On the Proof Theory of Applicative Theories

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Thomas Adrian Strahm

von Oberthal und Bern

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

1996

On the Proof Theory of Applicative Theories

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Thomas Adrian Strahm

von Oberthal und Bern

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

Von der Philosophisch-naturwissenschaftlichen Fakultät angenommen.

Der Dekan:

Bern, den 6. Juni 1996 Prof. Dr. H. Pfander

Contents

Introduction

In the mid seventies, Feferman [18, 20] introduced systems of explicit mathematics in order to provide an alternative foundation of constructive mathematics. More precisely, is was the origin of Feferman's program to give a logical account to Bishop's style of constructive mathematics. Right from the beginning, systems of explicit mathematics turned out to be of general interest for proof theory, mainly in connection with the proof-theoretic analysis of subsystems of first and second order arithmetic and set theory. Complete proof-theoretic information about the most prominent framework for explicit mathematics, T_0, is available since 1983 by the work of Feferman [18, 20], Feferman and Sieg [30], Jäger [45] and Jäger and Pohlers [49]. An excellent and uniform presentation of many of these results is contained, among other things, in Glaß' thesis [37].

More recently, systems of explicit mathematics have been used to develop a general logical framework for functional programming and type theory, where it is possible to derive such important properties of functional programs as termination and correctness. The programs considered are taken from functional programming languages, which are either based on the untyped λ calculus (e.g. SCHEME) or the polymorphic typed λ calculus (e.g. ML). Important references for the use of explicit mathematics as an abstract logical basis for functional programming and type theory are Feferman [25, 26, 27] and Jäger [47]. For frameworks closer to actual programming languages, cf. Hayashi and Nakano [40] and Talcott [69]. The former reference contains – among other things – the description of an experimental implementation for extracting programs from constructive proofs in a Feferman-style explicit mathematics setting.

There are two kinds of objects relevant in explicit mathematics, namely *operations* or *rules* and *classifications* or *types*. The former are characterized by the applicative axioms of a partial combinatory algebra plus natural numbers, giving rise to a standard interpretation in terms of the (indices of) partial recursive functions. Hence, an object is not only an input to a program, but acts as a program itself so that self-application is meaningful, though not necessarily total. Observe that our way of describing operations is clearly intensional. Classifications or types, on the other hand, must be thought of a being generated successively from preceding ones. Although they may define rather complicated properties, they are always explicitly given or represented by means of operations. Nevertheless, the two basic notions of operation and classification are asymmetric in the sense that the characteristic function of a type need not generally be given by an operation or rule. This ends our brief discussion on the two basic sorts of objects that are present in explicit mathematics. The reader is referred to Feferman [18, 20] and Jäger [46] for information concerning the ontology of operations and classifications.

In this thesis we exclusively focus on the applicative basis of explicit mathematics, i.e. the operation/rule part described above. The so-obtained frameworks are referred to as *applicative theories* in the sequel: these are well-known of already being of significant foundational interest for mathematics and computer science, cf. e.g. the text books Beeson [3] and Troelstra and Van Dalen [71] for extensive surveys.

Apart from some contributions concerning the syntax and semantics of theories with self-application, the main concern of this thesis is of proof-theoretic nature. We will give a broad discussion of the proof-theoretic strength of many applicative theories, where special emphasis will be put on the definition and analysis of feasible theories with self-application as well as the study of the non-constructive quantification operator μ or, more familiar, Kleene's type two functional E in an applicative framework. In the sequel let us try to outline some general aspects concerning these two central issues of our thesis.

In the last decade, there have been many activities in the field of so-called *bounded arithmetic*, emerging from Buss' important work in [5]. A huge variety of formal systems of arithmetic and their relationship to computational complexity has been investigated, and this research is still going on, cf. [7, 13, 12] for a survey and further references. A natural question – first posed by Feferman in [26] – is whether a similar program can be carried through in the context of explicit mathematics or applicative theories. Most interestingly, is it possible to set up a natural self-applicative framework in such a way that the provably total functions are exactly the polynomial time computable functions? We have shown in [67] that a direct mimicking of systems of bounded arithmetic in applicative theories does not work: this is due to the presence of unbounded recursion principles in the self-applicative setting. Hence, a theory had to be developed that is truly in the spirit of applicative systems. In this thesis, we present a first-order theory PTO of operations and binary words, which embodies the full expressive power of self-application, though its provably total functions are just the polytime ones. PTO includes a very natural induction principle on binary words, so-called set induction. The upper bound computation for PTO goes via a reduction to Ferreira's system of polynomial time computable arithmetic (cf. [32, 33]) plus a suitable reflection principle so that the distinctive feasibility w.r.t. provable Π^0_2 statements of this latter theory yields the desired bound for PTO. We will show how our approach can be extended in order to define applicative theories G_n that capture exactly the nth level of the Grzegorczyk hierarchy. In particular, the provably total function of G_3 are just the elementary functions in the sense of Kalmar.

Let us now turn to the non-constructive μ operator. In [18] Feferman has not only introduced the system T_0 mentioned above, but also a theory T_1 that is obtained from T_0 by strengthening the applicative axioms by the quantification operator μ. The standard interpretation of the operations is now given in terms of Π^1_1 recursion theory, namely the partial Π^1_1 functions. Only recently it has been possible to obtain complete proof-theoretic information about subsystems of T_1 by the work of Feferman and Jäger [28, 29] and Glaß and Strahm [38]. For the relevance of systems of explicit mathematics with μ for mathematical practice, cf. Feferman [19, 23].

In this thesis we give an extensive discussion of first-order applicative theories that are based on the non-constructive μ operator, starting off from the paper Feferman and Jäger [28]. The two principles of complete induction on the natural numbers studied there are set induction and full formula induction. In our investigations we will be particularly interested in forms of induction lying between these two induction principles, namely *positive* forms of induction on the natural numbers. The corresponding systems with μ turn out to have the proof-theoretic strength of the least primitive recursively closed ordinal, $\varphi\omega 0$. Moreover, special emphasis will be put on the role of the axiom of *totality* in our applicative setting with μ. By totality we mean that a applied to b always yields a value c for all objects a and b of our universe of discourse. Clearly, recursion-theoretic models are no longer possible in the presence of totality: the standard models for total applicative theories are term models with suitable forms of term reduction and Church Rosser properties. Such models provide a natural operational semantics of total applicative theories which is based on term reduction. The definition of term models in the presence of μ is not completely trivial: we will define so-called *infinitary* term models that include special reductions for μ and we show that the corresponding reduction relations satisfy the Church Rosser property. As in [28], the crucial tool for establishing upper bounds for applicative theories with μ are so-called fixed point theories over Peano arithmetic with ordinals. They have been introduced in Jäger [48] and extended in Jäger and Strahm [51] to second order theories with ordinals. In [28] ordinal theories have been used in order to describe the standard recursion-theoretic model of the partial Π^1_1 functions. Here we will demonstrate that the above mentioned infinitary term models together with their Church Rosser properties can be formalized in suitable fixed point theories with ordinals, thus establishing the desired proof-theoretic upper bounds for *total* applicative theories with μ. In particular, it will turn out that totality (and even extensionality) does not raise the proof-theoretic strength of all theories with μ studied in these investigations.

This finishes our very sketchy discussion of some central aspects of this thesis. We are doing without a detailed plan of our work at this point and refer the reader to the introductions of the individual chapters and sections as well as the table of contents. Finally, let us mention that throughout these investigations we have made free use of the papers Jäger and Strahm [52, 50] and Strahm [68, 66].

Acknowledgments

I am deeply grateful to Prof. Gerhard Jäger for introducing me to mathematical logic and proof theory and for his steady encouragement during the past years. He has not only been an excellent guide, but also a friend.

I am indebted to Prof. Solomon Feferman for many inspiring discussions on explicit mathematics and applicative theories and, especially, for inviting me to Stanford University last autumn.

I benefited greatly from frequent conversations with Prof. Andrea Cantini on weak applicative theories. Many thanks are due to him, too.

Last but not least, I want to thank all the present and former members as well as guests of our research group for sharing many mathematical and also other thoughts and for always providing such a friendly working atmosphere.

Bern, June 1996, Thomas Adrian Strahm

Chapter 1

Syntax and semantics of theories with self application

It is the aim of the first chapter of this thesis to give a very brief introduction into the syntax and semantics of theories with self-application, thereby laying the basis for the proof-theoretic investigations to be discussed in the remaining chapters.

In Section 1 we introduce the language and axioms of the basic theory BON of operations and numbers. The first part of Section 2 is centered around explicit definitions and recursion theorems in BON, and in the second part we address some basic inconsistency results concerning extensions of BON. Section 3 contains a short survey of various models of BON, where some emphasis is put on recursion-theoretic and term models. In Section 4 we will establish some undecidability results for total models of BON, thus generalizing the Scott-Curry undecidability theorem of the untyped λ calculus. Section 5 contains a very short discussion on finitary inductive data types as an alternative to the type of the natural numbers in BON. In Section 6, finally, we give a review of our paper [68] on partial applicative theories and explicit substitutions, thereby shedding some more light on crucial differences between partial and total applicative theories.

For more comprehensive introductions to applicative theories the reader is referred to Beeson [3], Cantini [9], Feferman [18, 20], and Troelstra [71].

1.1 The formal framework

Let us begin with describing the formal framework of the basic theory BON of operations and numbers, cf. Feferman and Jäger [28].

We start off from the language $\mathcal{L}_\mathsf{c}$ of partial combinatory logic. $\mathcal{L}_\mathsf{c}$ is a first order language of the logic of partial terms with *individual variables* $a, b, c, x, y, z, u, v, w, f, g, h, \dots$ (possibly with subscripts). In addition, $\mathcal{L}_\mathsf{c}$ includes *individual constants* k, s (combinators) and p, p_0, p_1 (pairing and unpairing). $\mathcal{L}_\mathsf{c}$ has a binary function symbol $\cdot$ for (partial) term application, a unary relation symbol $\downarrow$ (defined) as well as a binary relation symbol $=$ (equality). The language $\mathcal{L}_{\mathsf{N}}$ of basic operations and numbers comprises $\mathcal{L}_{\mathsf{c}}$ and, in addition, includes constants $\mathsf{0}$ (zero), $\mathsf{s_N}$ (numerical successor), $\mathsf{p_N}$ (numerical predecessor), $\mathsf{d_N}$ (definition by numerical cases), $\mathsf{r_N}$ (primitive recursion) and possibly other constants to be specified separately. Furthermore, $\mathcal{L}_{\mathsf{N}}$ has a unary relation symbol N (natural numbers).

The *individual terms* $(r, s, t, r_1, s_1, t_1, \ldots)$ of $\mathcal{L}_{\mathsf{N}}$ are inductively defined as follows:

1. The individual variables and individual constants are individual terms.
2. If s and t are individual terms, then so also is $(s \cdot t)$.

As usual $t[s/x]$ denotes the term t where each occurrence of the variable x is replaced by the term s. In the following we write (st) or just st instead of $(s \cdot t)$, and we adopt the convention of association to the left, i.e. $s_1 s_2 \ldots s_n$ stands for $(\ldots(s_1 s_2)\ldots s_n)$. We also write (t_1, t_2) for $\mathsf{p} t_1 t_2$ and $(t_1, t_2, \ldots, t_n)$ for $(t_1, (t_2, \ldots, t_n))$. Further we put $t' := \mathsf{s_N} t$ and $\mathsf{1} := \mathsf{0}'$. Finally, the numeral $\overline{n}$ is inductively given by $\overline{0} := \mathsf{0}$ and $\overline{n+1} := \mathsf{s_N}\overline{n}$.

The *formulas* $(A, B, C, A_1, B_1, C_1, \ldots)$ of $\mathcal{L}_{\mathsf{N}}$ are inductively defined as follows:

1. Each atomic formula $\mathsf{N}(t)$, $t\!\downarrow$ and $(s = t)$ is a formula.
2. If A and B are formulas, then so also are $\neg A$, $(A \vee B)$, $(A \wedge B)$ and $(A \to B)$.
3. If A is a formula, then so also are $(\exists x)A$ and $(\forall x)A$.

Our applicative theories are based on *partial* term application. Hence, it is not guaranteed that terms have a value, and $t\!\downarrow$ is read as "t is defined" or "t has a value". The *partial equality relation* $\simeq$ is introduced by

$$s \simeq t \ := \ (s\!\downarrow \vee\, t\!\downarrow) \to (s = t).$$

In addition, we write $(s \neq t)$ for $(s\!\downarrow \wedge\, t\!\downarrow \wedge\, \neg(s = t))$. Finally, we use the following abbreviations concerning the predicate N:

$$\begin{aligned}
t \in \mathsf{N} &\ :=\ \mathsf{N}(t),\\
(\exists x \in \mathsf{N})A &\ :=\ (\exists x)(x \in \mathsf{N} \wedge A),\\
(\forall x \in \mathsf{N})A &\ :=\ (\forall x)(x \in \mathsf{N} \to A),\\
(t \in \mathsf{N} \to \mathsf{N}) &\ :=\ (\forall x \in \mathsf{N})(tx \in \mathsf{N}),\\
(t \in \mathsf{N}^{m+1} \to \mathsf{N}) &\ :=\ (\forall x \in \mathsf{N})(tx \in \mathsf{N}^m \to \mathsf{N}).
\end{aligned}$$

Before we turn to the exact axiomatization of BON, let us give an informal interpretation of its syntax. The individual variables are conceived of as ranging over a universe V of computationally amenable objects, which can freely be applied to each other. Self-application is meaningful, but not necessarily total. V is assumed to be combinatory complete, due to the presence of the well-known combinators k

and s, and V is closed under pairing. There is an infinite collection of objects $\mathsf{N} \subset V$ which is generated from $\mathsf{0}$ by the successor operation $\mathsf{s_N}$; there exists a corresponding predecessor operation $\mathsf{p_N}$. Finally, $\mathsf{d_N}$ acts as a definition by cases operator on N, and an operation $\mathsf{r_N}$ guarantees closure under primitive recursion on N.

The underlying logic of BON is the *classical logic of partial terms* due to Beeson [3]; it corresponds to E^+ logic with strictness and equality of Troelstra and Van Dalen [70]. The non-logical axioms of BON are divided into the following five groups.

I. Partial combinatory algebra.

(1) $\mathsf{k}xy = x$,

(2) $\mathsf{s}xy\downarrow \wedge \mathsf{s}xyz \simeq xz(yz)$.

II. Pairing and projection.

(3) $\mathsf{p_0}(x, y) = x \wedge \mathsf{p_1}(x, y) = y$.

III. Natural numbers.

(4) $\mathsf{0} \in \mathsf{N} \wedge (\forall x \in \mathsf{N})(x' \in \mathsf{N})$,

(5) $(\forall x \in \mathsf{N})(x' \neq \mathsf{0} \wedge \mathsf{p_N}(x') = x)$,

(6) $(\forall x \in \mathsf{N})(x \neq \mathsf{0} \rightarrow \mathsf{p_N}x \in \mathsf{N} \wedge (\mathsf{p_N}x)' = x)$.

IV. Definition by numerical cases.

(7) $a \in \mathsf{N} \wedge b \in \mathsf{N} \wedge a = b \rightarrow \mathsf{d_N}xyab = x$,

(8) $a \in \mathsf{N} \wedge b \in \mathsf{N} \wedge a \neq b \rightarrow \mathsf{d_N}xyab = y$.

V. Primitive recursion on N.

(9) $(f \in \mathsf{N} \rightarrow \mathsf{N}) \wedge (g \in \mathsf{N}^3 \rightarrow \mathsf{N}) \rightarrow (\mathsf{r_N}fg \in \mathsf{N}^2 \rightarrow \mathsf{N})$,

(10) $(f \in \mathsf{N} \rightarrow \mathsf{N}) \wedge (g \in \mathsf{N}^3 \rightarrow \mathsf{N}) \wedge x \in \mathsf{N} \wedge y \in \mathsf{N} \wedge h = \mathsf{r_N}fg \rightarrow$
$hx\mathsf{0} = fx \wedge hx(y') = gxy(hxy)$.

It is important to observe that the theory BON does not include complete induction on the natural numbers N. We will discuss the effect of adding various induction principles to BON in the next chapters, where we will be studying the proof theory of applicative theories.

Before we turn to the discussion of basic consequences and facts about the axioms of BON, let us briefly recall the notion of a *subset of* N from [19, 28]. Sets of natural numbers are most naturally understood in our applicative context via their characteristic functions which are total on N. Accordingly, we define

$$f \in \mathcal{P}(\mathsf{N}) \;:=\; (\forall x \in \mathsf{N})(fx = \mathsf{0} \vee fx = \mathsf{1}),$$

with the intention that an object x belongs to the set $f \in \mathcal{P}(\mathsf{N})$ if and only if $(fx = \mathsf{0})$.

For now, we will be interested in three possible strengthenings of the applicative axioms, namely totality, extensionality, and full definition by cases. The *totality axiom* (Tot) expresses that application is always total, i.e.

$$(\mathsf{Tot}) \qquad (\forall x)(\forall y)(xy\downarrow).$$

In the sequel we will often write TON instead of BON + (Tot). The *extensionality axiom* (Ext) claims that operations are extensional in the following sense:

$$(\mathsf{Ext}) \qquad (\forall x)(fx \simeq gx) \rightarrow (f = g).$$

Finally, by *full definition by cases* $(\mathsf{D_V})$ we mean the strengthening of definition by numerical cases $\mathsf{d_N}$ on N to definition by cases $\mathsf{d_V}$ on the universes V:

$$(\mathsf{D_V}) \qquad (a = b \rightarrow \mathsf{d_V}xyab = x) \wedge (a \neq b \rightarrow \mathsf{d_V}xyab = y).$$

This finishes the description of the formal framework for applicative theories.

1.2 Basic consequences of the axioms

In this section we address some of the most important consequences of the axioms of BON. Furthermore, we collect some well-known inconsistency results concerning strengthenings of BON by (Tot), (Ext) and $(\mathsf{D_V})$, cf. [3, 9, 18, 20, 71].

1.2.1 Explicit definitions and recursion theorems

Let us start with the well-known notion of λ abstraction, which, due to the presence of k and s, is definable in BON.

Proposition 1 (Explicit definitions) *For each $\mathcal{L}_\mathsf{N}$ term t there exists an $\mathcal{L}_\mathsf{N}$ term $(\lambda x.t)$ whose free variables are those of t, excluding x, so that*

$$\mathsf{BON} \vdash (\lambda x.t)\downarrow \wedge (\lambda x.t)x \simeq t.$$

Proof. Define $(\lambda x.t)$ by induction on the complexity of t by: $(\lambda x.x) := \mathsf{skk}$; $(\lambda x.t) := \mathsf{k}t$ for t a variable different from x or a constant; $(\lambda x.t_1t_2) := \mathsf{s}(\lambda x.t_1)(\lambda x.t_2)$. $\Box$

In the following we write $\lambda x_1x_2\ldots x_n.t$ for $\lambda x_1.(\lambda x_2.(\ldots(\lambda x_n.t)\ldots))$.

Observe that our definition of λ abstraction is different from the well-known definition in the context of a *total* logic: there one usually defines $(\lambda x.t) := \mathsf{k}t$ if x does not occur in t; this definition, however, no longer guarantees $(\lambda x.t)\downarrow$ in a partial setting. As a consequence, the λ abstraction of our theorem behaves very badly with respect to substitutions, i.e. for $x \neq y$ and x not occurring in s we generally do not have

$$s\downarrow \to (\lambda x.t)[s/y] = \lambda x.t[s/y]. \qquad (*)$$

For example, $(\lambda x.y)[zz/y] = \mathsf{k}(zz)$, but $\lambda x.zz = \mathsf{s}(\mathsf{k}z)(\mathsf{k}z)$. The failure of $(*)$ is not really problematic, since $(\lambda x.t)[s/y]$ and $\lambda x.t[s/y]$ are easily shown to be *extensionally equal*, and this is actually enough for working in BON. For an extensive discussion about substitutions in partial applicative theories we refer to our paper Strahm [68]. A short overview of that paper is given in Section 1.6 of this chapter.

In a next step we give a partial and a total version of the recursion theorem. We begin with the fixed point theorem for BON.

Proposition 2 (Recursion theorem for BON) *There exists an $\mathcal{L}_\mathsf{c}$ term rec_p so that*

$$\mathsf{BON} \vdash \mathsf{rec}_p f\downarrow \wedge\ (\forall x)(\mathsf{rec}_p f x \simeq f(\mathsf{rec}_p f)x).$$

Proof. Choose rec_p as $\lambda f.(\lambda yz.f(yy)z)(\lambda yz.f(yy)z)$. $\Box$

Due to the presence of a total application operation, TON, i.e. $\mathsf{BON} + (\mathsf{Tot})$, proves a slightly stronger version of the recursion theorem, which literally corresponds to the fixed point theorem for the untyped lambda calculus. We will see in the next paragraph that this strengthening allows one to derive inconsistency results in the presence of totality.

Proposition 3 (Recursion theorem for TON) *There exists an $\mathcal{L}_\mathsf{c}$ term rec_t so that*

$$\mathsf{TON} \vdash \mathsf{rec}_t f = f(\mathsf{rec}_t f).$$

Proof. Choose rec_t as $\lambda f.(\lambda y.f(yy))(\lambda y.f(yy))$. $\Box$

As usual we get from both recursion theorems that the partial recursive functions are (numeralwise) representable by appropriate $\mathcal{L}_\mathsf{N}$ terms in BON and TON, respectively.

Corollary 4 *The partial recursive functions are (numeralwise) representable in* BON *and* TON.

At this point the reader may wonder why we did include the primitive recursor $\mathsf{r_N}$ in the theory BON: although such a combinator is definable by making use of the recursion theorem and its defining axiom (10) can be derived in BON, induction is required in order to establish the totality axiom (9) about $\mathsf{r_N}$; since we will be interested in very weak forms of complete induction on N that are not strong enough to derive (9), we claim the totality of $\mathsf{r_N}$ as an axiom of BON.

We finish this paragraph by giving an application of the recursion theorem in BON, which guarantees the existence of a combinator $\mathsf{not_N}$ that provably does not belong to N. This result is due to Kahle [53].

Proposition 5 *There exists an* $\mathcal{L}_\mathsf{N}$ *term* $\mathsf{not_N}$ *so that*

$$\mathsf{BON} \vdash \mathsf{not_N} \notin \mathsf{N}.$$

Proof. Choose $\mathsf{not_N}$ as $\mathsf{rec}_p(\lambda xy.\mathsf{d_N}10(xy)0)0$, and assume $\mathsf{not_N} \in \mathsf{N}$. Then easy calculations show that $\mathsf{not_N} = 0$ implies $\mathsf{not_N} = 1$, and $\mathsf{not_N} = 0$ follows from $\mathsf{not_N} \neq 0$. This is a contradiction and, therefore, $\mathsf{not_N} \notin \mathsf{N}$. □

1.2.2 Basic inconsistency results

In this paragraph we shortly address some well-known basic inconsistency results concerning extensions of BON and TON. The main tools to establish these inconsistencies are the recursion theorems.

In a first step we show that totality (Tot) is inconsistent with full definition by cases ($\mathsf{D_V}$).

Proposition 6 $\mathsf{TON} + (\mathsf{D_V})$ *is inconsistent.*

Proof. Let $s := \mathsf{rec}_t(\lambda x.\mathsf{d_V}10x0)$. Then easy calculations show that $s = 0$ implies $s = 1$, and $s \neq 0$ yields $s = 0$: contradiction. □

As an immediate consequence we get that totality plus "everything is a number" is inconsistent, too. Actually, this is also a corollary to Kahle's Proposition 5.

Corollary 7 $\mathsf{TON} + (\forall x)\mathsf{N}(x)$ *is inconsistent.*

In the following we see that not only totality (Tot), but also extensionality (Ext) is inconsistent with full definition by cases.

Proposition 8 $\mathsf{BON} + (\mathsf{Ext}) + (\mathsf{D_V})$ *is inconsistent.*

Proof. Let $s := \mathsf{rec}_p(\lambda yx.\mathsf{d_V}10y(\lambda z.0))$. Then we have $s\downarrow$ and for all x

$$sx \simeq \mathsf{d_V}10s(\lambda z.0).$$

Hence, if $s = (\lambda z.0)$, then $(\forall x)(sx = 1)$, which is impossible; therefore, $s \neq (\lambda z.0)$. But then $(\forall x)(sx = 0)$; this yields $s = (\lambda z.0)$ by extensionality (Ext): contradiction. □

Corollary 9 $\mathsf{BON} + (\mathsf{Ext}) + (\forall x)\mathsf{N}(x)$ *is inconsistent.*

By using similar techniques as above one can also show that both totality (Tot) and extensionality (Ext) are inconsistent with the following statements:

(i) there is a characteristic function for equality $=$ on the universe V;

(ii) there is an injective operation from the universe V into the numbers N.

Moreover, (Tot) is easily seen to be inconsistent with the statement

(iii) there is a characteristic function for the natural numbers N.

Is (Ext) inconsistent with (iii), too?

1.3 Models

In this section we briefly discuss some basic models of BON and its extensions. We mainly concentrate on recursion-theoretic and term models, since these will be important for our metamathematical investigations in the rest of this thesis.

1.3.1 Recursion-theoretic models

We give two recursion-theoretic models, namely one in terms of ordinary Σ^0_1 recursion theory and one in terms of Π^1_1 recursion theory.

Σ^0_1 recursion theory: the model *PRO*

The universe of the model *PRO* of partial recursive operations consists of the set of natural numbers $\mathbb{N}$. N is interpreted as $\mathbb{N}$ and application $\cdot$ as ordinary partial recursive function application, i.e. $x \cdot y$ means $\{x\}(y)$ in *PRO*, where $\{x\}$ is a standard enumeration of the partial recursive functions. It is easy to find appropriate interpretations of k and s such that the axioms of a partial combinatory algebra are satisfied, and the remaining constants of $\mathcal{L}_{\mathsf{N}}$ are straightforwardly interpreted, too. The sets in $\mathcal{P}(\mathsf{N})$ are exactly the recursive sets, and the total operations from N to N the total recursive functions. Observe that application in *PRO* is truly partial, hence $PRO \not\models (\mathsf{Tot})$. Moreover, $PRO \not\models (\mathsf{Ext})$ for trivial reasons, but $PRO \models (\mathsf{D_V})$, since $V = \mathsf{N}$ is true in *PRO*.

PRO provides a natural example of a domain where objects may be programs as well as inputs to programs. The model underlines the constructive and operational character of applicative theories.

Π^1_1 recursion theory: the model *ERO*

The model *ERO* is obtained from *PRO* by replacing ordinary partial recursive function application by application of partial Π^1_1 functions: these are obtained by Π^1_1 uniformization of a Π^1_1 enumeration of the Π^1_1 relations, cf. Rogers [57]. This interpretation of application is equivalent to partial recursive function application in Kleene's type two functional E, i.e., $x \cdot y$ means $\{x\}^E(y)$ in *ERO*, where

$$E(f) = \begin{cases} 0, & \text{if } (\exists x)(f(x) = 0), \\ 1, & \text{else,} \end{cases}$$

for f a total function from $\mathbb{N}$ to $\mathbb{N}$, cf. Hinman [42] for details. $\mathcal{P}(\mathsf{N})$ corresponds to the hyperarithemtic sets in this model, and the total operations from N to N are exactly the hyperarithmetic functions.

The model *ERO* of E recursive operations will be the standard model of the theory BON plus the so-called non-constructive μ operator, which we will study in detail in Chapter 4.

1.3.2 Term models

In the sequel we will address partial and total term models of BON and TON, respectively. For that purpose we need some general notions concerning reduction relations. We adopt the notation from Barendregt [2], pages 50ff.

A *notion of reduction* is just a binary relation R on the $\mathcal{L}_\mathsf{N}$ terms. If R_1 and R_2 are notions of reduction, then R_1R_2 denotes $R_1 \cup R_2$. A notion of reduction R induces the binary relation $\rightarrow_R$ of *one step R reduction* (the compatible closure of R) and the binary relation $\twoheadrightarrow_R$ of *R reduction* (the reflexive, transitive closure of $\rightarrow_R$).

The notion of reduction $\boldsymbol{\rho}$ that is appropriate for building models of our theories is just the usual notion of reduction for combinatory logic (cf. e.g. [2]) extended by reduction rules for the constants p, p_0, p_1, $\mathsf{s_N}$, $\mathsf{p_N}$, $\mathsf{d_N}$ and $\mathsf{r_N}$. The relation $\boldsymbol{\rho}$ is given by the following redex-contractum pairs, where t_0, t_1, t_2, s are $\mathcal{L}_\mathsf{N}$ terms and $m, n \in \mathbb{N}$ with $m \neq n$:

$$
\begin{array}{rcl}
\mathsf{k}t_0t_1 & \boldsymbol{\rho} & t_0, \\
\mathsf{s}t_0t_1t_2 & \boldsymbol{\rho} & t_0t_2(t_1t_2), \\
\mathsf{p}_0(\mathsf{p}t_0t_1) & \boldsymbol{\rho} & t_0, \\
\mathsf{p}_1(\mathsf{p}t_0t_1) & \boldsymbol{\rho} & t_1, \\
\mathsf{p_N}(\mathsf{s_N}\overline{m}) & \boldsymbol{\rho} & \overline{m}, \\
\mathsf{d_N}t_0t_1\overline{m}\,\overline{m} & \boldsymbol{\rho} & t_0, \\
\mathsf{d_N}t_0t_1\overline{m}\,\overline{n} & \boldsymbol{\rho} & t_1, \\
\mathsf{r_N}t_0t_1s\,\overline{0} & \boldsymbol{\rho} & t_0s, \\
\mathsf{r_N}t_0t_1s\,\overline{m+1} & \boldsymbol{\rho} & t_1s\,\overline{m}(\mathsf{r_N}t_0t_1s\,\overline{m}).
\end{array}
$$

Using the standard method of parallelization it is straightforward to prove that $\twoheadrightarrow_{\boldsymbol{\rho}}$ has the Church Rosser property (cf. e.g. [2]).

We start our discussion with *total* term models, since these are more general and familiar from the untyped λ calculus and combinatory logic.

The total term model CTT

The universe of CTT consists of all closed $\mathcal{L}_\mathsf{N}$ terms. The constants are interpreted by themselves, and term application means just juxtaposition of terms in CTT. Crucial is the treatment of equality: two $\mathcal{L}_\mathsf{N}$ terms are equal in CTT if and only if they have a common reduct with respect to $\twoheadrightarrow_{\boldsymbol{\rho}}$. Moreover, the natural numbers in CTT are those $\mathcal{L}_\mathsf{N}$ terms which reduce to some numeral $\overline{n}$ with respect to $\twoheadrightarrow_{\boldsymbol{\rho}}$. Indeed, CTT is a model of TON, thanks to the Church Rosser property of $\twoheadrightarrow_{\boldsymbol{\rho}}$. Moreover, it is not difficult to see that $CTT \not\models (\mathsf{Ext})$, cf. Beeson [3] for details. We also have $CTT \not\models (\mathsf{D_V})$ by definition, actually, by Proposition 6 we know that there exists no model of $\mathsf{TON} + (\mathsf{D_V})$.

An extensional version TTE of CTT can be obtained by considering the term model of the $\lambda\eta$ calculus (extended by reduction rules for the additional constants of $\mathcal{L}_\mathsf{N}$) and using the standard translation of combinatory logic into λ calculus. In the context of extensionality, the universe of a total term model consists of *all* terms of $\mathcal{L}_\mathsf{N}$ and not only the closed $\mathcal{L}_\mathsf{N}$ terms, of course.

In Chapter 4 we will define and study the term model analogue $CTT(\mu)$ of *ERO*. $CTT(\mu)$ is a so-called *infinitary* term model, which satisfies the axioms of the non-constructive μ operator as well as totality (Tot).

The normal term model ***CNT***

The universe of *CNT* consists of all closed $\mathcal{L}_\mathsf{N}$ terms in *normal form*, the constants of $\mathcal{L}_\mathsf{N}$ are interpreted by themselves, and $t_1 \cdot t_2$ means $\mathit{InFirst}(t_1 t_2)$; here $\mathit{InFirst}(t_1 t_2)$ denotes the uniquely determined normal term s provided that $t_1 t_2$ can be reduced to s according to the *leftmost minimal strategy*, $\mathit{InFirst}(t_1 t_2)$ is undefined otherwise. Using the leftmost minimal strategy, at each stage of a reduction sequence the leftmost minimal redex is contracted, where a redex is called *minimal*, if it does not contain any other redexes. It is necessary to use the leftmost minimal strategy in order to be consistent with the strictness axioms of BON. Finally, equality means identity in *CNT* and the natural numbers N are exactly the numerals of $\mathcal{L}_\mathsf{N}$. Obviously, $CNT \not\models$ (Tot), and also $CNT \not\models$ (Ext), cf. [3]. Moreover, it is easy to see that a slight modification of *CNT* satisfies ($\mathsf{D_V}$).

1.3.3 Continuous and other models

We finish this section by mentioning some *continuous models* of our theories, without giving their definition. Among them are Scott's D_∞, Plotkin's $P\omega$, Engeler's D_A, Barendregt's Böhm tree model, and Scott's information systems; for details and references, cf. [2, 3, 41, 71]. All these models satisfy the axiom of totality (Tot). Another interesting class of partial models are the so-called *generated models* of Feferman [20]; we will see a particular example of such a model construction in Chapter 3, where we will discuss polynomial time operations in applicative theories.

1.4 Complexity issues in models

In this section we address some complexity issues in models of TON, i.e. models with a *total* application operation. In particular, we are interested in the complexity of *application*, *equality*, and *natural numbers*.

There seems to be a striking difference in the complexities of these relations, depending on whether the application relation is partial or total: in the partial models *PRO* and *CNT* discussed in the last section, application is truly r.e., whereas the natural numbers and equality have trivial complexity; this is in sharp contrast to the total term model CTT, where application is trivial, but the natural numbers and equality are undecidable. Similar remarks apply to effective versions of continuous models. The question arises whether the above relations are generally undecidable in an *arbitrary* model of TON. In the following we show that if $\mathfrak{M}$ is a model of TON that can be described in $\mathbb{N}$, then equality and natural numbers are undecidable in $\mathfrak{M}$ as long as application is numeralwise definable in $\mathfrak{M}$. In particular, if application is r.e., then equality and numbers are undecidable.

In order to be able to discuss undecidability issues more easily, let us introduce the notion of a number-theoretic interpretation of $\mathcal{L}_\mathsf{N}$. The recursion-theoretic models and term models obviously fit this definition in a straightforward way.

Definition 10 A *number-theoretic interpretation* of $\mathcal{L}_\mathsf{N}$ is a structure

$$\mathfrak{M} = \langle U, App, Eq, Nat, \hat{\mathsf{k}}, \hat{\mathsf{s}}, \ldots, \hat{\mathsf{r}}_\mathsf{N} \rangle,$$

so that the following conditions are satisfied: (i) $Nat \subset U \subset \mathbb{N}$; (ii) $Eq \subset U^2$; (iii) App is a partial function from U^2 to U; (iv) $\hat{c} \in U$ for all constants c of $\mathcal{L}_\mathsf{N}$.

It is clear what it means for a number-theoretic interpretation to be a model of BON or TON. The notion of numeralwise definability within a number-theoretic interpretation $\mathfrak{M}$ is crucial for the theorem to be proved below.

Definition 11 Let $\mathfrak{M} = \langle U, \ldots \rangle$ be a number-theoretic interpretation of $\mathcal{L}_\mathsf{N}$. A function $F : U^n \to U$ is called *numeralwise definable in* $\mathfrak{M}$, if there exists an $f \in U$ so that

$$f\overline{m}_1 \ldots \overline{m}_n = \overline{F(m_1, \ldots, m_n)}$$

holds in $\mathfrak{M}$ for all $m_1, \ldots, m_n \in U$.

The following theorem is a generalization of the Scott-Curry undecidability theorem to arbitrary, not necessarily term models. Its proof very closely follows the one given in Hindley and Seldin [41].

Theorem 12 *Let* $\mathfrak{M} = \langle U, App, Eq, Nat, \ldots \rangle$ *be a number-theoretic model of* TON *so that App is numeralwise definable in* $\mathfrak{M}$*. Assume further that A and B are sets of natural numbers so that (i) $A, B \subset U$; (ii) $A \neq \emptyset, B \neq \emptyset$; (iii) A, B are closed under equality Eq in* $\mathfrak{M}$*. Then A and B are recursively inseparable.*

Proof. Let us assume that the hypothesis (i)–(iii) hold and assume by contradiction that A and B are recursively separable. Hence, there exists a recursive function E so that we have for all natural numbers n:

$$E(n) = 0 \lor E(n) = 1; \quad n \in A \Rightarrow E(n) = 1; \quad n \in B \Rightarrow E(n) = 0. \qquad (1)$$

Since $\mathfrak{M} \models$ TON and the recursive functions are representable in TON, there exists an $e \in U$ which numeralwise defines E in $\mathfrak{M}$. In the following we work in the model

$\mathfrak{M}$. Since *App* is numeralwise definable in $\mathfrak{M}$, there exists an $f \in U$ so that we have for all $m, n \in U$:

$$f\overline{m}\,\overline{n} = \overline{mn}. \tag{2}$$

Moreover, there is a $g \in U$ satisfying $g\overline{n} = \overline{\overline{n}}$ for all natural numbers n; the operation g exists in $\mathfrak{M}$, since $\mathfrak{M}$ is closed under recursion, and we have

$$g\overline{0} = \overline{\overline{0}}; \quad g\overline{n+1} = f\overline{\mathsf{s_N}}(g\overline{n}). \tag{3}$$

Since A and B are non-empty, there exists an $a \in A$ and a $b \in B$. Further, let h and j in U be given by:

$$h \;:=\; \lambda x.\mathsf{d_N} ab(e(fx(gx)))\mathsf{0}; \quad j \;:=\; h\overline{h}. \tag{4}$$

From the definition of j, (2) and (3) we get

$$j = \mathsf{d_N} ab(e(f\overline{h}(g\overline{h})))\mathsf{0} = \mathsf{d_N} ab(e(\overline{h\overline{h}}))\mathsf{0} = \mathsf{d_N} ab(e\overline{j})\mathsf{0}. \tag{5}$$

Assume now $E(j) = 0$. Then we have $e\overline{j} = \mathsf{0}$ and, hence, by (5) $j = a$. Since A is closed under equality, this implies $j \in A$: this is a contradiction, since now $E(j) = 1$ by (1). Analogously, we get a contradiction by assuming $E(j) = 1$. Since $E(j) = 0$ or $E(j) = 1$, we get an overall contradiction and, hence, A and B are recursively inseparable. □

Corollary 13 *Let* $\mathfrak{M} = \langle U, App, Eq, Nat, \ldots\rangle$ *be a number-theoretic model of* TON *so that App is numeralwise definable in* $\mathfrak{M}$. *Then the sets* $\{x \in U : Nat(x)\}$ *and* $\{(x, y) \in U^2 : Eq(x, y)\}$ *are not recursive.*

Proof. Assume that $\{x \in U : Nat(x)\}$ is recursive. Then $\{x \in U : Nat(x)\}$ and $\{x \in U : \neg Nat(x)\}$ are recursively separable. Observe that by Proposition 7, the second of these sets is non-empty. Furthermore, both sets are closed under equality *Eq*, since $\mathfrak{M} \models$ TON. All together, we have a contradiction to our theorem. The argument for $\{(x, y) \in U^2 : Eq(x, y)\}$ is similar. □

Corollary 14 *Let* $\mathfrak{M} = \langle U, App, Eq, Nat, \ldots\rangle$ *be a number-theoretic model of* TON *so that the graph of App is r.e. Then* $\{x \in U : Nat(x)\}$ *and* $\{(x, y) \in U^2 : Eq(x, y)\}$ *are not recursive.*

Proof. If the graph of *App* is r.e., then *App* is a partial recursive function and, hence, numeralwise definable in every model of TON. □

The question arises whether the result of the last corollary is sharp. In particular, is there a number-theoretic model of TON so that *App* is non r.e. but *Eq* and *Nat* are decidable? Moreover, is Corollary 13 indeed more general than Corollary 14?

1.5 Finitary inductive data types

In the following we indicate how to replace the natural numbers $\mathbb{N}$ by an arbitrary finitary inductive data type $\mathbb{D}$ in the formulation of our applicative theories.

Let us briefly recall the basic notions, cf. [72]. A *finitary inductive data type* $\mathbb{D}$ is generated from a finite set of *atoms* a_i $(i \in I)$, and a finite set of *constructors* c_j of arity n_j $(j \in J)$ as follows:

(1) $a_i \in \mathbb{D} \quad (i \in I)$;

(2) $x_1, \ldots, x_{n_j} \in \mathbb{D} \Rightarrow c_j(x_1, \ldots, x_{n_j}) \in \mathbb{D} \quad (j \in J)$.

The corresponding principle of $\mathbb{D}$ *induction* has the form

$$\frac{\{A(a_i)\}_{i \in I} \quad \{A(x_1) \wedge \cdots \wedge A(x_{n_j}) \;\Rightarrow\; A(c_j(x_1, \ldots, x_{n_j}))\}_{j \in J}}{(\forall x \in \mathbb{D}) A(x)},$$

and the scheme for function definition by *primitive recursion on* $\mathbb{D}$ has the following obvious clauses:

(1) $f(\vec{y}, a_i) = g_i(\vec{y}) \quad (i \in I)$;

(2) $f(\vec{y}, c_j(x_1, \ldots, x_{n_j})) = h_j(\vec{y}, x_1, \ldots, x_{n_j}, f(\vec{y}, x_1), \ldots, f(\vec{y}, x_{n_j})) \quad (j \in J)$.

It is easily seen that the natural numbers, lists, finitely branching trees, etc. fit the definition of an inductive date type $\mathbb{D}$.

A formulation of our applicative theories which is based on $\mathbb{D}$ instead on $\mathbb{N}$ is obtained in a straightforward manner as follows. The language $\mathcal{L}_{\mathsf{D}}$ extends $\mathcal{L}_{\mathsf{c}}$ by a unary relation symbol D, constants a_i for the atoms of $\mathbb{D}$, and constants c_j for the constructors of $\mathbb{D}$. Moreover, one adds for each constructor c_j corresponding deconstructor operations $\bar{\mathsf{c}}_{j,1}, \ldots, \bar{\mathsf{c}}_{j,n_j}$, and $\mathcal{L}_{\mathsf{D}}$ includes constants d_{D} for definition by cases on D, and r_{D} for primitive recursion on D. The respective axioms are formulated in the obvious way.

It is well-known that a finitary inductive date type $\mathbb{D}$ can be embedded into $\mathbb{N}$ by means of standard sequence coding. Based on this, it is straightforward to provide an interpretation of a partial applicative theory based on $\mathbb{D}$ in terms of ordinary recursion theory, i.e. we obtain an analogue of the model *PRO* for such a theory. Things get slightly more involved if we turn to systems based on $\mathbb{D}$ and including the axiom of totality (Tot). Here one can consider a modification of the term model *CTT* which includes suitable reductions for terms involving the date type $\mathbb{D}$, e.g.

$$\bar{\mathsf{c}}_{j,i}\mathsf{c}_j(t_1, \ldots, t_{n_j}) \;\boldsymbol{\rho}\; t_i \qquad (1 \leq i \leq n_j).$$

It is a matter of routine to establish the Church Rosser property for such a modified reduction relation.

Based on the above ideas, we will see in the next chapter that proof-theoretic upper bounds for theories with $\mathbb{N}$ also provide upper bounds for systems with a finitary inductive data type $\mathbb{D}$. In general, applicative theories based on $\mathbb{N}$ are proof-theoretically equivalent to those based on $\mathbb{D}$ as long as we have *unbounded* primitive recursors in our systems. The situation changes drastically if we consider applicative theories with a *bounded* primitive recursor built in. Such systems are generally not equivalent for all inductive data types $\mathbb{D}$. In Chapter 3 we will study an applicative theory for polynomial time operations based on the data type $\mathbb{W}$ of finite 0-1 sequences; the corresponding theory based on $\mathbb{N}$ is very likely to be of minor proof-theoretic strength, cf. the discussion in Section 3.5.3.

It has to be mentioned that a completely satisfactory treatment of inductively generated data types in explicit mathematics can only be achieved by considering appropriate flexible type theories above applicative theories. The reader is referred to Feferman [24, 25, 26, 27] for further information.

This finishes our short discussion on finitary inductive data types. We have seen that $\mathbb{N}$ plays a "universal" role among all inductive data types $\mathbb{D}$ and, hence, it is perfectly justified to stick to theories based on $\mathbb{N}$ in the sequel. As already mentioned, an exception will be our polynomial time applicative theory which will be based on the set $\mathbb{W}$ of finite 0-1 words.

1.6 Explicit substitutions

In this section we give a brief review of our paper [68] on partial applicative theories and explicit substitutions. We try to discuss the basic ideas rather than giving precise definitions and proofs of theorems.

It is a reasonable question to ask whether the axioms for a partial combinatory algebra of BON can be replaced by some form of the partial λ calculus in such a manner that BON and its partial λ version are mutually interpretable into each other so that the relevant models are preserved in a natural way. It is well-known that this question has a positive answer as long as the underlying logic is *total*, however, the situation changes drastically if one considers a *partial* application operation as it is our case with BON.

A straightforward λ analogue BON_λ of BON is obtained by allowing the possibility for building λ terms, and by replacing the axioms of a partial combinatory algebra by the two axioms

$$(\lambda x.t)\downarrow \qquad (\lambda x.t(x))y \simeq t(y).$$

Now the main problem encountered is that substitution in BON_λ is defined as usual in the metalanguage and, hence, the substitution property $(*)$ of Section 1.2.1 is satisfied for trivial reasons. On the other hand, we have seen that $(*)$ fails for the encoding of λ abstraction in partial combinatory logic and, therefore, the straightforward embedding of BON_λ into BON fails. As a consequence, the models *PRO* and *CNT* of BON do not readily carry over to models of BON_λ.

The question arises whether it is possible to find another encoding of λ in BON or our model *PRO* satisfying the substitution property $(*)$. Elena Pezzoli has found an elegant formal argument showing that the existence of a recursion-theoretic interpretation of BON_λ which has a partial recursive term evaluation function contradicts the undecidability of the halting problem, answering the above question in the negative. The proof of her theorem is included in [68].

Theorem 15 *It is not possible to make the partial recursive functions model PRO of* BON *into a model of* BON_λ *in such a way that the term evaluation function* $f(t, \alpha) \simeq \|t\|_\alpha$*, for* t *a* λ *term and* α *an assignment of variables, is partial recursive.*

The following corollary is immediate from the fact that *PRO* is a model of BON.

Corollary 16 *There is no recursive encoding of* λ *in* BON *validating* $(*)$.

We have seen that a stronger concept of substitution in the partial λ calculus makes its embedding into partial combinatory logic fail and, therefore, *PRO* is not preserved as a model of BON_λ. The deep reasons for the failure of this embedding have to be seen in the fact that pushing a substitution inside an abstraction is not consistent with a strongly *intensional* point of view. Indeed, the problems described above completely disappear in the presence of the extensionality axiom (Ext). In particular, $(*)$ holds in $\mathsf{BON} + (\mathsf{Ext})$. For further discussion on this point we refer again to [68].

In the following we briefly sketch a modified version $\mathsf{BON}_{\lambda\sigma}$ of BON_λ, which is equivalent to BON and, therefore, enjoys a recursion-theoretic interpretation. The novel point of $\mathsf{BON}_{\lambda\sigma}$ is the use of *explicit substitutions*. Accordingly, substitution is no longer a notion of the metalanguage, but an operation axiomatized in the theory under consideration itself. To be more precise, a substitution θ is a finite set $\{t_1/x_1, \dots, t_n/x_n\}$ so that the x_i are distinct variables and the t_i are terms; those are defined simultaneously with the substitutions by closing under application, λ abstraction, and substitution application $t\theta$ for arbitrary substitutions θ. Note that $t\theta$ is a purely syntactical object, which can only be evaluated by means of appropriate axioms to be specified now. The extended β axiom of $\mathsf{BON}_{\lambda\sigma}$ reads as

$$(\lambda x.t)\theta y \simeq t(y/x \cdot \theta),$$

where $(y/x \cdot \theta)$ is $\{y/x\} \cup \theta^{-x}$ and θ^{-x} is θ with any binding for x deleted. Other important axioms of $\mathsf{BON}_{\lambda\sigma}$ are the so-called substitution axioms; they allow a careful evaluation of substitutions in the following way:

$$\begin{array}{rcll} x\theta & \simeq & t & (t/x \in \theta) \\ (ts)\theta & \simeq & (t\theta)(s\theta) & \\ (t\theta)\sigma & \simeq & t(\theta\sigma) & \\ t(s/x \cdot \theta) & \simeq & t\theta & (x \notin \mathit{fvar}(t) \cup \mathit{dom}(\theta)) \\ t\varepsilon & \simeq & t. & \end{array}$$

Here $(\theta\sigma)$ denotes the composition of θ and σ, $fvar(t)$ the free variables of t, and $dom(\theta)$ the domain of the substitution θ; ε is the empty substitution. For precise definitions see [68]. It is crucial to observe that among our substitution axioms we do not have an axiom which allows us to push a substitution θ inside an abstraction $(\lambda x.t)$. This is exactly what we want to prevent. Terms of the form $(\lambda x.t)\theta$ can only be resolved if applied to another object, say y. This is reflected in the extended β axiom, where an interleaving substitution θ is allowed. If θ is the empty substitution ε, then we have the usual β axiom.

The following theorem states in a compact form the results which are proved in [68].

Theorem 17 1. *The theories $\mathsf{BON}_{\lambda\sigma}$ and BON are mutually interpretable into each other; in particular, $\mathsf{BON}_{\lambda\sigma}$ enjoys the standard recursion-theoretic interpretation PRO.*

2. *There is a suitable term rewriting system for $\mathsf{BON}_{\lambda\sigma}$ which has the Church Rosser property and whose substitution fragment is strongly normalizing.*

Let us mention that the theory of explicit substitutions has been treated in the literature before, but from a different point of view. The main work has been done in the context of implementation of functional programming languages, and application in those systems is always *total*. The very concern of the our work, however, is to study a *partial* application operation. A key reference for the previous work on explicit substitution is the paper by Abadi, Cardelli, Curien and Lévy [1].

We finish this section by mentioning some recent work by Robert Stärk. In [64] he establishes a very natural relationship between the number-free fragment of $\mathsf{BON}_{\lambda\sigma}$ and the programming language SCHEME. Moreover, in [65] he discusses a system LV, which is a partial form of the call-by-value λ calculus and is shown to be equivalent to partial combinatory logic. Although LV avoids explicit substitutions, it shows some drawbacks with respect to extensions to theories of operations and numbers.

Chapter 2

Basic proof theory of applicative theories

In this chapter we give a complete proof-theoretic characterization of BON in the presence of various induction principles on the natural numbers, and including the axioms of totality (Tot) and extensionality (Ext), cf. Jäger and Strahm [52]. We obtain various applicative theories of strength PRA and PA. Although the results concerning systems of strength PA are well-known (cf. e.g. [71]), we include them for reasons of completeness. The delineation of systems of strength PRA including totality and extensionality is independently due to Cantini [9]. Finally, we mention that corresponding results for applicative theories *without* totality and extensionality are established in Feferman [26, 27].

In the first section we introduce a bunch of induction principles on the natural numbers, and we briefly discuss their logical relationship. Section 2 contains a short definition of first order arithmetic and some subsystems; we mention Parson's result. In Section 3 we discuss various systems of strength PRA. In particular, we obtain upper bounds for such systems by means of formalized term model constructions; these are introduced in a rather general form, since we will use them again in Chapter 4, where we will discuss infinitary term models that are based on the non-constructive μ operator. Hence, the considerations of this section must also be seen as a preparation for the work in Chapter 4. Finally, in Section 4 we see that our methods extend to systems of strength PA in a straightforward manner.

2.1 Induction principles on the natural numbers

In this section we introduce various induction principles on the natural numbers N, whose exact proof-theoretic strength over BON will be determined in the sequel.

Let us start by defining the *positive* and the *negative* formulas of $\mathcal{L}_{\mathsf{N}}$ by a simultaneous inductive definition as follows.

Definition 18 (F^+ **and** F^- **formulas)**

1. Each atomic formula $\mathsf{N}(t)$, $t\downarrow$ and $(s = t)$ is an F^+ formula.
2. If A is an F^+ $[\mathsf{F}^-]$ formula, then $\neg A$ is an F^- $[\mathsf{F}^+]$ formula.
3. If A and B are F^+ $[\mathsf{F}^-]$ formulas, then $(A \vee B)$ and $(A \wedge B)$ are F^+ $[\mathsf{F}^-]$ formulas.
4. If A is an F^- $[\mathsf{F}^+]$ formula and B is an F^+ $[\mathsf{F}^-]$ formula, then $(A \to B)$ is an F^+ $[\mathsf{F}^-]$ formula.
5. If A is an F^+ $[\mathsf{F}^-]$ formula, then $(\exists x \in \mathsf{N})A$ and $(\forall x \in \mathsf{N})A$ are F^+ $[\mathsf{F}^-]$ formulas.

The Σ^+ and Π^- formulas of $\mathcal{L}_\mathsf{N}$ are defined in the very same way with the only exception that clause 5. is replaced by

5'. If A is a Σ^+ $[\Pi^-]$ formula, then $(\exists x \in \mathsf{N})A$ $[(\forall x \in \mathsf{N})A]$ is a Σ^+ $[\Pi^-]$ formula.

In the following we are interested in six forms of complete induction on the natural numbers, namely set induction, operation induction, N induction, Σ^+ induction, positive formula induction, and full formula induction.

Set induction on N, $(\mathsf{S}\text{-}\mathsf{I}_\mathsf{N})$.

$$f \in \mathcal{P}(\mathsf{N}) \wedge f0 = 0 \wedge (\forall x \in \mathsf{N})(fx = 0 \to f(x') = 0) \to (\forall x \in \mathsf{N})(fx = 0).$$

Operation induction on N, $(\mathsf{O}\text{-}\mathsf{I}_\mathsf{N})$.

$$f0 = 0 \wedge (\forall x \in \mathsf{N})(fx = 0 \to f(x') = 0) \to (\forall x \in \mathsf{N})(fx = 0).$$

N induction on N, $(\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$.

$$f0 \in \mathsf{N} \wedge (\forall x \in \mathsf{N})(fx \in \mathsf{N} \to f(x') \in \mathsf{N}) \to (\forall x \in \mathsf{N})(fx \in \mathsf{N}).$$

Σ^+ induction on N, $(\Sigma^+\text{-}\mathsf{I}_\mathsf{N})$. For all Σ^+ formulas $A(x)$ of $\mathcal{L}_\mathsf{N}$:

$$A(0) \wedge (\forall x \in \mathsf{N})(A(x) \to A(x')) \to (\forall x \in \mathsf{N})A(x).$$

Positive formula induction on N, $(\mathsf{F}^+\text{-}\mathsf{I}_\mathsf{N})$. The above scheme for all F^+ formulas.

Formula induction on N, $(\mathsf{F}\text{-}\mathsf{I}_\mathsf{N})$. The above scheme for all formulas of $\mathcal{L}_\mathsf{N}$.

Observe that it is trivial from the definitions that

$$(\mathsf{S}\text{-}\mathsf{I}_\mathsf{N}) \subset (\mathsf{O}\text{-}\mathsf{I}_\mathsf{N}) \subset (\Sigma^+\text{-}\mathsf{I}_\mathsf{N}) \subset (\mathsf{F}^+\text{-}\mathsf{I}_\mathsf{N}) \subset (\mathsf{F}\text{-}\mathsf{I}_\mathsf{N}) \quad \text{and} \quad (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N}) \subset (\Sigma^+\text{-}\mathsf{I}_\mathsf{N}).$$

Moreover, the following non-trivial result is proved in Kahle [53].

Proposition 19 $(\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$ *implies* $(\mathsf{S}\text{-}\mathsf{I}_\mathsf{N})$ *over* BON.

The exact relationship between $(\mathsf{O}\text{-}\mathsf{I}_\mathsf{N})$ and $(\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$ is much more delicate, and it seems that none of the two is implied by the other one over BON. Kahle [53] discusses a strengthening $\widehat{\mathsf{BON}}$ of BON so that $(\mathsf{O}\text{-}\mathsf{I}_\mathsf{N})$ and $(\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$ are equivalent over $\widehat{\mathsf{BON}}$, and he shows that $(\mathsf{O}\text{-}\mathsf{I}_\mathsf{N})$ and $(\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$ are even equivalent over BON in the presence of the quantification operator μ (cf. Proposition 81). This last result will be used in Chapter 4, where we will give a detailed discussion of μ. Nevertheless, we will see in the next section that $(\mathsf{O}\text{-}\mathsf{I}_\mathsf{N})$ and $(\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$ are *proof-theoretically* equivalent over BON.

2.2 Some systems of arithmetic

Let us briefly fix some terminology concerning well-known subsystems of Peano arithmetic PA.

In the following let $\mathcal{L}_1$ be the usual first order language of arithmetic with number variables $u, v, w, x, y, z, f, g, h, \ldots$ (possibly with subscripts), the constant 0, as well as function and relation symbols for all primitive recursive functions and relations. The number terms of $\mathcal{L}_1$ $(r, s, t, r_1, s_1, t_1, \ldots)$ are defined as usual.

We will use standard notation for coding sequences of natural numbers: $\langle \ldots \rangle$ is a primitive recursive function for forming n tuples $\langle t_0, \ldots, t_{n-1} \rangle$; *Seq* denotes the primitive recursive set of sequence numbers; $lh(t)$ gives the length of the sequence coded by t, i.e. if $t = \langle t_0, \ldots, t_{n-1} \rangle$ then $lh(t) = n$; $(t)_i$ denotes the ith component of the sequence coded by t if $i < lh(t)$. Furthermore, $\dot{-}$ is the usual primitive recursive cut-off difference on the naturals.

As usual PA denotes the system of Peano arithmetic formulated in $\mathcal{L}_1$: PA includes defining axioms for all primitive recursive functions and relations as well as all instances of complete induction on the natural numbers. PRA is the system of primitive recursive arithmetic and is obtained from PA by restricting induction to quantifier free $\mathcal{L}_1$ formulas. It is well-known from Parsons [55] that PRA is proof-theoretically equivalent to $\mathsf{PRA} + (\Sigma^0_1\text{-}\mathsf{I_N})$, i.e. the subsystem of PA with induction restricted to Σ^0_1 formulas; as usual, for B a quantifier free formula, A is called Σ^0_1 if it has the form $(\exists x)B$, and it is Π^0_2 if it has the shape $(\forall x)(\exists y)B$.

Proposition 20 $\mathsf{PRA}+(\Sigma^0_1\text{-}\mathsf{I_N})$ *is a conservative extension of* PRA *for* Π^0_2 *statements.*

2.3 Systems of strength PRA

In this section we show that the induction principles $(\mathsf{S\text{-}I_N})$, $(\mathsf{O\text{-}I_N})$, $(\mathsf{N\text{-}I_N})$, and $(\Sigma^+\text{-}\mathsf{I_N})$ over BON always yield systems of proof-theoretic strength PRA. More precisely, we establish lower bounds for the partial theories, i.e. systems based on BON, and upper bounds for systems including the axiom of totality (Tot). Moreover, we will always see that the methods for proving upper bounds for total applicative theories easily extend to systems including the axiom of extensionality (Ext).

2.3.1 Lower bounds

We obtain a natural embedding of the language $\mathcal{L}_1$ of arithmetic into the language $\mathcal{L}_\mathsf{N}$ as follows: the number variables of $\mathcal{L}_1$ are supposed to range over N; symbols for primitive recursive functions translate into their corresponding $\mathcal{L}_\mathsf{N}$ terms using the recursion operator $\mathsf{r_N}$ so that their totality is derivable in BON. Summing up, the translation $(\cdot)^\mathsf{N}$ from $\mathcal{L}_1$ to $\mathcal{L}_\mathsf{N}$ is such that

$$((\exists x)A(x))^\mathsf{N} = (\exists x \in \mathsf{N})A^\mathsf{N}(x),$$

and similarly for universal quantifiers. Moreover, it is straightforward to establish that each quantifier free formula of $\mathcal{L}_1$ can be represented in $\mathcal{L}_\mathsf{N}$ by a set in the sense of $\mathcal{P}(\mathsf{N})$ as follows.

Lemma 21 *For every quantifier free formula $A(\vec{x})$ of $\mathcal{L}_1$ with at most $\vec{x}$ free there exists an individual term t_A of $\mathcal{L}_\mathsf{N}$ so that*

1. $\mathsf{BON} \vdash (\forall \vec{x} \in \mathsf{N})(t_A \vec{x} = \mathsf{0} \vee t_A \vec{x} = \mathsf{1})$,
2. $\mathsf{BON} \vdash (\forall \vec{x} \in \mathsf{N})(A^\mathsf{N}(\vec{x}) \leftrightarrow t_A \vec{x} = \mathsf{0})$.

It is immediate from this lemma that quantifier free induction of PRA translates into set induction (S-$\mathsf{I_N}$) and, hence, we can state the following embedding.

Proposition 22 *We have for every $\mathcal{L}_1$ formula $A(\vec{x})$ with at most $\vec{x}$ free:*

$$\mathsf{PRA} \vdash A(\vec{x}) \quad \Longrightarrow \quad \mathsf{BON} + (\mathsf{S}\text{-}\mathsf{I_N}) \vdash \vec{x} \in \mathsf{N} \to A^\mathsf{N}(\vec{x}).$$

A fortiori, we have that PRA is contained in $\mathsf{BON} + (\mathsf{O}\text{-}\mathsf{I_N})$ and $\mathsf{BON} + (\Sigma^+\text{-}\mathsf{I_N})$. Moreover, $\mathsf{BON} + (\mathsf{N}\text{-}\mathsf{I_N})$ contains PRA by Kahle's Proposition 19.

2.3.2 Upper bounds

In the sequel we give an interpretation of $\mathsf{TON} + (\Sigma^+\text{-}\mathsf{I_N})$ into $\mathsf{PRA} + (\Sigma^0_1\text{-}\mathsf{I_N})$. The interpretation is based on a formalization of the total term model CTT that we have described in Section 1.3.2. We will see that a slight adaptation of our interpretation actually yields an upper bound for the system $\mathsf{TON} + (\mathsf{Ext}) + (\Sigma^+\text{-}\mathsf{I_N})$, too. At the end of this section we indicate how this last result may even further be strengthened.

In order to formalize term models we need a Gödelnumbering of the closed terms of the language $\mathcal{L}_\mathsf{N}$. Therefore, let us assign to each constant c of $\mathcal{L}_\mathsf{N}$ and the application symbol $\cdot$ natural numbers $\ulcorner c \urcorner$ and $\ulcorner \cdot \urcorner$ in some appropriate way. In particular, $\ulcorner c \urcorner$ and $\ulcorner \cdot \urcorner$ must not be elements of *Seq*. The Gödelnumber of a compound term (st) is then given in the obvious way by

$$\ulcorner st \urcorner = \langle \ulcorner \cdot \urcorner, \ulcorner s \urcorner, \ulcorner t \urcorner \rangle.$$

In the following, $CTer(x)$ denotes the primitive recursive predicate expressing that x is the Gödelnumber of a closed term of $\mathcal{L}_\mathsf{N}$. If $\vec{x} = x_1, \dots, x_n$ then we often write $CTer(\vec{x})$ instead of $CTer(x_1) \wedge \cdots \wedge CTer(x_n)$. Furthermore, let $Num : \mathbb{N} \to \mathbb{N}$ be the primitive recursive function satisfying $Num(x) = \ulcorner \overline{x} \urcorner$, i.e. $Num(x)$ is the Gödelnumber of the xth numeral of $\mathcal{L}_\mathsf{N}$.

Since we are going to use formalized term models again in Chapter 4, we introduce them in a rather general form right from the beginning. For that purpose, let us assume that R is a notion of reduction on the closed $\mathcal{L}_\mathsf{N}$ terms. In the sequel we will need formalized versions of R, $\to_R$ and $\twoheadrightarrow_R$, respectively, on the Gödelnumbers of closed terms of $\mathcal{L}_{\mathsf{N}}$. Therefore, let $\mathcal{L}$ be a first order language containing $\mathcal{L}_1$ and let $\mathit{RedCon}_R(x, y)$ be an $\mathcal{L}$ formula formalizing R. Then the formalized version $\mathit{Red1}_R(x, y)$ of $\rightarrow_R$ can be described by the following primitive recursive (in RedCon_R) definition:

$$\mathit{Red1}_R(x, y) \;:=\; \mathit{CTer}(x) \wedge \mathit{CTer}(y) \wedge \mathit{Red1}^*_R(x, y),$$

where $\mathit{Red1}^*_R(x, y)$ is the disjunction of the following formulas:

(1) $\mathit{RedCon}_R(x, y)$,

(2) $x = \langle \ulcorner \cdot \urcorner, (x)_1, (x)_2 \rangle \;\wedge\; y = \langle \ulcorner \cdot \urcorner, (x)_1, (y)_2 \rangle \;\wedge\; \mathit{Red1}_R((x)_2, (y)_2)$,

(3) $x = \langle \ulcorner \cdot \urcorner, (x)_1, (x)_2 \rangle \;\wedge\; y = \langle \ulcorner \cdot \urcorner, (y)_1, (x)_2 \rangle \;\wedge\; \mathit{Red1}_R((x)_1, (y)_1)$.

In order to formalize the reflexive, transitive closure $\twoheadrightarrow_R$ of $\rightarrow_R$ one defines an intermediate predicate $\mathit{RedSeq}_R(x, y, z)$ with the intended meaning that x codes a reduction sequence from the closed term with Gödelnumber y to the closed term with Gödelnumber z with respect to R:

$$\mathit{RedSeq}_R(x, y, z) \;:=\; \mathit{Seq}(x) \wedge \mathit{CTer}(y) \wedge \mathit{CTer}(z) \wedge \mathit{RedSeq}^*_R(x, y, z),$$

where $\mathit{RedSeq}^*_R(x, y, z)$ is the disjunction of the following formulas:

(1) $\mathit{lh}(x) = 1 \;\wedge\; x = \langle y \rangle \;\wedge\; y = z$,

(2) $\mathit{lh}(x) > 1 \;\wedge\; y = (x)_0 \;\wedge\; z = (x)_{\mathit{lh}(x) \dot{-} 1} \;\wedge\; (\forall i < \mathit{lh}(x) \dot{-} 1)\, \mathit{Red1}_R((x)_i, (x)_{i+1})$.

The formalization Red_R of $\twoheadrightarrow_R$ is then given in a straightforward manner as follows:

$$\mathit{Red}_R(x, y) \;:=\; (\exists z)\mathit{RedSeq}_R(z, x, y).$$

It is obvious that the formalization $\mathit{RedCon}_{\boldsymbol{\rho}}$ of our reduction relation $\boldsymbol{\rho}$ for BON is primitive recursive. Hence, we have the following observation.

Remark 23 The formula $\mathit{Red}_{\boldsymbol{\rho}}(x, y)$ is (equivalent in PRA to) a Σ^0_1 formula.

We are ready to describe our formalized term model construction in a general form. Assume that $\mathcal{L}$ is a first order language containing $\mathcal{L}_1$, and let RedCon_R be an $\mathcal{L}$ formula formalizing a notion of reduction R. Then Red_R is an $\mathcal{L}$ formula, and the translation $*$ from $\mathcal{L}_{\mathsf{N}}$ into $\mathcal{L}$, depending on Red_R, is given by the following clauses 1–8.

The $*$ translation t^* of an individual term t of $\mathcal{L}_{\mathsf{N}}$ is given as follows:

1. If t is an individual variable, then t^* is t.

2. If t is an individual constant, then t^* is $\ulcorner t \urcorner$.

3. If t is the individual term (rs), then t^* is $\langle \ulcorner \cdot \urcorner, r^*, s^* \rangle$.

The $*$ translation A^* of an $\mathcal{L}_{\mathsf{N}}$ formula A is given as follows:

4. If A is the formula $(s = t)$, then A^* is
$$(\exists x)(Red_R(s^*, x) \ \wedge \ Red_R(t^*, x)).$$

5. If A is the formula $\mathsf{N}(t)$, then A^* is
$$(\exists x)Red_R(t^*, Num(x)).$$

6. If A is the formula $\neg B$, then A^* is $\neg(B^*)$.

7. If A is the formula $(B\ j\ C)$ for $j \in \{\vee, \wedge, \rightarrow\}$, then A^* is $(B^*\ j\ C^*)$.

8. If A is the formula $(\mathcal{Q}x)B$ for $\mathcal{Q} \in \{\exists, \forall\}$, then A^* is $(\mathcal{Q}x)(CTer(x) \ \wedge \ A^*)$.

For a specific choice of a notion of reduction R it is essential to verify the Church Rosser property $CR(Red_R)$ of R, i.e. the statement
$$(\forall x, y_1, y_2)[Red(x, y_1) \wedge Red(x, y_2) \ \rightarrow \ (\exists z)(Red(y_1, z) \wedge Red(y_2, z))].$$

It is well-known that the proof of the Church Rosser property for a usual combinatory reduction relation uses finitary arguments only and, hence, can be formalized in PRA, cf. e.g. Cantini [9] or Girard [36].

Proposition 24 $\mathsf{PRA} \vdash CR(Red_{\boldsymbol{\rho}})$.

In the sequel we will work with the translation $*$ of $\mathcal{L}_{\mathsf{N}}$ into $\mathcal{L}_1$, depending on $Red_{\boldsymbol{\rho}}$. Before we state the final proof-theoretic reduction, we want to mention an important lemma, which is an immediate consequence of the (formalized) Church Rosser theorem.

Lemma 25 *We have for all $\mathcal{L}_{\mathsf{N}}$ formulas $A(x)$:*
$$\mathsf{PRA} \vdash Red_{\boldsymbol{\rho}}(x, y) \ \rightarrow \ (A^*(x) \leftrightarrow A^*(y)).$$

Corollary 26 *Let $\mathcal{Q} \in \{\exists, \forall\}$. Then we have for all $\mathcal{L}_{\mathsf{N}}$ formulas $A(x)$:*
$$\mathsf{PRA} \vdash ((\mathcal{Q}x \in \mathsf{N})A(x))^* \ \leftrightarrow \ (\mathcal{Q}x)A^*(Num(x)).$$

It is easy to verify the $*$ translation of each axiom of TON in $\mathsf{PRA} + (\Sigma^0_1\text{-}\mathsf{I_N})$. In particular, axiom (9) is verified by means of $(\Sigma^0_1\text{-}\mathsf{I_N})$. Moreover, $(\Sigma^+\text{-}\mathsf{I_N})$ translates into $(\Sigma^0_1\text{-}\mathsf{I_N})$ by the above corollary and the fact that the $*$ translation of a Σ^+ formula is (provably equivalent in PRA to) a Σ^0_1 formula by Remark 23. Hence, we can state the following theorem.

Theorem 27 *We have for all $\mathcal{L}_{\mathsf{N}}$ formulas $A(\vec{x})$ with at most $\vec{x}$ free:*
$$\mathsf{TON} + (\Sigma^+\text{-}\mathsf{I_N}) \vdash A(\vec{x}) \quad \Longrightarrow \quad \mathsf{PRA} + (\Sigma^0_1\text{-}\mathsf{I_N}) \vdash CTer(\vec{x}) \rightarrow A^*(\vec{x}).$$

Together with Propositions 20 and 22 we are now in a position to state the following proof-theoretic equivalences. Here "≡" denotes the usual notion of proof-theoretic equivalence as it is defined e.g. in Feferman [22].

Corollary 28 *We have the following proof-theoretic equivalences:*

$$\mathsf{TON} + (\mathsf{S}\text{-}\mathsf{I_N}) \equiv \mathsf{TON} + (\mathsf{O}\text{-}\mathsf{I_N}) \equiv \mathsf{TON} + (\mathsf{N}\text{-}\mathsf{I_N}) \equiv \mathsf{TON} + (\Sigma^+\text{-}\mathsf{I_N}) \equiv \mathsf{PRA} + (\Sigma^0_1\text{-}\mathsf{I_N}) \equiv \mathsf{PRA}.$$

From Proposition 20 and the fact that an $\mathcal{L}_\mathsf{N}$ formula of the form $(\forall\vec{x} \in \mathsf{N})(t\vec{x} \in \mathsf{N})$ translates into a Π^0_2 statement under $*$, we get the following corollary.

Corollary 29 *Suppose that t is a closed term of $\mathcal{L}_\mathsf{N}$ so that*

$$\mathsf{TON} + (\Sigma^+\text{-}\mathsf{I_N}) \vdash (\forall\vec{x} \in \mathsf{N})(t\vec{x} \in \mathsf{N}).$$

Then t defines a primitive recursive function.

Let us briefly argue that these results still hold if the extensionality axiom (Ext) is added to $\mathsf{TON} + (\Sigma^+\text{-}\mathsf{I_N})$. To see this, one formalizes the term model TTE of the $\lambda\eta$ calculus (cf. Section 1.3.2) instead of CTT, using the standard translation from combinatory logic into λ calculus. It is immediate that the proof of the Church Rosser property for the $\lambda\eta$ calculus (cf. [2]) is formalizable in PRA. Hence, we can state the following strengthening of Corollary 28.

Theorem 30 *We have the following proof-theoretic equivalences:*

$$\mathsf{TON} + (\mathsf{Ext}) + (\mathsf{S}\text{-}\mathsf{I_N}) \equiv \mathsf{TON} + (\mathsf{Ext}) + (\mathsf{O}\text{-}\mathsf{I_N}) \equiv \mathsf{TON} + (\mathsf{Ext}) + (\mathsf{N}\text{-}\mathsf{I_N}) \equiv \mathsf{TON} + (\mathsf{Ext}) + (\Sigma^+\text{-}\mathsf{I_N}) \equiv \mathsf{PRA} + (\Sigma^0_1\text{-}\mathsf{I_N}) \equiv \mathsf{PRA}.$$

Let us mention that the methods sketched so far also provide upper bounds for theories based on a finitary inductive data type $\mathbb{D}$ instead of $\mathbb{N}$. This is immediate by formalizing the corresponding total term models, cf. Section 1.5.

We finish this section by addressing possible strengthenings of our last corollary. Let us write $\mathsf{B}\Sigma^+$ for the least class of $\mathcal{L}_\mathsf{N}$ formulas which is obtained form the Σ^+ formulas by closing under the connectives $\neg$, $\vee$, $\wedge$ and $\rightarrow$. Hence, a formula is in $\mathsf{B}\Sigma^+$ if it is a *boolean combination of* Σ^+ formulas. In particular, we have that every *quantifier free*, not necessarily positive formula of $\mathcal{L}_\mathsf{N}$ is in $\mathsf{B}\Sigma^+$. If $(\mathsf{B}\Sigma^+\text{-}\mathsf{I_N})$ denotes the induction schema for such formulas, then we have that the system

$$\mathsf{TON} + (\mathsf{Ext}) + (\mathsf{B}\Sigma^+\text{-}\mathsf{I_N})$$

still has the same proof-theoretic strength as primitive recursive arithmetic PRA. This follows from the fact that $\mathsf{PRA} + (\Sigma^0_1\text{-}\mathsf{I_N})$ proves induction for formulas which are Σ^0_0 in Σ^0_1, in particular, boolean combinations of Σ^0_1 formulas. For a proof of this fact we refer to Hájek and Pudlák [39]. Since the $*$ translation of a formula in $\mathsf{B}\Sigma^+$ is exactly such a $\Sigma^0_0(\Sigma^0_1)$ formula, the methods of this section provide a reduction of the above theory to $\mathsf{PRA} + (\Sigma^0_1\text{-}\mathsf{I_N})$, and hence to PRA.

Theorem 31 *We have the following proof-theoretic equivalences:*

$$\mathsf{TON} + (\mathsf{Ext}) + (\mathsf{B}\Sigma^+\text{-}\mathsf{I_N}) \equiv \mathsf{PRA} + (\Sigma^0_1\text{-}\mathsf{I_N}) \equiv \mathsf{PRA}.$$

We finish this section be mentioning a very recent result by Cantini [10, 11]. A formula A of $\mathcal{L}_\mathsf{N}$ is called N positive or N^+, if the predicate N occurs in A positively only. Further, let $(\mathsf{N}^+\text{-}\mathsf{I_N})$ denote the corresponding scheme of induction in $\mathcal{L}_\mathsf{N}$. Then it follows from Cantini's work that closed terms of type $(\mathsf{N} \to \mathsf{N})$ of the system

$$\mathsf{TON} + (\mathsf{Ext}) + (\mathsf{N}^+\text{-}\mathsf{I_N})$$

give rise to primitive recursive algorithms. This is established by partial cut elimination and formalized asymmetric interpretation within $\mathsf{PRA} + (\Sigma^0_1\text{-}\mathsf{I_N})$. Observe that $(\mathsf{N}^+\text{-}\mathsf{I_N})$ no longer allows a inner model construction in $\mathsf{PRA} + (\Sigma^0_1\text{-}\mathsf{I_N})$.

Observe that the above two extensions actually are strengthenings into different directions: although we have that Σ^+ is contained in N^+, it is not the case that N^+ contains the class $\mathsf{B}\Sigma^+$. Is it even possible to allow boolean combinations of N^+ formulas without going beyond PRA?

2.4 Systems of strength PA

For completeness, let us very briefly indicate that the induction principles $(\mathsf{F}^+\text{-}\mathsf{I_N})$ and $(\mathsf{F}\text{-}\mathsf{I_N})$ yield systems of strength PA. More precisely, PA is contained in the system $\mathsf{BON} + (\mathsf{F}^+\text{-}\mathsf{I_N})$ and, moreover, $\mathsf{TON} + (\mathsf{Ext}) + (\mathsf{F}\text{-}\mathsf{I_N})$ can be interpreted in PA by the methods of the previous section.

2.4.1 Lower bounds

Let us work again with the translation $(\cdot)^\mathsf{N}$ from $\mathcal{L}_1$ into $\mathcal{L}_\mathsf{N}$. It is immediate from Lemma 21 that the translation of atomic formulas and negated atomic formulas is equivalent to a positive equation in BON. Hence, A^N is equivalent to an F^+ formula in BON for each $\mathcal{L}_1$ formula A. Hence, we can state the following proposition.

Proposition 32 *We have for every $\mathcal{L}_1$ formula $A(\vec{x})$ with at most $\vec{x}$ free:*

$$\mathsf{PA} \vdash A(\vec{x}) \quad\Longrightarrow\quad \mathsf{BON} + (\mathsf{F}^+\text{-}\mathsf{I_N}) \vdash \vec{x} \in \mathsf{N} \to A^\mathsf{N}(\vec{x}).$$

A fortiori, PA is a lower bound for the system $\mathsf{BON} + (\mathsf{F}\text{-}\mathsf{I_N})$.

2.4.2 Upper bounds

From the work done in the previous section it is immediate that our translation $*$ based on Red_ρ gives us an embedding of $\mathsf{TON} + (\mathsf{F}\text{-}\mathsf{I_N})$ into PA.

Theorem 33 *We have for all $\mathcal{L}_\mathsf{N}$ formulas $A(\vec{x})$ with at most $\vec{x}$ free:*

$$\mathsf{TON} + (\mathsf{F}\text{-}\mathsf{I_N}) \vdash A(\vec{x}) \quad \Longrightarrow \quad \mathsf{PA} \vdash CTer(\vec{x}) \rightarrow A^*(\vec{x}).$$

Together with Proposition 32 we have thus obtained the following proof-theoretic equivalences.

Corollary 34 *We have the following proof-theoretic equivalences:*

$$\mathsf{TON} + (\mathsf{F}^+\text{-}\mathsf{I_N}) \equiv \mathsf{TON} + (\mathsf{F}\text{-}\mathsf{I_N}) \equiv \mathsf{PA}.$$

Moreover, it is again possible to extend these results to include the extensionality axiom (Ext). Hence, we get the following analogue of Theorem 30.

Theorem 35 *We have the following proof-theoretic equivalences:*

$$\mathsf{TON} + (\mathsf{Ext}) + (\mathsf{F}^+\text{-}\mathsf{I_N}) \equiv \mathsf{TON} + (\mathsf{Ext}) + (\mathsf{F}\text{-}\mathsf{I_N}) \equiv \mathsf{PA}.$$

We have seen in this section that positive formula induction ($\mathsf{F}^+\text{-}\mathsf{I_N}$) and full formula induction ($\mathsf{F}\text{-}\mathsf{I_N}$) have the same proof-theoretic strength over BON, namely PA. This will be very different with respect to extensions of BON by the non-constructive μ operator, where ($\mathsf{F}\text{-}\mathsf{I_N}$) will turn out to be much stronger than ($\mathsf{F}^+\text{-}\mathsf{I_N}$).

Chapter 3

Polynomial time applicative theories and extensions

It is the aim of this chapter to propose a *first order* theory PTO of operations and binary words, which allows full self-application and whose provably total functions on $\mathbb{W} = \{0,1\}^*$ are exactly the polynomial time computable functions. In spite of its proof-theoretic weakness, PTO has an enormous expressive power due to the presence of full (partial) combinatory logic, i.e. there are terms for each partial recursive function. The main bulk of the material presented in this chapter will appear in our paper [66].

When trying to set up a theory with self-application of polynomial strength, one might first try to mimic first order systems of bounded arithmetic – say Buss' S^1_2 – in the applicative setting in a direct way. However, it is shown in Strahm [67] that this naive approach does not work, and one immediately ends up with systems of the same strength as primitive recursive arithmetic PRA; this is due to the presence of unbounded recursion principles in the applicative language. Hence, a direct translation of induction principles from bounded arithmetic is not successful, and a theory had to be found that is better tailored for the applicative framework.

The formulation of the proposed theory PTO is very much akin to the basic theory of operations and numbers BON; in particular, PTO contains the usual axioms for a partial combinatory algebra and, hence, all the results of the first chapter of this thesis directly apply to PTO. In fact, PTO can be viewed as the polynomial time analogue of the theory $\mathsf{BON} + (\mathsf{S}\text{-}\mathsf{I}_\mathsf{N})$ that we have studied in the previous chapter. The choice of a unary predicate W for binary words instead of a predicate N for natural numbers is not mandatory, but more natural in the context of polynomial time computability. Crucial in the formulation of PTO is the principle of *set induction on* W, $(\mathsf{S}\text{-}\mathsf{I}_\mathsf{W})$, which is very natural and – most important – in the spirit of applicative theories.

The proof of the fact that PTO captures exactly polynomial time is established along the lines of reductive proof theory. More precisely, we show that PTO contains Ferreira's system of polynomial time computable arithmetic PTCA (cf. [32, 33]) via a natural embedding. Furthermore, PTO is reducible to the theory $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$, where PTCA^+ denotes the extension of PTCA by NP induction and $(\Sigma\text{-}\mathsf{Ref})$ is the reflection principle for Σ formulas. Σ reflection $(\Sigma\text{-}\mathsf{Ref})$ is equivalent to the collection principle for bounded formulas, $(\Sigma^b_\infty\text{-}\mathsf{CP})$. $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$ is known to be a Π^0_2 conservative extension of PTCA^+ by the work of Buss [6], Cantini [8], or Ferreira [31]. Moreover, PTCA^+ is Π^0_2 conservative over PTCA by Buchholz and Sieg [4], Cantini [8], and Ferreira [33]. Summing up, the provably total functions of $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$ are exactly the polytime functions.

Finally, let us mention that our approach can easily be extended in order to provide applicative theories capturing the nth level $\mathcal{E}_n$ of the Grzegorczyk hierarchy.

The plan of this chapter is as follows. In the first section we give an exact formulation of the theory PTO. Section 2 is centered around the theory of polynomial time computable arithmetic PTCA^+ plus the Σ reflection principle, and some known proof-theoretic results are addressed. The exact proof-theoretic strength of PTO is established in Section 3: we give an embedding of PTCA into PTO and show how PTO can be reduced to $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$. Section 4 deals with various conservative extensions of PTO, and in Section 5 we briefly address some additional topics and open problems concerning PTO; in particular, we discuss the status of the totality axiom (Tot) in PTO. In the final section of this chapter we sketch a generalization of our approach to the Grzegorczyk hierarchy $\mathcal{E}_n$ $(n \geq 3)$.

3.1 The theory PTO

In this section we introduce the formal framework of the theory PTO for polynomial time operations on binary words. We start off from a modified base theory BOW of basic operations and binary words and define PTO to be BOW plus set induction with respect to binary words W.

The language $\mathcal{L}_\mathsf{W}$ of BOW is a language for the logic of partial terms. It includes the language $\mathcal{L}_\mathsf{c}$ of partial combinatory logic plus the following constants: $\epsilon, \mathsf{0}, \mathsf{1}$ (empty word, zero, one), $*, \times, \mathsf{p_W}$ (word concatenation and multiplication, word predecessor), $\mathsf{c}_\subseteq$ (initial subword relation), $\mathsf{d_W}$ (definition by cases on binary words), $\mathsf{r_W}$ (bounded primitive recursion). Finally, $\mathcal{L}_\mathsf{W}$ includes a unary relation symbol W (binary words).

The *individual terms* $(r, s, t, r_1, s_1, t_1, \ldots)$ and *formulas* $(A, B, C, A_1, B_1, C_1, \ldots)$ of $\mathcal{L}_\mathsf{W}$ are defined in the very same way as terms and formulas of the language $\mathcal{L}_\mathsf{N}$. Moreover, we adopt the same conventions concerning left association, pairing and the predicate W as for $\mathcal{L}_\mathsf{N}$. Further, we often use infix notation for $*$ and $\times$, i.e. $s * t$ abbreviates $*st$ and $s \times t$ stands for $\times st$. In addition, let us write $s \subseteq t$ instead of $\mathsf{c}_\subseteq st = \mathsf{0}$, and $s \leq t$ for $\mathsf{1} \times s \subseteq \mathsf{1} \times t$. Finally, $(s = t \,|\, r)$ is an abbreviation for

$$(r \leq t \wedge s \subseteq t \wedge \mathsf{1} \times s = \mathsf{1} \times r) \vee (t \leq r \wedge s = t).$$

Let us give a brief informal interpretation of the syntax of the basic theory of operations and binary words BOW to be introduced below. The individual variables are supposed to range over a universe V that forms a partial combinatory algebra closed under pairing. There is a subset $\mathsf{W} \subset V$, consisting of finite sequences of $\mathsf{0}$'s and $\mathsf{1}$'s; W is generated from ϵ, $\mathsf{0}$ and $\mathsf{1}$ by the operation $*$ of word concatenation. Furthermore, we have an operation $\times$ of word multiplication, where $w_1 \times w_2$ denotes the word w_1 concatenated with itself length of w_2 times. $\mathsf{p_W}$ is supposed to be a predecessor or destructor operation on W, and $\mathsf{c}_\subseteq$ denotes the characteristic function of the initial subword relation. $\mathsf{d_W}$ provides a definition by cases operator on W. The relation $w_1 \leq w_2$ means that the length of w_1 is less than or equal to the length of w_2; accordingly, $w_1 \,|\, w_2$ denotes the truncation of w_1 to the length of w_2. This gives meaning to the *bounded* recursor $\mathsf{r_W}$ on W, which provides an operation $\mathsf{r_W} fgb$ for primitive recursion from f and g with length bound b.

We are ready to introduce the theory BOW of basic operations and binary words. It is defined in complete analogy to the theory BON, where the natural numbers N are replaced by the binary words W, and primitive recursion $\mathsf{r_N}$ on N by *bounded* primitive recursion $\mathsf{r_W}$ on W. The axioms for a partial combinatory algebra and pairing remain unchanged in the theory PTO; we repeat them for completeness below. The underlying logic of BOW is again the classical logic of partial terms, and its non-logical axioms are divided into the following eight groups.

I. Partial combinatory algebra.

(1) $\mathsf{k}xy = x$,

(2) $\mathsf{s}xy\downarrow \;\wedge\; \mathsf{s}xyz \simeq xz(yz)$.

II. Pairing and projection.

(3) $\mathsf{p_0}(x, y) = x \;\wedge\; \mathsf{p_1}(x, y) = y$.

III. Binary words.

(4) $\epsilon \in \mathsf{W} \;\wedge\; \mathsf{0} \in \mathsf{W} \;\wedge\; \mathsf{1} \in \mathsf{W}$,

(5) $(* \in \mathsf{W}^2 \to \mathsf{W})$,

(6) $x \in \mathsf{W} \to x * \epsilon = x$,

(7) $x \in \mathsf{W} \wedge y \in \mathsf{W} \;\to\; x * (y * \mathsf{0}) = (x * y) * \mathsf{0} \;\wedge\; x * (y * \mathsf{1}) = (x * y) * \mathsf{1}$,

(8) $x \in \mathsf{W} \wedge y \in \mathsf{W} \;\to\; x * \mathsf{0} \neq y * \mathsf{1} \;\wedge\; x * \mathsf{0} \neq \epsilon \;\wedge\; x * \mathsf{1} \neq \epsilon$,

(9) $x \in \mathsf{W} \wedge y \in \mathsf{W} \wedge x * \mathsf{0} = y * \mathsf{0} \;\to\; x = y$,

(10) $x \in \mathsf{W} \wedge y \in \mathsf{W} \wedge x * \mathsf{1} = y * \mathsf{1} \;\to\; x = y$.

IV. Word multiplication.

(11) $\times \in \mathsf{W}^2 \to \mathsf{W}$,

(12) $x \in \mathsf{W} \to x \times \epsilon = \epsilon$,

(13) $x \in \mathsf{W} \wedge y \in \mathsf{W} \;\to\; x \times (y * \mathsf{0}) = (x \times y) * x \;\wedge\; x \times (y * \mathsf{1}) = (x \times y) * x$.

V. Predecessor on W.

(14) $\mathsf{p_W} \in \mathsf{W} \to \mathsf{W}$,

(15) $\mathsf{p_W}\epsilon = \epsilon$,

(16) $x \in \mathsf{W} \;\to\; \mathsf{p_W}(x * \mathsf{0}) = x \;\wedge\; \mathsf{p_W}(x * \mathsf{1}) = x$,

(17) $x \in \mathsf{W} \wedge x \neq \epsilon \;\to\; (\mathsf{p_W}x) * \mathsf{0} = x \;\vee\; (\mathsf{p_W}x) * \mathsf{1} = x$.

VI. Initial subword relation.

(18) $x \in \mathsf{W} \wedge y \in \mathsf{W} \;\to\; \mathsf{c}_\subseteq xy = \mathsf{0} \;\vee\; \mathsf{c}_\subseteq xy = \mathsf{1}$,

(19) $x \in \mathsf{W} \;\to\; (x \subseteq \epsilon \leftrightarrow x = \epsilon)$,

(20) $x \in \mathsf{W} \wedge y \in \mathsf{W} \wedge y \neq \epsilon \;\to\; (x \subseteq y \leftrightarrow x \subseteq \mathsf{p_W}y \vee x = y)$.

VII. Definition by cases on W.

(21) $a \in \mathsf{W} \wedge b \in \mathsf{W} \wedge a = b \;\to\; \mathsf{d_W}xyab = x$,

(22) $a \in \mathsf{W} \wedge b \in \mathsf{W} \wedge a \neq b \;\to\; \mathsf{d_W}xyab = y$.

VIII. Bounded primitive recursion on W.

(23) $(f \in \mathsf{W} \to \mathsf{W}) \;\wedge\; (g \in \mathsf{W}^3 \to \mathsf{W}) \;\wedge\; (b \in \mathsf{W}^2 \to \mathsf{W}) \;\to\; (\mathsf{r_W}fgb \in \mathsf{W}^2 \to \mathsf{W})$,

(24) $(f \in \mathsf{W} \to \mathsf{W}) \;\wedge\; (g \in \mathsf{W}^3 \to \mathsf{W}) \;\wedge\; (b \in \mathsf{W}^2 \to \mathsf{W}) \wedge$
$x \in \mathsf{W} \;\wedge\; y \in \mathsf{W} \;\wedge\; y \neq \epsilon \;\wedge\; h = \mathsf{r_W}fgb \;\to$
$hx\epsilon = fx \;\wedge\; hxy = gxy(hx(\mathsf{p_W}y)) \,|\, bxy$.

Observe that in the formulation of bounded primitive recursion $\mathsf{r_W}$ on W, we do *not* require b to be a polynomial, but only a total operation on W. This formulation is more natural, and we will see in Section 3.3.2 that it does not raise the proof-theoretic strength of the theory PTO to be introduced below. Moreover, the above formulation allows natural generalizations of our theory by stronger initial functions, cf. Section 3.6. Let us recall that bounded primitive recursion on $\mathbb{W} = \{0,1\}^*$ with (word) polynomials as initial functions exactly generates the polynomial time computable functions on $\mathbb{W}$, cf. Ferreira [32] and Cobham [14].

Sets of binary words are understood in our context in the same way as sets of natural numbers, namely via their total characteristic functions on W. Accordingly, we define $\mathcal{P}(\mathsf{W})$ by

$$f \in \mathcal{P}(\mathsf{W}) \ := \ (\forall x \in \mathsf{W})(fx = 0 \lor fx = 1).$$

Moreover, the principle of set induction on W reads in an analogous way as follows.

Set induction on W, $(\mathsf{S\text{-}I_W})$.

$$f \in \mathcal{P}(\mathsf{W}) \land f\epsilon = 0 \land (\forall x \in \mathsf{W})(f(\mathsf{p_W}x) = 0 \to fx = 0) \to (\forall x \in \mathsf{W})(fx = 0).$$

Our applicative theory of polynomial time operations PTO is now defined to be BOW plus set induction on W, i.e. $\mathsf{BOW} + (\mathsf{S\text{-}I_W})$. The principle of set induction $(\mathsf{S\text{-}I_W})$ is crucial for the proof-theoretic strength of PTO. We will see in Section 3.3.2 that the premise $f \in \mathcal{P}(\mathsf{W})$ allows one to treat set induction in a certain theory of arithmetic which has polynomial strength only.

Since BOW contains the usual axioms for a partial combinatory algebra, the results of Chapter 1 *mutis mutandis* carry over to BOW and PTO. In particular, we obtain a natural adaptation of the standard recursion theoretic model *PRO* in terms of ordinary recursion theory on $\mathbb{W}$, and explicit definitions and the partial form of the recursion theorem are available in our new setting. Accordingly, there exists a term t_f for each partial recursive function f, however, PTO does generally not prove the totality of f. For example, PTO includes a term t_{exp} which defines the exponential function.

Let us finish this section by making some brief comments concerning polynomial time functionals. Cook and Urquhart [16] introduced a class *BFF* of *basic feasible functionals* in all finite types in order to provide functional interpretations of feasibly constructive arithmetic. The type 1 functions of *BFF* coincide with the polynomial time computable functions. It is straightforward from the axioms of PTO and λ abstraction in PTO that there exists an $\mathcal{L}_\mathsf{W}$ term t_F for each functional F in *BFF* so that the defining equations and the well-typedness of F are derivable in PTO. Further work on *BFF* and feasible functionals in general can be found in Cook and Kapron [15] and Seth [61].

3.2 Polynomial time computable arithmetic and extensions

In this section we give an introduction into Ferreira's framework of polynomial time computable arithmetic. In particular, we introduce the three systems PTCA, PTCA^+ and $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$. We recapitulate the crucial result saying that all these theories prove the same Π^0_2 statements and, hence, their provably total functions are exactly the polytime functions. In Section 3.3 we will show that PTO contains PTCA and is reducible to $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$, thus establishing PTO's exact proof-theoretic strength.

3.2.1 The theories PTCA and PTCA$^+$

The theory PTCA of polynomial time computable arithmetic over binary strings was introduced by Ferreira [32, 33]. PTCA can be viewed as a polynomial time analogue of Skolem's system of primitive recursive arithmetic PRA. The theory PTCA is formulated in the first order language $\mathcal{L}_\mathsf{p}$, which is based on the elementary language $\mathcal{L}_\mathsf{e}$. The latter contains individual variables $a, b, c, x, y, z, u, v, w, f, g, h, \ldots$ (possibly with subscripts), constants ϵ, 0, 1, the binary function symbols $*$ and $\times$[1] as well as the binary relation symbols $=$ and $\subseteq$; the meaning of these symbols is identical to the one of the corresponding operations in $\mathcal{L}_\mathsf{W}$. Now $\mathcal{L}_\mathsf{p}$ is obtained from $\mathcal{L}_\mathsf{e}$ by adding a function symbol for each description of a polynomial time computable function, where the terms of $\mathcal{L}_\mathsf{e}$ act as bounding terms, similar to Cobham's characterization of the polytime functions. Terms $(r, s, t, \ldots)$ and formulas $(A, B, C, \ldots)$ of $\mathcal{L}_\mathsf{p}$ (both possibly with subscripts) are defined as usual. For the details the reader is referred to [32, 33].

There are two sorts of *bounded quantifiers* which are relevant in the sequel. The *sharply bounded quantifiers* have the form $(\exists x)(x \subseteq t \wedge \ldots)$ or $(\forall x)(x \subseteq t \rightarrow \ldots)$, and in the following we just write $(\exists x \subseteq t)(\ldots)$ and $(\forall x \subseteq t)(\ldots)$. Furthermore, we have *(generally) bounded quantifiers* $(\exists x)(x \leq t \wedge \ldots)$ and $(\forall x)(x \leq t \rightarrow \ldots)$, where $x \leq t$ reads as $1 \times x \subseteq 1 \times t$ as in the previous section. Again we use the usual shorthands as above. If A is an arbitrary $\mathcal{L}_\mathsf{p}$ formula, then we write A^t for the formula which is obtained from A by replacing each unbounded quantifier $(\mathcal{Q}x)$ by the corresponding bounded quantifier $(\mathcal{Q}x \leq t)$. The following definition contains important classes of $\mathcal{L}_\mathsf{p}$ formulas.

Definition 36 Let us define the following classes of $\mathcal{L}_\mathsf{p}$ formulas.

1. QF denotes the set of all quantifier free $\mathcal{L}_\mathsf{p}$ formulas.

2. A formula is called Δ^b_0 if all its quantifiers are *sharply* bounded.

3. A formula is in the class Σ^b_1 if it has the form $(\exists x \leq t)A$ for A a formula in QF.

4. A formula is called *extended* Σ^b_1 or $\mathsf{e}\Sigma^b_1$ if (i) all its positive existential and negative universal quantifiers are bounded, and (ii) all its positive universal and negative existential quantifiers are *sharply* bounded.

5. An $\mathcal{L}_\mathsf{p}$ formula is called Σ^b_∞ or *bounded* if all its quantifiers are bounded.

6. A Σ^0_1 formula has the form $(\exists x)A$ for A in QF; a Π^0_2 formula is of the shape $(\forall x)(\exists y)A$ for A in QF.

7. A formula is in the class Σ if all its positive universal and negative existential quantifiers are bounded. Π formulas are defined dually to Σ formulas, i.e. they are equivalent to negations of Σ formulas.

[1] We again use infix notation for $*$ and $\times$ and often write ts instead of $t * s$.

The Δ_0^b formulas are the polynomial time decidable matrices of [32, 33]. Furthermore, the Σ_1^b formulas define exactly the NP predicates and the Σ_∞^b formulas the predicates in the Meyer-Stockmeyer polynomial time hierarchy.

The theory of polynomial time computable arithmetic PTCA is a first order theory based on classical logic with equality, and comprising defining axioms for the base language $\mathcal{L}_\mathsf{e}$ as well as defining equations for each description of a polytime function in $\mathcal{L}_\mathsf{p}$. In addition, PTCA includes the notation induction scheme

$$A(\epsilon) \land (\forall x)(A(x) \to A(x0) \land A(x1)) \to (\forall x)A(x)$$

for each $\mathcal{L}_\mathsf{p}$ formula $A(x)$ in QF. It is well-known that PTCA proves induction for Δ_0^b formulas; for details we refer to [32, 33]. Furthermore, it is straightforward to establish that the provably total functions of PTCA are exactly the polytime functions (cf. [4, 33]).

Let PTCA^+ denote the extension of PTCA, where notation induction is allowed for NP predicates, i.e. formulas in Σ_1^b. The system PTCA^+ is closely related to Buss' system S_2^1 (cf. [5]). Induction is provable in PTCA^+ for *extended* Σ_1^b formulas (cf. [32, 33]). In analogy to Parson's result we obtain that PTCA^+ is a conservative extension of PTCA with respect to Π_2^0 statements. Proofs can be found in [4, 8, 33].

Proposition 37 *Suppose* $\mathsf{PTCA}^+ \vdash (\forall x)(\exists y)A(x,y)$*, where* A *is a* QF *formula. Then we have* $\mathsf{PTCA} \vdash (\forall x)(\exists y)A(x,y)$*.*

Corollary 38 *Suppose* $\mathsf{PTCA}^+ \vdash (\forall x)(\exists y)A(x,y)$*, where* A *is a* QF *formula. Then there exists an* $\mathcal{L}_\mathsf{p}$ *term* $t(x)$ *so that* $\mathsf{PTCA} \vdash (\forall x)A(x,t(x))$*.*

In the following we often write $|s|$ (the length of s) instead of $1 \times s$, $s \subset t$ instead of $s \subseteq t \land s \neq t$, and $s < t$ instead of $1 \times s \subset 1 \times t$. The abbreviation $s = t\,|\,r$ is understood in the same way as in the previous section. In addition, p denotes the obvious predecessor function on binary words and $c_\subseteq$ is the binary characteristic function of the initial subword relation. Finally, we use the trivial representation of the natural numbers as tally words, which is given by $\overline{0} = \epsilon$ and $\overline{n+1} = \overline{n}1$. We will write n instead of $\overline{n}$ whenever it is clear from the context that we mean n as a tally word and not as a natural number.

We finish this paragraph by adopting some conventions concerning polynomial time sequence coding within PTCA.[2] For the details the reader is again referred to Ferreira [32, 33]. Let $\langle \ldots \rangle$ denote a polytime function for forming n-sequences $\langle t_0, \ldots, t_{n-1} \rangle$ of binary words, and let $lh(t)$ denote the length of the sequence coded by t, i.e. if $t = \langle t_0, \ldots, t_{n-1} \rangle$, then $lh(t) = \overline{n}$. We write $Seq_n(t)$ for $Seq(t) \land lh(t) = \overline{n}$. There is a polytime projection function so that $(t)_m$ denotes the mth component of the sequence coded by t if $m \subset lh(t)$; we write $last(t)$ for $(t)_{p(lh(t))}$ and $(t)_{m,n}$ instead of

[2]Some notations coincide with those already introduced in the context of $\mathcal{L}_1$, however, this will never cause confusion.

$((t)_m)_n$. Furthermore, let $\frown$ denote the polytime sequence concatenation function. For example, if t is the sequence $\langle t_0, t_1, t_2, t_3 \rangle$, then $lh(t) = 1111$, $(t)_\epsilon = t_0$, $(t)_1 = t_1$, $(t)_{11} = t_2$, $(t)_{111} = t_3$, $last(t) = t_3$ and $t = \langle t_0, t_1 \rangle \frown \langle t_2, t_3 \rangle$. Finally, let $SqBd(a, b)$ denote a suitable $\mathcal{L}_\mathsf{p}$ term, so that PTCA proves

$$Seq(v) \wedge lh(v) \leq |b|1 \wedge (\forall w \subset lh(v))((v)_w \leq a) \rightarrow v \leq SqBd(a, b).$$

$SqBd$ is easily constructed from the terms in $\mathcal{L}_\mathsf{e}$.

3.2.2 The theory $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$

In order to interpret our theory of polynomial time operations on binary words PTO, we will need the crucial principle of Σ *reflection* $(\Sigma\text{-}\mathsf{Ref})$, which has the form

$$(\Sigma\text{-}\mathsf{Ref}) \qquad A \rightarrow (\exists x) A^x,$$

where A is a formula in Σ. It is not difficult to see that $(\Sigma\text{-}\mathsf{Ref})$ is equivalent to the *collection principle for bounded formulas* $(\Sigma^b_\infty\text{-}\mathsf{CP})$, which reads as

$$(\Sigma^b_\infty\text{-}\mathsf{CP}) \qquad (\forall x \leq t)(\exists y) A \rightarrow (\exists z)(\forall x \leq t)(\exists y \leq z) A,$$

where A is a Σ^b_∞ formula. It is known that adding Σ reflection (or equivalently bounded collection) to a suitable bounded theory yields a Π^0_2 conservative extension. This was first proved by Buss [6]. Another elementary model-theoretic proof is due to Ferreira [31]. Finally, a very perspicuous proof-theoretic proof making use of partial cut elimination and an *asymmetric interpretation* has recently been established by Cantini [8], cf. the sketch below.

Proposition 39 *Suppose* $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref}) \vdash (\forall x)(\exists y) A(x, y)$, *where* A *is a* Σ^b_∞ *formula. Then we have* $\mathsf{PTCA}^+ \vdash (\forall x)(\exists y) A(x, y)$.

As consequence we get by Corollary 38 the desired conservation result.

Corollary 40 *Suppose* $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref}) \vdash (\forall x)(\exists y) A(x, y)$, *where* A *is a* QF *formula. Then there exists an* $\mathcal{L}_\mathsf{p}$ *term* $t(x)$ *so that* $\mathsf{PTCA} \vdash (\forall x) A(x, t(x))$.

Let us mention that Σ reflection $(\Sigma\text{-}\mathsf{Ref})$ follows from Weak König's Lemma for trees defined by bounded formulas, $(\Sigma^b_\infty\text{-}\mathsf{WKL})$. In fact, the first order strength of $(\Sigma^b_\infty\text{-}\mathsf{WKL})$ is exactly $(\Sigma\text{-}\mathsf{Ref})$ (over the base theory PTCA^+), cf. Ferreira [34]. Furthermore, $(\Sigma^b_\infty\text{-}\mathsf{WKL})$ is a consequence of strict Π^1_1 reflection, which by Cantini [8] again yields a Π^0_2 conservative extension of PTCA.

Let us finish this section by giving a very brief sketch of Cantini's proof of Proposition 39. Actually, his proof in [8] is in the context of strict Π^1_1 reflection which implies $(\Sigma\text{-}\mathsf{Ref})$, however, the idea of the argument is the same. In a *first step* one reformulates $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$ in a Tait calculus T; in particular, Σ^b_1 induction and $(\Sigma\text{-}\mathsf{Ref})$ are stated by their corresponding rules. One immediately observes that the main formulas of the non-logical axioms and rules of T are Σ and, hence, each T derivation can be transformed into a quasinormal T derivation so that each cut formula is either Σ or Π; this is established by the usual (finite) cut elimination argument, cf. [56, 59]. In a *second step*, the Σ-Π fragment of of T is reduced to PTCA^+ by means of an *asymmetric interpretation* argument, where for each quasinormal derivation d in T one provides a term t_d of $\mathcal{L}_\mathsf{e}$ so that $t_d(x)$ is an appropriate bound for existential quantifiers provided x is a given bound for the universal quantifiers. (Σ-Ref) is trivially interpreted in this argument and Σ^b_1 induction is basically left untouched by our interpretation, since its induction formulas are bounded. The rule for the existential quantifier is treated in an obvious manner, and the cut rule is taken care of as usual in asymmetric interpretations by composition of given terms. This concludes our brief proof sketch of Proposition 39.

It is immediate from the proof sketched above that the argument works equally well for arbitrary *bounded* theories. This fact will be used in Section 3.6 of this chapter.

3.3 The proof-theoretic strength of PTO

In the following we address the exact proof-theoretic strength of PTO. In the first paragraph we show that Ferreira's systems of polynomial time computable arithmetic PTCA is contained in PTO. In the second paragraph we establish an interpretation of PTO into the system $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$, thus demonstrating the crucial role of Σ reflection. As a byproduct of our interpretation we obtain that closed terms of type $(\mathsf{W} \to \mathsf{W})$ of PTO give rise to polytime algorithms. In the final paragraph of this section we give a detailed proof of the fixed point theorem (Theorem 43), which is essential for our interpretation in the second paragraph.

3.3.1 Lower bounds

There is a natural embedding of the language $\mathcal{L}_\mathsf{p}$ into the language $\mathcal{L}_\mathsf{W}$, which is similar in spirit to our interpretation of $\mathcal{L}_1$ into $\mathcal{L}_\mathsf{N}$ in Section 2.3.1. Using the bounded recursion operator $\mathsf{r_W}$, each (description of) a polytime function can be represented in PTO by an $\mathcal{L}_\mathsf{W}$ term. Furthermore, the recursion equations and the totality of the corresponding function are derivable in PTO. Hence, we have an $\mathcal{L}_\mathsf{W}$ formula $A^\mathsf{W}(\vec{x})$ for each $\mathcal{L}_\mathsf{p}$ formula A, where the individual variables of $\mathcal{L}_\mathsf{p}$ are supposed to range over W, i.e.

$$((\exists y)A(\vec{x}, y))^\mathsf{W} \;=\; (\exists y \in \mathsf{W})A^\mathsf{W}(\vec{x}, y),$$

and similarly for universal quantifiers. Moreover, each quantifier free formula of $\mathcal{L}_\mathsf{p}$ can be represented in $\mathcal{L}_\mathsf{W}$ by a set in the sense of $P(\mathsf{W})$.

Lemma 41 *For every quantifier free formula $A(\vec{x})$ of $\mathcal{L}_\mathsf{p}$ with at most $\vec{x}$ free there exists an individual term t_A of $\mathcal{L}_\mathsf{W}$ so that*

1. $\mathsf{PTO} \vdash (\forall \vec{x} \in \mathsf{W})(t_A \vec{x} = \mathsf{0} \vee t_A \vec{x} = \mathsf{1})$,

2. $\mathsf{PTO} \vdash (\forall \vec{x} \in \mathsf{W})(A^{\mathsf{W}}(\vec{x}) \leftrightarrow t_A \vec{x} = \mathsf{0})$.

It is an immediate consequence of this lemma that notation induction for quantifier free formulas carries over to set induction in $\mathcal{L}_{\mathsf{W}}$. Hence, we have the following embedding of PTCA into PTO.

Proposition 42 *We have for every $\mathcal{L}_{\mathsf{p}}$ formula $A(\vec{x})$ with at most $\vec{x}$ free:*

$$\mathsf{PTCA} \vdash A(\vec{x}) \quad \Longrightarrow \quad \mathsf{PTO} \vdash \vec{x} \in \mathsf{W} \rightarrow A^{\mathsf{W}}(\vec{x}).$$

This finishes our discussion of the lower bound for PTO.

3.3.2 Upper bounds

In the following we show that PTO can be embedded into $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$, which is known to be a Π^0_2 conservative extension of PTCA by the results of Section 3.2.2. As a consequence, we obtain that the provably total functions of PTO are computable in polynomial time.

The main step in establishing an embedding of PTO into $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$ is to find an $\mathcal{L}_{\mathsf{p}}$ formula $App(x, y, z)$ which interprets $xy \simeq z$. Together with an interpretation of the constants of $\mathcal{L}_{\mathsf{W}}$ this will yield a translation of $\mathcal{L}_{\mathsf{W}}$ into $\mathcal{L}_{\mathsf{p}}$ in a standard way. In the definition of App we will make use of a construction similar to Feferman [20], p. 200, Feferman and Jäger [28], p. 258 or Beeson [3], p. 144. In particular, App will be represented as a fixed point of a Σ^0_1 positive inductive definition. The details of this construction are very relevant due to the weakness of $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$.

In order to describe a suitable inductive operator form below, it will be convenient to work with an extension $\mathcal{L}_{\mathsf{p}}(Q)$ of $\mathcal{L}_{\mathsf{p}}$ by a ternary relation symbol Q which does not belong to $\mathcal{L}_{\mathsf{p}}$. If $A(Q)$ is an $\mathcal{L}_{\mathsf{p}}(Q)$ formula and $B(x, y, z)$ an $\mathcal{L}_{\mathsf{p}}$ formula, then $A(B)$ denotes the result of substituting $B(r, s, t)$ for every occurrence of $Q(r, s, t)$ in the formula $A(Q)$.

In the following let us first turn to the interpretation of the recursion operator r_{W}. Toward this end, assume that $A(f, x, y)$ is a fixed $\mathcal{L}_{\mathsf{p}}(Q)$ formula with at most f, x, y free. Then we define for each natural number n greater than 0 an $\mathcal{L}_{\mathsf{p}}(Q)$ formula $A_n(f, x_1, \ldots, x_n, y)$ by recursion on n as follows:

$$\begin{aligned} A_1(f, x_1, y) &:= A(f, x_1, y), \\ A_{n+1}(f, x_1, \ldots, x_{n+1}, y) &:= (\exists z)(A_n(f, x_1, \ldots, x_n, z) \wedge A(z, x_{n+1}, y)). \end{aligned}$$

If $A(f, x, y)$ is assumed to interpret $fx \simeq y$, then $A_n(f, x_1, \ldots, x_n, y)$ interprets $fx_1, \ldots x_n \simeq y$. We will drop the subscript n whenever it is clear from the context.

Now we are ready to define the $\mathcal{L}_{\mathsf{p}}(Q)$ formula $Rec_A(f,g,b,x,y,z)$. It describes the graph of the function which is defined from f and g by bounded primitive recursion with length bound b in the sense of A. The exact formulation of Rec_A is as follows:

$$
\begin{aligned}
&Rec_A(f,g,b,x,y,z) \;:=\\
&\quad (\exists v)[Seq(v) \;\wedge\; lh(v) = |y|1 \;\wedge\; A(f,x,(v)_\epsilon) \;\wedge\\
&\qquad (\forall w \subseteq y)(w \neq \epsilon \to\\
&\qquad\quad (\exists u_1,u_2)[A_3(g,x,w,(v)_{|p(w)|},u_1) \wedge A_2(b,x,w,u_2) \wedge (v)_{|w|} = u_1|u_2])\\
&\qquad \wedge\; z = (v)_{|y|}].
\end{aligned}
$$

In a next step we define a Q positive $\mathcal{L}_{\mathsf{p}}(Q)$ formula $\mathcal{A}(Q,x,y,z)$, a so-called inductive operator form; a fixed point of $\mathcal{A}$ will later serve as an interpretation of the application operation. Let us choose pairwise different binary words $\hat{\mathsf{k}}$, $\hat{\mathsf{s}}$, $\hat{\mathsf{p}}$, $\hat{\mathsf{p}}_0$, $\hat{\mathsf{p}}_1$, $\hat{*}$, $\hat{\times}$, $\hat{\mathsf{p}}_{\mathsf{W}}$, $\hat{\mathsf{c}}_{\subseteq}$, $\hat{\mathsf{d}}_{\mathsf{W}}$ and $\hat{\mathsf{r}}_{\mathsf{W}}$, which do not belong to $Seq \cup \{\epsilon, 0, 1\}$. In addition, put $\hat{\epsilon} = \epsilon$, $\hat{0} = 0$ and $\hat{1} = 1$. Then we define $\mathcal{A}(Q,x,y,z)$ to be the disjunction of the following formulas (1)–(26):

(1) $x = \hat{\mathsf{k}} \wedge z = \langle \hat{\mathsf{k}}, y\rangle$,

(2) $Seq_2(x) \;\wedge\; (x)_0 = \hat{\mathsf{k}} \;\wedge\; (x)_1 = z$,

(3) $x = \hat{\mathsf{s}} \;\wedge\; z = \langle \hat{\mathsf{s}}, y\rangle$,

(4) $Seq_2(x) \;\wedge\; (x)_0 = \hat{\mathsf{s}} \;\wedge\; z = \langle \hat{\mathsf{s}}, (x)_1, y\rangle$,

(5) $Seq_3(x) \;\wedge\; (x)_0 = \hat{\mathsf{s}} \;\wedge\; (\exists v,w)\,(Q((x)_1,y,v) \wedge Q((x)_2,y,w) \wedge Q(v,w,z))$,

(6) $x = \hat{\mathsf{p}} \;\wedge\; z = \langle \hat{\mathsf{p}}, y\rangle$,

(7) $Seq_2(x) \;\wedge\; (x)_0 = \hat{\mathsf{p}} \;\wedge\; z = \langle (x)_1, y\rangle$,

(8) $x = \hat{\mathsf{p}}_0 \;\wedge\; y = \langle z, (y)_1\rangle$,

(9) $x = \hat{\mathsf{p}}_1 \;\wedge\; y = \langle (y)_0, z\rangle$,

(10) $x = \hat{*} \;\wedge\; z = \langle \hat{*}, y\rangle$,

(11) $Seq_2(x) \;\wedge\; (x)_0 = \hat{*} \;\wedge\; z = (x)_1 * y$,

(12) $x = \hat{\times} \;\wedge\; z = \langle \hat{\times}, y\rangle$,

(13) $Seq_2(x) \;\wedge\; (x)_0 = \hat{\times} \;\wedge\; z = (x)_1 \times y$,

(14) $x = \hat{\mathsf{p}}_{\mathsf{W}} \;\wedge\; z = p(y)$,

(15) $x = \hat{\mathsf{c}}_{\subseteq} \;\wedge\; z = \langle \hat{\mathsf{c}}_{\subseteq}, y\rangle$,

(16) $Seq_2(x) \;\wedge\; (x)_0 = \hat{\mathsf{c}}_{\subseteq} \;\wedge\; z = c_{\subseteq}((x)_1, y)$,

(17) $x = \hat{\mathsf{d}}_\mathsf{W} \wedge z = \langle \hat{\mathsf{d}}_\mathsf{W}, y \rangle$,

(18) $Seq_2(x) \wedge (x)_0 = \hat{\mathsf{d}}_\mathsf{W} \wedge z = \langle \hat{\mathsf{d}}_\mathsf{W}, (x)_1, y \rangle$,

(19) $Seq_3(x) \wedge (x)_0 = \hat{\mathsf{d}}_\mathsf{W} \wedge z = \langle \hat{\mathsf{d}}_\mathsf{W}, (x)_1, (x)_2, y \rangle$,

(20) $Seq_4(x) \wedge (x)_0 = \hat{\mathsf{d}}_\mathsf{W} \wedge (x)_3 = y \wedge z = (x)_1$,

(21) $Seq_4(x) \wedge (x)_0 = \hat{\mathsf{d}}_\mathsf{W} \wedge (x)_3 \neq y \wedge z = (x)_2$,

(22) $x = \hat{\mathsf{r}}_\mathsf{W} \wedge z = \langle \hat{\mathsf{r}}_\mathsf{W}, y \rangle$,

(23) $Seq_2(x) \wedge (x)_0 = \hat{\mathsf{r}}_\mathsf{W} \wedge z = \langle \hat{\mathsf{r}}_\mathsf{W}, (x)_1, y \rangle$,

(24) $Seq_3(x) \wedge (x)_0 = \hat{\mathsf{r}}_\mathsf{W} \wedge z = \langle \hat{\mathsf{r}}_\mathsf{W}, (x)_1, (x)_2, y \rangle$,

(25) $Seq_4(x) \wedge (x)_0 = \hat{\mathsf{r}}_\mathsf{W} \wedge z = \langle \hat{\mathsf{r}}_\mathsf{W}, (x)_1, (x)_2, (x)_3, y \rangle$,

(26) $Seq_5(x) \wedge (x)_0 = \hat{\mathsf{r}}_\mathsf{W} \wedge Rec_Q((x)_1, (x)_2, (x)_3, (x)_4, y, z)$.

This finishes the definition of the Q positive $\mathcal{L}_\mathsf{p}(Q)$ formula $\mathcal{A}(Q, x, y, z)$. Note that $\mathcal{A}$ is in fact a Σ^0_1 definition (modulo (Σ-Ref)). Hence, we know from standard recursion theory (cf. e.g. Hinman [42]) that the least fixed point of $\mathcal{A}$ is an r.e. set. The usual proof of this fact uses a careful construction *from below* by defining some sort of computability predicate, similar to the proof of Kleene's normal form theorem. Since we have all the sequence coding available in our weak setting, it is more or less straightforward to see that this construction can be carried through in PTCA^+. The details, however, are long and tedious. Moreover, one easily verifies that the so-obtained r.e. set - call it App - defines a fixed point of $\mathcal{A}$, where an obvious application of (Σ-Ref) is needed. $\mathsf{PTCA}^+ + (\Sigma\text{-Ref})$ does not prove the minimality of App, of course. Instead, it is not difficult to establish the functionality of App. Summing up, we have the following theorem, whose proof is contained in the next paragraph of this section.

Theorem 43 *There exists a Σ_1 formula $App(x, y, z)$ of $\mathcal{L}_\mathsf{p}$ with free variables as shown so that* $\mathsf{PTCA}^+ + (\Sigma\text{-Ref})$ *proves:*

1. $(\forall x, y, z)(\mathcal{A}(App, x, y, z) \leftrightarrow App(x, y, z))$.
2. $(\forall x, y, z_1, z_2)(App(x, y, z_1) \wedge App(x, y, z_2) \rightarrow z_1 = z_2)$.

Now the stage is set in order to describe a translation $*$ from $\mathcal{L}_\mathsf{W}$ into $\mathcal{L}_\mathsf{p}$. Let us first define an $\mathcal{L}_\mathsf{p}$ formula $V^*_t(x)$ for each individual term t of $\mathcal{L}_\mathsf{W}$ so that the variable x does not occur in t. The formula $V^*_t(x)$ expresses that x is the value of t under the interpretation $*$. The exact definition is by induction on the complexity of t:

1. If t is an individual variable, then $V^*_t(x)$ is $(t = x)$.
2. If t is an individual constant, then $V^*_t(x)$ is $(\hat{t} = x)$.

3. If t is the individual term (rs), then

$$V_t^*(x) \;:=\; (\exists y_1, y_2)(V_r^*(y_1) \;\wedge\; V_s^*(y_2) \;\wedge\; App(y_1, y_2, x)).$$

In a second step we define the $*$ translation of an $\mathcal{L}_{\mathsf{W}}$ formula A as follows:

4. If A is the formula $\mathsf{W}(t)$ or $t\downarrow$, then A^* is

$$(\exists x)V_t^*(x).$$

5. If A is the formula $(s = t)$, then A^* is

$$(\exists x)(V_s^*(x) \;\wedge\; V_t^*(x)).$$

6. If A is the formula $\neg B$, then A^* is $\neg(B^*)$.

7. If A is the formula $(B \; j \; C)$ for $j \in \{\vee, \wedge, \rightarrow\}$, then A^* is $(B^* \; j \; C^*)$.

8. If A is the formula $(\mathcal{Q}x)B$ for $\mathcal{Q} \in \{\exists, \forall\}$, then A^* is $(\mathcal{Q}x)B^*$.

This finishes the description of the translation $*$ from $\mathcal{L}_{\mathsf{W}}$ into $\mathcal{L}_{\mathsf{p}}$. In a further step we have to verify the $*$ translation of the BOW axioms (1)–(24) and of set induction on W, $(\mathsf{S}\text{-}\mathsf{I}_{\mathsf{W}})$, in the theory $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$. In the following we only discuss axiom (23) for bounded primitive recursion and set induction $(\mathsf{S}\text{-}\mathsf{I}_{\mathsf{W}})$. The remaining axioms are easily verified by making use of Theorem 43.

Let us first turn to the bounded recursor r_{W}, and let us show the totality of r_{W} in $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$. We will realize the crucial role of Σ reflection $(\Sigma\text{-}\mathsf{Ref})$ for the first time.

Lemma 44 *The $*$ translation of axiom (23) about r_{W} is provable in the theory $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$, i.e. $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$ proves*

$$[(f \in \mathsf{W} \rightarrow \mathsf{W}) \;\wedge\; (g \in \mathsf{W}^3 \rightarrow \mathsf{W}) \;\wedge\; (b \in \mathsf{W}^2 \rightarrow \mathsf{W}) \;\rightarrow\; (\mathsf{r}_{\mathsf{W}} fgb \in \mathsf{W}^2 \rightarrow \mathsf{W})]^*.$$

Proof. In the sequel we work informally in the theory $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$ and assume

$$(f \in \mathsf{W} \rightarrow \mathsf{W})^*, \tag{1}$$
$$(b \in \mathsf{W}^2 \rightarrow \mathsf{W})^*, \tag{2}$$
$$(g \in \mathsf{W}^3 \rightarrow \mathsf{W})^*. \tag{3}$$

If we spell out (1), (2) and (3) according to the translation $*$, we obtain

$$(\forall x)(\exists z)App(f, x, z), \tag{4}$$
$$(\forall x, w)(\exists z)App_2(b, x, w, z), \tag{5}$$
$$(\forall x, w, v)(\exists z)App_3(g, x, w, v, z). \tag{6}$$

It is our aim to show $(\mathsf{r_W} fgb \in \mathsf{W}^2 \to \mathsf{W})^*$, i.e.

$$(\forall x, w)(\exists z) App(\langle \hat{\mathsf{r}}_\mathsf{W}, f, g, b, x\rangle, w, z), \tag{7}$$

which by Theorem 43 is equivalent to

$$(\forall x, w)(\exists z) Rec_{App}(f, g, b, x, w, z). \tag{8}$$

In the sequel fix arbitrary x_0 and y_0. Furthermore, by (4) choose z_0 so that $App(f, x_0, z_0)$. Now we obtain from (5) and Σ reflection (Σ-Ref) an a_1 so that

$$(\forall w \subseteq y_0)(\exists z \leq a_1) App_2^{a_1}(b, x_0, w, z). \tag{9}$$

If we set $a_2 = z_0 a_1$, then (6) and another application of (Σ-Ref) provide us with an a_3 so that

$$(\forall w \subseteq y_0)(\forall v \leq a_2)(\exists z \leq a_3) App_3^{a_3}(g, x_0, w, v, z). \tag{10}$$

Now set $a_4 = SqBd(a_2, y_0)$ and consider the statement $\widetilde{Rec}_{App}(f, g, b, x_0, y, z)$, which is given by the formula

$$
\begin{aligned}
&\widetilde{Rec}_{App}(f, g, b, x_0, y, z) \ := \\
&\quad (\exists v \leq a_4)[Seq(v) \ \wedge \ lh(v) = |y|1 \ \wedge \ (v)_\epsilon = z_0 \ \wedge \\
&\qquad (\forall w \subseteq y)(w \neq \epsilon \to \\
&\qquad\quad (\exists u_1 \leq a_3)(\exists u_2 \leq a_1)[App_3^{a_3}(g, x_0, w, (v)_{|p(w)|}, u_1) \ \wedge \\
&\qquad\qquad App_2^{a_1}(b, x_0, w, u_2) \ \wedge \\
&\qquad\qquad\quad (v)_{|w|} = u_1 | u_2]) \\
&\qquad \wedge \ z = (v)_{|y|}].
\end{aligned}
$$

In the following let us write $A(y)$ for the $\mathcal{L}_\mathsf{p}$ formula which is given by

$$y \subseteq y_0 \ \to \ (\exists z \leq a_2)\widetilde{Rec}_{App}(f, g, b, x_0, y, z).$$

Then one easily verifies that (9) and (10) imply

$$A(\epsilon) \ \wedge \ (\forall y)(A(y) \ \to \ A(y0) \wedge A(y1)). \tag{11}$$

Since $A(y)$ is an extended Σ_1^b formula of $\mathcal{L}_\mathsf{p}$, induction is available in PTCA^+ for A. Hence, (11) implies $A(y_0)$, from which we immediately derive

$$(\exists z) Rec_{App}(f, g, b, x_0, y_0, z). \tag{12}$$

Since x_0 and y_0 were arbitrary, we have shown (8), and this finishes our proof. □

In a next step we show that the $*$ translation of set induction is provable in the system $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$. Again the presence of Σ reflection (Σ-Ref) is crucial: the requirement $f \in \mathcal{P}(\mathsf{W})$ allows one to "reflect" Σ_1^0 induction by Σ_1^b induction.

Lemma 45 *The $*$ translation of set induction* $(\mathsf{S\text{-}I_W})$ *is provable in the system* $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$, *i.e.* $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$ *proves*

$$[f \in \mathcal{P}(\mathsf{W}) \ \wedge\ f\epsilon = \mathsf{0} \ \wedge\ (\forall x \in \mathsf{W})(f(\mathsf{p_W}x) = \mathsf{0} \ \rightarrow\ fx = \mathsf{0}) \ \rightarrow\ (\forall x \in \mathsf{W})(fx = \mathsf{0})]^*.$$

Proof. Let us work informally in $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$. Assume the $*$ translations of $f \in \mathcal{P}(\mathsf{W})$, $f\epsilon = \mathsf{0}$ and $(\forall x \in \mathsf{W})(f(\mathsf{p_W}x) = \mathsf{0} \rightarrow fx = \mathsf{0})$. Hence, we get

$$(\forall x)(\exists! y) App(f, x, y), \tag{1}$$

$$App(f, \epsilon, 0), \tag{2}$$

$$(\forall x)[App(f, x, 0) \ \rightarrow\ App(f, x0, 0) \wedge App(f, x1, 0)]. \tag{3}$$

Now fix an arbitrary x_0. By Σ reflection $(\Sigma\text{-}\mathsf{Ref})$ there exists an a so that

$$(\forall x \subseteq x_0)(\exists y \leq a) App^a(f, x, y). \tag{4}$$

As an immediate consequence we get that

$$(\forall x \subseteq x_0)(\forall y)[App(f, x, y) \ \leftrightarrow\ App^a(f, x, y)]. \tag{5}$$

Let us now write $A(x)$ for the extended Σ^b_1 statement

$$x \subseteq x_0 \ \rightarrow\ App^a(f, x, 0).$$

Then one easily derives $(\forall x)A(x)$ by Σ^b_1 induction, making use of (2), (3) and (5). Hence, we have obtained

$$App^a(f, x_0, 0), \tag{6}$$

and since x_0 was arbitrary, we have derived the $*$ translation of $(\forall x \in \mathsf{W})(fx = \mathsf{0})$ in $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$. This finishes our proof. □

The reader may have noticed that in the proofs of Lemma 44 and Lemma 45 we did not make use of the full strength of the Σ reflection principle $(\Sigma\text{-}\mathsf{Ref})$. In fact, reflection is only needed for formulas of the shape $(\forall x \leq y)A$, so that each positive universal and each negative existential quantifier in A is sharply bounded. We can also dispense with the initial universal bounded quantifier, expect for obtaining the bound a_3 in equation (10) of the proof of Lemma 44. Similar remarks will apply to the treatment of the theory PTO^+ in Section 3.4, cf. the proof of Lemma 58. However, the *full* Σ reflection principle will be needed for analyzing the theory $\mathsf{PTO}^+ + (\Sigma^+\text{-}\mathsf{CP_W})$ at the end of Section 3.4. For reasons of notational simplicity, we refrained from displaying the fine structure of Σ reflection in the formulation of theorems and proofs. This is perfectly justified by the fact that full Σ reflection does not take us beyond polynomial strength, cf. Corollary 40.

We are now in a position to state the following embedding theorem.

Theorem 46 *We have for all* $\mathcal{L}_\mathsf{W}$ *formulas* A*:*

$$\mathsf{PTO} \vdash A \quad \Longrightarrow \quad \mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref}) \vdash A^*.$$

From Corollary 40 and Proposition 42 we get the following equivalences. Here "≡" denotes a natural adaptation to our weak setting of Feferman's [22] notion of proof-theoretic equivalence.

Corollary 47 *We have the following proof-theoretic equivalences:*

$$\mathsf{PTO} \equiv \mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref}) \equiv \mathsf{PTCA}.$$

From Corollary 40 and the fact that an $\mathcal{L}_\mathsf{W}$ formula of the form $(\forall \vec{x} \in \mathsf{W})(t\vec{x} \in \mathsf{W})$ translates into a Π^0_2 statement under $*$, we get the following crucial corollary.

Corollary 48 *Suppose that t is a closed term of $\mathcal{L}_\mathsf{W}$ so that*

$$\mathsf{PTO} \vdash (\forall \vec{x} \in \mathsf{W})(t\vec{x} \in \mathsf{W}).$$

Then t defines a polytime function on $\mathbb{W}$.

3.3.3 The proof of the fixed point theorem

In this paragraph we give a detailed proof of Theorem 43. In particular, we show that the operator form $\mathcal{A}(Q, x, y, z)$ has a Σ^0_1 fixed point App which is functional, provably in $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$. As already indicated, App will be constructed from below by making use of a specific computability predicate $Comp_\mathcal{A}(c)$, expressing that c is a computation sequence with respect to the operator form $\mathcal{A}$. Informally, a computation sequence c with respect to $\mathcal{A}$ is a sequence $c = \langle (c)_0, \ldots, (c)_{p(lh(c))} \rangle$ so that each $(c)_a$ is a sequence $\langle (c)_{a,0}, (c)_{a,1}, (c)_{a,2} \rangle$ of length 3 with the intended meaning that $(c)_{a,0}$ applied to $(c)_{a,1}$ yields $(c)_{a,2}$ in the sense of $\mathcal{A}$ and, moreover, this is computed or "proved" by $\langle (c)_0, \ldots, (c)_{p(a)} \rangle$.

Let us first define $\mathcal{L}_\mathsf{p}$ formulas $App_n(f, x_1, \ldots, x_n, y, a, c)$[3] for each $n \geq 1$ by induction on n as follows:

$$\begin{aligned} App_1(f, x_1, y, a, c) &:= (\exists b \subset a)((c)_b = \langle f, x_1, y \rangle), \\ App_{n+1}(f, x_1, \ldots, x_{n+1}, y, a, c) &:= \\ &(\exists z \leq c)(\exists b \subset a)[App_n(f, x_1, \ldots, x_n, z, a, c) \wedge (c)_b = \langle z, x_{n+1}, y \rangle]. \end{aligned}$$

The intended meaning of $App_n(f, x_1, \ldots, x_n, y, a, c)$ is that $f x_1 \ldots x_n \simeq y$ with respect to the sequence c restricted to the entries with index smaller than a.

Remark 49 *$App_n(f, x_1, \ldots, x_n, y, a, c)$ is an extended Σ^b_1 formula.*

In a next step we define an $\mathcal{L}_\mathsf{p}$ formula $Rec_{App}(f, g, b, x, y, z, a, c)$. It defines the graph of the function which is defined from f and g by bounded primitive recursion

[3] In the sequel it will always be clear from the number of parameters shown whether we mean $App_n(f, x_1, \ldots, x_n, y, a, c)$ or $App_n(f, x_1, \ldots, x_n, y)$.

with length bound b in the sense of the computation sequence c with entry indices smaller than a.

$$
\begin{aligned}
&Rec_{App}(f,g,b,x,y,z,a,c) \;:= \\
&\quad (\exists v \le c)[Seq(v) \;\wedge\; lh(v) = |y|1 \;\wedge\; App_1(f,x,(v)_\epsilon,a,c) \;\wedge \\
&\qquad (\forall w \subseteq y)(w \neq \epsilon \rightarrow \\
&\qquad\quad (\exists u_1,u_2)[App_3(g,x,w,(v)_{|p(w)|},u_1,a,c) \;\wedge\; App_2(b,x,w,u_2,a,c) \\
&\qquad\qquad \wedge\; (v)_{|w|} = u_1|u_2]) \\
&\qquad \wedge\; z = (v)_{|y|}].
\end{aligned}
$$

Remark 50 *$Rec_{App}(f,g,b,x,y,z,a,c)$ is an extended Σ^b_1 formula.*

In the following let us write $\mathcal{A}_i(x,y,z)$ for the ith clause of the operator form $\mathcal{A}$ for $i \neq 5$ and $i \neq 26$. We are ready to define the $\mathcal{L}_\mathsf{p}$ formula $Comp_\mathcal{A}$, which expresses that c is a computation sequence in the sense of the operator form $\mathcal{A}$.

$$Comp_\mathcal{A}(c) \;:=\; Seq(c) \;\wedge\; (\forall a \subset lh(c))[Seq_3((c)_a) \;\wedge\; C((c)_{a,0},(c)_{a,1},(c)_{a,2},a)],$$

where $C(x,y,z,a)$ is the disjunction of the $\mathcal{A}_i(x,y,z)$ for $i \neq 5$ and $i \neq 26$ plus the two disjuncts

(5') $Seq_3(x) \;\wedge\; (x)_0 = \hat{\mathsf{s}} \;\wedge$
$(\exists v,w \le c)[App_1((x)_1,y,v,a,c) \wedge App_1((x)_2,y,w,a,c) \wedge App_1(v,w,z,a,c)]$,

(26') $Seq_5(x) \;\wedge\; (x)_0 = \hat{\mathsf{r}}_\mathsf{W} \;\wedge\; Rec_{App}((x)_1,(x)_2,(x)_3,(x)_4,y,z,a,c)$.

Remark 51 *$Comp_\mathcal{A}(c)$ is an extended Σ^b_1 formula.*

Now we are in a position to define the $\mathcal{L}_\mathsf{p}$ formula $App(x,y,z)$, which expresses that there is a computation sequence c whose last entry is $\langle x,y,z\rangle$.

$$App(x,y,z) \;:=\; (\exists c)[Comp_\mathcal{A}(c) \;\wedge\; last(c) = \langle x,y,z\rangle].$$

Remark 52 *$App(x,y,z)$ is equivalent to a Σ^0_1 formula, provably in PTCA^+.*

Remark 53 The reader might ask why we did at all make use of the operator form $\mathcal{A}(Q,x,y,z)$ in Section 3.3.2 instead of giving the above definition directly. The reason is conceptual clarity: the only properties that we have used in order to establish the embedding of PTO into $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$ are the *fixed point* property and the *functionality* property, i.e. the two claims of Theorem 43. This is in full accordance with previous treatments of applicative theories, cf. e.g. Feferman and Jäger [28].

It remains to show that (i) *App* is a fixed point of the operator form $\mathcal{A}$, and (ii) *App* is functional, and in addition, (i) and (ii) are provable in $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$. In the following we work informally in the theory $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$, and we first want to show that *App* is functional.

Lemma 54 $\mathsf{PTCA} \vdash (\forall x, y, z_1, z_2)(App(x, y, z_1) \wedge App(x, y, z_2) \rightarrow z_1 = z_2)$.

Proof. We assume $Comp_{\mathcal{A}}(b) \wedge Comp_{\mathcal{A}}(c)$ and show the Δ_0^b statement

$$v \subset lh(c) \rightarrow (\forall u \subseteq v)(\forall w \subset lh(b))[(b)_w = \langle (c)_{u,0}, (c)_{u,1}, (b)_{w,2} \rangle \rightarrow (b)_{w,2} = (c)_{u,2}]$$

by induction on v. Then our claim immediately follows. If $v = \epsilon$, then one of the clauses $\mathcal{A}_i$ for some i different from 5 and 26 applies, and our assertion is immediate. For the induction step let us assume that our assertion holds for some v; in order to verify it for $v1$, we have to distinguish several cases. If we are again in the case of one of the clauses $\mathcal{A}_i$ for i different from 5 and 26, then our claim follows as above. If clause (5') for the s combinator applies, then we are immediately done by the induction hypothesis. Finally, if we are in the case of clause (26') for $\mathsf{r_W}$, then our assertion follows from the induction hypothesis and an obvious subsidiary induction. This settles our claim about the functionality of App. $\square$

It remains to show that $App(x, y, z)$ defines a fixed point of the positive operator $\mathcal{A}(Q, x, y, z)$, provably in $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$. We split the proof of the fixed point property into the two implications (i) $\mathcal{A}(App, x, y, z) \rightarrow App(x, y, z)$, and (ii) $App(x, y, z) \rightarrow \mathcal{A}(App, y, z)$.

Lemma 55 $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref}) \vdash (\forall x, y, z)(\mathcal{A}(App, x, y, z) \rightarrow App(x, y, z))$.

Proof. Let us assume $\mathcal{A}(App, x, y, z)$. Then exactly one of the clauses (1)–(26) applies. If we have $\mathcal{A}_i(x, y, z)$ for an i different from 5 and 26, then we are done by the computation sequence $c = \langle\langle x, y, z \rangle\rangle$. Now suppose that clause (5) applies. Then we have $Seq_3(x) \wedge (x)_0 = \hat{\mathsf{s}}$, and there exist binary words v and w so that

$$App((x)_1, y, v) \wedge App((x)_2, y, w) \wedge App(v, w, z).$$

The above three conjuncts provide $\mathcal{A}$ computation sequences c_0, c_1 and c_2, and obviously the sequence $c = c_0 \frown c_1 \frown c_2 \frown \langle\langle x, y, z \rangle\rangle$ witnesses $App(x, y, z)$ as desired. Finally, we have to consider clause (26) for $\mathsf{r_W}$. Therefore, assume

$$Seq_5(x) \wedge (x)_0 = \hat{\mathsf{r}}_{\mathsf{W}} \wedge Rec_{App}((x)_1, (x)_2, (x)_3, (x)_4, y, z).$$

Then there exists a v, and by $(\Sigma\text{-}\mathsf{Ref})$ an a so that we have

$$\begin{aligned} & Seq(v) \wedge lh(v) = |y|1 \wedge App^a((x)_1, (x)_4, (v)_\epsilon) \wedge \\ & \quad (\forall w \subseteq y)(w \neq \epsilon \rightarrow \\ & \qquad (\exists u_1, u_2 \leq a)[App_3^a((x)_2, (x)_4, w, (v)_{|p(w)|}, u_1) \wedge App_2^a((x)_3, (x)_4, w, u_2) \wedge \\ & \qquad\qquad (v)_{|w|} = u_1|u_2]) \\ & \quad \wedge z = (v)_{|y|}. \end{aligned}$$

Now it is straightforward to establish the statement

$$\begin{aligned} & y' \subseteq y \rightarrow \\ & \quad (\exists c \leq t(y', a))[Comp_{\mathcal{A}}(c) \wedge Rec_{App}((x)_1, (x)_2, (x)_3, (x)_4, y', (v)_{|y'|}, p(lh(c)), c)] \end{aligned}$$

by induction on y', where $t(y', a)$ is a suitable $\mathcal{L}_{\mathsf{e}}$ term which provides an upper bound for the length of c (as a binary word). For example, choose the term $t(y', a)$ as $(aaaaa\overline{8} \times y'1)$. By setting $y' = y$, there now exists an $\mathcal{A}$ computation sequence c_y so that $Rec_{App}((x)_1, (x)_2, (x)_3, (x)_4, y, z, p(lh(c_y)), c_y)$. Our argument is finished, since the sequence $c'_y = c_y \frown \langle\langle x, y, z\rangle\rangle$ witnesses $App(x, y, z)$. $\square$

Our last aim is to show the other direction of the fixed point property.

Lemma 56 $\mathsf{PTCA} \vdash (\forall x, y, z)(App(x, y, z) \to \mathcal{A}(App, x, y, z))$.

Proof. Suppose $App(x, y, z)$ holds for some binary words x, y and z. Hence, there exists a sequence c so that

$$Comp_{\mathcal{A}}(c) \wedge last(c) = \langle x, y, z\rangle.$$

If $\mathcal{A}_i(x, y, z)$ holds for some i different from 5 and 26, then our claim is trivial. If $\langle x, y, z\rangle$ was computed according to clause (5'), then an obvious decomposition of c yields the desired result. Finally, let us consider the case where we have

$$Seq_5(x) \wedge (x)_0 = \hat{\mathsf{r}}_{\mathsf{W}} \wedge Rec_{App}((x)_1, (x)_2, (x)_3, (x)_4, y, z, p(lh(c)), c).$$

Then an easy decomposition of c yields $Rec_{App}((x)_1, (x)_2, (x)_3, (x)_4)$ as desired. $\square$

This ends the proof of Theorem 43.

3.4 The theory PTO^+

In this paragraph we propose an extension PTO^+ of PTO, which results from PTO by strengthening set induction to a form of complete induction on W which is related to NP induction, though it is formally much stronger. Furthermore, we briefly address a collection principle which does not raise the proof-theoretic strength of PTO^+.

In the following let the $\mathcal{L}_{\mathsf{W}}$ formula $NP(f, g, x)$ be given by

$$NP(f, g, x) \;:=\; (\exists y \leq fx)(gxy = \mathsf{0}).^4$$

In addition, $\mathcal{P}(\mathsf{W}^2)$ denotes the obvious generalization of $\mathcal{P}(\mathsf{W})$ to binary (curried) characteristic functions on W, i.e.

$$f \in \mathcal{P}(\mathsf{W}^2) \;:=\; (\forall x, y \in \mathsf{W})(fxy = \mathsf{0} \vee fxy = \mathsf{1}).$$

Then PTO^+ is defined to be PTO, where set induction (S-I_{W}) is replaced by the NP induction axiom (NP-I_{W}):

$$f \in \mathsf{W} \to \mathsf{W} \wedge g \in \mathcal{P}(\mathsf{W}^2) \wedge NP(f, g, \epsilon) \wedge (\forall x \in \mathsf{W})(NP(f, g, \mathsf{p_W}x) \to NP(f, g, x))$$
$$\to (\forall x \in \mathsf{W})NP(f, g, x).$$

[4] Bounded quantifiers are understood to be restricted to W.

It is easy to see that set induction ($\mathsf{S\text{-}I_W}$) in fact follows from the above induction principle ($\mathsf{NP\text{-}I_W}$).

We know from Proposition 42 that PTCA is contained in PTO via the translation $(\cdot)^{\mathsf{W}}$. By making use of Lemma 41, it is now straightforward to verify that PTO^+ validates the *NP* induction principle of PTCA^+ with respect to $(\cdot)^{\mathsf{W}}$. Hence, the following analogue of Proposition 42 holds.

Proposition 57 *We have for every $\mathcal{L}_\mathsf{p}$ formula $A(\vec{x})$ with at most $\vec{x}$ free:*

$$\mathsf{PTCA}^+ \vdash A(\vec{x}) \quad \Longrightarrow \quad \mathsf{PTO}^+ \vdash \vec{x} \in \mathsf{W} \to A^{\mathsf{W}}(\vec{x}).$$

On the other hand, we will now show that PTO^+ is not stronger than PTO. In particular, we establish the $*$ translation of ($\mathsf{NP\text{-}I_W}$) in $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$.

Lemma 58 *The $*$ translation of* ($\mathsf{NP\text{-}I_W}$) *is provable in* $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$.

Proof. In the following let us work informally in $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$ and assume the $*$ translation of the premise of ($\mathsf{NP\text{-}I_W}$). The assumptions $(f \in \mathsf{W} \to \mathsf{W})^*$ and $(g \in P(\mathsf{W}^2))^*$ yield

$$(\forall x)(\exists! z) App(f, x, z), \tag{1}$$

$$(\forall x, y)(\exists! z) App_2(g, x, y, z). \tag{2}$$

In the sequel fix an arbitrary x_0. By (1) and (Σ-Ref) there exists an a_1 so that

$$(\forall x \subseteq x_0)(\exists z \leq a_1) App^{a_1}(f, x, z). \tag{3}$$

In addition, (2) and (Σ-Ref) provide us with an a_2 so that

$$(\forall x \subseteq x_0)(\forall y \leq a_1)(\exists z \leq a_2) App_2^{a_2}(g, x, y, z). \tag{4}$$

In the following we write $A(f, g, x)$ for the formula

$$(\exists z \leq a_1)(\exists y \leq z)[App_1^{a_1}(f, x, z) \ \wedge \ App_2^{a_2}(g, x, y, 0)].$$

Then it is straightforward to check from (3) and (4) that

$$(\forall x \subseteq x_0)[NP^*(f, g, x) \ \leftrightarrow \ A(f, g, x)]. \tag{5}$$

On the other hand, we have assumed

$$NP^*(f, g, \epsilon), \tag{6}$$

$$(\forall x)(NP^*(f, g, x) \ \to \ NP^*(f, g, x0) \wedge NP^*(f, g, x1)). \tag{7}$$

Hence, we can derive $(\forall x)B(x)$ by Σ_1^b induction from (5), (6) and (7), where $B(x)$ denotes the formula

$$x \subseteq x_0 \ \to \ A(f, g, x).$$

We have shown $NP^*(f, g, x_0)$, and since x_0 was arbitrary, this finishes our proof. □

The following analogue of Theorem 46 has been established.

Theorem 59 *We have for all $\mathcal{L}_\mathsf{W}$ formulas A:*

$$\mathsf{PTO}^+ \vdash A \quad \Longrightarrow \quad \mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref}) \vdash A^*.$$

From Corollary 40 and Proposition 57 we can derive the same corollaries as in the previous section.

Corollary 60 *We have the following proof-theoretic equivalences:*

$$\mathsf{PTO}^+ \equiv \mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref}) \equiv \mathsf{PTCA}.$$

Corollary 61 *Suppose that t is a closed term of $\mathcal{L}_\mathsf{W}$ so that*

$$\mathsf{PTO}^+ \vdash (\forall \vec{x} \in \mathsf{W})(t\vec{x} \in \mathsf{W}).$$

Then t defines a polytime function on $\mathbb{W}$.

We finish this section by formulating a collection principle in $\mathcal{L}_\mathsf{W}$ which does not raise the proof-theoretic strength of PTO^+ either. The class of Σ^+ formulas of $\mathcal{L}_\mathsf{W}$ is defined similarly to the class of Σ^+ formulas of $\mathcal{L}_\mathsf{N}$ (cf. Definition 18):[5]

1. Each atomic formula $\mathsf{W}(t)$, $t\downarrow$ and $(s = t)$ is a Σ^+ formula.
2. If A and B are Σ^+ formulas, then so also are $(A \vee B)$ and $(A \wedge B)$.
3. If A is a Σ^+ formula, then so also are $(\forall x \leq y)A$ and $(\exists x \in \mathsf{W})A$.

Further, the scheme of Σ^+ *collection on* W, $(\Sigma^+\text{-}\mathsf{CP}_\mathsf{W})$, has the form

$$(\Sigma^+\text{-}\mathsf{CP}_\mathsf{W}) \qquad (\forall x \leq y)(\exists z \in \mathsf{W})A \rightarrow (\exists u \in \mathsf{W})(\forall x \leq y)(\exists z \leq u)A,$$

where A is a Σ^+ formula of $\mathcal{L}_\mathsf{W}$.

Now it is easy to verify that $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$ validates the $*$ translation of each instance of $(\Sigma^+\text{-}\mathsf{CP}_\mathsf{W})$ and, therefore, $\mathsf{PTO}^+ + (\Sigma^+\text{-}\mathsf{CP}_\mathsf{W})$ does not go beyond polynomial strength, too. Here the full strength of $(\Sigma\text{-}\mathsf{Ref})$ is needed in order to handle $(\Sigma^+\text{-}\mathsf{CP}_\mathsf{W})$, of course.

Theorem 62 *We have the following proof-theoretic equivalences:*

$$\mathsf{PTO}^+ + (\Sigma^+\text{-}\mathsf{CP}_\mathsf{W}) \equiv \mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref}) \equiv \mathsf{PTCA}.$$

3.5 Additional topics and open problems

In this section we first address the very interesting open problem concerning the axiom of totality (Tot) in the context of PTO, thereby encountering the question whether the Church Rosser property for combinatory logic is provable in feasible arithmetic. In the second paragraph we discuss the status of the principles of operation induction on W and W induction on W over PTO. Finally, the last paragraph contains very brief thoughts on a reformulation of PTO which is based on N instead of W.

[5] To be precise, we should define simultaneously Σ^+ and Π^- formulas.

3.5.1 Totality

The theories PTO and PTO^+ are based on a *partial* form of term application, and the proof-theoretic reduction described in Section 3.3.2 makes substantial use of this fact. The question arises whether the assumption of a *total* application operation does raise the strength of PTO. More precisely, what is the exact proof-theoretic strength of $\mathsf{PTO} + (\mathsf{Tot})$?

We have seen in Section 2.3.2 that totality (Tot) does not raise the strength of various applicative theories of strength at least PRA, and we will prove in the next chapter that this is also true for systems including the non-constructive μ operator. For our upper bound argument in Section 2.3.2 we have made substantial use of the fact that the *Church Rosser property* of our reduction relation $\boldsymbol{\rho}$ is derivable in primitive recursive arithmetic PRA, cf. Proposition 24. If we consider a suitable total term model of PTO which is based on the usual reduction relation for total combinatory logic, then we do not know whether the corresponding Church Rosser property is provable in $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$. Certainly, we do not need full PRA in order to formalize the Church Rosser theorem: recently, Duccio Pianigiani (manuscript in preparation) has shown that a careful formalization of the usual Church Rosser proof can be carried through in $\mathsf{I\Delta_0} + (\mathsf{exp})$. However, it is not clear at the moment how to avoid the use of the exponential function and whether this is possible at all.

Hence, we are left with the two questions whether the Church Rosser property is derivable in a feasible system, and whether $\mathsf{PTO} + (\mathsf{Tot})$ is conservative over PTO. A positive answer to the first of these questions would almost certainly yield a positive answer to the second by combining methods of Sections 2.3.2 and 3.3.2. Moreover, we strongly conjecture that the answer to the second question is positive; in particular, the provably total functions of $\mathsf{PTO} + (\mathsf{Tot})$ are still computable in polynomial time.

Recently, Cantini [10] has established – among other things – that the provably total functions of the system $\mathsf{PTO} + (\mathsf{Tot})$ have *polynomial growth rate* only.[6] His analysis of $\mathsf{PTO} + (\mathsf{Tot})$ makes use of partial cut elimination and an asymmetric interpretation with respect to the W predicate. However, it does not follow from Cantini's argument that the provably total functions of $\mathsf{PTO} + (\mathsf{Tot})$ are computable in *polynomial time*.

3.5.2 The theories $\mathsf{PTO} + (\mathsf{W}\text{-}\mathsf{I_W})$ and $\mathsf{PTO} + (\mathsf{O}\text{-}\mathsf{I_W})$

In the sequel we briefly address extensions of PTO by W induction on W, $(\mathsf{W}\text{-}\mathsf{I_W})$, and operation induction on W, $(\mathsf{O}\text{-}\mathsf{I_W})$. These induction principles are defined in complete analogy to the corresponding principles in $\mathcal{L}_\mathsf{N}$, cf. Section 2.1. We repeat them for reasons of completeness here.

[6] Actually, Cantini establishes this result for a substantial extension of $\mathsf{PTO} + (\mathsf{Tot})$.

W induction on W, $(\mathsf{W}\text{-}\mathsf{I_W})$.

$$f\epsilon \in \mathsf{W} \wedge (\forall x \in \mathsf{W})(f(\mathsf{p_W}x) \in \mathsf{W} \rightarrow fx \in \mathsf{W}) \rightarrow (\forall x \in \mathsf{W})(fx \in \mathsf{W}).$$

Operation induction on W, $(\mathsf{O}\text{-}\mathsf{I_W})$.

$$f\epsilon = 0 \wedge (\forall x \in \mathsf{W})(f(\mathsf{p_W}x) = 0 \rightarrow fx = 0) \rightarrow (\forall x \in \mathsf{W})(fx = 0).$$

We first turn to the discussion of W induction in PTO, cf. also the introduction to this chapter. It follows from our work [67] that $\mathsf{PTO} + (\mathsf{W}\text{-}\mathsf{I_W})$ proves the totality of every *primitive recursive* function and, hence, contains PRA. This is due to the fact that *exponentiation* can be defined in PTO by means of the recursion theorem (Proposition 2), and its totality is provable by $(\mathsf{W}\text{-}\mathsf{I_W})$. Once exponentiation is available, one establishes in a straightforward manner the totality of every primitive recursive function in $\mathsf{PTO} + (\mathsf{W}\text{-}\mathsf{I_W})$, where another application of the recursion theorem provides the corresponding $\mathcal{L}_\mathsf{W}$ terms. Moreover, the methods of Section 2.3.2 allow one to establish an upper bound of $\mathsf{PTO} + (\mathsf{W}\text{-}\mathsf{I_W})$ in $\mathsf{PRA} + (\Sigma^0_1\text{-}\mathsf{I_N})$ so that (Tot) and (Ext) are validated by the corresponding interpretation. Hence, we can state the following theorem.

Theorem 63 *We have the following proof-theoretic equivalences:*

$$\mathsf{PTO} + (\mathsf{W}\text{-}\mathsf{I_W}) \equiv \mathsf{PTO} + (\mathsf{Tot}) + (\mathsf{Ext}) + (\mathsf{W}\text{-}\mathsf{I_W}) \equiv \mathsf{PRA} + (\Sigma^0_1\text{-}\mathsf{I_N}) \equiv \mathsf{PRA}.$$

In the second half of this paragraph we briefly address operation induction on W, $(\mathsf{O}\text{-}\mathsf{I_W})$. One immediately observes that our upper bound argument for PTO does not carry over to $\mathsf{PTO} + (\mathsf{O}\text{-}\mathsf{I_W})$: the requirement $f \in \mathcal{P}(\mathsf{W})$ in the proof of Lemma 45 is essential in order to apply Σ reflection $(\Sigma\text{-}\mathsf{Ref})$. Hence, the question arises whether $\mathsf{PTO} + (\mathsf{O}\text{-}\mathsf{I_W})$ is stronger than PTO. In partial answer to that question, it follows again from Cantini's work in [10] that the provably total functions of the system $\mathsf{PTO} + (\mathsf{O}\text{-}\mathsf{I_W})$ have *polynomial growth rate* only.

It is interesting to compare this last result with Kahle's work in [53]. There he introduces a strengthening $\widehat{\mathsf{BON}}$ of BON so that $(\mathsf{O}\text{-}\mathsf{I_N})$ and $(\mathsf{N}\text{-}\mathsf{I_N})$ are equivalent over $\widehat{\mathsf{BON}}$. The corresponding extension $\widehat{\mathsf{PTO}}$ of PTO is obtained by adding the following strengthening of definition by cases $\mathsf{d_W}$ on W:

$$\mathsf{d_W}xyab \in \mathsf{W} \ \rightarrow \ a \in \mathsf{W} \wedge b \in \mathsf{W}.$$

Then $(\mathsf{O}\text{-}\mathsf{I_W})$ and $(\mathsf{W}\text{-}\mathsf{I_W})$ are equivalent over $\widehat{\mathsf{PTO}}$ by Kahle [53] and, hence, the system $\widehat{\mathsf{PTO}} + (\mathsf{O}\text{-}\mathsf{I_W})$ proves the totality of every primitive recursive function by Theorem 63. Further, one easily verifies that our upper bound argument for PTO carries over to $\widehat{\mathsf{PTO}}$, and the one for $\mathsf{PTO} + (\mathsf{W}\text{-}\mathsf{I_W})$ to $\widehat{\mathsf{PTO}} + (\mathsf{W}\text{-}\mathsf{I_W})$. Summing up, the system $\widehat{\mathsf{PTO}} + (\mathsf{O}\text{-}\mathsf{I_W})$ is strictly stronger than $\widehat{\mathsf{PTO}}$, which in turn is equivalent to PTO. Moreover, $\mathsf{PTO} + (\mathsf{O}\text{-}\mathsf{I_W})$ is strictly contained in $\widehat{\mathsf{PTO}} + (\mathsf{O}\text{-}\mathsf{I_W})$ by [10]. In particular, the methods used by Cantini do not validate the above strengthening of definition by cases $\mathsf{d_W}$ on W. This immediately becomes clear by looking at the asymmetric interpretation argument given in [10].

Theorem 64 *We have the following proof-theoretic equivalences:*

$$\widehat{\mathsf{PTO}} + (\mathsf{O}\text{-}\mathsf{I}_\mathsf{W}) \equiv \widehat{\mathsf{PTO}} + (\mathsf{Tot}) + (\mathsf{Ext}) + (\mathsf{O}\text{-}\mathsf{I}_\mathsf{W}) \equiv \mathsf{PTO} + (\mathsf{W}\text{-}\mathsf{I}_\mathsf{W}) \equiv \mathsf{PRA}.$$

3.5.3 W versus N in PTO

Let us very briefly address the possibility of replacing the basic type W in PTO by the type for the natural numbers N. Our remarks are rather tentative and details are not worked out yet.

We obtain a theory based on the language $\mathcal{L}_\mathsf{N}$ which is formulated analogously to PTO as follows: we have as our base functions addition and multiplication, and we formulate a bounded primitive recursor in the same way as the recursor r_W of PTO, making use of the $\leq$ relation on the naturals to be included in the system under discussion. In addition, we allow set induction on N, $(\mathsf{S}\text{-}\mathsf{I}_\mathsf{N})$. The so-obtained system is strongly conjectured to capture the second level $\mathcal{E}_2$ of the Grzegorczyk hierarchy. It is well-known that a number-theoretic function belongs to $\mathcal{E}_2$, if and only if, it is computable in *linear space*, cf. Rose [58].

3.6 Extension to the Grzegorczyk hierarchy

In this section we address an extension of our approach to the levels $\mathcal{E}_n$ $(n \geq 3)$ of the Grzegorczyk hierarchy. In particular, we sketch applicative theories G_n $(n \geq 3)$ so that the provably total functions of G_n are exactly the number-theoretic functions in $\mathcal{E}_n$. As the formulation of the theories and the corresponding upper bound argument run in complete analogy to PTO, we restrict ourselves to the discussion of the most central aspects of the new theories G_n.

In the following we use a characterization of $\mathcal{E}_n$ which is due to Ritchie (cf. Sieg [62]). For a detailed discussion on the Grzegorczyk hierarchy the reader is referred to Rose [58]. Let us first define the following sequence $(A_n)_{n\in\mathbb{N}}$ of number-theoretic functions:

$$\begin{aligned}
A_0(x,y) &= y+1, \\
A_{n+1}(x,0) &= \begin{cases} x & \text{if } n=0 \\ 0 & \text{if } n=1 \\ 1 & \text{if } 2 \leq n, \end{cases} \\
A_{n+1}(x,y+1) &= A_n(x, A_{n+1}(x,y)).
\end{aligned}$$

A_n denotes the nth branch of the Ackermann function. In particular, A_1, A_2 and A_3 denote addition, multiplication and exponentiation, respectively. Then $\mathcal{E}_n$ is defined to be the smallest class of number-theoretic functions that contains the usual base functions, A_m $(m \leq n)$, and is closed under substitution and bounded primitive recursion. $\mathcal{E}_3$ corresponds to the Kalmar elementary functions.

The applicative theories G_n $(n \geq 3)$ are now obtained as follows. They are based on a straightforward modification of BON, where the recursor $\mathsf{r_N}$ is replaced by a corresponding *bounded* primitive recursor; we include the $\leq$ relation on the naturals in order to formulate this new recursor. Moreover, G_n includes defining equations and totality axioms for the new operations a_m for A_m for each $m \leq n$. Finally, we allow set induction on N, $(\mathsf{S\text{-}I_N})$, in each of the theories G_n.

The systems of arithmetic corresponding to G_n are described in Sieg [62]. These are extension $\mathsf{I\Delta_0} + (\mathcal{U}_n)$ of $\mathsf{I\Delta_0}$ by function symbols for the elements of the set $\mathcal{U}_n := \{A_m : 3 \leq m \leq n\}$, and each number-theoretic function in $\mathcal{E}_n$ can be introduced in $\mathsf{I\Delta_0} + (\mathcal{U}_n)$; hence, $\mathsf{I\Delta_0} + (\mathcal{E}_n)$ is a definitional extension of $\mathsf{I\Delta_0} + (\mathcal{U}_n)$. Moreover, each $\Delta_0(\mathcal{E}_n)$ formula is equivalent to a quantifier free formula in the extended language, provably in $\mathsf{I\Delta_0} + (\mathcal{E}_n)$, cf. [62]. Hence, we see that $\mathsf{I\Delta_0} + (\mathcal{E}_n)$ is contained in G_n by the very methods of Section 3.3.1.

Let us now turn to the discussion of the upper bound of the theories G_n $(n \geq 3)$. The theories $\mathsf{I\Delta_0} + (\mathcal{E}_n)$ are not suited for an upper bound argument, and we need again the principle of Σ reflection, $(\Sigma\text{-}\mathsf{Ref})$; this is formulated in the very same way as in the context of polynomial time computable arithmetic. An inspection of our sketch of Cantini's argument at the end of Section 3.2.2 is convincing enough to see that $\mathsf{I\Delta_0} + (\mathcal{E}_n) + (\Sigma\text{-}\mathsf{Ref})$ is a conservative extension of $\mathsf{I\Delta_0} + (\mathcal{E}_n)$ for Π^0_2 statements, and this is in full accordance with the results of Buss [6] and Ferreira [31]. Now it is immediate to check that our upper bound argument for PTO in $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$ can be adapted in a straightforward manner in order to give a reduction of G_n to $\mathsf{I\Delta_0} + (\mathcal{E}_n) + (\Sigma\text{-}\mathsf{Ref})$ for $n \geq 3$; the details of the argument are almost literally the same as for PTO, and in some sense even easier due to the presence of exponentiation. Hence, we can state the following theorem.

Theorem 65 *Let $n \geq 3$. Then we have the following proof-theoretic equivalences:*

$$\mathsf{G}_n \;\equiv\; \mathsf{I\Delta_0} + (\mathcal{E}_n) + (\Sigma\text{-}\mathsf{Ref}) \;\equiv\; \mathsf{I\Delta_0} + (\mathcal{E}_n).$$

Formulas of the form $(\forall \vec{x} \in \mathsf{N})(t\vec{x} \in \mathsf{N})$ translate into a Π^0_2 statement under our interpretation. Moreover, provable Π^0_2 statements of $\mathsf{I\Delta_0} + (\mathcal{E}_n) + (\Sigma\text{-}\mathsf{Ref})$ are already derivable in $\mathsf{I\Delta_0} + (\mathcal{E}_n)$ by our discussion above. Since the provably total functions of $\mathsf{I\Delta_0} + (\mathcal{E}_n)$ are exactly the functions in $\mathcal{E}_n$ (cf. [62]), we have established the following corollary.

Corollary 66 *Let $n \geq 3$, and suppose that t is a closed term so that*

$$\mathsf{G}_n \vdash (\forall \vec{x} \in \mathsf{N})(t\vec{x} \in \mathsf{N}).$$

Then t defines a function in the nth level $\mathcal{E}_n$ of the Grzegorczyk hierarchy.

Let us finish this section by briefly addressing the question of whether totality (Tot) can conservatively be added to G_n for each $n \geq 3$. The answer to this question is positive: we have seen in Section 3.5.1 that – due to Duccio Pianigiani – the Church

Rosser property for the combinatory reduction relation is derivable in $\mathsf{I\Delta_0} + (\mathsf{exp})$ or, equivalently, $\mathsf{I\Delta_0} + (\mathcal{E}_3)$. Now it is possible to obtain an upper bound of $\mathsf{G}_n + (\mathsf{Tot})$ in $\mathsf{I\Delta_0} + (\mathcal{E}_n) + (\Sigma\text{-}\mathsf{Ref})$ by combining methods of Sections 2.3.2 and 3.3.2 so that we can state the following strengthening of Theorem 65.

Theorem 67 *Let $n \geq 3$. Then we have the following proof-theoretic equivalences:*

$$\mathsf{G}_n + (\mathsf{Tot}) \equiv \mathsf{I\Delta_0} + (\mathcal{E}_n) + (\Sigma\text{-}\mathsf{Ref}) \equiv \mathsf{I\Delta_0} + (\mathcal{E}_n).$$

Moreover, it seems very likely that the methods of Pianigiani's Church Rosser proof in $\mathsf{I\Delta_0} + (\mathsf{exp})$ extend to the corresponding extensional reduction relations so that the last theorem should hold in the presence of extensionality (Ext), too. The details, however, still have to be checked.

Chapter 4

The non-constructive μ operator

The central theme of this last chapter of our thesis are applicative theories based on the non-constructive quantification operator μ or, more familiar, Kleene's type two quantification functional E. We establish the proof-theoretic strength of BON plus μ in the presence of various induction principles on the natural numbers, most interestingly, set induction $(\mathsf{S}\text{-}\mathsf{I}_\mathsf{N})$, various forms of positive induction, and full formula induction $(\mathsf{F}\text{-}\mathsf{I}_\mathsf{N})$. Special emphasis will be put on theories including totality (Tot) and extensionality (Ext), as well as intermediate forms of complete induction, cf. Jäger and Strahm [52, 50]. Corresponding results for theories based on a *partial* form of term application plus $(\mathsf{S}\text{-}\mathsf{I}_\mathsf{N})$ and $(\mathsf{F}\text{-}\mathsf{I}_\mathsf{N})$ are due to Feferman and Jäger [28].

As in Feferman and Jäger [28], upper bounds for applicative theories with μ operator will be obtained via so-called fixed point theories over Peano arithmetic with ordinals; these have been introduced in Jäger [48] and extended in Jäger and Strahm [51] to second order theories with ordinals. In contrast to [28], ordinal theories will not be used in order to describe recursion-theoretic models, but so-called *infinitary term models* together with their corresponding Church Rosser properties. Such models validate totality (Tot) and, possibly, extensionality (Ext).

The exact plan of this chapter is as follows. First, we state the exact axiomatization of μ, and we discuss the standard recursion-theoretic model *ERO* of $\mathsf{BON}(\mu)$ (cf. Section 1.3.1) and, most important for our purpose, the infinitary term model $CTT(\mu)$ of $\mathsf{TON}(\mu)$. In Section 2 we briefly review some predicative subsystems of analysis based on iterated arithmetic comprehension; these will be used in order to measure the proof-theoretic strength of $\mathsf{BON}(\mu)$ plus various induction principles. Section 3 is dedicated to lower bound computations. We briefly review the lower bounds for $\mathsf{BON}(\mu)$ in the presence $(\mathsf{S}\text{-}\mathsf{I}_\mathsf{N})$ and $(\mathsf{F}\text{-}\mathsf{I}_\mathsf{N})$ of [28], and then concentrate on intermediate forms of induction. In particular, we give a detailed well-ordering proof for the systems $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$. In Section 4 we introduce the framework of fixed point theories over Peano arithmetic with ordinals. We state the proof-theoretic strength of all the theories that are relevant in the sequel, and we exemplary carry through the exact proof-theoretic analysis of two such systems. Upper bounds for theories based on μ and totality are established in Section 5 by formalizing the infinitary term model $CTT(\mu)$ in the previously introduced ordinal theories. At the end of that section, we give a complete Church Rosser proof for the reduction relation of $CTT(\mu)$, and we convince ourselves that this proof can be formalized in a suitable fixed point theory with ordinals. Finally, Section 6 contains some short remarks concerning an axiomatization that is directly based on E instead of μ.

4.1 The quantification operator μ

In this section we introduce the axioms for the non-constructive quantification operator μ, and we address recursion-theoretic as well as term models of the theory $\mathsf{BON}(\mu)$.

4.1.1 The theory $\mathsf{BON}(\mu)$

In the sequel we assume that our language $\mathcal{L}_\mathsf{N}$ contains an additional constant μ for the non-constructive quantification operator μ. We follow its axiomatization in Jäger and Strahm [52], which is a slight strengthening of the formulation given in Feferman [19] and Feferman and Jäger [28, 29]. For an axiomatization that is directly based on Kleene's type two functional E instead of μ, we refer to our brief discussion in the last section of this chapter.

The unbounded minimum operator

(μ.1) $\quad (f \in \mathsf{N} \to \mathsf{N}) \leftrightarrow \mu f \in \mathsf{N}$,

(μ.2) $\quad (f \in \mathsf{N} \to \mathsf{N}) \wedge (\exists x \in \mathsf{N})(fx = 0) \to f(\mu f) = 0$.

We write $\mathsf{BON}(\mu)$ instead of $\mathsf{BON} + (\mu.1, \mu.2)$; $\mathsf{TON}(\mu)$ reads similarly.

In the sequel we will be interested in the proof-theoretic strength of the following theories, where special emphasis will be put on the presence of totality (Tot) and extensionality (Ext):

$$\begin{array}{lll} \mathsf{BON}(\mu) + (\mathsf{S}\text{-}\mathsf{I}_\mathsf{N}) & \mathsf{BON}(\mu) + (\mathsf{O}\text{-}\mathsf{I}_\mathsf{N}) & \mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N}) \\ \mathsf{BON}(\mu) + (\Sigma^+\text{-}\mathsf{I}_\mathsf{N}) & \mathsf{BON}(\mu) + (\mathsf{F}^+\text{-}\mathsf{I}_\mathsf{N}) & \mathsf{BON}(\mu) + (\mathsf{F}\text{-}\mathsf{I}_\mathsf{N}). \end{array}$$

Before we begin with our proof-theoretic investigations, let us briefly discuss some models of the theory $\mathsf{BON}(\mu)$.

4.1.2 Models of $\mathsf{BON}(\mu)$

Let us briefly address the standard models of the theory $\mathsf{BON}(\mu)$, namely recursion-theoretic models and term models.

The recursion-theoretic model

We have already met the standard recursion-theoretic model *ERO* of $\mathsf{BON}(\mu)$ in Section 1.3.1. It is based on application of partial Π^1_1 functions or, alternatively, partial recursive function application in Kleene's type two functional E. Recall that sets in the sense of $\mathcal{P}(\mathsf{N})$ are exactly the hyperarithmetic sets, and semi-sets (r.e. sets) the Π^1_1 sets in *ERO*. Observe that E and μ are interdefinable in this standard recursion-theoretic framework, of course. For a (formalized) description of *ERO* which is directly based on μ, cf. Feferman and Jäger [28]. Obviously, $ERO \not\models (\mathsf{Tot})$ and $ERO \not\models (\mathsf{Ext})$.

Term models

Let us now turn to the description of the total term model $CTT(\mu)$ of the theory $\mathsf{BON}(\mu)+(\mathsf{Tot})$. This is defined in complete analogy to CTT, with the only exception that it is based on the notion of reduction $\boldsymbol{\rho} \cup \boldsymbol{\mu}$, or $\boldsymbol{\rho\mu}$ for short, where $\boldsymbol{\mu}$ is a new set of redex-contractum pairs.

Definition 68 The stages $\boldsymbol{\mu}_\alpha$ of the μ redex-contractum pairs are defined by transfinite recursion on the ordinals and generated by the following two clauses (1) and (2), where t is a closed $\mathcal{L}_\mathsf{N}$ term and $k, l, m \in \mathbb{N}$:

(1) If $t\overline{m}\ \boldsymbol{\tau}_\alpha\ \overline{0}$ and $(\forall k)(\exists l)[t\overline{k}\ \boldsymbol{\tau}_\alpha\ \overline{l} \wedge (k < m \rightarrow l > 0)]$, then $\mu t\ \boldsymbol{\mu}_\alpha\ \overline{m}$,

(2) If $(\forall k)(\exists l > 0)(t\overline{k}\ \boldsymbol{\tau}_\alpha\ \overline{l})$, then $\mu t\ \boldsymbol{\mu}_\alpha\ \overline{0}$,

where

$$\boldsymbol{\tau}_\alpha = \bigcup_{\beta<\alpha} \twoheadrightarrow_{\boldsymbol{\rho\mu}_\beta} .$$

This finishes the specification of our new reduction relation $\twoheadrightarrow_{\boldsymbol{\rho\mu}}$ by taking $\boldsymbol{\mu}$ as $\bigcup_\alpha \boldsymbol{\mu}_\alpha$.

Now it is crucial to verify that $\twoheadrightarrow_{\boldsymbol{\rho\mu}}$ has the Church Rosser property. Since we will need a formalized version of $CTT(\mu)$ in Section 4.5 for proving upper bounds of theories based on $\mathsf{BON}(\mu)$ plus totality, we postpone the Church Rosser proof until Section 4.5; then we will provide the necessary details of this long and tedious proof together with indications concerning its formalization in a certain fixed point theory with ordinals. Nevertheless, let us state the theorem for now.

Theorem 69 *The reduction relation* $\twoheadrightarrow_{\boldsymbol{\rho\mu}}$ *has the Church Rosser property.*

As a corollary we get that $CTT(\mu)$ is a model of $\mathsf{BON}(\mu)$, and the axiom of totality is satisfied in $CTT(\mu)$, of course.

Corollary 70 $CTT(\mu) \models \mathsf{BON}(\mu) + (\mathsf{Tot})$.

We finish this section by mentioning that we again obtain an extensional version $TTE(\mu)$ of $CTT(\mu)$ by strengthening $\boldsymbol{\rho}$ to its extensional version $\boldsymbol{\beta\eta}$. The Church Rosser proof for $\twoheadrightarrow_{\boldsymbol{\rho\mu}}$ then extends to $\twoheadrightarrow_{\boldsymbol{\beta\eta\mu}}$.

4.2 Predicative subsystems of analysis

For the sake of completeness, let us briefly recapitulate the definition of some well-known subsystems of second order arithmetic, which will be relevant in the sequel.

Let $\mathcal{L}_2$ denote the usual language of second order arithmetic, which extends $\mathcal{L}_1$ by set variables $X, Y, Z, \ldots$ (possibly with subscripts) and the binary relation symbol $\in$ for elementhood between numbers and sets. Terms and formulas of $\mathcal{L}_2$ are defined as usual. We write $s \in (X)_t$ for $\langle s, t\rangle \in X$. An $\mathcal{L}_2$ formula is called *arithmetic*, if it does not contain bound set variables; let Π^0_∞ denote the class of arithmetic $\mathcal{L}_2$ formulas.

For the definition of theories with iterated arithmetic comprehension we refer to a primitive recursive standard well-ordering $\prec$ of order type Γ_0 with least element 0 and field $\mathbb{N}$, and the reader is assumed to be familiar with the Veblen functions φ_α, cf. Pohlers [56] or Schütte [59] for precise definitions. Furthermore, let $\prec_n$ denote the restriction of $\prec$ to $\{m : m \prec n\}$, and let us write $TI(\alpha, A)$ for the formula

$$(\forall x)((\forall y \prec x)A(y) \to A(x)) \;\to\; (\forall x \prec n)A(x),$$

provided that the order type of $\prec_n$ is α.

Let $A(X, y)$ be an arithmetic $\mathcal{L}_2$ formula with at most X, y free. Then the *A jump hierarchy along $\prec_n$ starting with X* is given by the following transfinite recursion:

$$\begin{aligned}(Y)_0 &:= X,\\ (Y)_i &:= \{\langle m, j\rangle : j \prec i \wedge A((Y)_j, m)\}\end{aligned}$$

for all $0 \prec i \prec n$, and we write $\mathcal{H}_A(X, Y, n)$ for the arithmetic $\mathcal{L}_2$ formula which formalizes this definition.

If α is an ordinal less than Γ_0, then we write $(\Pi^0_\infty\text{-}\mathsf{CA})_\alpha$ for the system of second order arithmetic which extends PA by $TI(\alpha, A)$ for all $\mathcal{L}_2$ formulas A plus

$$(\forall X)(\exists Y)\mathcal{H}_B(X, Y, n)$$

for all *arithmetic* $\mathcal{L}_2$ formulas $B(X, y)$ with at most X, y free, where the order type of $\prec_n$ is α. $(\Pi^0_\infty\text{-}\mathsf{CA})_{<\alpha}$ denotes the union of the theories $(\Pi^0_\infty\text{-}\mathsf{CA})_\beta$ for $\beta < \alpha$. For more information about theories with iterated comprehension the reader is referred to [17, 35, 63].

We finish this section by mentioning a theorem concerning the proof-theoretic ordinal of theories with iterated arithmetic comprehension. The proof follows by the methods of Schütte [59].

Theorem 71 *Let α be a limit ordinal less than Γ_0. Then the system of second order arithmetic $(\Pi^0_\infty\text{-}\mathsf{CA})_{<\omega^\alpha}$ has proof-theoretic ordinal $\varphi\alpha 0$.*

In particular, the proof-theoretic ordinals of $(\Pi^0_\infty\text{-}\mathsf{CA})_{<\omega^\omega}$ and $(\Pi^0_\infty\text{-}\mathsf{CA})_{<\varepsilon_0}$ are $\varphi\omega 0$ and $\varphi\varepsilon_0 0$, respectively.

4.3 Lower bounds

In this section we give a thorough discussion of the proof-theoretic lower bounds for $\mathsf{BON}(\mu)$ plus various forms of complete induction on the natural numbers. In the first paragraph we extend our embedding $(\cdot)^{\mathsf{N}}$ of $\mathcal{L}_1$ to an embedding of $\mathcal{L}_2$ into $\mathcal{L}_{\mathsf{N}}$; we address a crucial application of the unbounded μ operator, namely elimination of number quantifiers. The second part of this section contains a brief sketch of the lower bounds for $\mathsf{BON}(\mu)$ in the presence of set induction $(\mathsf{S}\text{-}\mathsf{I}_{\mathsf{N}})$ and full formula induction $(\mathsf{F}\text{-}\mathsf{I}_{\mathsf{N}})$ as they are established in Feferman and Jäger [28]. In particular, $\mathsf{BON}(\mu) + (\mathsf{S}\text{-}\mathsf{I}_{\mathsf{N}})$ contains PA and $(\Pi^0_\infty\text{-}\mathsf{CA})_{<\varepsilon_0}$ is interpretable into $\mathsf{BON}(\mu) + (\mathsf{F}\text{-}\mathsf{I}_{\mathsf{N}})$. The third paragraph, finally, contains a lower bound of $\mathsf{BON}(\mu)$ in the presence of intermediate forms of induction, cf. [50]. In particular, we give a wellordering proof for each initial segment of $\varphi\omega 0$ in the system $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_{\mathsf{N}})$; together with a recent result of Kahle [53], this will also determine a lower bound for $\mathsf{BON}(\mu) + (\mathsf{O}\text{-}\mathsf{I}_{\mathsf{N}})$. Hence, $(\Pi^0_\infty\text{-}\mathsf{CA})_{<\omega^\omega}$ is reducible to $\mathsf{BON}(\mu)$ plus $(\mathsf{N}\text{-}\mathsf{I}_{\mathsf{N}})$ and $(\mathsf{O}\text{-}\mathsf{I}_{\mathsf{N}})$, respectively.

4.3.1 Embedding $\mathcal{L}_2$ into $\mathcal{L}_{\mathsf{N}}$

In Section 2.3.1 we have discussed an embedding $(\cdot)^{\mathsf{N}}$ of the language of first order arithmetic $\mathcal{L}_1$ into the language $\mathcal{L}_{\mathsf{N}}$. We now extended this embedding to the language $\mathcal{L}_2$ as follows. The set variables of $\mathcal{L}_2$ are supposed to range over $\mathcal{P}(\mathsf{N})$ and, accordingly, an atomic $\mathcal{L}_2$ formula $(x \in Y)$ is translated into $(yx = \mathsf{0})$, where x and y are the variables of $\mathcal{L}_{\mathsf{N}}$ which are associated to the variables x and Y of $\mathcal{L}_2$, respectively. Hence, the extended translation $(\cdot)^{\mathsf{N}}$ is such that

$$((\exists X)A(X))^{\mathsf{N}} = (\exists x \in \mathcal{P}(\mathsf{N}))A^{\mathsf{N}}(x),$$

and similarly for universal quantifiers. In order to simplify the notation, we identify terms and formulas of $\mathcal{L}_2$ and their translation into $\mathcal{L}_{\mathsf{N}}$ when there is no danger of confusion. In addition, we freely use symbols for primitive recursive relations, which are introduced as usual via their characteristic functions.

This is the right place to mention a crucial application of the unbounded μ operator, namely elimination of number quantifiers (cf. [28]). The following lemma is the analogue of Lemma 21 in the context of μ. Its proof proceeds straightforwardly by induction on the complexity of A.

Lemma 72 For every arithmetic $\mathcal{L}_2$ formula $A(\vec{X}, \vec{y})$ with at most $\vec{X}, \vec{y}$ free there exists an individual term t_A of $\mathcal{L}_{\mathsf{N}}$ so that

1. $\mathsf{BON}(\mu) \vdash (\forall \vec{x} \in \mathcal{P}(\mathsf{N}))(\forall \vec{y} \in \mathsf{N})(t_A\vec{x}\vec{y} = \mathsf{0} \vee t_A\vec{x}\vec{y} = \mathsf{1})$,
2. $\mathsf{BON}(\mu) \vdash (\forall \vec{x} \in \mathcal{P}(\mathsf{N}))(\forall \vec{y} \in \mathsf{N})(A^{\mathsf{N}}(\vec{x}, \vec{y}) \leftrightarrow t_A\vec{x}\vec{y} = \mathsf{0})$.

We finish this paragraph with a remark concerning our notion of set in the applicative framework. Sometimes it will be convenient to work with a slightly more general notion of set (cf. Section 4.3.3). According to this generalization, a set is not necessarily an element of $\mathcal{P}(\mathsf{N})$ but an element of $(\mathsf{N} \to \mathsf{N})$, and as before, an object x belongs to a set $(f \in \mathsf{N} \to \mathsf{N})$ if and only if $(fx = \mathsf{0})$. It is easily seen that these two notions of a set are equivalent. In particular, $\mathsf{BON} + (\mathsf{S\text{-}I_N})$ proves set induction for "extended sets",

$$(f \in \mathsf{N} \to \mathsf{N}) \wedge f\mathsf{0} = \mathsf{0} \wedge (\forall x \in \mathsf{N})(fx = \mathsf{0} \to f(x') = \mathsf{0}) \to (\forall x \in \mathsf{N})(fx = \mathsf{0}).$$

Moreover, Lemma 21 and Lemma 72 are easily seen to hold for extended sets, too. Therefore, we will tacitly use both $\mathcal{P}(\mathsf{N})$ and $(\mathsf{N} \to \mathsf{N})$ as our notion of set, whichever is more convenient.

4.3.2 Lower bounds for $\mathsf{BON}(\mu) + (\mathsf{S\text{-}I_N})$ and $\mathsf{BON}(\mu) + (\mathsf{F\text{-}I_N})$

In this paragraph we briefly address the lower bounds of the systems $\mathsf{BON}(\mu)+(\mathsf{S\text{-}I_N})$ and $\mathsf{BON}(\mu) + (\mathsf{F\text{-}I_N})$. The lower bound PA for $\mathsf{BON}(\mu) + (\mathsf{S\text{-}I_N})$ is immediate, and a detailed reduction of $(\Pi^0_\infty\text{-}\mathsf{CA})_{<\varepsilon_0}$ to $\mathsf{BON}(\mu) + (\mathsf{F\text{-}I_N})$ is given in Feferman and Jäger [28]. We only give a very brief sketch of this proof here; similar techniques will be presented in detail when we give the well-ordering proof for the system $\mathsf{BON}(\mu) + (\mathsf{N\text{-}I_N})$ in the next paragraph.

Let us first turn to the system $\mathsf{BON}(\mu) + (\mathsf{S\text{-}I_N})$. By Lemma 72 we know that each $\mathcal{L}_1$ formula A can be represented as a set in the sense of $\mathcal{P}(\mathsf{N})$ in $\mathsf{BON}(\mu)$. Hence, induction for $\mathcal{L}_1$ formulas in PA translates into $(\mathsf{S\text{-}I_N})$ under the translation $(\cdot)^{\mathsf{N}}$, and we can state the following extension of Proposition 22, cf. [28].

Proposition 73 *We have for every $\mathcal{L}_1$ formula $A(\vec{x})$ with at most $\vec{x}$ free:*

$$\mathsf{PA} \vdash A(\vec{x}) \quad \Longrightarrow \quad \mathsf{BON}(\mu) + (\mathsf{S\text{-}I_N}) \vdash \vec{x} \in \mathsf{N} \to A^{\mathsf{N}}(\vec{x}).$$

In a next step we briefly discuss the embedding of the system $(\Pi^0_\infty\text{-}\mathsf{CA})_{<\varepsilon_0}$ into $\mathsf{BON}(\mu) + (\mathsf{F\text{-}I_N})$. For the precise proof the reader is referred to Feferman and Jäger [28]. We must prove in $\mathsf{BON}(\mu) + (\mathsf{F\text{-}I_N})$ that the A jump hierarchy up to a fixed ordinal $\alpha < \varepsilon_0$ exists for each arithmetic $\mathcal{L}_2$ formula $A(X, y)$. First of all, we have a term t_A for each such arithmetic A by Lemma 72 so that t_A represents A and, moreover, t_A behaves as a set in the sense of $\mathcal{P}(\mathsf{N})$ provided the parameters of A are known to be numbers and sets, respectively. Hence, one hopes to build iterations of A by means of the recursion theorem for BON (Proposition 2). How do we know that such iterations yield sets in $\mathsf{BON}(\mu) + (\mathsf{F\text{-}I_N})$? It follows from standard proof theory that $\mathsf{BON}(\mu)+(\mathsf{F\text{-}I_N})$ proves transfinite induction with respect to $\mathcal{L}_\mathsf{N}$ formulas along each initial segment of ε_0 and, hence, we prove by transfinite induction up to each $\alpha < \varepsilon_0$ that the hierarchy up to α generated by its corresponding term really is a hierarchy of sets in the sense of $\mathcal{P}(\mathsf{N})$. Summing up, if α is less than ε_0 and $n \in \mathbb{N}$ so that the order type of $\prec_n$ is α, then for each elementary $\mathcal{L}_2$ formula $A(X, y)$ there exists an $\mathcal{L}_\mathsf{N}$ term h so that

$$\mathsf{BON}(\mu) + (\mathsf{F\text{-}I_N}) \vdash x \in \mathcal{P}(\mathsf{N}) \to hx \in \mathcal{P}(\mathsf{N}) \wedge \mathcal{H}_A(x, hx, n).$$

The remaining axioms of $(\Pi^0_\infty\text{-}\mathsf{CA})_{<\varepsilon_0}$ are easily dealt with, too. We are ready to state the embedding of $(\Pi^0_\infty\text{-}\mathsf{CA})_{<\varepsilon_0}$ into $\mathsf{BON}(\mu) + (\mathsf{F}\text{-}\mathsf{I_N})$, cf. [28].

Theorem 74 *We have for every $\mathcal{L}_2$ formula $A(\vec{X}, \vec{y})$ with at most $\vec{X}, \vec{y}$ free:*

$$(\Pi^0_\infty\text{-}\mathsf{CA})_{<\varepsilon_0} \vdash A(\vec{X}, \vec{y}) \implies \mathsf{BON}(\mu) + (\mathsf{F}\text{-}\mathsf{I_N}) \vdash \vec{x} \in \mathcal{P}(\mathsf{N}) \wedge \vec{y} \in \mathsf{N} \rightarrow A^{\mathsf{N}}(\vec{x}, \vec{y}).$$

This finishes our short discussion of the lower bound for μ systems based on set and formula induction. In the next paragraph we turn to discussion of lower bounds in the presence of intermediate forms of induction.

4.3.3 The wellordering proof for $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I_N})$

In the sequel we show that $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I_N})$ proves transfinite induction along each initial segment of $\varphi\omega 0$. This yields the lower bound $(\Pi^0_\infty\text{-}\mathsf{CA})_{<\omega^\omega}$ for the system $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I_N})$. Moreover, a result of Kahle to be mentioned below shows that $(\mathsf{O}\text{-}\mathsf{I_N})$ and $(\mathsf{N}\text{-}\mathsf{I_N})$ are equivalent over $\mathsf{BON}(\mu)$ and, hence, a lower bound for $\mathsf{BON}(\mu) + (\mathsf{O}\text{-}\mathsf{I_N})$ will be available, too.

Throughout this paragraph we are implicitly working with our embedding $(\cdot)^{\mathsf{N}}$ of $\mathcal{L}_2$ into $\mathcal{L}_{\mathsf{N}}$ which we have described at the beginning of this section. Recall that $\prec$ is a primitive recursive standard wellordering of order type Γ_0 with field $\mathbb{N}$ and least element 0. As usual we set (with respect to $\prec$):

$$\begin{aligned} \mathit{Prog}(A) &:= (\forall x)((\forall y \prec x)A(y) \rightarrow A(x)), \\ \mathit{TI}(a, A) &:= \mathit{Prog}(A) \rightarrow (\forall x \prec a)A(x). \end{aligned}$$

Moreover, $\mathit{TI}(a, f)$ abbreviates $\mathit{TI}(a, fx = \mathsf{0})$. In the sequel we want to show that

$$\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I_N}) \vdash (\forall f \in \mathsf{N} \rightarrow \mathsf{N})\mathit{TI}(a, f)$$

for each $a \prec \varphi\omega 0$ or, equivalently, for each $a = \varphi k 0$ $(k < \omega)$.

In order to make the wellordering proof work, we need a certain amount of transfinite induction with respect to formulas of the form $tx \in \mathsf{N}$. More precisely, we have to extend N induction $(\mathsf{N}\text{-}\mathsf{I_N})$ to N transfinite induction up to ω^k for each $k < \omega$. This can be established in a straightforward manner, however, there is one point where attention is needed: in the proof of Lemma 76 below we will use the fact that the class of formulas $tx \in \mathsf{N}$ is closed under universal quantifiers of the form $(\forall x \prec s)$, a closure property which is not immediately obvious. Observe that in the proof of the following lemma we make essential use of the non-constructive μ operator for the first time.

Lemma 75 *For every $\mathcal{L}_{\mathsf{N}}$ term s there exists an $\mathcal{L}_{\mathsf{N}}$ term t so that*

$$\mathsf{BON}(\mu) \vdash (\forall x \in \mathsf{N})((\forall y \prec x)sy \in \mathsf{N} \leftrightarrow tx \in \mathsf{N}).$$

Proof. Let r be an $\mathcal{L}_\mathsf{N}$ term for the characteristic function of $\prec$. For a given $\mathcal{L}_\mathsf{N}$ term s choose the term t' of $\mathcal{L}_\mathsf{N}$ given by

$$t' := \lambda y.\mathsf{d_N}(sy)\mathsf{0}(ryx)\mathsf{0}. \tag{1}$$

Then it is straightforward to verify that

$$\mathsf{BON} \vdash x \in \mathsf{N} \rightarrow [(\forall y \in \mathsf{N})(ryx = \mathsf{0} \rightarrow sy \in \mathsf{N}) \leftrightarrow (t' \in \mathsf{N} \rightarrow \mathsf{N})]. \tag{2}$$

Using the axiom (μ.1) for the non-constructive μ operator we have

$$\mathsf{BON}(\mu) \vdash (t' \in \mathsf{N} \rightarrow \mathsf{N}) \leftrightarrow \mu t' \in \mathsf{N}. \tag{3}$$

Hence, we can take $t := \lambda x.\mu t'$ and read off our assertion from (2) and (3). $\square$

This finishes our preparation for the following lemma, which guarantees N transfinite induction up to ω^k for each $k < \omega$.

Lemma 76 *We have for all $k < \omega$:*

$$\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I_N}) \vdash TI(\omega^k, fx \in \mathsf{N}).$$

Proof. We prove the claim by induction on k. The case $k = 0$ is trivial. For the induction step assume that the assertion is true for some $k < \omega$. Let us work informally in $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I_N})$ and make free use of the preceding lemma. We show

$$A(y) := (\forall x \prec \omega^k \cdot y) fx \in \mathsf{N} \tag{1}$$

by $(\mathsf{N}\text{-}\mathsf{I_N})$ induction on y, assuming $Prog(fx \in \mathsf{N})$. This will immediately yield the induction step. $A(\mathsf{0})$ is trivially satisfied. So assume $A(y)$ and show $A(y')$. First, one easily verifies

$$(\forall a)[(\forall b \prec a)(\forall x \prec \omega^k \cdot y + b) fx \in \mathsf{N} \rightarrow (\forall x \prec \omega^k \cdot y + a) fx \in \mathsf{N}] \tag{2}$$

by making use of the assumptions $A(y)$ and $Prog(fx \in \mathsf{N})$. Furthermore, by applying the (meta) induction hypothesis to (2) we obtain

$$(\forall a \prec \omega^k)(\forall x \prec \omega^k \cdot y + a) fx \in \mathsf{N}. \tag{3}$$

From (3) and $Prog(fx \in \mathsf{N})$ we can conclude

$$(\forall a \prec \omega^k) f(\omega^k \cdot y + a) \in \mathsf{N}, \tag{4}$$

which together with $A(y)$ yields $A(y')$ as desired. This finishes our proof. $\square$

On the other hand, we already know that $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I_N})$ proves transfinite induction up to each ordinal less than ε_0 with respect to *sets*: according to Proposition 19, $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I_N})$ proves set induction $(\mathsf{S}\text{-}\mathsf{I_N})$ and, moreover, $\mathsf{BON}(\mu) + (\mathsf{S}\text{-}\mathsf{I_N})$ contains PA via the embedding $(\cdot)^\mathsf{N}$ by Proposition 73.

In the sequel we need primitive recursive auxiliary functions p and e on our ordinal notations, which satisfy

(i) $p(0) = e(0) = 0$; $p(\omega^a) = 0$ and $e(\omega^a) = a$;

(ii) if $a = \omega^{a_1} + \cdots + \omega^{a_n}$ for more than one summand so that $a_n \preceq \cdots \preceq a_1$, then $p(a) = \omega^{a_1} + \cdots + \omega^{a_{n-1}}$ and $e(a) = a_n$.

In addition, let us define some sort of jump operator J, which is given by the following arithmetic definition:

$$J(X, a) \ := \ (\forall y)((\forall x \prec y)(x \in X) \rightarrow (\forall x \prec y + a)(x \in X)).$$

Let $(f \in \mathsf{N} \rightarrow \mathsf{N})$ be a set. In order to prove $TI(a, f)$ for each $a \prec \varphi\omega 0$, we build up a hierarchy of sets $(H_b)_{b \prec \omega^k}$ for each $k < \omega$. The definition of the hierarchy corresponds to the formulas $\mathcal{R}(P, Q, t)$ of Schütte [59], p. 184ff. More precisely,

$$\begin{aligned} H_0 &= f, \\ H_a &= \{y : (\forall z)\,(p(a) \preceq z \prec a \rightarrow J(H_z, \varphi(e(a), y)))\}, \qquad (0 \prec a). \end{aligned}$$

In order to formalize $(H_b)_{b \prec \omega^k}$ in $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$, we need some preliminary considerations. The arithmetic $\mathcal{L}_2$ formula $A(X, a, y)$ is given by

$$A(X, a, y) \ := \ (\forall z)\,(p(a) \preceq z \prec a \rightarrow J((X)_z, \varphi(e(a), y))).$$

According to Lemma 72 (cf. also the comments following it), there exists an $\mathcal{L}_\mathsf{N}$ term t_A so that $\mathsf{BON}(\mu)$ proves:

$$(\forall x \in \mathsf{N} \rightarrow \mathsf{N})(\forall a, y \in \mathsf{N})(t_A x a y \in \mathsf{N}),$$
$$(\forall x \in \mathsf{N} \rightarrow \mathsf{N})(\forall a, y \in \mathsf{N})(A^\mathsf{N}(x, a, y) \leftrightarrow t_A x a y = \mathsf{0}).$$

An application of the same lemma provides us with a term s so that the following is derivable in $\mathsf{BON}(\mu)$:

$$(\forall x, y \in \mathsf{N})(sxy \in \mathsf{N}),$$
$$(\forall x, y \in \mathsf{N})((x = \langle (x)_0, (x)_1 \rangle \wedge (x)_1 \prec y) \leftrightarrow sxy = \mathsf{0}).$$

Finally, the operation g is given by

$$g \ := \ \lambda xyz.(\mathsf{d_N}(x(z)_1(z)_0)1(szy)\mathsf{0}).$$

If x is assumed to be an operation which enumerates the sets xb, then gxa is a characteristic function of the disjoint union of the sets $(xb)_{b \prec a}$.

We have prepared the ground in order to introduce an operation h so that hfa represents the ath level of the H hierarchy with initial set f. It is given by the recursion theorem to satisfy

$$hfay \ \simeq \ \begin{cases} fy, & \text{if } a = \mathsf{0}, \\ t_A(g(hf)a)ay, & \text{otherwise.} \end{cases}$$

So far we do not know that hfa represents a set in $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$. This is the content of the following crucial lemma.

Lemma 77 *We have for all $k < \omega$:*

$$\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N}) \vdash (\forall f \in \mathsf{N} \to \mathsf{N})(\forall a \prec \omega^k)(hfa \in \mathsf{N} \to \mathsf{N}).$$

Proof. Let us first fix a $k < \omega$ and an $(f \in \mathsf{N} \to \mathsf{N})$. We work informally in $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$ and show that

$$Prog(ra \in \mathsf{N}), \tag{1}$$

where r is defined to be the term $\lambda a.\mu(hfa)$. Then our assertion immediately follows from (1), Lemma 76 and an application of the axiom $(\mu.1)$. In order to prove (1) let us assume

$$(\forall b \prec a)(rb \in \mathsf{N}), \tag{2}$$

i.e. $(\forall b \prec a)(\mu(hfb) \in \mathsf{N})$. The equivalence $(\mu.1)$ yields

$$(\forall b \prec a)(hfb \in \mathsf{N} \to \mathsf{N}). \tag{3}$$

It is our aim to show $(hfa \in \mathsf{N} \to \mathsf{N})$, which by $(\mu.1)$ yields $ra \in \mathsf{N}$ as desired. If $a = \mathsf{0}$, then $(hf\mathsf{0} \in \mathsf{N} \to \mathsf{N})$ holds since it is $(f \in \mathsf{N} \to \mathsf{N})$ by assumption. Otherwise, we have to show $(t_A(g(hf)a)a \in \mathsf{N} \to \mathsf{N})$. But this is immediate, since (3) implies $(g(hf)a \in \mathsf{N} \to \mathsf{N})$, and t_A maps sets and numbers into sets according to our discussion above. This finishes the proof of (1), and hence our assertion follows as shown. $\Box$

We have established the existence of the hierarchy $(H_a)_{a \prec \omega^k}$ as a hierarchy of *sets* in $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$ for each $k < \omega$, and its defining properties can be proved there.

The next lemma is essential in the wellordering proof for $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$. It corresponds to Lemma 9 of Schütte [59], and its proof is very similar to the proof of Lemma 9. A careful but straightforward formalization of that proof only uses set induction $(\mathsf{S}\text{-}\mathsf{I}_\mathsf{N})$, which is available in $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$ by Proposition 19. For the details the reader is referred to [59].

Lemma 78 *We have for all $k < \omega$:*

$$\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N}) \vdash (f \in \mathsf{N} \to \mathsf{N}) \wedge \mathsf{0} \prec a \prec \omega^k \wedge (\forall b \prec a) Prog(hfb) \ \to \ Prog(hfa).$$

We are now able to show that $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$ proves transfinite induction up to $\varphi k0$ for each $k < \omega$. This will immediately yield the desired lower bound.

Theorem 79 *We have for all $k < \omega$:*

$$\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N}) \vdash (\forall f \in \mathsf{N} \to \mathsf{N}) TI(\varphi k0, f).$$

Proof. In the following we work informally in the theory $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$. Let us choose $k < \omega$ and an arbitrary $(f \in \mathsf{N} \to \mathsf{N})$. By Lemma 77 we have a hierarchy of sets $(hfa)_{a \prec \omega^{k+1}}$ with initial set $hf\mathsf{0} = f$. Hence, we trivially have

$$Prog(f) \ \to \ Prog(hf\mathsf{0}). \tag{1}$$

A combination of (1) and the previous lemma yields

$$Prog(f) \wedge a \prec \omega^{k+1} \wedge (\forall b \prec a) Prog(hfb) \; \rightarrow \; Prog(hfa). \tag{2}$$

If we abbreviate $B(a) \; := \; a \prec \omega^{k+1} \rightarrow Prog(hfa)$, then (2) amounts to

$$Prog(f) \; \rightarrow \; Prog(B). \tag{3}$$

Furthermore, it is easily seen that B can be represented as a set t_B, provably in $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$, for example, choose

$$t_B \; := \; \lambda a.\mathsf{d}_\mathsf{N}(prog(hfa))1(ra\omega^{k+1})\mathsf{0}, \tag{4}$$

where *prog* is the set corresponding to *Prog* according to Lemma 72, and r represents the characteristic function of $\prec$. Therefore, we can conclude from (3) and set transfinite induction up to ω^{k+1} that

$$Prog(f) \; \rightarrow \; Prog(hf\omega^k). \tag{5}$$

In addition, we trivially have

$$Prog(hf\omega^k) \; \rightarrow \; (hf\omega^k\mathsf{0} = \mathsf{0}). \tag{6}$$

Since $p(\omega^k) = 0$ and $e(\omega^k) = k$ we get by the definition of $hf\omega^k$ that

$$(hf\omega^k\mathsf{0} = \mathsf{0}) \; \rightarrow \; J(f, \varphi k0). \tag{7}$$

Furthermore, it is immediate from the definition of J that

$$J(f, \varphi k0) \; \rightarrow \; (\forall x \prec \varphi k0)(fx = \mathsf{0}). \tag{8}$$

If we combine (5)–(8) we obtain $TI(f, \varphi k0)$ as desired. □

When measuring the proof-theoretic ordinal $|\mathsf{S}|$ of a formal system S as usual by means of transfinite induction, this is most naturally done for an anonymous (free) relation variable U. In our applicative framework we assume that $\mathsf{U} \subset \mathsf{N}$ and, moreover, there is a total characteristic function $\mathsf{c_U}$ of U on N: $\mathsf{c_U} \in \mathcal{P}(\mathsf{N})$. With this definition in mind we are now able to derive the following corollary from our theorem.

Corollary 80 $\varphi\omega 0 \leq |\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})|$.

Instead of giving a wellordering proof for $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$, it would have also been possible to provide a direct embedding of the second order system $(\Pi^0_\infty\text{-}\mathsf{CA})_{<\omega^\omega}$ or, more comfortably, $(\Pi^0_1\text{-}\mathsf{CA})_{<\omega^\omega}$, into $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$. Such an embedding makes use of formalized recursion theory and the same techniques for building hierarchies of sets as in the wellordering proof above. For a similar embedding in a different setting the reader is referred to Jäger and Strahm [50].

We finish this section by briefly addressing the lower bound of $\mathsf{BON}(\mu) + (\mathsf{O}\text{-}\mathsf{I}_\mathsf{N})$. It is immediately available by our lower bound for $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$ and a result by Kahle [53], which says that $(\mathsf{O}\text{-}\mathsf{I}_\mathsf{N})$ and $(\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$ are equivalent over $\mathsf{BON}(\mu)$. We mention his theorem from [53] without proof.

Proposition 81 $(\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$ *and* $(\mathsf{O}\text{-}\mathsf{I}_\mathsf{N})$ *are equivalent over* $\mathsf{BON}(\mu)$*.*

Together with Corollary 80 we thus obtain the following lower bound result for the system $\mathsf{BON}(\mu) + (\mathsf{O}\text{-}\mathsf{I}_\mathsf{N})$.

Corollary 82 $\varphi\omega 0 \leq |\mathsf{BON}(\mu) + (\mathsf{O}\text{-}\mathsf{I}_\mathsf{N})|$.

This finishes our discussion on lower bounds for systems with μ plus intermediate forms of induction on the natural numbers.

4.4 Fixed point theories with ordinals

In this section we discuss fixed point theories over Peano arithmetic with ordinals as they have been introduced in Jäger [48]. They will be used in the next section in order to determine proof-theoretic upper bounds of applicative theories with μ operator.

In the first paragraph we introduce the fixed point theories $\mathsf{PA}^\mathsf{r}_\Omega$, $\mathsf{PA}^\mathsf{s}_\Omega$, $\mathsf{PA}^\mathsf{w}_\Omega$ and PA_Ω, and we state the main proof-theoretic equivalences. The remaining paragraphs contain a detailed discussion of the upper bounds of $\mathsf{PA}^\mathsf{s}_\Omega$ and $\mathsf{PA}^\mathsf{w}_\Omega$, cf. Jäger [48] and Jäger and Strahm [50]. These two systems contain Δ^Ω_0 induction on the ordinals plus Σ^Ω and full induction on the natural numbers, respectively. In particular, we introduce an infinitary system T_∞, which we subsequently use in order to provide asymmetric interpretations of suitable Tait-style reformulations T_1 and T_2 of $\mathsf{PA}^\mathsf{s}_\Omega$ and $\mathsf{PA}^\mathsf{w}_\Omega$, respectively.

4.4.1 The formal framework and main results

In this paragraph we introduce the formal framework for fixed point theories over Peano arithmetic with ordinals; we state the main proof-theoretic results, some of which will be established in some detail in the remaining paragraphs of this section.

The language $\mathcal{L}_\Omega$

We first introduce the notion of an inductive operator form. Let P be an n-ary relation symbol which does not belong to the language $\mathcal{L}_1$, and let $\mathcal{L}_1(P)$ denote the extension of $\mathcal{L}_1$ by P. An $\mathcal{L}_1(P)$ formula is called *P positive* if each occurrence of P in it is positive. We call P positive formulas which contain at most $\vec{x} = x_1, \ldots, x_n$ free *inductive operator forms*; we let $\mathcal{A}(P, \vec{x})$ range over such forms.

Now we extend $\mathcal{L}_1$ to a new first order language $\mathcal{L}_\Omega$ by adding a new sort of *ordinal variables* $(\sigma, \tau, \eta, \xi, \ldots)$, new binary relation symbols $<$ and $=$ for the less relation and the equality relation on the ordinals[1] and an $(n+1)$-ary relation symbol $P_\mathcal{A}$ for each inductive operator form $\mathcal{A}(P, \vec{x})$ for which P is n-ary.

[1] In general it will be clear from the context whether $<$ and $=$ denote the less and equality relation on the nonnegative integers or on the ordinals.

The *number terms* of $\mathcal{L}_\Omega$ are the number terms of $\mathcal{L}_1$; the *ordinal terms* of $\mathcal{L}_\Omega$ are the ordinal variables of $\mathcal{L}_\Omega$. The *formulas* of $\mathcal{L}_\Omega$ $(A, B, C, \ldots)$ are inductively defined as follows:

1. If R is an n-ary relation symbol of $\mathcal{L}_1$, then $R(s_1, \ldots, s_n)$ is an atomic formula of $\mathcal{L}_\Omega$.
2. The formulas $(\sigma < \tau)$, $(\sigma = \tau)$ and $P_\mathcal{A}(\sigma, \vec{s})$ are atomic formulas of $\mathcal{L}_\Omega$.
3. If A and B are $\mathcal{L}_\Omega$ formulas, then so also are $\neg A$, $(A \vee B)$, $(A \wedge B)$ and $(A \rightarrow B)$.
4. If A is an $\mathcal{L}_\Omega$ formula, then so also are $(\exists x)A$ and $(\forall x)A$.
5. If A is an $\mathcal{L}_\Omega$ formula, then so also are $(\exists \xi < \sigma)A$, $(\forall \xi < \sigma)A$, $(\exists \xi)A$ and $(\forall \xi)A$.

For every $\mathcal{L}_\Omega$ formula A we write A^σ to denote the $\mathcal{L}_\Omega$ formula which is obtained by replacing all unbounded ordinal quantifiers $(\mathcal{Q}\xi)$ in A by $(\mathcal{Q}\xi < \sigma)$. Additional abbreviations are:

$$\begin{aligned} P_\mathcal{A}^\sigma(\vec{s}) &:= P_\mathcal{A}(\sigma, \vec{s}), \\ P_\mathcal{A}^{<\sigma}(\vec{s}) &:= (\exists \xi < \sigma) P_\mathcal{A}^\xi(\vec{s}), \\ P_\mathcal{A}(\vec{s}) &:= (\exists \xi) P_\mathcal{A}^\xi(\vec{s}). \end{aligned}$$

Finally, the following classes of $\mathcal{L}_\Omega$ formulas are crucial in the framework of Peano arithmetic with ordinals.

Definition 83 (Δ_0^Ω formulas) The Δ_0^Ω formulas of $\mathcal{L}_\Omega$ are inductively defined as follows:

1. Every atomic formula of $\mathcal{L}_\Omega$ is a Δ_0^Ω formula.
2. If A and B are Δ_0^Ω formulas, then so also are $\neg A$, $(A \vee B)$, $(A \wedge B)$ and $(A \rightarrow B)$.
3. If A is a Δ_0^Ω formula, then so also are $(\exists x)A$ and $(\forall x)A$.
4. If A is a Δ_0^Ω formula, then so also are $(\exists \xi < \sigma)A$ and $(\forall \xi < \sigma)A$.

Definition 84 (Σ^Ω and Π^Ω formulas) The Σ^Ω and Π^Ω formulas are inductively generated as follows:

1. Every Δ_0^Ω formula is a Σ^Ω and Π^Ω formula.
2. If A is a Σ^Ω [Π^Ω] formula, then $\neg A$ is a Π^Ω [Σ^Ω] formula.
3. If A and B are Σ^Ω [Π^Ω] formulas, then $(A \vee B)$ and $(A \wedge B)$ are Σ^Ω [Π^Ω] formulas.

4. If A is a Π^Ω [Σ^Ω] formula and B is a Σ^Ω [Π^Ω] formula, then $(A \rightarrow B)$ is a Σ^Ω [Π^Ω] formula.

5. If A is a Σ^Ω [Π^Ω] formula, then $(\exists x)A$ and $(\forall x)A$ are Σ^Ω [Π^Ω] formulas.

6. If A is a Σ^Ω [Π^Ω] formula, then $(\exists \xi < \sigma)A$ and $(\forall \xi < \sigma)A$ are Σ^Ω [Π^Ω] formulas.

7. If A is a Σ^Ω formula, then $(\exists \xi)A$ is a Σ^Ω formula; if A is a Π^Ω formula, then $(\forall \xi)A$ is a Π^Ω formula.

This finishes our description of the syntax of $\mathcal{L}_\Omega$. In a next step we turn to the axioms of our fixed point theories with ordinals.

The theories $\mathsf{PA}^\mathsf{r}_\Omega$, $\mathsf{PA}^\mathsf{s}_\Omega$, $\mathsf{PA}^\mathsf{w}_\Omega$ and PA_Ω

In the following we introduce four fixed point theories with ordinals; the first three of those will be relevant in the sequel for the proof-theoretic analysis of applicative theories wit μ.

We start off with the specification of the weakest of our fixed point theories, the system $\mathsf{PA}^\mathsf{r}_\Omega$. It is based on the usual two-sorted predicate calculus with equality and classical logic. The non-logical axioms of $\mathsf{PA}^\mathsf{r}_\Omega$ are divided into the following six groups:

I. Number-theoretic axioms. The axioms of Peano arithmetic PA with the exception of complete induction on the natural numbers.

II. Inductive operator axioms. For all inductive operator forms $\mathcal{A}(P, \vec{x})$:

$$P^\sigma_\mathcal{A}(\vec{s}) \leftrightarrow \mathcal{A}(P^{<\sigma}_\mathcal{A}, \vec{s}).$$

III. Σ^Ω Reflection axioms, (Σ^Ω-Ref). For all Σ^Ω formulas A:

$$A \rightarrow (\exists \xi)A^\xi.$$

IV. Linearity axioms.

$$\sigma \not< \sigma \;\wedge\; (\sigma < \tau \wedge \tau < \eta \rightarrow \sigma < \eta) \;\wedge\; (\sigma < \tau \vee \sigma = \tau \vee \tau < \sigma).$$

V. Δ^Ω_0 induction on the natural numbers, (Δ^Ω_0-I_N). For all Δ^Ω_0 formulas $A(x)$:

$$A(0) \wedge (\forall x)(A(x) \rightarrow A(x')) \;\rightarrow\; (\forall x)A(x).$$

VI. Δ^Ω_0 induction on the ordinals, (Δ^Ω_0-I_Ω). For all Δ^Ω_0 formulas $A(\xi)$:

$$(\forall \xi)[(\forall \eta < \xi)A(\eta) \rightarrow A(\xi)] \;\rightarrow\; (\forall \xi)A(\xi).$$

This finishes the description of the theory $\mathsf{PA}^\mathsf{r}_\Omega$. $\mathsf{PA}^\mathsf{s}_\Omega$ is now defined to be $\mathsf{PA}^\mathsf{r}_\Omega$ plus induction on the natural numbers for Σ^Ω formulas, i.e. $\mathsf{PA}^\mathsf{s}_\Omega$ is $\mathsf{PA}^\mathsf{r}_\Omega + (\Sigma^\Omega$-$\mathsf{I}_\mathsf{N})$.

Moreover, $\mathsf{PA}^{\mathsf{w}}_{\Omega}$ is obtained from $\mathsf{PA}^{\mathsf{r}}_{\Omega}$ by allowing induction on the natural numbers for arbitrary $\mathcal{L}_{\Omega}$ formulas, i.e. $\mathsf{PA}^{\mathsf{w}}_{\Omega}$ is $\mathsf{PA}^{\mathsf{r}}_{\Omega} + (\mathsf{F\text{-}I_N})$. Finally, we obtain the full theory PA_{Ω} by adding formula induction on the ordinals to $\mathsf{PA}^{\mathsf{w}}_{\Omega}$, i.e. PA_{Ω} is $\mathsf{PA}^{\mathsf{w}}_{\Omega} + (\mathsf{F\text{-}I}_{\Omega})$. Observe that we trivially have

$$\mathsf{PA}^{\mathsf{r}}_{\Omega} \subset \mathsf{PA}^{\mathsf{s}}_{\Omega} \subset \mathsf{PA}^{\mathsf{w}}_{\Omega} \subset \mathsf{PA}_{\Omega}.$$

We finish this paragraph by mentioning the following crucial fixed point lemma, which says that the Σ^{Ω} formulas $P_{\mathcal{A}}$ describe fixed points of the inductive operator form $\mathcal{A}(P, \vec{x})$:

Lemma 85 *We have for all inductive operator forms $\mathcal{A}(P, \vec{x})$:*

$$\mathsf{PA}^{\mathsf{r}}_{\Omega} \vdash (\forall \vec{x})(P_{\mathcal{A}}(\vec{x}) \leftrightarrow \mathcal{A}(P_{\mathcal{A}}, \vec{x})).$$

Proof. Immediate from the inductive operator and Σ^{Ω} reflection axioms. □

Main results

In the sequel let us briefly mention the main results concerning the proof-theoretic strength of the four fixed point theories $\mathsf{PA}^{\mathsf{r}}_{\Omega}$, $\mathsf{PA}^{\mathsf{s}}_{\Omega}$, $\mathsf{PA}^{\mathsf{w}}_{\Omega}$ and PA_{Ω}. We will provide the detailed upper bound computations for the theories $\mathsf{PA}^{\mathsf{s}}_{\Omega}$ and $\mathsf{PA}^{\mathsf{w}}_{\Omega}$ in the rest of this section, and sketch the relevant arguments for $\mathsf{PA}^{\mathsf{r}}_{\Omega}$ and PA_{Ω} immediately after the following theorem. 1., 3. and 4. of the theorem are due to Jäger [48] where, in particular, a detailed treatment of $\mathsf{PA}^{\mathsf{r}}_{\Omega}$ is carried through; 2. is established in Jäger and Strahm [50]. Below, $\widehat{\mathsf{ID}}_1$ denotes the well-known fixed point theory for positive operators (cf. Feferman [21]) and $\mathsf{ID}^{\#}_1$ is the subsystem of $\widehat{\mathsf{ID}}_1$ where induction on the natural numbers is restricted to formulas which are positive in the fixed point constants, cf. [50]. Hence, it is immediate from Lemma 85 that the theories $\mathsf{ID}^{\#}_1$, $\widehat{\mathsf{ID}}_1$, and ID_1 are contained in $\mathsf{PA}^{\mathsf{s}}_{\Omega}$, $\mathsf{PA}^{\mathsf{w}}_{\Omega}$, and PA_{Ω}, respectively. Alternatively, the lower bounds for the ordinal theories $\mathsf{PA}^{\mathsf{s}}_{\Omega}$ and $\mathsf{PA}^{\mathsf{w}}_{\Omega}$ follow from the results of Sections 4.3 and 4.5.

Theorem 86 *We have the following proof-theoretic equivalences:*

1. $\mathsf{PA}^{\mathsf{r}}_{\Omega} \equiv \mathsf{PA}$.
2. $\mathsf{PA}^{\mathsf{s}}_{\Omega} \equiv \mathsf{ID}^{\#}_1 \equiv (\Pi^0_{\infty}\text{-}\mathsf{CA})_{<\omega^{\omega}}$.
3. $\mathsf{PA}^{\mathsf{w}}_{\Omega} \equiv \widehat{\mathsf{ID}}_1 \equiv (\Pi^0_{\infty}\text{-}\mathsf{CA})_{<\varepsilon_0}$.
4. $\mathsf{PA}_{\Omega} \equiv \mathsf{ID}_1$.

Let us now briefly sketch the argument which shows that $\mathsf{PA}^{\mathsf{r}}_{\Omega}$ is a conservative extension of Peano arithmetic PA. For the details, the reader is referred to [48]. In a *first step*, one reformulates $\mathsf{PA}^{\mathsf{r}}_{\Omega}$ in a Tait-style calculus T in a straightforward manner and observes that the main formulas of the non-logical axioms and rules of T are Σ^{Ω}.

Hence, we obtain by (finite) partial cut elimination for T that all but Σ^Ω and Π^Ω cuts can be eliminated from T derivations. In a *second step*, the Σ^Ω-Π^Ω fragment of T is reduced to PA via an *asymmetric interpretation*: ordinal variables are replaced by finite ordinals so that a formula $P_{\mathcal{A}}(m, \vec{s})$ with $m \in \mathbb{N}$ translates into an $\mathcal{L}_1$ formula that describes the build up in stages of the corresponding inductive definition, and if m is a bound for universal ordinal quantifiers, then $m + 2^n$ provides a bound for existential ordinal quantifiers, where n is the length of a given T derivation. The so-obtained asymmetric interpretation validates Σ^Ω and Π^Ω cuts as well as Σ^Ω reflection. In addition, Δ_0^Ω induction on the natural numbers, (Δ_0^Ω-I_N), translates into complete induction for arbitrary $\mathcal{L}_1$ formulas. This finishes our brief sketch of the conservativity of $\mathsf{PA}_\Omega^\mathsf{r}$ over PA. A similar interpretation will be used in the upper bound argument for the system $\mathsf{PA}_\Omega^\mathsf{w}$, cf. Section 4.4.4.

As to the full theory PA_Ω, it is straightforward that PA_Ω can be embedded into the system of Kripke-Platek set theory KPu, which by Jäger [44] is proof-theoretically equivalent to ID_1. A direct treatment of PA_Ω is possible, too.

In the following paragraphs we establish the proof-theoretic upper bounds for the systems $\mathsf{PA}_\Omega^\mathsf{s}$ and $\mathsf{PA}_\Omega^\mathsf{w}$ in some more detail.

4.4.2 The system T_∞

In this section we introduce the infinitary system T_∞ which will be used for the proof-theoretic treatment of $\mathsf{PA}_\Omega^\mathsf{s}$ and $\mathsf{PA}_\Omega^\mathsf{w}$ below. It is based on the language $\mathcal{L}_\infty$, which extends $\mathcal{L}_\Omega$ by constants $\bar{\alpha}$ for all ordinals $\alpha < \varepsilon_0$ (in the sense of the notation system).[2] In addition, we assume that $\mathcal{L}_\infty$ contains an anonymous (free) relation symbol U, cf. the comments after Theorem 79. The *ordinal terms* ($\theta, \theta_0, \theta_1, \ldots$) of $\mathcal{L}_\infty$ are the ordinal variables and the ordinal constants of $\mathcal{L}_\infty$. The *literals* of $\mathcal{L}_\infty$ are the literals of $\mathcal{L}_\Omega$ extended to the language $\mathcal{L}_\infty$ plus the literals involving U. To simplify the notation we often write $A(\alpha)$ instead of $A(\bar{\alpha})$ if α is an ordinal less than ε_0.

The *formulas* of $\mathcal{L}_\infty$ are inductively generated as follows:

1. Every literal of $\mathcal{L}_\infty$ is an $\mathcal{L}_\infty$ formula.
2. If A and B are $\mathcal{L}_\infty$ formulas, then so also are $(A \vee B)$ and $(A \wedge B)$.
3. If A is an $\mathcal{L}_\infty$ formula, so also are $(\exists x)A$, $(\forall x)A$, $(\exists \xi < \theta)A$ and $(\forall \xi < \theta)A$.

Since T_∞ is a Tait-style system, we assume that the negation $\neg A$ of an $\mathcal{L}_\infty$ formula A is defined as usual by making use of De Morgan's laws and the law of double negation. Notice that $\mathcal{L}_\infty$ formulas do not contain unbounded ordinal quantifiers. The $\mathcal{L}_\infty^\mathsf{c}$ formulas are the $\mathcal{L}_\infty$ formulas which do not contain free number and free

[2] We can allow all ordinals from our notation system here, but we will only need those below ε_0 in the sequel.

ordinal variables. Furthermore, a literal of $\mathcal{L}_\infty^{\mathsf{c}}$ is called *primitive* if it is not of the form $U(s)$, $\neg U(s)$, $P_\mathcal{A}^\alpha(\vec{s})$ or $\neg P_\mathcal{A}^\alpha(\vec{s})$. Obviously, every primitive literal of $\mathcal{L}_\infty^{\mathsf{c}}$ is either true or false, and in the following we write TRUE for the set of true primitive literals.

In order to measure the complexity of cuts in T_∞ we assign a rank to each $\mathcal{L}_\infty^{\mathsf{c}}$ formula. This definition is tailored so that the process of building up stages of an inductive definition is reflected by the rank of the formulas $P_\mathcal{A}^\alpha(\vec{s})$.

Definition 87 The *rank* $rn(A)$ of a $\mathcal{L}_\infty^{\mathsf{c}}$ formula A is inductively defined as follows:

1. If A is a literal $R(\vec{s})$, $\neg R(\vec{s})$, $U(s)$, $\neg U(s)$, $(\alpha < \beta)$, $(\alpha \not< \beta)$, $(\alpha = \beta)$ or $(\alpha \neq \beta)$, then $rn(A) := 0$.
2. If A is a literal $P_\mathcal{A}^\alpha(\vec{s})$ or $\neg P_\mathcal{A}^\alpha(\vec{s})$, then $rn(A) := \omega(\alpha + 1)$.
3. If A is a formula $(B \vee C)$ or $(B \wedge C)$ so that $rn(B) = \beta$ and $rn(C) = \gamma$, then $rn(A) := max(\beta, \gamma) + 1$.
4. If A is a formula $(\exists x)B(x)$ or $(\forall x)B(x)$ so that $rn(B(0)) = \alpha$, then $rn(A) := \alpha + 1$.
5. If A is a formula $(\exists \xi < \alpha)B(\xi)$ or $(\forall \xi < \alpha)B(\xi)$, then
$$rn(A) := sup\{rn(B(\beta)) + 1 : \beta < \alpha\}.$$

We write $oc(B)$ for the set of ordinal constants which occur in the $\mathcal{L}_\infty$ formula B. The proof of the following lemma is a matter of routine (cf. [51, 50]).

Lemma 88 *We have for all inductive operator forms $\mathcal{A}(P, \vec{x})$, all $\mathcal{L}_\infty^{\mathsf{c}}$ formulas A and all ordinals $\alpha < \varepsilon_0$:*

1. $rn(\mathcal{A}(P_\mathcal{A}^{<\alpha}, \vec{s})) < rn(P_\mathcal{A}^\alpha(\vec{s}))$.
2. *If $\beta < \alpha$ for all $\beta \in oc(A)$, then $rn(A) < \omega\alpha + \omega$.*

The system T_∞ is formulated as a Tait-style calculus for finite sets $(\Gamma, \Lambda, \ldots)$ of $\mathcal{L}_\infty^{\mathsf{c}}$ formulas (cf. e.g. [60]). If A is an $\mathcal{L}_\infty^{\mathsf{c}}$ formula, then Γ, A is a shorthand for $\Gamma \cup \{A\}$, and similarly for expressions like Γ, A, B. T_∞ contains the following axioms and rules of inference.

I. Axioms. For all finite sets Γ of $\mathcal{L}_\infty^{\mathsf{c}}$ formulas, all closed number terms s and t with identical value, and all literals A in TRUE:

$$\Gamma,\ \neg U(s),\ U(t) \quad \text{and} \quad \Gamma,\ A.$$

II. Propositional rules. For all finite sets Γ of $\mathcal{L}_\infty^{\mathsf{c}}$ formulas and all $\mathcal{L}_\infty^{\mathsf{c}}$ formulas A and B:

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

III. Number quantifier rules. For all finite sets Γ of $\mathcal{L}^{\mathsf{c}}_{\infty}$ formulas and all $\mathcal{L}^{\mathsf{c}}_{\infty}$ formulas $A(s)$:

$$\frac{\Gamma,\ A(s)}{\Gamma,\ (\exists x)A(x)}, \qquad \frac{\Gamma,\ A(t) \text{ for all closed number terms } t}{\Gamma,\ (\forall x)A(x)} \quad (\omega).$$

IV. Ordinal quantifier rules. For all finite sets Γ of $\mathcal{L}^{\mathsf{c}}_{\infty}$ formulas, all $\mathcal{L}^{\mathsf{c}}_{\infty}$ formulas $A(\alpha)$ and all ordinals β with $\alpha < \beta < \varepsilon_0$:

$$\frac{\Gamma,\ A(\alpha)}{\Gamma,\ (\exists \xi < \beta)A(\xi)}, \qquad \frac{\Gamma,\ A(\gamma) \text{ for all } \gamma < \beta}{\Gamma,\ (\forall \xi < \beta)A(\xi)}.$$

V. Inductive operator rules. For all finite sets Γ of $\mathcal{L}^{\mathsf{c}}_{\infty}$ formulas, all inductive operator forms $\mathcal{A}(P, \vec{x})$, all closed number terms $\vec{s}$ and all ordinals $\alpha < \varepsilon_0$:

$$\frac{\Gamma,\ \mathcal{A}(P^{<\alpha}_{\mathcal{A}}, \vec{s})}{\Gamma,\ P^{\alpha}_{\mathcal{A}}(\vec{s})}, \qquad \frac{\Gamma,\ \neg\mathcal{A}(P^{<\alpha}_{\mathcal{A}}, \vec{s})}{\Gamma,\ \neg P^{\alpha}_{\mathcal{A}}(\vec{s})}.$$

VI. Cut rules. For all finite sets Γ of $\mathcal{L}^{\mathsf{c}}_{\infty}$ formulas and all $\mathcal{L}^{\mathsf{c}}_{\infty}$ formulas A:

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

The formulas A and $\neg A$ are the cut formulas of this cut; the rank of a cut is the rank of its cut formulas.

As usual, for α and ρ less than Γ_0, we write $\mathsf{T}_{\infty} \,|\!\frac{\alpha}{\rho}\, \Gamma$ if Γ is provable in T_{∞} by a proof of depth less than or equal to α so that all cuts in this proof have rank *less* than ρ, cf. e.g. [56] for a precise definition.

It is easy to check that the assignment of ranks and the rules of inference are tailored so that the methods of predicative proof theory yield full cut elimination for T_{∞}. Therefore, we omit the proof of the following theorem and refer to Pohlers [56] or Schütte [59].

Theorem 89 (Cut elimination for T_{∞}) *We have for all finite sets Γ of $\mathcal{L}^{\mathsf{c}}_{\infty}$ formulas and all ordinals α, β, ρ less than Γ_0:*

$$\mathsf{T}_{\infty} \,|\!\frac{\alpha}{\beta+\omega^{\rho}}\, \Gamma \quad \Longrightarrow \quad \mathsf{T}_{\infty} \,|\!\frac{\varphi\rho\alpha}{\beta}\, \Gamma.$$

We finish this paragraph by mentioning the following crucial persistency lemma. It is proved by a straightforward induction on α.

Lemma 90 (Persistency) *We have for all finite sets Γ of $\mathcal{L}^{\mathsf{c}}_{\infty}$ formulas, all Σ^{Ω} formulas $A(\vec{\xi}, \vec{x})$ and Π^{Ω} formulas $B(\vec{\xi}, \vec{x})$ of $\mathcal{L}_{\Omega}(U)$ with free variables among those indicated, all closed number terms $\vec{r}$, and all ordinals $\alpha, \rho < \Gamma_0$ and $\beta, \vec{\gamma}, \delta < \varepsilon_0$ so that $\beta \leq \delta$:*

1. $\mathsf{T}_{\infty} \,|\!\frac{\alpha}{\rho}\, \Gamma,\ A^{\beta}(\vec{\gamma}, \vec{r}) \quad \Longrightarrow \quad \mathsf{T}_{\infty} \,|\!\frac{\alpha}{\rho}\, \Gamma,\ A^{\delta}(\vec{\gamma}, \vec{r})$.
2. $\mathsf{T}_{\infty} \,|\!\frac{\alpha}{\rho}\, \Gamma,\ B^{\delta}(\vec{\gamma}, \vec{r}) \quad \Longrightarrow \quad \mathsf{T}_{\infty} \,|\!\frac{\alpha}{\rho}\, \Gamma,\ B^{\beta}(\vec{\gamma}, \vec{r})$.

In the following two paragraphs we use the framework presented so far in order to give a proof-theoretic analysis of $\mathsf{PA}^{\mathsf{s}}_{\Omega}$ and $\mathsf{PA}^{\mathsf{w}}_{\Omega}$.

4.4.3 The proof-theoretic strength of $\mathsf{PA}^{\mathsf{s}}_{\Omega}$

In the sequel we indicate the main lines of the proof-theoretic analysis of the system $\mathsf{PA}^{\mathsf{s}}_{\Omega}$, i.e. $\mathsf{PA}^{\mathsf{r}}_{\Omega} + (\Sigma^{\Omega}\text{-}\mathsf{I}_{\mathsf{N}})$. The analysis proceeds in two steps: first, one observes that a straightforward Tait-style reformulation T_1 of $\mathsf{PA}^{\mathsf{s}}_{\Omega}$ enjoys partial cut elimination so that the only cuts which are needed are in $\Sigma^{\Omega} \cup \Pi^{\Omega}$. Subsequently, the Σ^{Ω}-Π^{Ω} fragment of T_1 is reduced to T_{∞} via an asymmetric interpretation; the required upper bound $\varphi\omega 0$ for $\mathsf{PA}^{\mathsf{s}}_{\Omega}$ is obtained by full cut elimination for T_{∞}.

Let us start off with describing the first step of the above procedure. We define the degree $dg(A)$ of an $\mathcal{L}_{\Omega}$ formula A in order to measure the complexity of cuts below.

Definition 91 The degree $dg(A)$ of an $\mathcal{L}_{\Omega}$[3] formulas A is inductively defined as follows:

1. If A is a Σ^{Ω} or Π^{Ω} formula, then $dg(A) := 0$. Below we assume that 1. does not apply.

2. If A is a formula $(B \vee C)$ or $(B \wedge C)$ so that $dg(B) = m$ and $dg(C) = n$, then $dg(A) := max(m, n) + 1$.

3. If A is a formula $(\exists x)B$, $(\forall x)B$, $(\exists \xi)B$ or $(\forall \xi)B$ so that $dg(B) = n$, then $dg(A) := n + 1$.

4. If A is a formula $(\exists \xi < \sigma)B$ or $(\forall \xi < \sigma)B$ so that $dg(B) = n$, then $dg(A) := n + 2$.

The Tait-style calculus T_1 for $\mathsf{PA}^{\mathsf{s}}_{\Omega}$ is formulated in the language $\mathcal{L}_{\Omega}$ and comprises the following axioms and rules of inference.

I. Axioms. For all finite sets Γ of $\mathcal{L}_{\Omega}$ formulas, all Δ^{Ω}_0 formulas A and all Δ^{Ω}_0 formulas B which are axioms of $\mathsf{PA}^{\mathsf{s}}_{\Omega}$:

$$\Gamma,\, \neg A,\, A \quad \text{and} \quad \Gamma,\, B.$$

II. Propositional and quantifier rules. These include the usual Tait-style inference rules for the propositional connectives and all sorts of quantifiers.

III. Σ^{Ω} Reflection. For all finite sets Γ of $\mathcal{L}_{\Omega}$ formulas and for all Σ^{Ω} formulas A:

$$\frac{\Gamma,\, A}{\Gamma,\, (\exists \xi) A^{\xi}}.$$

IV. Σ^{Ω} induction on the natural numbers. For all finite sets Γ of $\mathcal{L}_{\Omega}$ formulas, all Σ^{Ω} formulas $A(x)$ and all number variables u which do not occur in Γ, $A(0)$:

$$\frac{\Gamma,\, A(0) \qquad \Gamma,\, \neg A(u),\, A(u')}{\Gamma,\, (\forall x) A(x)}.$$

[3] For our Tait-style treatment below, we assume that negation is defined in $\mathcal{L}_{\Omega}$.

V. Δ_0^Ω induction on the ordinals. For all finite sets Γ of $\mathcal{L}_\Omega$ formulas, all Δ_0^Ω formulas $A(\sigma)$ and all ordinals variables ξ which do not occur in Γ, $A(\sigma)$:

$$\frac{\Gamma,\, \neg(\forall \eta < \xi)A(\eta),\, A(\xi)}{\Gamma,\, A(\sigma)}.$$

VI. Cut rules. These are formulated in the same way as for T_∞.

The notion $\mathsf{T}_1 \overset{n}{\underset{k}{\vdash}} \Gamma$ is now defined in a straightforward manner, cf. the previous paragraph. Since the main formulas of all non-logical axioms and rules of T_1 are Σ^Ω, we obtain the following partial cut elimination theorem for T_1; here $2_k(n)$ is given as usual by $2_0(n) = n$ and $2_{k+1}(n) = 2^{2_k(n)}$.

Theorem 92 (Partial cut elimination for T_1) *We have for all finite sets Γ of $\mathcal{L}_\Omega$ formulas and all natural numbers n and k:*

$$\mathsf{T}_1 \overset{n}{\underset{k}{\vdash}} \Gamma \quad \Longrightarrow \quad \mathsf{T}_1 \overset{2_k(n)}{\underset{1}{\vdash}} \Gamma.$$

Together with the obvious embedding of $\mathsf{PA}^{\mathsf{s}}_\Omega$ into T_1 we thus obtain the following corollary.

Corollary 93 *Let A be an $\mathcal{L}_\Omega$ formula which is provable in $\mathsf{PA}^{\mathsf{s}}_\Omega$. Then there exists a natural number n so that $\mathsf{T}_1 \overset{n}{\underset{1}{\vdash}} A$.*

This finishes our preliminary treatment of $\mathsf{PA}^{\mathsf{s}}_\Omega$. In a second step we now provide an asymmetric interpretation of the Σ^Ω-Π^Ω fragment of T_1 into T_∞. For that purpose, we introduce the crucial notion of an (α, β) instance. Let Γ be a finite set of $\mathcal{L}_\Omega$ formulas, Λ a finite set of $\mathcal{L}^{\mathsf{c}}_\infty$ formulas and $\alpha, \beta < \varepsilon_0$. Then Λ is called an (α, β) *instance* of Γ if it results from Γ by replacing

(i) each free number variable by a closed number term and each free ordinal variable by an ordinal less than α;

(ii) each universal ordinal quantifier $(\forall \xi)$ in the formulas of Γ by $(\forall \xi < \alpha)$;

(iii) each existential ordinal quantifier $(\exists \xi)$ in the formulas of Γ by $(\exists \xi < \beta)$.

We are ready to state the asymmetric interpretation theorem.

Theorem 94 (Asymmetric interpretation of T_1 into T_∞) *Assume that Γ is a finite set of Σ^Ω and Π^Ω formulas of $\mathcal{L}_\Omega$ so that $\mathsf{T}_1 \overset{n}{\underset{1}{\vdash}} \Gamma$ for some natural number n. Then we have for all ordinals $\alpha < \omega^\omega$ and every $(\alpha, \alpha + \omega^n)$ instance Λ of Γ:*

$$\mathsf{T}_\infty \overset{\omega^{\alpha+\omega^n}}{\underset{\omega(\alpha+\omega^n)}{\vdash}} \Lambda.$$

Proof. The theorem is proved by induction on n. Apart from Σ^Ω induction on the natural numbers all axioms and rules of inference are treated as in similar asymmetric interpretations, cf. e.g. Jäger [43, 48] or Schütte [59]. In the following argument we make tacitly use of Lemma 90.

Now suppose that Γ is the conclusion of the rule for Σ^Ω induction on the natural numbers. Then there exists a Σ^Ω formula $A(x)$ and $n_0, n_1 < n$ so that

$$\mathsf{T}_1 \vdash_1^{n_0} \Gamma,\ A(0), \tag{1}$$

$$\mathsf{T}_1 \vdash_1^{n_1} \Gamma,\ \neg A(u),\ A(u'). \tag{2}$$

Let m be the maximum of n_0 and n_1 and set $\beta_k := \alpha + \omega^m(k+1)$ for all natural numbers k. We show by side induction on k that

$$\mathsf{T}_\infty \vdash_{\omega(\alpha+\omega^n)}^{\omega^{\beta_k}+1} \Lambda,\ A^{\beta_k}(k).^{4} \tag{3}$$

If $k = 0$ then (3) follows from (1) and the main induction hypothesis. If $k > 0$ then the side induction hypothesis yields

$$\mathsf{T}_\infty \vdash_{\omega(\alpha+\omega^n)}^{\omega^{\beta_{k-1}}+1} \Lambda,\ A^{\beta_{k-1}}(k-1). \tag{4}$$

Now we apply the main induction hypothesis to (2) with α replaced by β_{k-1} and obtain

$$\mathsf{T}_\infty \vdash_{\omega(\alpha+\omega^n)}^{\omega^{\beta_k}} \Lambda,\ \neg A^{\beta_{k-1}}(k-1),\ A^{\beta_k}(k). \tag{5}$$

Hence (4), (5) and a cut imply

$$\mathsf{T}_\infty \vdash_{\omega(\alpha+\omega^n)}^{\omega^{\beta_k}+1} \Lambda,\ A^{\beta_k}(k). \tag{6}$$

This finishes the proof of (3). A further application of Lemma 90 to (3) gives

$$\mathsf{T}_\infty \vdash_{\omega(\alpha+\omega^n)}^{\omega^{\beta_k}+1} \Lambda,\ A^{\beta}(k) \tag{7}$$

for $\beta := \alpha + \omega^n$ and all natural numbers k. In (7) we can replace k by an arbitrary closed term with value k. Hence, we are in a position to apply the inference rule for numerical universal quantification and conclude

$$\mathsf{T}_\infty \vdash_{\omega(\alpha+\omega^n)}^{\omega^{\alpha+\omega^n}} \Lambda,\ (\forall x)A^{\beta}(x). \tag{8}$$

Since the formula $(\forall x)A^\beta(x)$ is contained in Λ, the treatment of Σ^Ω induction on the natural numbers is complete. $\square$

Together with Corollary 93 and Theorem 89 we have thus obtained the following reduction of the Σ^Ω fragment of $\mathsf{PA}^{\mathsf{s}}_\Omega$ to the cut free part of T_∞.

[4] To be more precise, we mean the instance of $A^{\beta_k}(k)$ where all free variables are replaced according to Λ.

Theorem 95 *Let A be a closed Σ^Ω formula which is provable in $\mathsf{PA}^{\mathsf{s}}_\Omega$. Then there exists an $\alpha < \varphi\omega 0$ and a $\beta < \omega^\omega$ so that $\mathsf{T}_\infty \overset{\alpha}{\underset{0}{\vdash}} A^\beta$.*

As usual this result also gives an upper bound for the proof-theoretic ordinal of the theory $\mathsf{PA}^{\mathsf{s}}_\Omega$, cf. e.g. Pohlers [56] or Schütte [59].

Corollary 96 $|\mathsf{PA}^{\mathsf{s}}_\Omega| \leq \varphi\omega 0$.

Moreover, a careful formalization of the above procedure yields a proof-theoretic reduction (in the sense of Feferman [22]) of $\mathsf{PA}^{\mathsf{s}}_\Omega$ to $(\Pi^0_\infty\text{-}\mathsf{CA})_{<\omega^\omega}$ or $\mathsf{ID}^\#_1$. Hence, we have established the upper bound part of Theorem 86, 2.

4.4.4 The proof-theoretic strength of $\mathsf{PA}^{\mathsf{w}}_\Omega$

We finish this section by showing how to adapt the argument of the proof-theoretic analysis of $\mathsf{PA}^{\mathsf{s}}_\Omega$ to the fixed point theory with *full* induction on the natural numbers, $\mathsf{PA}^{\mathsf{w}}_\Omega$. As the procedure is very similar (and in some sense even simpler) to the one of $\mathsf{PA}^{\mathsf{s}}_\Omega$, we restrict ourselves to mentioning the main lines of the argument only.

The first step in the treatment of $\mathsf{PA}^{\mathsf{w}}_\Omega$ consists in eliminating complete induction on the natural numbers. For that purpose, we introduce an infinitary Tait-calculus T_2. In T_2 we derive finite sets of *simple* $\mathcal{L}_\infty$ formulas, or $\mathcal{L}^{\mathsf{s}}_\infty$ formulas for short: such formulas contain neither free number variables nor constants for ordinals; however, *ordinal variables* are allowed to occur in $\mathcal{L}^{\mathsf{s}}_\infty$ formulas. The axioms and rules of inference of T_2 are essentially the same as those of T_1 with the only crucial exception that the rule for induction on the natural numbers is omitted in favor of the ω rule. The degree $dg(A)$ of an $\mathcal{L}^{\mathsf{s}}_\infty$ formulas A is defined in the same way as in Definition 91, and once more we have the standard derivability relation $\mathsf{T}_2 \overset{\alpha}{\underset{k}{\vdash}} \Gamma$ for $\alpha < \Gamma_0$ and $k < \omega$. Summing up, induction on the natural numbers becomes provable in T_2 at the prize of infinite derivation lengths so that we obtain the following embedding of $\mathsf{PA}^{\mathsf{w}}_\Omega$ into T_2.

Proposition 97 (Embedding of $\mathsf{PA}^{\mathsf{w}}_\Omega$ into T_2) *Let A be a numerically closed $\mathcal{L}_\Omega$ formula which is provable in $\mathsf{PA}^{\mathsf{w}}_\Omega$. Then there exist $\alpha < \omega + \omega$ and $k < \omega$ so that $\mathsf{T}_2 \overset{\alpha}{\underset{k}{\vdash}} A$.*

Similar to Theorem 92, we obtain partial cut elimination for T_2.

Theorem 98 (Partial cut elimination for T_2) *We have for all finite sets Γ of $\mathcal{L}^{\mathsf{s}}_\infty$ formulas, all ordinals $\alpha < \Gamma_0$ and all natural numbers $k < \omega$:*

$$\mathsf{T}_2 \overset{\alpha}{\underset{k}{\vdash}} \Gamma \quad \Longrightarrow \quad \mathsf{T}_2 \overset{2_k(\alpha)}{\underset{1}{\vdash}} \Gamma.$$

From Proposition 97 we derive the following corollary.

Corollary 99 *Let A be a numerically closed $\mathcal{L}_\Omega$ formula which is provable in $\mathsf{PA}^{\mathsf{w}}_\Omega$. Then there exists an ordinal $\alpha < \varepsilon_0$ so that $\mathsf{T}_2 \overset{\alpha}{\underset{1}{\vdash}} A$.*

The second step in the analysis of $\mathsf{PA}^{\mathsf{w}}_{\Omega}$ is an asymmetric interpretation of the Σ^{Ω}-Π^{Ω} fragment of T_2 into T_{∞}. The notion of an (α, β) instance of a finite set Γ of $\mathcal{L}^{\mathsf{s}}_{\infty}$ formulas is defined in the same way as in the previous paragraph. The asymmetric interpretation of T_2 is slightly simpler than the one of T_1, since an asymmetric treatment of Σ^{Ω} induction on the natural numbers is no longer needed. In particular, we can do with the function $2^{\bullet}$ instead of $\omega^{\bullet}$ for bounding existential quantifiers. The exact formulation of the theorem reads as follows.

Theorem 100 (Asymmetric interpretation of T_2 into T_{∞}) *Assume that Γ is a finite set of Σ^{Ω} and Π^{Ω} formulas of $\mathcal{L}^{\mathsf{s}}_{\infty}$ so that $\mathsf{T}_2 \mid\frac{\alpha}{1} \Gamma$ for some ordinal $\alpha < \varepsilon_0$. Then we have for all ordinals $\beta < \varepsilon_0$ and every $(\beta, \beta + 2^{\alpha})$ instance Λ of Γ:*

$$\mathsf{T}_{\infty} \mid\frac{\omega^{\beta+\omega^{\alpha}}}{\omega(\beta+\omega^{\alpha})} \Lambda.$$

From Corollary 99 and Theorem 89 we can now derive the following reduction theorem for $\mathsf{PA}^{\mathsf{w}}_{\Omega}$, together with a corollary concerning the upper bound for its proof-theoretic ordinal.

Theorem 101 *Let A be a closed Σ^{Ω} formula which is provable in $\mathsf{PA}^{\mathsf{w}}_{\Omega}$. Then there exists an $\alpha < \varphi\varepsilon_0 0$ and a $\beta < \varepsilon_0$ so that $\mathsf{T}_{\infty} \mid\frac{\alpha}{0} A^{\beta}$.*

Corollary 102 $|\mathsf{PA}^{\mathsf{w}}_{\Omega}| \leq \varphi\varepsilon_0 0$.

Once more, formalization of the above arguments yields a proof-theoretic reduction of $\mathsf{PA}^{\mathsf{w}}_{\Omega}$ to $(\Pi^0_{\infty}\text{-}\mathsf{CA})_{<\varepsilon_0}$ or $\widehat{\mathsf{ID}}_1$, thus establishing the upper bound part of the main Theorem 86, 3.

4.5 Upper bounds

This section contains the exact upper bound computations for $\mathsf{TON}(\mu)$ plus various forms of complete induction on the natural numbers. The relevant proof-theoretic reductions are obtained by formalizing the infinitary term model $CTT(\mu)$ of $\mathsf{TON}(\mu)$ in fixed point theories with ordinals, cf. Jäger and Strahm [52]. The methods presented easily extend to systems with the extensionality axiom (Ext). Corresponding results in the context of a *partial* application operation plus set induction and full formula induction are due to Feferman and Jäger [28]. For the treatment of positive formula induction, cf. Jäger and Strahm [50].

In the first paragraph we show how the reduction relation $\twoheadrightarrow_{\boldsymbol{\rho\mu}}$ can be represented as a fixed point of a suitable operator form. The second paragraph describes reductions of $\mathsf{TON}(\mu) + (\mathsf{S}\text{-}\mathsf{I}_{\mathsf{N}})$, $\mathsf{TON}(\mu) + (\mathsf{F}^{+}\text{-}\mathsf{I}_{\mathsf{N}})$, and $\mathsf{TON}(\mu) + (\mathsf{F}\text{-}\mathsf{I}_{\mathsf{N}})$ to $\mathsf{PA}^{\mathsf{r}}_{\Omega}$, $\mathsf{PA}^{\mathsf{s}}_{\Omega}$, and $\mathsf{PA}^{\mathsf{w}}_{\Omega}$, respectively, via the formalized term model $CTT(\mu)$. In the last paragraph, finally, we give a complete Church Rosser proof for $\twoheadrightarrow_{\boldsymbol{\rho\mu}}$, together with indications concerning its formalization in the fixed point theory $\mathsf{PA}^{\mathsf{r}}_{\Omega}$.

4.5.1 Formalizing $CTT(\mu)$ in $\mathcal{L}_\Omega$

In the sequel we show how our infinitary term model $CTT(\mu)$ of $\mathsf{TON}(\mu)$ (cf. Definition 68) can be represented in the language $\mathcal{L}_\Omega$ so that the Church Rosser property of $\twoheadrightarrow_{\boldsymbol{\rho\mu}}$, $CR(\twoheadrightarrow_{\boldsymbol{\rho\mu}})$, is derivable in the weakest of our fixed point theories with ordinals, $\mathsf{PA}^{\mathsf{r}}_\Omega$. The proof of this last claim will be given in Section 4.5.3.

Let us start off with the formalization of $\twoheadrightarrow_{\boldsymbol{\rho\mu}}$ in $\mathcal{L}_\Omega$. In particular, it is our aim to represent $\twoheadrightarrow_{\boldsymbol{\rho\mu}}$ as a fixed point $P_{Red_{\boldsymbol{\rho\mu}}}$ of a binary inductive operator form $Red_{\boldsymbol{\rho\mu}}$ to be specified now.

Let P be a new binary relation symbol. Then the μ redex-contractum pairs w.r.t. P are given as follows:

$$RedCon_{\boldsymbol{\mu}}(P,x,y) \ :=\ CTer(x) \wedge CTer(y) \wedge RedCon^*_{\boldsymbol{\mu}}(P,x,y),$$

where $RedCon^*_{\boldsymbol{\mu}}(P,x,y)$ is the disjunction of the following formulas:

(1) $x = \langle \ulcorner\cdot\urcorner, \ulcorner\mu\urcorner, (x)_2\rangle \wedge$
$(\exists z)[Num(z) = y \ \wedge\ P(\langle \ulcorner\cdot\urcorner, (x)_2, y\rangle, \ulcorner 0\urcorner) \wedge$
$(\forall u)(\exists v)(P(\langle \ulcorner\cdot\urcorner, (x)_2, Num(u)\rangle, Num(v)) \ \wedge\ (u < z 1 v > 0))]$,

(2) $x = \langle \ulcorner\cdot\urcorner, \ulcorner\mu\urcorner, (x)_2\rangle \ \wedge\ y = \ulcorner 0\urcorner \wedge (\forall u)(\exists v > 0)P(\langle \ulcorner\cdot\urcorner, (x)_2, Num(u)\rangle, Num(v))$.

Recall from Section 2.3.2 that $RedCon_{\boldsymbol{\rho}}(x,y)$ denotes a primitive recursive formalization of the $\boldsymbol{\rho}$ redex-contractum pairs. Then the following formula describes the $\boldsymbol{\rho\mu}$ redex-contractum pairs w.r.t. P:

$$RedCon_{\boldsymbol{\rho\mu}}(P,x,y) \ :=\ RedCon_{\boldsymbol{\rho}}(x,y) \ \vee\ RedCon_{\boldsymbol{\mu}}(P,x,y).$$

Once we have given the formula $RedCon_{\boldsymbol{\rho\mu}}(P,x,y)$, the formulas $Red1_{\boldsymbol{\rho\mu}}(P,x,y)$, $RedSeq_{\boldsymbol{\rho\mu}}(P,x,y,z)$, and finally $Red_{\boldsymbol{\rho\mu}}(P,x,y)$ are defined exactly as in Section 2.3.2 with the only difference of containing the parameter P.

Remark 103 *The formula $Red_{\boldsymbol{\rho\mu}}(P,x,y)$ is a binary inductive operator form.*

We are ready to put down the formal representation of $\twoheadrightarrow_{\boldsymbol{\rho\mu}}$ in $\mathcal{L}_\Omega$ as a fixed point $P_{Red_{\boldsymbol{\rho\mu}}}(x,y)$ of $Red_{\boldsymbol{\rho\mu}}(P,x,y)$, i.e. as the formula

$$(\exists\alpha)P^{\alpha}_{Red_{\boldsymbol{\rho\mu}}}(x,y).$$

We have already stated in Theorem 69 that $\twoheadrightarrow_{\boldsymbol{\rho\mu}}$ has the Church Rosser property. The next theorem says that the corresponding proof can be carried through in the system $\mathsf{PA}^{\mathsf{r}}_\Omega$ for the formalization $P_{Red_{\boldsymbol{\rho\mu}}}$ of $\twoheadrightarrow_{\boldsymbol{\rho\mu}}$. The detailed arguments of this fact will be given in the last paragraph of this section. Let us only mention now that the presence of Δ^Ω_0 induction on the ordinals in $\mathsf{PA}^{\mathsf{r}}_\Omega$, $(\Delta^\Omega_0\text{-}\mathsf{I}_\Omega)$, is crucial for the Church Rosser proof.

Theorem 104 $\mathsf{PA}^{\mathsf{r}}_\Omega \vdash CR(P_{Red_{\boldsymbol{\rho\mu}}})$.

This finishes the formalization of $\twoheadrightarrow_{\boldsymbol{\rho\mu}}$ in the language $\mathcal{L}_\Omega$.

4.5.2 Embedding theories with μ into fixed point theories with ordinals

In this paragraph we establish embeddings of $\mathsf{TON}(\mu)+(\mathsf{S}\text{-}\mathsf{I}_\mathsf{N})$, $\mathsf{TON}(\mu)+(\mathsf{F}^+\text{-}\mathsf{I}_\mathsf{N})$, and $\mathsf{TON}(\mu)+(\mathsf{F}\text{-}\mathsf{I}_\mathsf{N})$ into $\mathsf{PA}^\mathsf{r}_\Omega$, $\mathsf{PA}^\mathsf{s}_\Omega$, and $\mathsf{PA}^\mathsf{w}_\Omega$, respectively, via the formalized term model construction $CTT(\mu)$ of $\mathsf{TON}(\mu)$. It is possible to obtain corresponding results for the extensional version $TTE(\mu)$ of $CTT(\mu)$ so that the above embeddings carry over to the presence of extensionality (Ext).

In the sequel we work with the translation $*$ that we have described in Section 2.3.2, now depending on the formalization $P_{Red_{\rho\mu}}$ of $\twoheadrightarrow_{\rho\mu}$. Hence, $*$ translates $\mathcal{L}_\mathsf{N}$ into $\mathcal{L}_\Omega$. First, it is easy to derive the following analogue of Lemma 25 from the (formalized) Church Rosser theorem in $\mathsf{PA}^\mathsf{r}_\Omega$ (Theorem 104).

Lemma 105 *We have for all $\mathcal{L}_\mathsf{N}$ formulas $A(x)$:*

$$\mathsf{PA}^\mathsf{r}_\Omega \vdash P_{Red_{\rho\mu}}(x,y) \;\rightarrow\; (A^*(x) \leftrightarrow A^*(y)).$$

Corollary 106 *Let $\mathcal{Q} \in \{\exists, \forall\}$. Then we have for all $\mathcal{L}_\mathsf{N}$ formulas $A(x)$:*

$$\mathsf{PA}^\mathsf{r}_\Omega \vdash ((\mathcal{Q}x \in \mathsf{N})A(x))^* \;\leftrightarrow\; (\mathcal{Q}x)A^*(Num(x)).$$

From this last corollary it is immediate that positive formula induction on the natural numbers, $(\mathsf{F}^+\text{-}\mathsf{I}_\mathsf{N})$, carries over to Σ^Ω induction on the natural numbers under $*$ and, of course, the $*$ translation of each instance of full formula induction, $(\mathsf{F}\text{-}\mathsf{I}_\mathsf{N})$, is verifiable by complete induction on the natural numbers for arbitrary $\mathcal{L}_\Omega$ formulas. The treatment of set induction $(\mathsf{S}\text{-}\mathsf{I}_\mathsf{N})$ by means of Δ^Ω_0 induction on the natural numbers is the content of the following lemma.

Lemma 107 *The $*$ translation of $(\mathsf{S}\text{-}\mathsf{I}_\mathsf{N})$ is provable in $\mathsf{PA}^\mathsf{r}_\Omega$, i.e. $\mathsf{PA}^\mathsf{r}_\Omega$ proves*

$$[f \in \mathcal{P}(\mathsf{N}) \wedge f\mathsf{0} = \mathsf{0} \wedge (\forall x \in \mathsf{N})(fx = \mathsf{0} \rightarrow f(x') = \mathsf{0}) \;\rightarrow\; (\forall x \in \mathsf{N})(fx = \mathsf{0})]^*.$$

Proof. In the following we work informally in $\mathsf{PA}^\mathsf{r}_\Omega$. Assume $(f \in \mathcal{P}(\mathsf{N}))^*$, $(f\mathsf{0} = \mathsf{0})^*$ and $[(\forall x \in \mathsf{N})(fx = \mathsf{0} \rightarrow f(x') = \mathsf{0})]^*$. From the first premise and Theorem 104 we conclude

$$(\forall x)(\exists! y)P_{Red_{\rho\mu}}(\langle \ulcorner \cdot \urcorner, f, Num(x)\rangle, Num(y)). \tag{1}$$

The other two premises yield

$$P_{Red_{\rho\mu}}(\langle \ulcorner \cdot \urcorner, f, \ulcorner \mathsf{0} \urcorner\rangle, \ulcorner \mathsf{0} \urcorner), \tag{2}$$

$$(\forall x)(P_{Red_{\rho\mu}}(\langle \ulcorner \cdot \urcorner, f, Num(x)\rangle, \ulcorner \mathsf{0} \urcorner) \rightarrow P_{Red_{\rho\mu}}(\langle \ulcorner \cdot \urcorner, f, Num(x+1)\rangle, \ulcorner \mathsf{0} \urcorner)). \tag{3}$$

From (1) we get by Σ^Ω reflection the existence of an ordinal α so that we have

$$(\forall x, y)(P_{Red_{\rho\mu}}(\langle \ulcorner \cdot \urcorner, f, Num(x)\rangle, Num(y)) \leftrightarrow P^{<\alpha}_{Red_{\rho\mu}}(\langle \ulcorner \cdot \urcorner, f, Num(x)\rangle, Num(y))). \tag{4}$$

Combining (2), (3) and (4) this amounts to

$$P^{<\alpha}_{Red_{\rho\mu}}(\langle\ulcorner\cdot\urcorner, f, \ulcorner \mathsf{0}\urcorner\rangle, \ulcorner \mathsf{0}\urcorner), \tag{5}$$

$$(\forall x)(P^{<\alpha}_{Red_{\rho\mu}}(\langle\ulcorner\cdot\urcorner, f, Num(x)\rangle, \ulcorner \mathsf{0}\urcorner) \imath P^{<\alpha}_{Red_{\rho\mu}}(\langle\ulcorner\cdot\urcorner, f, Num(x+1)\rangle, \ulcorner \mathsf{0}\urcorner)). \tag{6}$$

Now recall that we have Δ^Ω_0 induction on the natural numbers available in the system $\mathsf{PA}^\mathsf{r}_\Omega$ and, therefore, (5) and (6) imply

$$(\forall x) P^{<\alpha}_{Red_{\rho\mu}}(\langle\ulcorner\cdot\urcorner, f, Num(x)\rangle, \ulcorner \mathsf{0}\urcorner). \tag{7}$$

But from (7) we immediately obtain $[(\forall x \in \mathsf{N})(fx = \mathsf{0})]^*$. This finishes our treatment of $(\mathsf{S}\text{-}\mathsf{I}_\mathsf{N})$ in $\mathsf{PA}^\mathsf{r}_\Omega$. □

Finally, it is not difficult either to verify that the $*$ translation of each axiom of $\mathsf{TON}(\mu)$ is provable in $\mathsf{PA}^\mathsf{r}_\Omega$. In particular, $\mathsf{PA}^\mathsf{r}_\Omega$ proves axioms (9), (μ.1) and (μ.2), where in each case the presence of Σ^Ω reflection is crucial. Therefore, we have established the proof-theoretic reduction of $\mathsf{TON}(\mu) + (\mathsf{S}\text{-}\mathsf{I}_\mathsf{N})$, $\mathsf{TON}(\mu) + (\mathsf{F}^+\text{-}\mathsf{I}_\mathsf{N})$, and $\mathsf{TON}(\mu) + (\mathsf{F}\text{-}\mathsf{I}_\mathsf{N})$ to $\mathsf{PA}^\mathsf{r}_\Omega$, $\mathsf{PA}^\mathsf{s}_\Omega$, and $\mathsf{PA}^\mathsf{w}_\Omega$, respectively.

Theorem 108 *We have for all $\mathcal{L}_\mathsf{N}$ formulas $A(\vec{x})$ with at most $\vec{x}$ free:*

1. $\mathsf{TON}(\mu) + (\mathsf{S}\text{-}\mathsf{I}_\mathsf{N}) \vdash A(\vec{x}) \implies \mathsf{PA}^\mathsf{r}_\Omega \vdash CTer(\vec{x}) \to A^*(\vec{x})$.
2. $\mathsf{TON}(\mu) + (\mathsf{F}^+\text{-}\mathsf{I}_\mathsf{N}) \vdash A(\vec{x}) \implies \mathsf{PA}^\mathsf{s}_\Omega \vdash CTer(\vec{x}) \to A^*(\vec{x})$.
3. $\mathsf{TON}(\mu) + (\mathsf{F}\text{-}\mathsf{I}_\mathsf{N}) \vdash A(\vec{x}) \implies \mathsf{PA}^\mathsf{w}_\Omega \vdash CTer(\vec{x}) \to A^*(\vec{x})$.

From Proposition 73, Theorem 74, Corollary 80, Corollary 82 and Theorem 86 we are now able to derive the following proof-theoretic equivalences.

Corollary 109 *We have the following proof-theoretic equivalences:*

1. $\mathsf{TON}(\mu) + (\mathsf{S}\text{-}\mathsf{I}_\mathsf{N}) \equiv \mathsf{PA}^\mathsf{r}_\Omega \equiv \mathsf{PA}$.
2. $\mathsf{TON}(\mu) + (\mathsf{O}\text{-}\mathsf{I}_\mathsf{N}) \equiv \mathsf{TON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N}) \equiv \mathsf{TON}(\mu) + (\Sigma^+\text{-}\mathsf{I}_\mathsf{N}) \equiv$
   $\mathsf{TON}(\mu) + (\mathsf{F}^+\text{-}\mathsf{I}_\mathsf{N}) \equiv \mathsf{PA}^\mathsf{s}_\Omega \equiv \mathsf{ID}^\#_1 \equiv (\Pi^0_\infty\text{-}\mathsf{CA})_{<\omega^\omega}$.
3. $\mathsf{TON}(\mu) + (\mathsf{F}\text{-}\mathsf{I}_\mathsf{N}) \equiv \mathsf{PA}^\mathsf{w}_\Omega \equiv \widehat{\mathsf{ID}}_1 \equiv (\Pi^0_\infty\text{-}\mathsf{CA})_{<\varepsilon_0}$.

The corresponding proof-theoretic ordinals are ε_0, $\varphi\omega 0$, and $\varphi\varepsilon_0 0$, respectively.

As we have already indicated in Section 4.1.2, we obtain an extensional version $TTE(\mu)$ of $CTT(\mu)$ in a straightforward manner as follows: the reduction relation $\twoheadrightarrow_{\beta\eta\mu}$ on (not necessarily closed) λ terms is defined in the same way as $\twoheadrightarrow_{\rho\mu}$ except that $\boldsymbol{\beta\eta}$ is used instead of $\boldsymbol{\rho}$ at each stage of the corresponding inductive definition. The proof of the Church Rosser theorem for $\twoheadrightarrow_{\beta\eta\mu}$ given in the next paragraph is easily seen to extend to $\twoheadrightarrow_{\beta\eta\mu}$ so that the relevant arguments are formalizable in $\mathsf{PA}^\mathsf{r}_\Omega$. Hence, Theorem 104 holds for the extensional reduction relation $\twoheadrightarrow_{\beta\eta\mu}$, too. All together, we are now in a position to state the following strengthening of Corollary 109.

Theorem 110 *We have the following proof-theoretic equivalences:*

1. $\mathsf{TON}(\mu) + (\mathsf{Ext}) + (\mathsf{S}\text{-}\mathsf{I_N}) \equiv \mathsf{PA}^{\mathsf{r}}_{\Omega} \equiv \mathsf{PA}$.

2. $\mathsf{TON}(\mu) + (\mathsf{Ext}) + (\mathsf{O}\text{-}\mathsf{I_N}) \equiv \mathsf{TON}(\mu) + (\mathsf{Ext}) + (\mathsf{N}\text{-}\mathsf{I_N}) \equiv$
   $\mathsf{TON}(\mu) + (\mathsf{Ext}) + (\Sigma^{+}\text{-}\mathsf{I_N}) \equiv \mathsf{TON}(\mu) + (\mathsf{Ext}) + (\mathsf{F}^{+}\text{-}\mathsf{I_N}) \equiv$
   $\mathsf{PA}^{\mathsf{s}}_{\Omega} \equiv \mathsf{ID}^{\#}_{1} \equiv (\Pi^0_\infty\text{-}\mathsf{CA})_{<\omega^\omega}$.

3. $\mathsf{TON}(\mu) + (\mathsf{Ext}) + (\mathsf{F}\text{-}\mathsf{I_N}) \equiv \mathsf{PA}^{\mathsf{w}}_{\Omega} \equiv \widehat{\mathsf{ID}}_1 \equiv (\Pi^0_\infty\text{-}\mathsf{CA})_{<\varepsilon_0}$.

The corresponding proof-theoretic ordinals are ε_0, $\varphi\omega 0$, *and* $\varphi\varepsilon_0 0$, *respectively.*

This finishes our discussion of the upper bound argument for applicative theories with μ operator. In the next paragraph we will provide the still missing Church Rosser proof for $\twoheadrightarrow_{\boldsymbol{\rho\mu}}$.

4.5.3 The Church Rosser proof for $\twoheadrightarrow_{\boldsymbol{\rho\mu}}$

In this paragraph we give a proof of Theorem 104. In particular, we show that $\twoheadrightarrow_{\boldsymbol{\rho\mu}}$ has the Church Rosser property, and we argue that our proof can be carried through in the system $\mathsf{PA}^{\mathsf{r}}_{\Omega}$ for the formalization $P_{Red_{\boldsymbol{\rho\mu}}}$ of $\twoheadrightarrow_{\boldsymbol{\rho\mu}}$.

The main idea is to prove that each stage $\twoheadrightarrow_{\boldsymbol{\rho\mu}_\alpha}$ of $\twoheadrightarrow_{\boldsymbol{\rho\mu}}$ is confluent. This is achieved by combining the $\boldsymbol{\rho}$ and $\boldsymbol{\mu}_\alpha$ reductions using the well-known Lemma of Hindley and Rosen (Lemma 115).

As in Section 4.1.2 we put

$$\boldsymbol{\tau}_\alpha = \bigcup_{\beta<\alpha} \twoheadrightarrow_{\boldsymbol{\rho\mu}_\beta} .$$

Furthermore, let us call a closed $\mathcal{L}_\mathsf{N}$ term t N *singular w.r.t.* $\boldsymbol{\tau}_\alpha$, if there is no $n \in \mathbb{N}$ so that $t \; \boldsymbol{\tau}_\alpha \; \overline{n}$. The following lemma states an important property of terms μt which are N singular w.r.t. $\boldsymbol{\tau}_\alpha$. We do not give the proof of the lemma here, but it is important to mention that the (formalized) proof only uses Δ^Ω_0 induction on the ordinals, which is available in the system $\mathsf{PA}^{\mathsf{r}}_{\Omega}$.

Lemma 111 *Let* $s(x)$ *be an* $\mathcal{L}_\mathsf{N}$ *term with at most* x *free, and let* μt *be a closed* $\mathcal{L}_\mathsf{N}$ *term which is* N *singular w.r.t.* $\boldsymbol{\tau}_\alpha$. *Furthermore, assume that* $s(\mu t) \; \boldsymbol{\tau}_\alpha \; \overline{m}$[5] *for some* $m \in \mathbb{N}$. *Then we have* $s(t') \; \boldsymbol{\tau}_\alpha \; \overline{m}$ *for all closed* $\mathcal{L}_\mathsf{N}$ *terms* t'.

We will also need the following observation, the proof of which is straightforward and, therefore, we omit it.

Lemma 112 *We have for all closed* $\mathcal{L}_\mathsf{N}$ *terms* t *and all* $m \in \mathbb{N}$: *if* $\mu t \; \boldsymbol{\tau}_\alpha \; \overline{m}$, *then* $\mu t \; \boldsymbol{\mu}_\alpha \; \overline{m}$.

[5]Observe that here $s(\mu t)$ denotes $s(x)$ where μt is substituted for x and not (necessarily) s applied to μt.

The next lemma tells us that a $\boldsymbol{\mu}_\alpha$ stage has the Church Rosser property provided that $\twoheadrightarrow_{\boldsymbol{\rho}\boldsymbol{\mu}_\beta}$ is confluent for all $\beta < \alpha$. Again it is easy to see that the proof of this lemma can be formalized in the system $\mathsf{PA}^{\mathsf{r}}_\Omega$.

Lemma 113 $(\forall \beta < \alpha) CR(\twoheadrightarrow_{\boldsymbol{\rho}\boldsymbol{\mu}_\beta}) \implies CR(\twoheadrightarrow_{\boldsymbol{\mu}_\alpha})$.

Proof. Let us assume $(\forall \beta < \alpha) CR(\twoheadrightarrow_{\boldsymbol{\rho}\boldsymbol{\mu}_\beta})$, which immediately implies $CR(\boldsymbol{\tau}_\alpha)$, of course. Since $CR(\twoheadrightarrow_{\boldsymbol{\mu}_\alpha})$ follows from $CR(\stackrel{=}{\rightarrow}_{\boldsymbol{\mu}_\alpha})$ by an easy diagram chase, it is enough to show $CR(\stackrel{=}{\rightarrow}_{\boldsymbol{\mu}_\alpha})$. Here $\stackrel{=}{\rightarrow}_{\boldsymbol{\mu}_\alpha}$ denotes the reflexive closure of $\rightarrow_{\boldsymbol{\mu}_\alpha}$. First of all it is an easy consequence of $CR(\boldsymbol{\tau}_\alpha)$ that the following holds for all closed $\mathcal{L}_\mathsf{N}$ terms t and all $m, n \in \mathbb{N}$:

$$\mu t \; \boldsymbol{\mu}_\alpha \; \overline{m} \wedge \mu t \; \boldsymbol{\mu}_\alpha \; \overline{n} \implies m = n. \tag{1}$$

The second critical case comes up if we have terms $s(x), t$ and $m, n \in \mathbb{N}$ so that

$$\mu s(\mu t) \; \boldsymbol{\mu}_\alpha \; \overline{m}, \qquad \mu s(\mu t) \rightarrow_{\boldsymbol{\mu}_\alpha} \mu s(\overline{n}), \tag{2}$$

where $\mu t \; \boldsymbol{\mu}_\alpha \; \overline{n}$. Then we have to show that $\mu s(\overline{n}) \; \boldsymbol{\mu}_\alpha \; \overline{m}$. Assume that $\mu s(\mu t) \; \boldsymbol{\mu}_\alpha \; \overline{m}$ holds because of clause (1) of the definition of $\boldsymbol{\mu}_\alpha$ (Definition 68). Then we have

$$s(\mu t)\overline{m} \; \boldsymbol{\tau}_\alpha \; \overline{0}, \tag{3}$$

and for each k there exists a k' so that

$$s(\mu t)\overline{k} \; \boldsymbol{\tau}_\alpha \; \overline{k'}, \tag{4}$$

where $k' > 0$ if $k < m$. Let us first assume that the term μt is N singular w.r.t. $\boldsymbol{\tau}_\alpha$. Then we can conclude from (3), (4) and Lemma 111 that

$$s(\overline{n})\overline{m} \; \boldsymbol{\tau}_\alpha \; \overline{0}, \qquad s(\overline{n})\overline{k} \; \boldsymbol{\tau}_\alpha \; \overline{k'} \tag{5}$$

for all $k \in \mathbb{N}$, which immediately implies $\mu s(\overline{n}) \; \boldsymbol{\mu}_\alpha \; \overline{m}$ by the very definition of $\boldsymbol{\mu}_\alpha$. If μt is not N singular w.r.t. $\boldsymbol{\tau}_\alpha$, then there exists an $l \in \mathbb{N}$ with $\mu t \; \boldsymbol{\tau}_\alpha \; \overline{l}$. Using the previous lemma this implies $\mu t \; \boldsymbol{\mu}_\alpha \; \overline{l}$. Since $\mu t \; \boldsymbol{\mu}_\alpha \; \overline{n}$ holds by hypothesis, this amounts to $l = n$ according to (1). We have shown $\mu t \; \boldsymbol{\tau}_\alpha \; \overline{n}$. From this we conclude for all $k \in \mathbb{N}$:

$$s(\mu t)\overline{m} \; \boldsymbol{\tau}_\alpha \; s(\overline{n})\overline{m}, \qquad s(\mu t)\overline{k} \; \boldsymbol{\tau}_\alpha \; s(\overline{n})\overline{k}. \tag{6}$$

Using (3), (4), (6) and $CR(\boldsymbol{\tau}_\alpha)$ we can immediately derive

$$s(\overline{n})\overline{m} \; \boldsymbol{\tau}_\alpha \; \overline{0}, \qquad s(\overline{n})\overline{k} \; \boldsymbol{\tau}_\alpha \; \overline{k'} \tag{7}$$

for all $k \in \mathbb{N}$. But (7) implies $\mu s(\overline{n}) \; \boldsymbol{\mu}_\alpha \; \overline{m}$ by the definition of $\boldsymbol{\mu}_\alpha$ as desired. The case where $\mu s(\mu t) \; \boldsymbol{\mu}_\alpha \; \overline{m}$ has been derived by clause (2) of the definition of $\boldsymbol{\mu}_\alpha$ is treated in a similar way. This finishes the proof of the lemma. □

In order to apply the Lemma of Hindley and Rosen below we have to introduce the following terminology. Let R_1 and R_2 be two binary relations on a set X. Then R_1 and R_2 *commute*, if

$$(\forall x, x_1, x_2 \in X)[x\ R_1\ x_1 \wedge x\ R_2\ x_2 \rightarrow (\exists x_3 \in X)(x_1\ R_2\ x_3 \wedge x_2\ R_1\ x_3)].$$

The next lemma is *the* essential step towards the use of the lemma of Hindley and Rosen. Again its proof can easily be formalized in the system $\mathsf{PA}^{\mathsf{r}}_{\Omega}$.

Lemma 114 *Assume that* $(\forall \beta < \alpha) CR(\twoheadrightarrow_{\boldsymbol{\rho\mu}_\beta})$. *Then the reduction relations* $\twoheadrightarrow_{\boldsymbol{\rho}}$ *and* $\twoheadrightarrow_{\boldsymbol{\mu}_\alpha}$ *commute.*

Proof. From $(\forall \beta < \alpha) CR(\twoheadrightarrow_{\boldsymbol{\rho\mu}_\beta})$ we immediately get $CR(\boldsymbol{\tau}_\alpha)$. We show that the following holds for all closed $\mathcal{L}_{\mathsf{N}}$ terms t, t_1 and t_2: if $t \rightarrow_{\boldsymbol{\rho}} t_1$ and $t \rightarrow_{\boldsymbol{\mu}_\alpha} t_2$, then there exists a closed $\mathcal{L}_{\mathsf{N}}$ term t_3 so that

$$t_2 \overset{=}{\rightarrow}_{\boldsymbol{\rho}} t_3, \qquad t_1 \twoheadrightarrow_{\boldsymbol{\mu}_\alpha} t_3. \tag{1}$$

From this the claim of the lemma follows by an easy diagram chase. In the sequel we will discuss the only critical case, namely where we have terms $s(x), t$ and an $n \in \mathbb{N}$ so that

$$s(\mu t) \rightarrow_{\boldsymbol{\mu}_\alpha} s(\overline{n}), \qquad s(\mu t) \rightarrow_{\boldsymbol{\rho}} s(\mu t'), \tag{2}$$

where $\mu t\ \boldsymbol{\mu}_\alpha\ \overline{n}$ and $t \rightarrow_{\boldsymbol{\rho}} t'$. Then it is easy to check that $\mu t'\ \boldsymbol{\mu}_\alpha\ \overline{n}$ also holds, since we know $CR(\boldsymbol{\tau}_\alpha)$. Hence, we can derive

$$s(\mu t') \rightarrow_{\boldsymbol{\mu}_\alpha} s(\overline{n}), \tag{3}$$

and we are done. This finishes the sketch of the proof of this lemma. □

We have prepared the grounds in order to apply the Lemma of Hindley and Rosen. For reasons of completeness, we give its detailed formulation below. For a proof the reader is referred to Barendregt [2], where one easily sees that the proof there only uses finitary arguments.

Lemma 115 (Hindley and Rosen) *Let* R_1 *and* R_2 *be two notions of reduction and suppose that*

(1) $\twoheadrightarrow_{R_1}$ *and* $\twoheadrightarrow_{R_2}$ *are Church Rosser,*

(2) $\twoheadrightarrow_{R_1}$ *commutes with* $\twoheadrightarrow_{R_2}$.

Then $\twoheadrightarrow_{R_1 R_2}$ *has the Church Rosser property, too.*

Taking R_1 as $\boldsymbol{\rho}$ and R_2 as $\boldsymbol{\mu}_\alpha$ and assuming $(\forall \beta < \alpha) CR(\twoheadrightarrow_{\boldsymbol{\rho\mu}_\beta})$ the assumptions (1) and (2) of the Lemma of Hindley and Rosen are satisfied by Lemma 113, Lemma 114 and the fact that $\twoheadrightarrow_{\boldsymbol{\rho}}$ is Church Rosser. Hence, we can state the following lemma.

Lemma 116 $(\forall \beta < \alpha) CR(\twoheadrightarrow_{\rho\mu_\beta}) \implies CR(\twoheadrightarrow_{\rho\mu_\alpha})$.

We have shown that $CR(\twoheadrightarrow_{\rho\mu_\beta})$ is progressive and, hence, $(\forall \alpha) CR(\twoheadrightarrow_{\rho\mu_\alpha})$ follows by transfinite induction. Furthermore, $(\forall \alpha) CR(\twoheadrightarrow_{\rho\mu_\alpha})$ implies $CR(\twoheadrightarrow_{\rho\mu})$.

Corollary 117 $CR(\twoheadrightarrow_{\rho\mu})$.

This finishes our proof that $\twoheadrightarrow_{\rho\mu}$ is confluent. Notice that the formalization of $CR(\twoheadrightarrow_{\rho\mu_\beta})$ in $\mathcal{L}_\Omega$ is a Δ_0^Ω formula and, therefore, only transfinite induction for Δ_0^Ω statements is used in the argument above. Together with our previous remarks concerning the formalization of our Church Rosser proof, we have sketched that all arguments can be carried through in the system PA_Ω^r. This finishes the considerations of this paragraph.

4.6 On μ versus E

In this section let us briefly address the possibility of replacing the non-constructive μ operator by Kleene's type two functional E, cf. e.g. Hinman [42]. We assume that the axiomatization of E is given as follows; here E is supposed to be a new constant of the language $\mathcal{L}_\mathsf{N}$.

The quantification functional E

(E.1) $(f \in \mathsf{N} \to \mathsf{N}) \leftrightarrow \mathsf{E}f \in \mathsf{N}$,

(E.2) $(f \in \mathsf{N} \to \mathsf{N}) \to ((\exists x \in \mathsf{N})(fx = 0) \leftrightarrow \mathsf{E}f = 0)$.

Let us first discuss the *logical* relationship between E and μ. Very recently, Kahle [54] has established that E is definable in $\mathsf{BON}(\mu)$ without any use of induction. His definition of E from μ is very tricky: the difficulty is to guarantee the backward direction in the axiom (E.1). As far as the definability of μ from E is concerned, it seems that even in the presence of full formula induction, it is not possible to obtain μ from E over BON: the problem is again the backward direction of the axiom (μ.1). However, it is not difficult to verify that μ is definable from E in the theory $\widehat{\mathsf{BON}}$ (cf. Kahle [53] or Section 3.5.2) plus a certain amount of complete induction on the natural numbers.

Let us now turn to the relationship between E and μ as far as the *proof-theoretic* strength of the theories studied in this chapter is concerned. First of all, since E is definable from μ, the upper bounds for E follow from those for μ. Moreover, one readily verifies that in the lower bound computations for $\mathsf{BON}(\mu) + (\mathsf{S}\text{-}\mathsf{I}_\mathsf{N})$, $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$, and $\mathsf{BON}(\mu) + (\mathsf{F}\text{-}\mathsf{I}_\mathsf{N})$, it is always possible to replace μ by E. Hence, it remains to show that Kahle's Proposition 81 holds in the presence of E instead of μ. This has been recently done by Kahle [54] so that operation induction $(\mathsf{O}\text{-}\mathsf{I}_\mathsf{N})$ and N induction $(\mathsf{N}\text{-}\mathsf{I}_\mathsf{N})$ are equivalent over BON in the presence of E, too. Summing up, all lower bounds for systems with μ carry over to E straightforwardly and, therefore, the proof-theoretic strength of all theories studied in this chapter remains the same if μ is replaced by E. This ends our discussion on μ versus E.

Proof-theoretic equivalences

The following table gives a short survey of proof-theoretic equivalences. The left column contains a typical reference system of arithmetic or analysis, in the middle we list a bunch of applicative theories of the corresponding strength, and the right column includes pointers to relevant theorems, propositions or corollaries.

PTCA	PTO	Corollary 47
	PTO^+	Corollary 60
	$\mathsf{PTO}^+ + (\Sigma^+\text{-}\mathsf{CP_W})$	Theorem 62
$\mathsf{I}\Delta_0 + (\mathcal{E}_n)$	G_n	Theorem 65
	$\mathsf{G}_n + (\mathsf{Tot})$	Theorem 67
PRA	$\mathsf{BON} + (\mathsf{S}\text{-}\mathsf{I_N})$	Proposition 22, Corollary 28
	$\mathsf{BON} + (\mathsf{O}\text{-}\mathsf{I_N})$	Proposition 22, Corollary 28
	$\mathsf{BON} + (\mathsf{N}\text{-}\mathsf{I_N})$	Proposition 22, Corollary 28
	$\mathsf{BON} + (\Sigma^+\text{-}\mathsf{I_N})$	Proposition 22, Corollary 28
	$\mathsf{BON} + (\mathsf{B}\Sigma^+\text{-}\mathsf{I_N})$	Proposition 22, Theorem 31
	$\mathsf{TON} + (\mathsf{Ext}) + (\mathsf{S}\text{-}\mathsf{I_N})$	Theorem 30
	$\mathsf{TON} + (\mathsf{Ext}) + (\mathsf{O}\text{-}\mathsf{I_N})$	Theorem 30
	$\mathsf{TON} + (\mathsf{Ext}) + (\mathsf{N}\text{-}\mathsf{I_N})$	Theorem 30
	$\mathsf{TON} + (\mathsf{Ext}) + (\Sigma^+\text{-}\mathsf{I_N})$	Theorem 30
	$\mathsf{TON} + (\mathsf{Ext}) + (\mathsf{B}\Sigma^+\text{-}\mathsf{I_N})$	Theorem 31
	$\mathsf{PTO} + (\mathsf{W}\text{-}\mathsf{I_W})$	Theorem 63
	$\widehat{\mathsf{PTO}} + (\mathsf{O}\text{-}\mathsf{I_W})$	Theorem 64
	$\mathsf{PTO} + (\mathsf{Tot}) + (\mathsf{Ext}) + (\mathsf{W}\text{-}\mathsf{I_W})$	Theorem 63
	$\widehat{\mathsf{PTO}} + (\mathsf{Tot}) + (\mathsf{Ext}) + (\mathsf{O}\text{-}\mathsf{I_W})$	Theorem 64

PA	BON + (F$^+$-I$_\mathsf{N}$)	Proposition 32, Corollary 34
	BON + (F-I$_\mathsf{N}$)	Proposition 32, Corollary 34
	TON + (Ext) + (F$^+$-I$_\mathsf{N}$)	Theorem 35
	TON + (Ext) + (F-I$_\mathsf{N}$)	Theorem 35
	BON(μ) + (S-I$_\mathsf{N}$)	Proposition 73, Corollary 109
	TON(μ) + (Ext) + (S-I$_\mathsf{N}$)	Theorem 110
$(\Pi^0_\infty$-CA$)_{<\omega^\omega}$	BON(μ) + (O-I$_\mathsf{N}$)	Corollary 82, Corollary 109
	BON(μ) + (N-I$_\mathsf{N}$)	Corollary 80, Corollary 109
	BON(μ) + (Σ^+-I$_\mathsf{N}$)	Corollary 80, Corollary 109
	BON(μ) + (F$^+$-I$_\mathsf{N}$)	Corollary 80, Corollary 109
	TON(μ) + (Ext) + (O-I$_\mathsf{N}$)	Theorem 110
	TON(μ) + (Ext) + (N-I$_\mathsf{N}$)	Theorem 110
	TON(μ) + (Ext) + (Σ^+-I$_\mathsf{N}$)	Theorem 110
	TON(μ) + (Ext) + (F$^+$-I$_\mathsf{N}$)	Theorem 110
$(\Pi^0_\infty$-CA$)_{<\varepsilon_0}$	BON(μ) + (F-I$_\mathsf{N}$)	Theorem 74, Corollary 109
	TON(μ) + (Ext) + (F-I$_\mathsf{N}$)	Theorem 110

List of symbols

The following list of symbols is divided into three separate tables: basic systems, axioms and rules, and other symbols. The symbols in the first two tables are listed alphabetically (disregarding non-roman characters), the ones in the last table are given in the order of their appearance in the text.

A Basic systems

BON, 9	basic theory of operations and numbers
$\widehat{\mathsf{BON}}$, 53	BON plus N strict definition by cases
BON_λ, 19	λ version of BON
$\mathsf{BON}_{\lambda\sigma}$, 20	λ version of BON with explicit substitutions
$\mathsf{BON}(\mu)$, 58	BON plus axioms for μ
BOW, 33	basic theory of operations and binary words
$(\Pi^0_\infty\text{-}\mathsf{CA})_{<\alpha}$, 60	iterated arithmetic comprehension below α
$(\Pi^0_\infty\text{-}\mathsf{CA})_{\alpha}$, 60	iterated arithmetic comprehension up to α
E^+, 9	E^+ logic of partial terms
G_n, 55	applicative theories for Grzegorczyk classes
$\mathsf{I\Delta_0}$, 52, 55	bounded first order arithmetic
ID_1, 71	one iterated inductive definitions
$\widehat{\mathsf{ID}}_1$, 71	fixed point theory over Peano arithmetic
$\mathsf{ID}_1^{\#}$, 71	sharp $\widehat{\mathsf{ID}}_1$
KPu, 72	Kripke Platek set theory with urelements
PA, 24	Peano arithmetic
PA_Ω, 71	Peano arithmetic with ordinals
$\mathsf{PA}^{\mathsf{r}}_\Omega$, 70	restricted Peano arithmetic with ordinals
$\mathsf{PA}^{\mathsf{s}}_\Omega$, 70	$\mathsf{PA}^{\mathsf{r}}_\Omega$ plus Σ^Ω induction on the natural numbers
$\mathsf{PA}^{\mathsf{w}}_\Omega$, 71	weak Peano arithmetic with ordinals
PRA, 24	primitive recursive arithmetic
PTCA, 37	polynomial time computable arithmetic
PTCA^+, 37	PTCA plus Σ^b_1 induction
PTO, 35	applicative theory of polynomial time operations
$\widehat{\mathsf{PTO}}$, 53	PTO plus W strict definition by cases
PTO^+, 49	PTO plus $(\mathsf{NP\text{-}I_W})$

T_∞, 73	(infinitary) ramified system for ordinal theories
T_1, 75	Tait-style system for $\mathsf{PA}^{\mathsf{s}}_\Omega$
T_2, 78	(infinitary) Tait-style system for $\mathsf{PA}^{\mathsf{w}}_\Omega$
TON, 10	BON plus totality (Tot)
$\mathsf{TON}(\mu)$, 58	TON plus axioms for μ

B Axioms and rules

$(\mu.1)$, 58	first axiom for μ operator
$(\mu.2)$, 58	second axiom for μ operator
(ω), 74	ω rule
$(\mathsf{B}\Sigma^+\text{-}\mathsf{I_N})$, 28	$\mathsf{B}\Sigma^+$ induction on the natural numbers
$(\Sigma^b_\infty\text{-}\mathsf{CP})$, 38	bounded collection
$(\Sigma^+\text{-}\mathsf{CP_W})$, 51	Σ^+ collection on W
$(\mathsf{D_V})$, 10	full definition by cases
$(\mathsf{E}.1)$, 86	first axiom for E
$(\mathsf{E}.2)$, 86	second axiom for E
$(\mathcal{E}_n)$, 55	defining equations for the nth level of the Grzegorczyk hierarchy
(exp), 52	exponentiation
(Ext), 10	extensionality axiom
$(\mathsf{F}^+\text{-}\mathsf{I_N})$, 23	positive formula induction on the natural numbers
$(\mathsf{F}\text{-}\mathsf{I_N})$, 23, 71	full formula induction on the natural numbers
$(\mathsf{F}\text{-}\mathsf{I}_\Omega)$, 71	full formula induction on the ordinals
$(\Sigma^+\text{-}\mathsf{I_N})$, 23	Σ^+ induction on the natural numbers
$(\Sigma^0_1\text{-}\mathsf{I_N})$, 24	Σ^0_1 induction on the natural numbers
$(\Delta^\Omega_0\text{-}\mathsf{I_N})$, 70	Δ^Ω_0 induction on the natural numbers
$(\Sigma^\Omega\text{-}\mathsf{I_N})$, 70	Σ^Ω induction on the natural numbers
$(\Delta^\Omega_0\text{-}\mathsf{I}_\Omega)$, 70	Δ^Ω_0 induction on the ordinals
$(\mathsf{N}\text{-}\mathsf{I_N})$, 23	N induction on the natural numbers
$(\mathsf{NP}\text{-}\mathsf{I_W})$, 49	NP induction on binary words
$(\mathsf{O}\text{-}\mathsf{I_N})$, 23	operation induction on the natural numbers
$(\mathsf{O}\text{-}\mathsf{I_W})$, 53	operation induction on binary words
$(\Sigma\text{-}\mathsf{Ref})$, 38	Σ reflection
$(\Sigma^\Omega\text{-}\mathsf{Ref})$, 70	Σ^Ω reflection
$(\mathsf{S}\text{-}\mathsf{I_N})$, 23	set induction on the natural numbers
$(\mathsf{S}\text{-}\mathsf{I_W})$, 35	set induction on binary words
(Tot), 10	totality axiom
$(\mathcal{U}_n)$, 55	defining equations for $\{A_m : 3 \leq m \leq n\}$
$(\mathsf{W}\text{-}\mathsf{I_W})$, 53	W induction on binary words
$(\Sigma^b_\infty\text{-}\mathsf{WKL})$, 38	Weak König's lemma for Σ^b_∞ formulas

C Other symbols

Bibliography

[1] Abadi, M., Cardelli, L., Curien, P.-L., and Lévy, J.-J. Explicit substitutions. *Journal of Functional Programming 1*, 4 (1991), 375–416.

[2] Barendregt, H. P. *The Lambda Calculus*, revised ed. North Holland, Amsterdam, 1984.

[3] Beeson, M. J. *Foundations of Constructive Mathematics: Metamathematical Studies*. Springer, Berlin, 1985.

[4] Buchholz, W., and Sieg, W. A note on polynomial time computable arithmetic. *Contemporary Mathematics 106* (1990), 51–55.

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

[6] Buss, S. R. A conservation result concerning bounded theories and the collection axiom. *Proceedings of the American Mathematical Society 100*, 4 (1987), 709–715.

[7] Buss, S. R., and Scott, P. J., Eds. *Feasible Mathematics*, vol. 9 of *Progress in Computer Science and Applied Logic*. Birkhäuser, Basel, 1990.

[8] Cantini, A. Asymmetric interpretation for bounded theories. *Mathematical Logic Quarterly*. To appear.

[9] Cantini, A. *Logical Frameworks for Truth and Abstraction*. To appear.

[10] Cantini, A. On the computational content of theories of operations with total application, June 1995. Handwritten notes.

[11] Cantini, A. Proof-theoretic aspects of self-referential truth, September 1995. Draft.

[12] Clote, P., and Krajíček, J., Eds. *Arithmetic, Proof Theory and Computational Complexity*. Claredon Press, Oxford, 1993.

[13] Clote, P., and Remmel, J., Eds. *Feasible Mathematics II*, vol. 13 of *Progress in Computer Science and Applied Logic*. Birkhäuser, Basel, 1995.

[14] Cobham, A. The intrinsic computational difficulty of functions. In *Logic, Methodology and Philosophy of Science II*. North Holland, Amsterdam, 1964, pp. 24–30.

[15] Cook, S. A., and Kapron, B. M. Characterizations of the basic feasible functionals of finite type. In *Feasible Mathematics*, S. R. Buss and P. J. Scott, Eds. Birkhäuser, Basel, 1990, pp. 71–95.

[16] Cook, S. A., and Urquhart, A. Functional interpretations of feasibly constructive arithmetic. *Annals of Pure and Applied Logic 63*, 2 (1993), 103–200.

[17] Feferman, S. Formal theories for transfinite iteration of generalized inductive definitions and some subsystems of analysis. In *Intuitionism and Proof Theory, Proceedings of the Summer Conference at Buffalo, New York, 1968*, A. Kino, J. Myhill, and R. E. Vesley, Eds. North Holland, Amsterdam, 1970, pp. 303–326.

[18] Feferman, S. A language and axioms for explicit mathematics. In *Algebra and Logic*, J. Crossley, Ed., vol. 450 of *Lecture Notes in Mathematics*. Springer, Berlin, 1975, pp. 87–139.

[19] Feferman, S. A theory of variable types. *Revista Colombiana de Matemáticas 19* (1975), 95–105.

[20] Feferman, S. Constructive theories of functions and classes. In *Logic Colloquium '78*, M. Boffa, D. van Dalen, and K. McAloon, Eds. North Holland, Amsterdam, 1979, pp. 159–224.

[21] Feferman, S. Iterated inductive fixed-point theories: application to Hancock's conjecture. In *The Patras Symposium*, G. Metakides, Ed. North Holland, Amsterdam, 1982, pp. 171–196.

[22] Feferman, S. Hilbert's program relativized: proof-theoretical and foundational studies. *Journal of Symbolic Logic*, 53 (1988), 364–384.

[23] Feferman, S. Weyl vindicated: "Das Kontinuum" 70 years later. In *Temi e prospettive della logica e della filosofia della scienza contemporanee*. CLUEB, Bologna, 1988, pp. 59–93.

[24] Feferman, S. Finitary inductively presented logics. In *Logic Colloquium '88*, R. Ferro, C. Bonotto, S. Valentini, and A. Zanardo, Eds. North Holland, Amsterdam, 1989, pp. 191–220.

[25] Feferman, S. Polymorphic typed lambda-calculi in a type-free axiomatic framework. In *Logic and Computation*, W. Sieg, Ed., vol. 106 of *Contemporary Mathematics*. American Mathematical Society, Providence, Rhode Island, 1990, pp. 101–136.

[26] Feferman, S. Logics for termination and correctness of functional programs. In *Logic from Computer Science*, Y. N. Moschovakis, Ed., vol. 21 of *MSRI Publications*. Springer, Berlin, 1991, pp. 95–127.

[27] Feferman, S. Logics for termination and correctness of functional programs II: Logics of strength PRA. In *Proof Theory*, P. Aczel, H. Simmons, and S. S. Wainer, Eds. Cambridge University Press, Cambridge, 1992, pp. 195–225.

[28] Feferman, S., and Jäger, G. Systems of explicit mathematics with non-constructive μ-operator. Part I. *Annals of Pure and Applied Logic 65*, 3 (1993), 243–263.

[29] Feferman, S., and Jäger, G. Systems of explicit mathematics with non-constructive μ-operator. Part II. *Annals of Pure and Applied Logic*. To appear.

[30] Feferman, S., and Sieg, W. Proof-theoretic equivalences between classical and constructive theories for analysis. In *Iterated Inductive Definitions and Subsystems of Analysis: Recent Proof-Theoretical Studies*, W. Buchholz, S. Feferman, W. Pohlers, and W. Sieg, Eds., vol. 897 of *Lecture Notes in Mathematics*. Springer, Berlin, 1981, pp. 78–142.

[31] Ferreira, F. A note on a result of Buss concerning bounded theories and the collection scheme. Submitted to *Portugaliae Mathematica*.

[32] Ferreira, F. *Polynomial Time Computable Arithmetic and Conservative Extensions*. PhD thesis, Pennsylvania State University, 1988.

[33] Ferreira, F. Polynomial time computable arithmetic. *Contemporary Mathematics 106* (1990), 137–156.

[34] Ferreira, F. A feasible theory for analysis. *Journal of Symbolic Logic 59*, 3 (1994), 1001–1011.

[35] Friedman, H. Iterated inductive definitions and Σ^1_2-AC. In *Intuitionism and Proof Theory, Proceedings of the Summer Conference at Buffalo, New York, 1968*, A. Kino, J. Myhill, and R. E. Vesley, Eds. North Holland, Amsterdam, 1970, pp. 435–442.

[36] Girard, J.-Y. *Proof Theory and Logical Complexitiy*. Bibliopolis, Napoli, 1987.

[37] Glass, T. *Standardstrukturen für Systeme Expliziter Mathematik*. PhD thesis, Westfälische Wilhelms-Universität Münster, 1993.

[38] Glass, T., and Strahm, T. Systems of explicit mathematics with non-constructive μ-operator and join. *Annals of Pure and Applied Logic*. To appear.

[39] Hájek, P., and Pudlák, P. *Metamathematics of First-Order Arithmetic*. Perspectives in Mathematical Logic. Springer, 1993.

[40] Hayashi, S., and Nakano, H. *PX: A Computational Logic*. MIT Press, Cambridge, MA, 1988.

[41] Hindley, J. R., and Seldin, J. P. *Introduction to Combinators and λ-Calculus*. Cambridge University Press, 1986.

[42] Hinman, P. G. *Recursion-Theoretic Hierarchies*. Springer, Berlin, 1978.

[43] Jäger, G. Beweistheorie von *KPN*. *Archiv für mathematische Logik und Grundlagenforschung 20* (1980), 53–64.

[44] Jäger, G. Zur Beweistheorie der Kripke-Platek-Mengenlehre über den natürlichen Zahlen. *Archiv für mathematische Logik und Grundlagenforschung 22* (1982), 121–139.

[45] Jäger, G. A well-ordering proof for Feferman's theory T_0. *Archiv für mathematische Logik und Grundlagenforschung 23* (1983), 65–77.

[46] Jäger, G. Induction in the elementary theory of types and names. In *Computer Science Logic '87*, E. Börger et al., Ed., vol. 329 of *Lecture Notes in Computer Science*. Springer, Berlin, 1988, pp. 118–128.

[47] Jäger, G. Type theory and explicit mathematics. In *Logic Colloquium '87*, H.-D. Ebbinghaus, J. Fernandez-Prida, M. Garrido, M. Lascar, and M. R. Artalejo, Eds. North Holland, Amsterdam, 1989, pp. 117–135.

[48] Jäger, G. Fixed points in Peano arithmetic with ordinals. *Annals of Pure and Applied Logic 60*, 2 (1993), 119–132.

[49] Jäger, G., and Pohlers, W. Eine beweistheoretische Untersuchung von $(\Delta^1_2\text{-CA}) + (\text{BI})$ und verwandter Systeme. In *Sitzungsberichte der Bayerischen Akademie der Wissenschaften, Mathematisch-naturwissenschaftliche Klasse*. 1982, pp. 1–28.

[50] Jäger, G., and Strahm, T. Some theories with positive induction of ordinal strength $\varphi\omega 0$. *Journal of Symbolic Logic*. To appear.

[51] Jäger, G., and Strahm, T. Second order theories with ordinals and elementary comprehension. *Archive for Mathematical Logic 34*, 6 (1995), 345–375.

[52] Jäger, G., and Strahm, T. Totality in applicative theories. *Annals of Pure and Applied Logic 74*, 2 (1995), 105–120.

[53] Kahle, R. Weak inductions in theories of operations and numbers. Tech. rep., Institut für Informatik und angewandte Mathematik, Universität Bern, 1995.

[54] Kahle, R. Personal communication, March 1996.

[55] Parsons, C. On a number theoretic choice schema and its relation to induction. In *Intuitionism and Proof Theory, Proceedings of the Summer Conference at Buffalo N.Y., 1968*, J. Myhill, A. Kino, and R. Vesley, Eds. North Holland, Amsterdam, 1970, pp. 459–473.

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

[57] Rogers, H. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill, New York, 1967.

[58] Rose, H. E. *Subrecursion: Functions and Hierarchies*, vol. 9 of *Oxford Logic Guides*. Clarendon Press, Oxford, 1984.

[59] Schütte, K. *Proof Theory*. Springer, Berlin, 1977.

[60] Schwichtenberg, H. Proof theory: Some applications of cut-elimination. In *Handbook of Mathematical Logic*, J. Barwise, Ed. North Holland, Amsterdam, 1977, pp. 867–895.

[61] Seth, A. *Complexity Theory of Higher Type Functionals*. PhD thesis, Tata Institute of Fundamental Research, Bombay, 1994.

[62] Sieg, W. Herbrand analyses. *Archive for Mathematical Logic*, 30 (1991).

[63] Simpson, S. G. Subsystems of second order arithmetic. Tech. rep., Pennsylvania State University, 1986. Chapters II–V and VII.

[64] Stärk, R. F. Applicative theories with explicit substitutions and call-by-value evaluation. Draft Notes, January 1995.

[65] Stärk, R. F. Partial combinatory logic and the call-by-value λ-calculus. Draft Notes, March 1995.

[66] Strahm, T. Polynomial time operations in explicit mathematics. *Journal of Symbolic Logic*. To appear.

[67] Strahm, T. Theories with self-application of strength PRA. Master's thesis, Institut für Informatik und angewandte Mathematik, Universität Bern, 1992.

[68] Strahm, T. Partial applicative theories and explicit substitutions. *Journal of Logic and Computation 6*, 1 (1996).

[69] Talcott, C. A theory for program and data type specification. *Theoretical Computer Science 104* (1992), 129–159.

[70] Troelstra, A., and van Dalen, D. *Constructivism in Mathematics*, vol. I. North-Holland, Amsterdam, 1988.

[71] Troelstra, A., and van Dalen, D. *Constructivism in Mathematics*, vol. II. North Holland, Amsterdam, 1988.

[72] Wainer, S. S. Four lectures on primitive recursion. In *Logic and Algebra of Specification*, F. L. Bauer, W. Brauer, and H. Schwichtenberg, Eds., vol. 94 of *NATO ASI Series*. Springer, Berlin, 1993, pp. 377–410.