, computing

$$\mathit{pred}(\omega_{n+3}(0), \mathsf{0}) = 0 + \omega^{\mathit{pred}(\omega_{n+2}(0), \mathsf{0})} \cdot 1 = 0 + \omega^{\omega_n(0)} \cdot 1 = \omega_{n+1}(0).$$

Returning to the proof of $\mathrm{TI}(A, \mathsf{0}, \omega_n(0))$, this is trivial for the base cases $n = 0$ and $n = 1$, because we have $\neg\delta \prec_{\mathsf{0}} 0$ and $\delta \prec_{\mathsf{0}} \langle 0, 0, 1\rangle \leftrightarrow \delta \preceq_{\mathsf{0}} \mathit{pred}(\langle 0, 0, 1\rangle, \mathsf{0}) \leftrightarrow \delta = 0$. For the induction step we assume $\forall\delta.\mathit{Ord}(\delta) \to \mathrm{Prog}(A, \mathsf{0}, \delta)$. This by the induction hypothesis gives $\forall\delta \prec_e \omega_n(0).A[\delta]$ and therefore, as $\mathit{Ord}(\omega_n(0))$ holds, $A[\omega_n(0)]$ as well. These together say $\forall\delta \preceq_{\mathsf{0}} \omega_{n+2}(0).A[\delta]$ which by the remarks above is all we have to show. □

With the help of lemma 34 we can apply a similar procedure to see that we could rephrase all results on input bounded (codes of) ordinals to hold for all $\delta \prec_x \omega_n(0)$. We perform this for a specific example that we will need in the next chapter.

Lemma 36. *$EA(;)$ proves, for all n,*

$$\forall\delta \preceq_x \omega_n(0).(\delta = 0 \vee \mathit{Lim}(\delta) \vee \mathit{Succ}(\delta)) \wedge (\delta \neq 0 \to \mathit{pred}(\delta, x)\!\downarrow \wedge\, \mathit{pred}(\delta, x) \prec_x \delta).$$

Proof. Let $A(\delta, e) :\equiv (\delta = 0 \vee \mathit{Lim}(\delta) \vee \mathit{Succ}(\delta)) \wedge (\delta \neq 0 \to \mathit{pred}(\delta, e)\!\downarrow \wedge\, \mathit{pred}(\delta, e) \prec_e \delta)$. The first step is to observe $(G[\alpha, e, x] \wedge \delta \preceq_e \alpha) \to \mathit{Ord}(\delta) \wedge \delta \leq x$ which follows directly from lemmas 28.1 and 28.3. Then $(G[\alpha, e, x] \wedge \delta \preceq_e \alpha) \to A(\delta, e)$ is immediate from lemmas 20, 23 and 24. Apply lemma 34, and we can remove premises and substitute x for e exactly the same way as in the proof of the theorem. □

Chapter 6

Transfinite Induction, Upper Bounds

6.1 The Slow-Growing Hierarchy

In this chapter we define a notion of provable ordinal for two-sorted arithmetic and show that the slow-growing hierarchy along these ordinals is always provably total. This implies that the lower bound ε_0 given in the previous chapter is a sharp one, as the slow-growing hierarchy up to (and including) ε_0 is a non-elementary function, see [8] for details. On the other hand we know from Ostrin and Wainer [18, Theorem 3.5] that all provably total functions of $EA(;)$ are elementary.

Our definition of a provable ordinal is based on structured tree ordinals rather than set theoretic ordinals. There are several reasons why this choice seems to be more adequate in this context. Following Wainer's program (as in [18]), introducing variable separation gives rise to a proof theory based on the slow-growing hierarchy, and we can see tree ordinals as a slow-growing counterpart to set theoretical ordinals. Secondly, set theoretical ordinals are simply too coarse. Sommer ([22], [23]) shows that the proof-theoretic ordinal in the classical sense must be ω^2 for all theories between $I\Delta_0$ (or T^2_1 respectively) and $I\Sigma_1$, see Beckmann [2, p. 4] for more details. This in particular means that classical ordinal analysis can't separate any of these theories. But $EA(;)$, as well as it's fragments and extensions which are also of interest, lie all in that range, at least when comparing the respective provably total functions: Linspace for $I\Delta_0$, elementary time for $EA(;)$, and all primitive recursive functions for $I\Sigma_1$. A more technical reason finally is that the lower bound we gave in theorem 35 didn't well-order any set-theoretical ordinal at all, but rather a family of (increasing) suborderings $\prec_n$. This perfectly fits the correspondance between tree ordinals and their sets of n-predecessors $\alpha[n]$, cf. Fairtlough and Wainer [8].

Our definition of the structured tree ordinals exactly follows [8], but for convenience we repeat the most relevant facts in the following definition and in lemma 38, which is copied from their corollary 2.9.

Definition 37. The set Ω of the *countable tree ordinals* is inductively defined as the closure of the zero ordinal 0 under successor $\alpha + 1 := \alpha \cup \{\alpha\}$ and limits $\sup(\alpha_x) := \langle \alpha_x \rangle_{x \in \mathbb{N}}$. The letter λ will denote limit ordinals $\lambda = \langle \lambda_x \rangle$.

This definition gives raise to a natural (partial) ordering $\prec$ on the set of the countable tree ordinals, defined as the transitive closure of the rules $\alpha \prec \alpha + 1$ and $\lambda_m \prec \langle \lambda_x \rangle$ (for all natural numbers m).

For each $\alpha \in \Omega$ the restriction $\prec_\alpha$ of this ordering below α is wellfounded. The ordinal height $|\alpha|$ of α is defined to be the set-theoretic ordinal height of $\prec_\alpha$.

For each natural number n the set of *n-predecessors* $\alpha[n]$ of a tree ordinal α is defined inductively by

- $0[n] := \emptyset$,
- $(\alpha + 1)[n] := \alpha[n] \cup \{\alpha\}$,
- $\lambda[n] := \lambda_n[n]$.

A tree ordinal α is called a *structured tree ordinal* if every $\lambda \preceq \alpha$ satisfies, for all n, the inclusion $\lambda_n \in \lambda[n+1]$.

Lemma 38. *For each non-zero structured tree ordinal α the set $\{\beta : \beta \prec \alpha\}$ is well-ordered by $\prec$, with least element 0 and such that $\beta \prec \alpha$ implies $\beta + 1 \preceq \alpha$. This well-ordering is the direct union of its finite sub-orderings $\alpha[n]$ for $n \in \mathbb{N}$.*

Definition 39.

1. A model of $EA(;)$ is called a *standard model*, if both the input and output variables are interpreted as (standard) natural numbers, the symbols $\mathsf{0}$, s and $=$ are interpreted as the constant 0, the successor function and the equality relation on the natural numbers, and the function symbols are assigned any partial functions on the natural numbers that make all equations in P true.

2. An *arithmetization of a tree ordinal* α consists of formulas *Ord*, *Succ*, *Lim* and $A_\prec$ in the language of $EA(;)$, an equational program P containing a binary function *pred* and a constant 0, and an injection $\lceil \cdot \rceil$ from $\{\beta : \beta \preceq \alpha\}$ into the closed terms of the language of $EA(;) + P$ such that

   (a) For all $\delta, \beta \prec \alpha$ and for all natural numbers n, $\delta \in \beta[n]$ if and only if $A_\prec[\lceil \delta \rceil, \lceil \beta \rceil, \overline{n}]$ is true in every standard model. Here $\overline{n}$ denotes the n-th numeral, and we usually write $a \prec_e b$ for $A_\prec[a, b, e]$.

   (b) For all $\beta, \lambda \prec \alpha$ and for all n, both $Succ(\lceil \beta + 1 \rceil) \wedge pred(\lceil \beta + 1 \rceil, \overline{n}) = \lceil \beta \rceil$ and $Lim(\lceil \lambda \rceil) \wedge pred(\lceil \lambda \rceil, \overline{n}) = \lceil \lambda_n \rceil$ are true in every standard model.

(c) For all $\beta \in \alpha[n]$ and for all n, $EA(;)$ proves $Ord(\lceil\beta\rceil, \overline{n})$.

(d) $EA(;)$ proves both $\forall\delta.Ord(\delta, x) \to (\delta = 0 \vee Lim(\delta) \vee Succ(\delta))$ and $\forall\delta.Ord(\delta, x) \wedge \delta \neq 0 \to (pred(\delta, x)\downarrow \wedge A_{\prec}[pred(\delta, x), \delta, x])$.

3. An ordinal α is *provable in* $EA(;)$ if and only if there is an arithmetization of α such that $EA(;) \vdash \mathrm{TI}(A, \prec, x, \lceil\alpha\rceil)$ for all formulas A, where

$$\mathrm{TI}(A, \prec, e, \alpha) :\equiv (\forall\delta.Ord(\delta, e) \to \mathrm{Prog}(A, \prec, e, \delta)) \to \forall\delta \prec_e \alpha.A[\delta]$$

and

$$\mathrm{Prog}(A, \prec, e, \beta) :\equiv (\forall\alpha \prec_e \beta.A[\alpha]) \to A[\beta].$$

We don't require that $EA(;)$ proves $\prec_x \subseteq \prec_{\mathsf{s}x}$ (theorem 2.8 of Fairtlough and Wainer [8]) for α being provable. For then, $EA(;)$ would even prove the wellfoundedness of $\prec$, i.e. transfinite induction over set theoretic ordinals up to ε_0, which would be much too strong a result.

We notice that all ordinals up to ε_0 are indeed provable in $EA(;)$ in the sense of definition 39 above, if we take them as being the closure of 0 under successor, addition and exponentiation with base $\omega := \langle x\rangle_x$. This is clear from lemma 36 and theorem 35, when we choose $\delta \preceq_e \lceil\alpha\rceil$ as the formula $Ord(\delta, e)$ in definition 39 for any such ordinal α.

Lemma 40. *Let α be a provable ordinal of $EA(;)$, and assume that P contains the equation*

$$g(\beta, a) = \begin{cases} \mathsf{0} & \beta = 0, \\ \mathsf{s}g(pred(\beta, a), a) & Succ(\beta), \\ g(pred(\beta, a), a) & Lim(\beta). \end{cases}$$

Then $\vdash g(\lceil\alpha\rceil, x)\downarrow$.

Proof. Let $A[\beta] :\equiv g(\beta, x)\downarrow$, we want to apply $\mathrm{TI}(A, \prec, x, \lceil\alpha\rceil)$. In order to prove the premise assume $Ord(\beta, x)$ and $\forall\delta \prec_x \beta.A[\delta]$, and show $A[\beta]$. By the assumptions on the arithmetization of α we can distinguish the cases $\beta = 0$, $Lim(\beta)$ and $Succ(\beta)$. Furthermore, if $\beta \neq 0$, then $pred(\beta, x)\downarrow$ and $pred(\beta, x) \prec_x \beta$, thus $g(pred(\beta, x), x)$ is defined by the induction hypothesis. Therefore all three cases satisfy $A[\beta]$. This means that we have shown $\forall\beta.Ord(\beta, x) \to \mathrm{Prog}(A, \prec, e, \beta)$. Now we can apply $\mathrm{TI}(\alpha, \prec, x, \lceil\alpha\rceil)$ to obtain $\forall\delta \prec_x \lceil\alpha\rceil.A[\delta]$. As $\lceil\alpha\rceil$, being a closed term, is defined, and $Ord(\lceil\alpha\rceil, x)$ holds, we have also $\mathrm{Prog}(A, \lceil\alpha\rceil)$ which implies $A[\lceil\alpha\rceil]$. □

As shown in the introduction to this chapter this result provides an upper bound for the provably total ordinals of $EA(;)$:

Corollary 41. *If a structured tree ordinal α is a provable ordinal of $EA(;)$, then its height $|\alpha|$ is smaller than ε_0.*

Proof. By (part 2(b) of) definition 39 $g(\lceil \alpha \rceil, \cdot)$, indeed defines the slow-growing hierarchy at level α. But for $|\alpha| \geq \varepsilon_0$ this is a non-elementary function! □

Corollary 42. *Define the proof theoretic ordinal of a two-sorted theory T to be*

$$\sup_{\alpha \in \Omega} \{|\alpha| : \alpha \text{ is provable in } T\}.$$

Then the proof theoretic ordinal of $EA(;)$ is ε_0.

Bibliography

[1] Aehlig, K., Berger, U., Hofmann, M., and Schwichtenberg, H. An arithmetic for non-size-increasing polynomial-time computation. *Theor. Comput. Sci. 318*, 1-2 (2004), 3–27.

[2] Beckmann, A. *Separating Fragments of Bounded Arithmetic*. PhD thesis, Univerität Münster, 1996.

[3] Bellantoni, S., and Cook, S. A new recursion-theoretic characterization of the polytime functions. *Computational Complexity 2* (1992), 97–110.

[4] Bellantoni, S., and Hofmann, M. A new "feasible" arithmetic. *J. Symb. Log. 67*, 1 (2002), 104–116.

[5] Bellantoni, S. J., Niggl, K.-H., and Schwichtenberg, H. Higher type recursion, ramification and polynomial time. *Ann. Pure Appl. Logic 104*, 1-3 (2000), 17–30.

[6] Bloch, S. A. Functional characterizations of uniform log-depth and polylog-depth circuit families. In *Structure in Complexity Theory Conference* (1992), pp. 193–206.

[7] Buss, S. R. *Bounded Arithmetic*. Bibliopolis, Napoli, 1986.

[8] Fairtlough, M., and Wainer, S. S. Hierarchies of provably recursive functions. In *Handbook of Proof Theory*, S. R. Buss, Ed. Elsevier Science Publishers, Amsterdam, 1998, pp. 149–207.

[9] Irwin, R. J., Royer, J. S., and Kapron, B. M. On characterizations of the basic feasible functionals (part i). *J. Funct. Program. 11*, 1 (2001), 117–153.

[10] Leivant, D. A foundational delineation of computational feasability. In *Proceedings of the Sixth IEEE Conference on Logic in Computer Science (Amsterdam)*. IEEE Computer Society Press, Washington, 1991.

[11] Leivant, D. Predicative recurrence in finite types. In *LFCS* (1994), A. Nerode and Y. Matiyasevich, Eds., vol. 813 of *Lecture Notes in Computer Science*, Springer, pp. 227–239.

[12] Leivant, D. Intrinsic theories and computational complexity. In *Logic and Computational Complexity*, D. Leivant, Ed., vol. 960 of *Lecture Notes in Computer Science*. Springer-Verlag, 1995.

[13] Leivant, D. Ramified recurrence and computational complexity iii: Higher type recurrence and elementary complexity. *Ann. Pure Appl. Logic 96*, 1-3 (1999), 209–229.

[14] Leivant, D. Calibrating computational feasibility by abstraction rank. In *LICS* (2002), pp. 345–.

[15] Nelson, E. *Predicative Arithmetic*. Princeton University Press, Princeton, N.J., 1986.

[16] Oitavem, I. Implicit characterizations of Pspace. In *Proof Theory in Computer Science* (2001), pp. 170–190.

[17] Ostrin, G. E., and Wainer, S. S. Proof theoretic complexity. In *Logic and Computational Complexity*, J. Tiuryn and R. Steinbruggen, Eds., vol. 960 of *Proof and System Reliability*. Kluwer, 2002, pp. 369–397.

[18] Ostrin, G. E., and Wainer, S. S. Elementary arithmetic. *Ann. Pure Appl. Logic 133*, 1-3 (2005), 275–292.

[19] Pohlers, W. *Proof Theory: An Introduction*, vol. 1407 of *Lecture Notes in Mathematics*. Springer, Berlin, 1988.

[20] Schwichtenberg, H. An arithmetic for polynomial-time computation. to appear.

[21] Simmons, H. The realm of primitive recursion. *Archive for Mathematical Logic 27* (1988), 177–188.

[22] Sommer, R. *Transfinite Induction and Hierarchies Generated by Transfinite Recursion within Peano Arithmetic*. PhD thesis, UC Berkeley, 1990.

[23] Sommer, R. Ordinals in bounded arithmetic. In *Arithmetic, Proof Theory, and Complexity*, P. Clote and J. Krajíček, Eds., vol. 23 of *Oxford Logic Guides*. Oxford University Press, 1993, pp. 320–363.

[24] Wainer, S. S., and Williams, R. S. Inductive definitions over a predicative arithmetic. *Ann. Pure Appl. Logic* (to appear).

[25] Williams, R. S. *Proof Theoretic Ordinals and Hierarchies*. PhD thesis, University of Leeds, 2005.

Index