



A flexible type system for the small Veblen ordinal

Florian Ranzi¹ · Thomas Strahm¹ 

Received: 28 January 2016 / Accepted: 14 January 2019 / Published online: 29 January 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract

We introduce and analyze two theories for typed (accessible part) inductive definitions and establish their proof-theoretic ordinal to be the small Veblen ordinal $\vartheta\Omega^\omega$. We investigate on the one hand the applicative theory FIT of functions, (accessible part) inductive definitions, and types. It includes a simple type structure and is a natural generalization of S. Feferman's system $QL(F_0\text{-}IR_N)$. On the other hand, we investigate the arithmetical theory TID of typed (accessible part) inductive definitions, a natural subsystem of ID_1 , and carry out a wellordering proof within TID that makes use of fundamental sequences for ordinal notations in an ordinal notation system based on the finitary Veblen functions. The essential properties for describing the ordinal notation system are worked out.

Keywords Proof theory · Inductive definitions · Applicative theories · Small Veblen ordinal · Finitary Veblen functions · Metapredicativity · Higher types · Subsystems of second-order arithmetic

Mathematics Subject Classification 03F03 · 03F15 · 03F35 · 03F50

1 Introduction

Feferman [10], introduced a two-sorted quantificational logic and showed that it has the same strength as (Skolem's system of) primitive recursive arithmetic. The characteristics of this two-sorted quantificational logic are that it is an *applicative theory* augmented by type variables as the second sort and with a refined notion of comprehension terms, so-called type and function terms. In particular, this theory embodies a rule $(F_0\text{-}IR_N)$ that is called the function-induction rule on N (where N is interpreted

Florian Ranzi's research was supported by the Swiss National Science Foundation.

✉ Thomas Strahm
strahm@inf.unibe.ch
<http://www.inf.unibe.ch/~strahm>

¹ Institut für Informatik, Universität Bern, Neubrückstr. 10, 3012 Bern, Switzerland

as the type for the natural numbers). It was shown to be closed under a strengthening of this rule to *finitary* inductively generated types I , called $(F_0\text{-IR}_I)$.

This kind of theory strongly influenced the shape of our applicative theory FIT that we are going to introduce in this article. Our motivation to examine the theory in [10] was to find a natural theory for carrying out metapredicative¹ wellordering proofs in the spirit of higher type functionals for ordinals. It seemed to provide a suitable environment for doing so. But soon, we realized that aside from this, the theory gave rise to the question of what consequence a function-induction rule for *infinitary* inductively generated types would have on the one side and to the idea of implementing the wellordering proofs through accessible part inductive definitions on the other side (having in mind our desire for metapredicative wellordering proofs). Hence, we tackle this question on infinitary inductively generated types only for inductively generated types that correspond to the (inductively defined) *accessible part* $I_{\mathbb{P}, \mathbb{Q}}$ for a (binary) relation \mathbb{Q} on a domain \mathbb{P} . In fact, our methods implicitly suggest that we get the same result for the variant where we allow for general inductively generated types.

FIT stands for “theory for function(al)s, non-iterated inductive definitions, and types (of level 1)”, and it represents the first step for a generalization of the theory in [10] which turns out to have the small Veblen ordinal as measure for its proof-theoretic strength, i.e., $\vartheta \Omega^\omega$ when using the terminology of [20]. Theories that have $\vartheta \Omega^\omega$ as proof-theoretic strength are for instance $\Pi_2^1\text{-BI}_0$ from [20] or more recently $\text{RCA}_0 + (\Pi_1^1(\Pi_3^0)\text{-CA}_0)^-$ from [25]. While these theories are analyzed by *impredicative* proof-theoretic methods, our treatment of FIT uses *metapredicative* methods for the lower bound. For the upper bound, we use an embedding into $\Pi_3^1\text{-RFN}_0$ and get a desired upper bound result in the realm of metapredicative proof-theory due to D. Probst’s *modular ordinal analysis* from [17] that determines by metapredicative methods the proof-theoretic ordinal of various theories with strength below (and reaching to) the Bachmann–Howard ordinal $\vartheta \varepsilon_{\Omega+1}$. One of these theories is $\Pi_3^1\text{-RFN}_0$ (which is denoted by $\mathfrak{p}_3(\text{ACA}_0)$ in [17]) and determined to have the proof-theoretic strength of the small Veblen ordinal. Furthermore, we mention the system $\text{KPi}^0 + (\Pi_3\text{-Ref})$ from [15] which is also related to $\Pi_3^1\text{-RFN}_0$. In particular, Jäger and Strahm [15] explains how the proof-theoretic strength of $\text{KPi}^0 + (\Pi_3\text{-Ref})$ can be determined to be $\vartheta \Omega^\omega$ by metapredicative methods.

1.1 Results on the theory FIT

We now explain the methods used in this article for the ordinal analysis of FIT. First, we shall consider a canonical implementation of FIT as a subsystem of ID_1 (while the latter is an arithmetical first-order theory for non-iterated general inductive definitions, see [3]) in which metapredicative wellordering proofs can be carried out in a perspicuous way and where the interpretation back into FIT is straight-forward. This subsystem of ID_1 is called TID for “theory of typed (accessible part) inductive definitions (of level

¹ The notion *metapredicativity* is meant in general for the approach to use proof-theoretic methods from the realm of *predicative* proof-theory instead of *impredicative* methods. In particular for wellordering proofs, we aim to avoid the use of so-called collapsing functions. We refer to [24] or [13]. For further reading on *metapredicativity*, we refer to [12, 15].

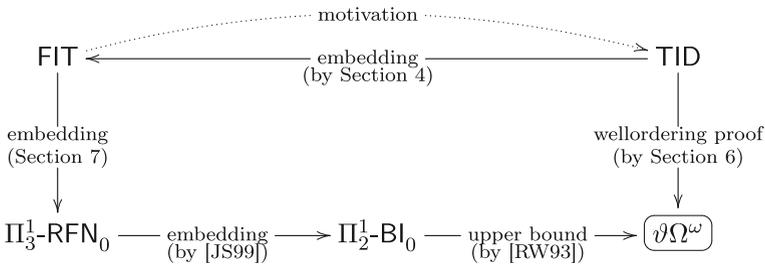


Fig. 1 Strategy to determine the proof-theoretic ordinal $\vartheta\Omega^\omega$ of FIT

1)” and essentially arises from ID_1 by restricting to accessible part inductive definitions and adapting the closure axioms, its induction scheme on the natural numbers, and its generalized induction scheme (ID) to (the translation of) the function types of FIT, akin to the restriction of ID_1 to the theory $ID_1^* \upharpoonright$ from [16].

For the (proof-theoretic) upper bound of FIT (and hence for TID), we shall embed it into the system $\Pi_3^1\text{-RFN}_0$ of second-order arithmetic for Π_3^1 ω -model reflection. In order to obtain the desired upper bound $\vartheta\Omega^\omega$, we shall use the results from [20] by impredicative methods, noting that the (meta)predicative treatment from [17] has not been published yet. Figure 1 depicts the abovementioned approaches accordingly. Furthermore, we give in the conclusion of this article (Sect. 8) some remarks on the canonical generalization of TID to a theory TID^f for general typed inductive definitions with the full range of positive arithmetical operator forms, leading to the same proof-theoretic strength of TID and TID^f . We refer here also to [18] for an extension TID_1^+ of TID with proof-theoretic ordinal $\vartheta\Omega^\Omega$, i.e., the large Veblen ordinal.

2 General notational framework

We work with three conceptually different kinds of logical frameworks: In Sect. 3 with the two-sorted theory FIT (with language \mathcal{L}_{FIT}) that is an applicative theory enhanced by a type system; in Sect. 4 with the first-order theory TID (with language \mathcal{L}_{TID}) that is an extension of Peano arithmetic PA (with language \mathcal{L}_{PA}) by new predicates; and in Sect. 7 with subsystems of second order arithmetic (with language \mathcal{L}_{PA}^2). Hence, we work with up to two sorts of (countably many) variables and use

$$a, b, c, d, u, v, w, x, y, z$$

as syntactic variables for the first sort,

$$U, V, W, X, Y, Z$$

as syntactic variables for the second sort, and

$$=, \neg, \rightarrow, \vee, \wedge, \exists, \forall$$

as basic logical symbols.

Assume for now that the notion of \mathcal{L} terms and \mathcal{L} formulas has been already introduced for \mathcal{L} being one of the languages \mathcal{L}_{FIT} , \mathcal{L}_{TID} , \mathcal{L}_{PA} , or $\mathcal{L}_{\text{PA}}^2$. If \mathcal{L} is clear from the context, we may drop the reference to \mathcal{L} by just using the notions term and formula. We use s, t, r as syntactic variables for \mathcal{L} terms and A, B, C, D, E, F as syntactic variables to denote \mathcal{L} formulas. *Literals* are atomic \mathcal{L} formulas or their negated version. Introducing an \mathcal{L} formula as $A(a)$ means A denotes this formula and that the variable a may occur freely in A . $\text{FV}(A)$ denotes the set of free variables of the first sort of A . We shall use the following notational conventions that a, b, c, d, u, v, w denote free variables of the first sort within an \mathcal{L} formula and that k, l, m, n, p, q denote variables in our meta-theory, i.e., as ranging there over the natural numbers. Parentheses may be used to make expressions unambiguous or more readable. We prefer *infix notation* rather than prefix notation when dealing with binary function and relation symbols. For \rightarrow , we follow the usual convention of right-associativity, e.g., $A \rightarrow B \rightarrow C$ denotes $A \rightarrow (B \rightarrow C)$. We write $A \leftrightarrow B$ to denote $(A \rightarrow B) \wedge (B \rightarrow A)$. Moreover, \wedge binds stronger than \rightarrow .

2.1 Vector notations

If $*$ denotes one of the syntactic variables that will be introduced in this article, then we allow the usual annotations such as $*', \tilde{*}$, or subscripts $*_i$ (for $i \in \mathbb{N}$, i.e., for natural numbers i).

For subscripts, we also use the vector notation $\vec{*}$ to denote lists of the form $*_1, \dots, *_n$ for some $n \in \mathbb{N}$. Moreover, when working with ordinal notations based on the finitary Veblen functions, the following notation will come in handy. We write

$$\vec{*}^{(n)}$$

for $n \in \mathbb{N}$ to denote the list $*_1, \dots, *_n$, and we may write $\vec{*}^{(k)}$ for any $k \in \mathbb{N}$ in order to denote $*_1, \dots, *_{\min(k,n)}$. In rare cases we write for specific constants c (e.g., for 0) the expression $\vec{c}^{(n)}$ to denote the list c, \dots, c of length n , and hence we read \vec{c} analogously. If $n = 0$, then $\vec{*}^{(n)}$ and $\vec{*}$ denote the empty list.

Applications of all these notations will be obvious, following common conventions—for instance $\forall \vec{x}^{(3)} A$ shall abbreviate $\forall x_1 \forall x_2 \forall x_3 A$ as usual, and $\forall \vec{x} A$ is just A if \vec{x} is the empty list. Also when writing $f \vec{t}^{(n)}$ for a list of terms $\vec{t}^{(n)}$ and an n -ary function symbol f , it is usually meant to abbreviate $f t_1 \dots t_n$ rather than $f t_1, \dots, t_n$.

2.2 Class terms and substitution

\mathcal{L} class terms are objects of the form

$$\Lambda a. A$$

for any \mathcal{L} formula A and we use $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ as syntactic variables for \mathcal{L} class terms. Sometimes, class terms are also called comprehension terms, and we do not use the

more common notations $\{a : A\}$ or $\lambda a.A$ because these notions are already reserved in our setting of the applicative theory FIT.

Substitution of a variable a in an \mathcal{L} formula A by an \mathcal{L} term t is denoted by $A(t/a)$ and $A_a(t)$, or just by $A(t)$ in case A has been introduced in the form $A(a)$, and as usual we assume (if necessary) an appropriate renaming of bound variables in A to avoid a clash of bound variables. Then for \mathcal{A} being $\Lambda a.A$, we set

$$\mathcal{A}(t) := t \in \mathcal{A} := A(t/a)$$

for any \mathcal{L} term t and write ambiguously $\mathcal{A} \in \mathcal{L}$ to stress that \mathcal{A} is an \mathcal{L} class term. Moreover, we also extend this to lists of variables $\vec{a} = a_1, \dots, a_n$ and have objects of the form $\Lambda a_1 \dots \Lambda a_n.A$ or $\Lambda \vec{a}.A$ for short with $(\Lambda \vec{a}.A)(\vec{t}) := A(\vec{t}/\vec{a})$ for terms $\vec{t} = t_1, \dots, t_n$ and where $A(\vec{t}/\vec{a})$ is obtained by simultaneously replacing in A all free occurrences of the variables \vec{a} by \vec{t} , while renaming bound variables may be necessary to avoid a clash of variables. Analogously for $k < n$, we use $(\Lambda \vec{a}^{(n)}.A)(\vec{t}^{(k)})$ to denote $\Lambda a_{k+1} \dots \Lambda a_n.(A(\vec{t}^{(k)}/\vec{a}^{(k)}))$ while tacitly assuming a reasonable variable-renaming to avoid clashes of variables.

In case that \mathcal{L} also embodies variables X, Y, Z of the second sort, we mean by *substitution of a variable X* in an \mathcal{L} formula A by an \mathcal{L} class term \mathcal{B} the expression $A(\mathcal{B}/X)$ which is obtained from A by substituting any atomic formula Xt with $\mathcal{B}(t)$ while a renaming of bound variables may be necessary as usual. If A has been introduced in the form $A(X)$, we may also just write $A(\mathcal{B})$ for $A(\mathcal{B}/X)$.

In case R is a unary relation symbol in \mathcal{L} or a second sort variable, we also define $A(R/X) := A(\mathcal{B}/X)$ for $\mathcal{B} := \Lambda a.Ra$. Furthermore, if \mathcal{A} is an \mathcal{L} class term $\Lambda a.A$, then we set $\mathcal{A}(\mathcal{B}/X) := \Lambda a.A(\mathcal{B}/X)$. Accordingly, we let substitution for number variables be defined by $\mathcal{A}_z(t) := A(t/z) := \Lambda a.A(t/z)$ assuming a renaming of the variable a in case of clashes of variables.

3 The theory FIT for functions, inductive definitions, and types

We introduce a basic language that is needed for the applicative part of FIT before defining the full language \mathcal{L}_{FIT} in Sect. 3.1.

Definition 3.1 The following *basic language* of FIT is based on a first sort of variables called *individual variables* and a second sort variables called *type variables*.

- (a) *Constants of the first sort:*
 $k, s, p, p_0, p_1, 0, s_N, p_N, d_N$, denoting the usual applicative constants.
- (b) *Constants of the second sort:*
 N, \bar{N}, U , for the natural numbers, its complement, and a symbol without further interpretation but needed for proof-theoretic investigations.
- (c) *Relation symbols of the first sort:*
 $=, \downarrow$, denoting equality and definedness for individual terms.²

² Individual terms will be defined in Definition 3.2.

- (d) *Further symbols:*
 \cdot, \in , denoting a binary function symbol for first sort term application and a binary relation symbol between individual terms and types.³

Definition 3.2 *Individual terms* s, t, r are defined inductively from individual variables and constants by use of the binary function symbol \cdot as usual.

Definition 3.3 The following notions and abbreviations will serve as basic applicative tools.

- (a) $t' := s_N t$ and $1 := 0'$.
- (b) $st_1 \dots t_n$ or also $s(t_1, \dots, t_n)$ denotes *term application on n inputs* and is defined recursively on $n \geq 0$ as s for $n = 0$ and $(s \cdot t_1)t_2 \dots t_n$ otherwise.
- (c) $\langle s_0, \dots, s_{n-1} \rangle^{\text{FIT}}$ denotes *general n -tupling* and is defined recursively on $n \geq 0$ as 0 if $n = 0$ and $\text{ps}_0 \langle s_1, \dots, s_{n-1} \rangle^{\text{FIT}}$ otherwise. Write also $\langle s_0, \dots, s_{n-1} \rangle$ if the meaning is clear from the context (e.g., in the proof of Lemma 4.23).
- (d) $(s)_n^{\text{FIT}}$ denotes *n -th projection* and is defined recursively on $n \geq 0$ as $\text{p}_0 s$ if $n = 0$ and $(\text{p}_1 s)_{n-1}^{\text{FIT}}$ otherwise. Write also $(s)_n$ if the meaning is clear from the context.
- (e) $\lambda x.t$ denotes *lambda abstraction of a variable x on a term t* and is defined in the usual way by making use of the combinators s and k , while $\lambda x.t$ does not contain the variable x . In general, $\lambda \vec{x}.t$ for a list of variables $\vec{x} = x_1, \dots, x_n$ denotes recursively $\lambda x_1.(\lambda x_2. \dots \lambda x_n.t)$ if $n > 0$ and t otherwise. See [18, Definition 5.3.(e)] for details.

3.1 Syntax of FIT

Definition 3.4 The language \mathcal{L}_{FIT} is defined simultaneously and inductively with the notions for *formulas* (For), *positive formulas* (For⁺), *types* (Ty), *restricted types* (Ty|), and *terms of the second sort*:

- (a) \mathcal{L}_{FIT} extends the basic language from Definition 3.1 by the following new (syntactically different) kinds of *terms of the second sort* while demanding $A \in \text{For}^+$ and $\mathbb{P}, \mathbb{Q} \in \text{Ty}|$:

$$\{x : A\} \quad \text{and} \quad \mathbb{P}\mathbb{Q}$$

- (b) For denotes the collection of *formulas* A, B, C, D , which consists of the expressions

$$t \in \mathbb{P} \quad t \in \mathbb{U} \quad t \downarrow \quad s = t$$

$$\neg A \quad A \rightarrow B \quad A \vee B \quad A \wedge B \quad \exists x A \quad \forall x A \quad \exists X A \quad \forall X A$$

and we demand here $\mathbb{P} \in \text{Ty}$. We sometimes write $A \in \mathcal{L}_{\text{FIT}}$ ambiguously for $A \in \text{For}$.

- (c) For⁺ denotes the collection of *positive (elementary) formulas*, i.e., $A \in \text{For}$ such that

³ Types will be defined in Definition 3.4.

- quantifications of type variables do not occur and
 - expressions of the form $t \in \mathbb{P}$ for types $\mathbb{P} \in \text{Ty}$ occur at most positively.⁴
- (d) Ty denotes the collection of *types* $\mathbb{P}, \mathbb{Q}, \mathbb{R}$ (also called *positive types*), i.e., expressions

$$X, Y, Z, \dots \text{ (i.e., type variables)}$$

$$\mathbb{N} \ \bar{\mathbb{N}} \ \{x : A\} \ \mathbb{I}_{\mathbb{P}\mathbb{Q}}$$

while demanding $A \in \text{For}^+$ and $\mathbb{P}, \mathbb{Q} \in \text{Ty} \uparrow$. Note that \mathbb{U} itself is not treated as a type.

- (e) $\text{Ty} \uparrow$ denotes the collection of *restricted types*, i.e., types such that
- no type variables and
 - no expressions of the form $\mathbb{I}_{\mathbb{P}\mathbb{Q}}$ occur

Definition 3.5 Let \rightarrow be a new distinguished symbol in our meta-language. The collection FT of *function types* $\mathbb{F}, \mathbb{G}, \mathbb{H}$ denotes objects in the meta-language that are defined inductively to consist of expressions of the form

$$(\mathbb{P}_1 \rightarrow (\dots (\mathbb{P}_{n-1} \rightarrow \mathbb{P}_n) \dots))$$

for any $\mathbb{P}_1, \dots, \mathbb{P}_n \in \text{Ty}$. We let \rightarrow be right-associative and we write just $\mathbb{P}_1 \rightarrow \dots \mathbb{P}_{n-1} \rightarrow \mathbb{P}_n$.

Remark 3.6 Note that function types are defined as objects in the *meta-language*. Within \mathcal{L}_{FIT} formulas, these new expressions will only occur in combination with terms in the form $t \in \mathbb{P} \rightarrow \mathbb{F}$, i.e., as abbreviations for proper \mathcal{L}_{FIT} formulas as in Notation 3.8. For remarks on how to make function types first-class members of \mathcal{L}_{FIT} , see [10] and [18, Remark 5.10.(b)]. Further, we do not need \mathbb{U} to be a type because $t \in \mathbb{U}$ can be obtained from $\{x : x \in \mathbb{U}\}$ and (CA^+) in FIT (see Definition 3.10).

Definition 3.7 The notion of $\text{FV}(A)$ is extended to the notion of atomic formulas $t \in \mathbb{P}$ for $\mathbb{P} \in \text{Ty}$ recursively in the canonical way as the union of the expressions involved. In particular, $\mathbb{I}_{\mathbb{P}\mathbb{Q}}$ is $\text{FV}(\mathbb{P}') \cup \text{FV}(\mathbb{Q}')$. With this extension explained, the *substitution of individual and type variables* is defined as in Sect. 2.

Notation 3.8 We have the following abbreviations for some formulas and types:

- $s \simeq t$ is $(s \downarrow \vee t \downarrow) \rightarrow s = t$.
- $s \neq t$ is $s \downarrow \wedge t \downarrow \wedge \neg(s = t)$.
- $t \in \mathbb{P} \rightarrow \mathbb{F}$ is recursively $\forall x(x \in \mathbb{P} \rightarrow tx \in \mathbb{F})$.
- $\mathbb{N}^{n+1} \rightarrow \mathbb{F}$ is recursively $\mathbb{N} \rightarrow (\mathbb{N}^n \rightarrow \mathbb{F})$ where $\mathbb{N}^0 \rightarrow \mathbb{F}$ is \mathbb{F} .
- $t \notin \mathbb{F}$ is $\neg(t \in \mathbb{F})$.

⁴ Positive is meant in the usual way: $t \in \mathbb{P}$ is called *positive* in $A \in \text{For}$ if it does not occur in negated form $\neg(t \in \mathbb{P})$ in A' , while A' shall be the translation of A where first each subformula of the form $B_1 \rightarrow B_2$ is transformed to $\neg B_1 \vee B_2$ and where we then move the negation symbol \neg next to atomic formulas, while making use of De Morgan's laws and the law of double negation.

- $(\exists x \in \mathbb{F})B$ is $\exists x(x \in \mathbb{F} \wedge B)$.
- $(\forall x \in \mathbb{F})B$ is $\forall x(x \in \mathbb{F} \rightarrow B)$.
- $\text{Cl}_{\mathbb{P}, \mathbb{Q}}(\mathcal{A})$ is $\forall x((x \in \mathbb{P} \wedge (\forall y \in \mathbb{P})(\langle y, x \rangle \in \mathbb{Q} \rightarrow \mathcal{A}(y))) \rightarrow \mathcal{A}(x))$.
We assume as usual for such notational abbreviations that x, y are supposed to not occur in \mathcal{A}, \mathbb{P} , and \mathbb{Q} . This shall hold analogously for similar such abbreviations for formulas.
- $A(\mathbb{F}/X)$ for the formula obtained by substituting any occurrence of $t \in X$ in A by $t \in \mathbb{F}$.

3.2 The theory FIT

Definition 3.9 The logic of FIT is a two-sorted logic whose first-order part (i.e., for individual variables) is based on the *classical logic of partial terms* LPT due to Beeson [2]:

- *Propositional axioms and rules* The usual propositional axioms and rules, based on some sound Hilbert calculus for classical propositional logic.
- *Quantificational logic for the first sort* For A being an \mathcal{L}_{FIT} formula and t an individual term, we have

$$(\forall x A \wedge t \downarrow) \rightarrow A(t/x) \quad (A(t/x) \wedge t \downarrow) \rightarrow \exists x A$$

and for A, B being \mathcal{L}_{FIT} formulas and $x \notin \text{FV}(A)$, we have the following figures:

$$\frac{A \rightarrow B}{A \rightarrow \forall x B} \quad \frac{B \rightarrow A}{\exists x B \rightarrow A}$$

- *Quantificational logic for the second sort* For A, B being \mathcal{L}_{FIT} formulas and \mathbb{P} a type, we have

$$\forall X A \rightarrow A(\mathbb{P}/X) \quad A(\mathbb{P}/X) \rightarrow \exists X A$$

and for A, B being \mathcal{L}_{FIT} formulas and X not occurring free in A , we have the following figures:

$$\frac{A \rightarrow B}{A \rightarrow \forall X B} \quad \frac{B \rightarrow A}{\exists X B \rightarrow A}$$

- *Equality axioms*

$$x = x \quad (x_1 = y_1 \wedge \dots \wedge x_n = y_n \wedge A) \rightarrow (\dots (A(y_1/x_1)) \dots (y_n/x_n))$$

- *Definedness axioms* For constants c of the first sort of \mathcal{L}_{FIT} , types \mathbb{P} , and individual terms t , we have

$$c \downarrow \wedge x \downarrow \quad (st) \downarrow \rightarrow (s \downarrow \wedge t \downarrow) \quad s = t \rightarrow (s \downarrow \wedge t \downarrow) \quad t \in \mathbb{P} \rightarrow t \downarrow \quad t \in \mathbf{U} \rightarrow t \downarrow$$

Writing $\vdash A$ for any \mathcal{L}_{FIT} formula A denotes the derivability of A in the logic of FIT.

Definition 3.10 FIT is the two-sorted applicative theory based on the logic of partial terms LPT (and on [10]). Its non-logical axioms are as follows:

I. Applicative axioms

I.1 Partial combinatory algebra

$$kxy = x \quad sxy\downarrow \wedge sxyz \simeq (xz)(yz)$$

I.2 Pairing and projection

$$p_0(pxy) = x \wedge p_1(pxy) = y$$

I.3 Definition by numerical cases

$$x \in \mathbb{N} \wedge y \in \mathbb{N} \wedge x = y \rightarrow d_{\mathbb{N}z_1z_2}xy = z_1$$

$$x \in \mathbb{N} \wedge y \in \mathbb{N} \wedge x \neq y \rightarrow d_{\mathbb{N}z_1z_2}xy = z_2$$

I.4 Axioms about \mathbb{N} and $\bar{\mathbb{N}}$

$$0 \in \mathbb{N} \wedge (x \in \mathbb{N} \rightarrow x' \in \mathbb{N}) \qquad x \in \mathbb{N} \rightarrow (x' \neq 0 \wedge p_{\mathbb{N}}(x') = x)$$

$$(x \in \mathbb{N} \wedge x \neq 0) \rightarrow (p_{\mathbb{N}}x \in \mathbb{N} \wedge (p_{\mathbb{N}}x)' = x) \quad x \in \bar{\mathbb{N}} \leftrightarrow x \notin \mathbb{N}$$

II. Induction on \mathbb{N} for $\mathbb{F} \in \text{FT}$

$$(\text{FT-Ind}) \quad t0 \in \mathbb{F} \wedge (\forall x \in \mathbb{N})(tx \in \mathbb{F} \rightarrow tx' \in \mathbb{F}) \rightarrow t \in (\mathbb{N} \rightarrow \mathbb{F})$$

III. Positive comprehension for $A \in \text{For}^+$

$$(\text{CA}^+) \quad y \in \{x : A\} \leftrightarrow A(y/x)$$

IV. Axioms about $\mathbb{I}_{\mathbb{P}\mathbb{Q}}$ for $\mathbb{F} \in \text{FT}$ and $\mathbb{P}, \mathbb{Q} \in \text{Ty}$

$$(\text{FT-Cl}) \quad \text{Cl}_{\mathbb{P},\mathbb{Q}}(\Lambda z.z \in \mathbb{I}_{\mathbb{P}\mathbb{Q}}) \quad (\text{FT-ID}) \quad \text{Cl}_{\mathbb{P},\mathbb{Q}}(\Lambda z.tz \in \mathbb{F}) \rightarrow t \in (\mathbb{I}_{\mathbb{P}\mathbb{Q}} \rightarrow \mathbb{F})$$

Writing $\text{FIT} \vdash A$ for any \mathcal{L}_{FIT} formula A denotes the derivability of A from these axioms in the logic of FIT given in Definition 3.9.

Lemma 3.11

(a) *Lambda abstraction: For all \mathcal{L}_{FIT} terms t, s and $\vec{s} = s_1, \dots, s_n$, and all individual variables y and $\vec{x} = x_1, \dots, x_n$ with $y \notin \{x_1, \dots, x_n\}$, we have the following:*

1. $\text{FIT} \vdash (\lambda \vec{x}.t)\downarrow \wedge (\lambda \vec{x}.t)\vec{x} \simeq t.$
2. $\text{FIT} \vdash (s_1\downarrow \wedge \dots \wedge s_n\downarrow) \rightarrow (\lambda \vec{x}.t)\vec{s} \simeq t(\vec{s}/\vec{x}).$
3. $\text{FIT} \vdash (\lambda \vec{x}.t)(s/y)x \simeq (\lambda \vec{x}.t(s/y))x.$

- (b) *Fixed-point:* There exists a closed term fix such that $\text{FIT} \vdash \text{fix}x\downarrow \wedge \text{fix}yx \simeq y(\text{fix}y)x$ holds for all number variables x, y .
- (c) *Pairs and tupling:* For all \mathcal{L}_{FIT} variables x_0, \dots, x_n and each $0 \leq i \leq n$, we have $\text{FIT} \vdash (s_0\downarrow \wedge \dots \wedge s_n\downarrow) \rightarrow ((s_0, \dots, s_n))_i = s_i$.

Proof The applicative part of FIT corresponds to the standard axioms and constants that appear in applicative theories. For details on (a) and (b), see for instance [9]. In particular for (c), we remark that this follows easily from the axioms of FIT, in particular by making use of **I.2** from Definition 3.10 and the definedness axioms. \square

3.3 Informal interpretation of FIT

Since FIT directly evolved from Feferman's theory $\text{QL}(F_0\text{-IR}_N)$, we refer for a thorough motivation and informal interpretation of FIT to [10, Sections 2 and 5]. Moreover, the special constant U can be interpreted as a subset of the natural numbers, having no further interpretation. It is needed for proof-theoretic investigations.

Regarding the definition of axiom (FT-ID), we can motivate this as an induction principle with respect to the totality notion of a (type 1) function, while noting hereby the restriction $\mathbb{F} \in \text{FT}$ in the definition of (FT-ID). The restriction $\mathbb{P}, \mathbb{Q} \in \text{Ty}|$ can be seen as accounting for non-iterated inductive definitions.

4 The theory TID for typed inductive definitions

FIT is a natural theory for specifying the behaviour of an applicative term t by use of types, say by a function type $\mathbb{P}_1 \rightarrow \dots \rightarrow \mathbb{P}_{n+1}$ that consists of types. For checking this behaviour, we have the axiom schemes (FT-lnd) and (FT-ID) at hand. The latter allows the discussion of the behaviour of an operation t that acts on the inductively defined accessible part of a given binary relation (e.g., if \mathbb{P}_1 is $\mathbb{I}_{\mathbb{P}, \mathbb{Q}}$ in the example above). This gives an idea for the following definition of the theory TID for typed inductive definitions as a subtheory of ID_1 . Before turning to the definition of TID, we introduce basic notions that we are going to use in combination with arithmetical theories.

Definition 4.1 \mathcal{L}_{PA} is the first-order language of Peano arithmetic with just one sort of variables x , referred to as (*number*) *variables*, a unary relation symbol U (without further interpretation and that is needed for proof-theoretic investigations), a symbol $=$ for equality, function symbols for all primitive recursive functions (and we denote by PR^n the collection of those function symbols that have arity $n \in \mathbb{N}$). Moreover, \mathcal{L}_{PA} contains for each function symbol $f \in \text{PR}^n$ with $n \neq 0$ also relation symbols R_f of arity n . Let $\text{PR} := \bigcup_{n \in \mathbb{N}} \text{PR}^n$ and see [18, Definition 1.1.] for more details.

The relation symbols R_f in \mathcal{L}_{PA} are needed for technical reasons to ease the embedding from TID into FIT in Sect. 4.2 (cf., Remark 4.20).

Definition 4.2 The language $\mathcal{L}_{\text{PA}}^2$ denotes the language of second-order arithmetic, i.e., the extension of \mathcal{L}_{PA} by a second sort of variables X , referred to as *set variables* or just *sets*.

Definition 4.3 The usual operations on the natural numbers are denoted as follows: S is the *successor function*, $0_{\mathbb{N}}$ is the zero constant, $+_{\mathbb{N}}$ is the binary function symbol for addition, $<_{\mathbb{N}}$ is the binary less-than relation. Then, $1_{\mathbb{N}}, 2_{\mathbb{N}}, \dots$ abbreviate $S0_{\mathbb{N}}, S(S0_{\mathbb{N}}), \dots$ as usual. If the meaning becomes clear from the context, we may drop the subscript \mathbb{N} . Moreover, $s \leq t$ is used in the obvious way to denote $s < t \vee s = t$.

$\langle t_1, \dots, t_k \rangle$ denotes for any $k \in \mathbb{N}$ and terms t_1, \dots, t_k the result of $\langle \rangle_k(t_1, \dots, t_k)$, while $\langle \rangle_k$ shall be the k -ary function symbol in \mathcal{L}_{PA} corresponding to one of the usual primitive recursive injective functions $\mathbb{N}^k \rightarrow \mathbb{N}$ for mapping finite lists of natural numbers of length k into the natural numbers.

Usual primitive recursive functions for projection $(m, n) \mapsto (m)_n$, list construction $(m, n) \mapsto \text{cons}(m, n)$, list concatenation $(m, n) \mapsto m * n$, and for computing the length $n \mapsto \text{lh}(n)$ of a list shall be used ambiguously to denote the application of its corresponding function symbol in \mathcal{L}_{PA} to terms. In particular, we have: $\langle n_1, \dots, n_k \rangle = 0$ if and only if $k = 0$. If $n \neq 0$ holds, then there is exactly one $k \neq 0$ and natural numbers n_1, \dots, n_k such that $n = \langle n_1, \dots, n_k \rangle$ holds. For each $i < \text{lh}(n)$, we have $(n)_i < n$. For each $0 \leq i \leq k$, we have $(\langle n_0, \dots, n_k \rangle)_i = n_i$. We have also $\text{lh}(\langle n_1, \dots, n_k \rangle) = k$, $\text{cons}(n, \langle n_1, \dots, n_k \rangle) = \langle n, n_1, \dots, n_k \rangle$, and $\langle n_1, \dots, n_k \rangle * \langle m_1, \dots, m_l \rangle = \langle n_1, \dots, n_k, m_1, \dots, m_l \rangle$.

Convention 4.4 \mathcal{L} will denote in the following either \mathcal{L}_{PA}^2 or any extension of \mathcal{L}_{PA} by new relation symbols. We will introduce common notions for such languages \mathcal{L} .

Definition 4.5 \mathcal{L} terms s, t, r are defined as usual inductively from function symbols and number variables. Since in this section \mathcal{L} extends \mathcal{L}_{PA} only by relation symbols or variables of the second sort, all such terms are \mathcal{L}_{PA} terms. A *constant* is a nullary function symbol. If f is an n -ary function symbol of \mathcal{L}_{PA} and $\vec{t} = t_1, \dots, t_n$ is a list of terms, then we set $f(\vec{t}) := f(t_1, \dots, t_n) := f\vec{t} := ft_1 \dots t_n$ and this holds analogously for lists introduced by the $\vec{t}^{(n)}$ notation. For closed terms t , we mean by $t^{\mathbb{N}}$ the *numerical value* of t , i.e., the canonical valuation of t in the standard model \mathbb{N} .

Definition 4.6 \mathcal{L} *formulas* are defined inductively as usual by use of parentheses and the basic logical symbols and we write ambiguously $A \in \mathcal{L}$ to stress that A is an \mathcal{L} formula. For terms s, t , we may sometimes write $s \neq t$ for $\neg(s = t)$. *Atomic \mathcal{L} formulas* are equations $s = t$ and formulas $Rt_1 \dots t_n$, while $R \in \mathcal{L}$ is an n -ary relation symbol and t_1, \dots, t_n are terms.

For the case that \mathcal{L} is \mathcal{L}_{PA}^2 , then also Xt is an atomic formula for any set variable X and term t . \mathcal{L}_{PA}^2 formulas further allow for *quantification over set variables* and we call an \mathcal{L}_{PA}^2 formula *arithmetical* if it does not contain such a quantification (but set variables may still occur).

For n -ary relation symbols or set variables R of \mathcal{L} , a formula A is *positive in R* if it occurs only positively in the usual sense, i.e., no atomic formula of the form $R(t_1, \dots, t_n)$ occurs negated in the formula which is obtained from A by translating first each subformula of the form $B_1 \rightarrow B_2$ to $\neg B_1 \vee B_2$ and where we then move every negation symbol \neg towards atomic formulas, while making use of De Morgan's laws and the law of double negation.⁵

⁵ Compare this definition of positive formula with the definition of For^+ in the setting of FIT.

Definition 4.7

- (a) For any language \mathcal{L} that is \mathcal{L}_{PA}^2 or (possibly) extends \mathcal{L}_{PA} by new relation or function symbols, a standard derivability notion \vdash shall be given that is based on a Hilbert-style deduction system for classical logic with equality axioms (in the first sort). In particular for \mathcal{L}_{PA}^2 , we assume besides modus ponens the usual axioms and rules for quantification over set variables.
- (b) We write $\vdash A$ for any $A \in \mathcal{L}$ to denote the derivability of A in this logic. Moreover, if T is a theory (i.e., a collection of non-logical axioms) with language \mathcal{L}_T , then writing $T \vdash A$ for any $A \in \mathcal{L}_T$ denotes the derivability of A from the axioms of T and this logic. For any set of formulas Γ , we write $\vdash \Gamma$ and $T \vdash \Gamma$ in order to denote that $\vdash A$ and $T \vdash A$ hold, respectively, for each $A \in \Gamma$.

Notation 4.8 For an n -ary relation symbol R with $n \geq 1$ and $\vec{t} = t_1 \dots t_n$, we write $R(\vec{t})$ for $Rt_1 \dots t_n$. and if $n = 1$, we also introduce the following notation: $t \in R := Rt$ and $t \notin R := \neg Rt$. Then $(\forall x \in R)A$ and $(\exists x \in R)A$ stand for $\forall x(R(x) \rightarrow A)$ and $\exists x(R(x) \wedge A)$, respectively. These conventions shall hold analogously also for set variables X . If \triangleleft is a binary relation symbol, we use expressions $(\forall x \triangleleft t)A$ and $(\exists x \triangleleft t)A$ to abbreviate $\forall x(x \triangleleft t \rightarrow A)$ and $\exists x(x \triangleleft t \wedge A)$, respectively.

Definition 4.9 The first-order theory PA is based on the language \mathcal{L}_{PA} and its non-logical axioms are the usual axioms of Peano arithmetic with complete induction, while for each relation symbol R_f that stems from a function symbol f of arity $n \geq 1$, we have for $\vec{x} = x_1, \dots, x_n$ the axiom $\forall \vec{x}(R_f \vec{x} \leftrightarrow f\vec{x} = 0)$. There is no non-logical axiom for the unary relation symbol U (besides in an instance of complete induction).

Definition 4.10 (*Arithmetical*) operator forms are objects of the form

$$\Lambda X.A$$

for \mathcal{L}_{PA}^2 class terms of the form $\mathcal{A} = \Lambda x.A$ such that A is an arithmetical formula with X being the *only* set variable that may occur in it (compare also with Sect. 2) and x is the *only* free number variable that may occur in it.⁶ Note that the unary relation symbol U may occur in A . We use $\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \mathfrak{D}$ as syntactic variables for operator forms. For each \mathcal{L} class terms \mathcal{B} , we set $(\Lambda X.A)(\mathcal{B}) := \mathcal{A}(\mathcal{B}/X)$ while note that the expression $\mathcal{A}(\mathcal{B}/X)$ may yield an \mathcal{L} formula here. We write $\mathfrak{A}(R)$ to denote $\mathfrak{A}(\Lambda x.Rx)$ when R is a unary relation symbol in \mathcal{L} or a set variable. *Positive operator forms* are operator forms $\mathfrak{A} := \Lambda X.\Lambda x.A$ where X occurs only positively in A .

Notation 4.11 We have the following abbreviations for some formulas and operator forms:

- $Cl_{\mathfrak{A}}(\mathcal{A}) := \forall x(\mathfrak{A}(\mathcal{A}, x) \rightarrow \mathcal{A}(x))$ for each operator form \mathfrak{A} and \mathcal{L} class term \mathcal{A} .

and for a binary relation symbol \triangleleft in \mathcal{L}_{PA} and any class term \mathcal{A} , we also have

- $Acc_{\triangleleft} := \Lambda X.\Lambda x.\forall y \triangleleft x(Xy)$,

⁶ Recall that $\mathfrak{A} := \Lambda X.\Lambda x.A$ is intended to define an operator $\Phi_{\mathfrak{A}}: \mathcal{P}(\mathbb{N}) \rightarrow \mathcal{P}(\mathbb{N})$ where $\mathfrak{A}(X, x)$ corresponds to “ $x \in \Phi_{\mathfrak{A}}(X)$ ” for some interpretation $X \subseteq \mathbb{N}$ and $x \in \mathbb{N}$ of x and X .

- $\text{Prog}_{\triangleleft}(\mathcal{A}) := \text{Cl}_{\triangleleft}(\mathcal{A}) := \text{Cl}_{\text{Acc}_{\triangleleft}}(\mathcal{A})$,
- $\text{TI}_{\triangleleft} := \Lambda X. \Lambda x. (\text{Prog}_{\triangleleft}(X) \rightarrow \forall y \triangleleft x(Xy))$, and

Note that we shall usually write $\text{Prog}_{\triangleleft}$ instead of $\text{Cl}_{\triangleleft}$. If \triangleleft is clear from the context, we may just write Acc , Cl , Prog , and TI instead of $\text{Acc}_{\triangleleft}$, $\text{Cl}_{\triangleleft}$, $\text{Prog}_{\triangleleft}$, and $\text{TI}_{\triangleleft}$, respectively.

4.1 The accessible part theory TID

Definition 4.12

- For each operator form \mathfrak{A} , let $P_{\mathfrak{A}}$ denote a new unary relation symbol not in \mathcal{L}_{PA} . Then, P_{\triangleleft} abbreviates $P_{\text{Acc}_{\triangleleft}}$ for any binary relation symbol \triangleleft in \mathcal{L}_{PA} .
- $\mathcal{L}_{\text{TID}} := \mathcal{L}_{\text{PA}} \cup \{P_{\triangleleft} : \triangleleft \text{ is a binary relation symbol in } \mathcal{L}_{\text{PA}}\}$ defines the language of TID.

Definition 4.13 Pos_0 denotes the collection of formulas $A \in \mathcal{L}_{\text{TID}}$ such that P_{\triangleleft} occurs at most positively in A for any binary relation symbol \triangleleft in \mathcal{L}_{PA} . Then define $\text{Pos}_1(a)$ for any number variable a as the collection of \mathcal{L}_{TID} formulas A such that one of the following cases holds:

- $A \in \text{Pos}_0$ or
- $A = \forall \vec{x}(B_1 \rightarrow B_2)$ for $a \notin \text{FV}(B_1)$ and $B_1, B_2 \in \text{Pos}_0$,

while \vec{x} is a (possibly empty) list of variables. This is motivated by FT from the setting of FIT. We write $\Lambda a. A \in \text{Pos}_1$ in order to denote $A \in \text{Pos}_1(a)$.⁷ Let $\text{Neg}_0 := \{A \in \mathcal{L}_{\text{TID}} : \neg A \in \text{Pos}_0\}$.

Remark 4.14 The variable condition $a \notin \text{FV}(B_1)$ in the definition of $\text{Pos}_1(a)$ is not necessary (see [18, Section 8.3]). We keep it for a more direct relationship between TID and FIT.

Definition 4.15 TID arises from the axioms of PA without complete induction by adding the following axioms and axiom schemes for binary relation symbols $\triangleleft \in \mathcal{L}_{\text{PA}}$ and $\mathcal{B} \in \text{Pos}_1$:

$$\begin{aligned} (\text{Ind}) \quad & \mathcal{B}(0) \wedge \forall x(\mathcal{B}(x) \rightarrow \mathcal{B}(Sx)) \rightarrow \forall x \mathcal{B}(x) \\ (\text{Cl}) \quad & \text{Prog}_{\triangleleft}(P_{\triangleleft}) \\ (\text{TID}) \quad & \text{Prog}_{\triangleleft}(\mathcal{B}) \rightarrow \forall x(P_{\triangleleft}x \rightarrow \mathcal{B}(x)) \end{aligned}$$

where (Cl) is called *closure* and (TID) is called *typed inductive definition*.

Remark 4.16 From (Cl) and (TID), we get a fixed-point principle

$$(\text{FP}) \quad \forall x(P_{\triangleleft}x \leftrightarrow \text{Acc}_{\triangleleft}(P_{\triangleleft}, x))$$

for each binary relation symbol \triangleleft in \mathcal{L}_{PA} . We write (Cl) ambiguously for (FP).

⁷ As an example, let $A := \forall y(P_{\triangleleft}y \rightarrow \forall x \triangleleft a(P_{\triangleleft}fxy))$ for some binary function symbol f in \mathcal{L}_{PA} and a a number variable. Then we have $A \in \text{Pos}_1(a)$ and $\Lambda a. A \in \text{Pos}_1$.

Remark 4.17 Instead of (Ind), we have also a course-of-value variant of complete induction $\forall x(\forall x_0 <_{\mathbb{N}} x \mathcal{B}(x_0) \rightarrow \mathcal{B}(x)) \rightarrow \forall x \mathcal{B}(x)$ at hand for all $\mathcal{B} \in \text{Pos}_1$ which we shall tacitly use.

4.2 Embedding TID into FIT

Theorem 4.18 *For each n -ary function symbol $f \in \mathcal{L}_{\text{PA}}$, there is a closed \mathcal{L}_{FIT} term pr_f such that the following holds:*

- (a) FIT proves the reformulation of every defining equation of f from PA with respect to pr_f , while interpreting number variables x as individual variables x with $x \in \mathbb{N}$.
- (b) $\text{FIT} \vdash \text{pr}_f \in \mathbb{N}^n \rightarrow \mathbb{N}$.

Proof pr_f can be obtained by making use of fix from Lemma 3.11(b) and the induction principle (FT-Ind). See [18, Section 6.2] for more details. □

Definition 4.19 Based on pr_f from Theorem 4.18, we define for each \mathcal{L}_{PA} term t the translation t^\bullet to an \mathcal{L}_{FIT} term recursively on the build-up of t : x if t is a variable x , pr_c if t is a constant c , and $\text{pr}_f t_1^\bullet \dots t_n^\bullet$ if t is $f t_1 \dots t_n$ with $f \in \text{PR}^n$ and $n \geq 1$.

The translation on terms is now extended to \mathcal{L}_{TID} formulas A . We define the \mathcal{L}_{FIT} formula A^\bullet recursively on the build-up of an \mathcal{L}_{TID} formula A (and where $\mathbb{Q}_{\triangleleft} := \{(x, y) : (x \triangleleft y)^\bullet\}$):

$$\begin{array}{ll}
 (s = t)^\bullet := (s^\bullet = t^\bullet) & (R_f t_1 \dots t_n)^\bullet := (\text{pr}_f t_1^\bullet \dots t_n^\bullet = 0) \\
 (Ut)^\bullet := t^\bullet \in U & (P_{\triangleleft} t)^\bullet := t^\bullet \in \text{In}_{\mathbb{Q}_{\triangleleft}} \\
 (\neg B)^\bullet := \neg(B^\bullet) & (B \circ C)^\bullet := B^\bullet \circ C^\bullet \quad (\text{for } \circ \in \{\vee, \wedge, \rightarrow\}) \\
 (\forall x B)^\bullet := \forall x(x \in \bar{\mathbb{N}} \vee B^\bullet) & (\exists x B)^\bullet := \exists x(x \in \mathbb{N} \wedge B^\bullet)
 \end{array}$$

The expression $\{(x, y) : (x \triangleleft y)^\bullet\}$ for $\mathbb{Q}_{\triangleleft}$ abbreviates the expression $\{z : z = \langle (z)_0, (z)_1 \rangle \wedge (x \triangleleft y)^\bullet((z)_0/x, (z)_1/y)\}$, i.e., $\{z : z = \langle (z)_0, (z)_1 \rangle \wedge R_f(z)_0(z)_1\}$ where f is such that \triangleleft is R_f .

Remark 4.20

- (a) Note that we defined A^\bullet in case of A being $\forall x B$ in such a way that we can show that $A \in \text{For}^+$ implies $A^\bullet \in \text{Pos}_0$ in Lemma 4.21. If we would have defined it to be $\forall x(x \in \mathbb{N} \rightarrow B^\bullet)$, then apparently \mathbb{N} would occur negatively in A^\bullet .
- (b) It can be readily checked that A^\bullet is indeed a \mathcal{L}_{FIT} formula. Moreover, A and A^\bullet have the same free variables. In particular, note that $(x \triangleleft y)^\bullet((z)_0/x, (z)_1/y)$ contains only z as a free variable.
- (c) We will use the expression $\mathbb{Q}_{\triangleleft}$ without further mentioning in order to denote the type that we introduced in the definition of $(P_{\triangleleft} t)^\bullet$. Recall also that $(x \triangleleft y)^\bullet$ equals $\text{pr}_f x y = 0$ for some binary function symbol $f \in \mathcal{L}_{\text{PA}}$ because the binary relation symbol $\triangleleft \in \mathcal{L}_{\text{PA}}$ is of the form R_f for such an f .

Lemma 4.21

- (a) For each $A \in \text{Pos}_0$ there exists some $A' \in \text{For}^+$ such that $\text{FV}(A^\bullet) = \text{FV}(A')$ and $\text{FIT} \vdash A^\bullet \leftrightarrow A'$ hold.
- (b) For each $A \in \text{Neg}_0$ there exists some $A' \in \text{For}^+$ such that $\text{FV}(A^\bullet) = \text{FV}(A')$ and $\text{FIT} \vdash A^\bullet \leftrightarrow \neg A'$ hold.

Proof By simultaneous induction on the build-up of A . □

Definition 4.22 For every $A \in \mathcal{L}_{\text{TID}}$, we define A_N^\bullet as A^\bullet if $\text{FV}(A) = \emptyset$ and otherwise as $x_1 \in \mathbb{N} \rightarrow \dots \rightarrow x_n \in \mathbb{N} \rightarrow A^\bullet$, given $\text{FV}(A) = \{x_1, \dots, x_n\}$ for some $n \neq 0$.

Lemma 4.23 For each $B \in \text{Pos}_1(a)$, there is an \mathcal{L}_{FIT} -term t and a function type $\mathbb{F} \in \text{FT}$ such that $\text{FIT} \vdash \forall x (tx \in \mathbb{F} \leftrightarrow B^\bullet(x/a))$ holds.

Proof We distinguish the following cases on $B \in \text{Pos}_1(a)$:

1. If $B \in \text{Pos}_0$, then Lemma 4.21 provides some $B' \in \text{For}^+$ such that $\text{FIT} \vdash B^\bullet \leftrightarrow B'$ holds, so for $\mathbb{F} := \{a : B'\}$ we have $\mathbb{F} \in \text{FT}$. Moreover, with $t := \lambda x.x$, we get the claim.
2. If B is of the form $\forall \vec{y} (B_1 \rightarrow B_2)$ with $a \notin \text{FV}(B_1)$, $\vec{y} = y_1, \dots, y_n$, and $B_1, B_2 \in \text{Pos}_0$, we first get $B'_1, B'_2 \in \text{For}^+$ from Lemma 4.21 such that $B_i^\bullet \leftrightarrow B'_i$ and $\text{FV}(B_i^\bullet) = \text{FV}(B'_i)$ holds for $i = 1, 2$. Then we set $\mathbb{Q}_1 := \{z : z = \langle (z)_0, \dots, (z)_{n-1} \rangle \wedge B'_1((z)_0/y_1, \dots, (z)_{n-1}/y_n)\}$, $\mathbb{Q}_2 := \{z : z = \langle (z)_0, \dots, (z)_n \rangle \wedge B'_2((z)_n/a, (z)_0/y_1, \dots, (z)_{n-1}/y_n)\}$, $\mathbb{F} := \mathbb{Q}_1 \rightarrow \mathbb{Q}_2$, and $t := \lambda x, z. \langle x, (z)_0, \dots, (z)_{n-1} \rangle$. Obviously $\mathbb{F} \in \text{FT}$ holds and then similar as in [10, 6.3], we have over FIT and for any x that $tx \in \mathbb{F}$ is equivalent to $\forall z (z \in \mathbb{Q}_1 \rightarrow txz \in \mathbb{Q}_2)$ which is equivalent to $\forall \vec{y} (B_1^\bullet \rightarrow tx \langle y_1, \dots, y_n \rangle \in \mathbb{Q}_2)$ which is equivalent to $\forall \vec{y} (B_1^\bullet \rightarrow \langle x, y_1, \dots, y_n \rangle \in \mathbb{Q}_2)$ which is equivalent to $\forall \vec{y} (B_1^\bullet \rightarrow B_2^\bullet(x/a))$ which is equivalent to $B^\bullet(x/a)$. Hence, we get the claim. Note that $n = 0$ is possible, so $\forall \vec{y} (B_1^\bullet \rightarrow tx \langle y_1, \dots, y_n \rangle \in \mathbb{Q}_2)$ denotes then $B_1^\bullet \rightarrow (tx \langle \rangle \in \mathbb{Q}_2)$. □

Theorem 4.24 If A is an instance of (Ind), (Cl), or (TID), then we have $\text{FIT} \vdash A^\bullet$.

Proof Let A be an instance of (Ind), (Cl), or (TID). We have to show $\text{FIT} \vdash A^\bullet$.

1. For (Cl): If $A = \text{Prog}_{\triangleleft}(P_{\triangleleft})$ holds for some \triangleleft , then we have that A^\bullet is logically equivalent over FIT to $\text{Cl}_{\mathbb{N}, \mathbb{Q}_{\triangleleft}}(\Lambda z.z \in \mathbb{I}_{\mathbb{N}, \mathbb{Q}_{\triangleleft}})$, and this is an instance of (FT-Cl). We have over FIT:

$$\begin{aligned}
 & (\text{Prog}_{\triangleleft}(P_{\triangleleft}))^\bullet \\
 & \leftrightarrow (\forall x (\text{Acc}_{\triangleleft}(P_{\triangleleft}, x) \rightarrow P_{\triangleleft}x))^\bullet \\
 & \leftrightarrow \forall x (x \in \bar{\mathbb{N}} \vee ((\text{Acc}_{\triangleleft}(P_{\triangleleft}, x))^\bullet \rightarrow x \in \mathbb{I}_{\mathbb{N}, \mathbb{Q}_{\triangleleft}})) \\
 & \leftrightarrow \forall x (x \in \bar{\mathbb{N}} \vee (\forall y (y \in \bar{\mathbb{N}} \vee ((y \triangleleft x)^\bullet \rightarrow y \in \mathbb{I}_{\mathbb{N}, \mathbb{Q}_{\triangleleft}})) \rightarrow x \in \mathbb{I}_{\mathbb{N}, \mathbb{Q}_{\triangleleft}})) \\
 & \leftrightarrow \forall x (x \in \bar{\mathbb{N}} \vee (\forall y (y \in \bar{\mathbb{N}} \vee ((y, x) \in \mathbb{Q}_{\triangleleft} \rightarrow y \in \mathbb{I}_{\mathbb{N}, \mathbb{Q}_{\triangleleft}})) \rightarrow x \in \mathbb{I}_{\mathbb{N}, \mathbb{Q}_{\triangleleft}})) \\
 & \leftrightarrow (\forall x \in \mathbb{N}) ((\forall y \in \mathbb{N}) ((y, x) \in \mathbb{Q}_{\triangleleft} \rightarrow y \in \mathbb{I}_{\mathbb{N}, \mathbb{Q}_{\triangleleft}}) \rightarrow x \in \mathbb{I}_{\mathbb{N}, \mathbb{Q}_{\triangleleft}}) \\
 & \leftrightarrow \text{Cl}_{\mathbb{N}, \mathbb{Q}_{\triangleleft}}(\Lambda z.z \in \mathbb{I}_{\mathbb{N}, \mathbb{Q}_{\triangleleft}})
 \end{aligned}$$

2. For (Ind) and (TID): Let $B \in \text{Pos}_1(a)$ be arbitrary. By Lemma 4.23 some \mathcal{L}_{FIT} -term t and function type $\mathbb{F} \in \text{FT}$ exist such that we have

$$\text{FIT} \vdash \forall x (tx \in \mathbb{F} \leftrightarrow B^\bullet(x/a)) \tag{1}$$

2.1. If $A = \mathcal{B}(0) \wedge \forall x (\mathcal{B}(x) \rightarrow \mathcal{B}(Sx)) \rightarrow \forall x \mathcal{B}(x)$ holds for $\mathcal{B} = \Lambda a.B$: We note that for $B_1 := B(a/Sa)$ one can prove (by induction on the build-up of B) that B_1^\bullet is $B^\bullet(a/s_Na)$. So, with $\mathcal{B}(Sx)^\bullet$ being $(B(a/Sx))^\bullet$ this becomes $(B_1(x))^\bullet$, i.e., we get $B_1^\bullet(a/x)$ and hence $(B^\bullet(a/s_Na))(a/x)$. So, we obtain that $\mathcal{B}(Sx)^\bullet$ is $B^\bullet(a/s_Nx)$, while note that for any $B' \in \mathcal{L}_{\text{TID}}$, we have that B' and B'^\bullet share the same first-order variables. For proving A^\bullet , we can therefore assume that

$$B^\bullet(0/a) \tag{2}$$

$$\forall x (x \in \bar{N} \vee (B^\bullet(x/a) \rightarrow B^\bullet(s_Nx/a))) \tag{3}$$

holds, and we have to show $\forall x (x \in \bar{N} \vee B^\bullet(x/a))$, while this is equivalent to $t \in N \rightarrow \mathbb{F}$ due to (1). Now we can directly apply (FT-Ind) because (2) is equivalent to $t0 \in \mathbb{F}$ and (3) is equivalent to $(\forall x \in N)(tx \in \mathbb{F} \rightarrow t(s_Nx) \in \mathbb{F})$.

2.2. If $A = \text{Prog}_{\triangleleft}(\mathcal{B}) \rightarrow \forall x (P_{\triangleleft}x \rightarrow \mathcal{B}(x))$ holds for $\mathcal{B} = \Lambda a.B$: With $(\text{Prog}_{\triangleleft}(\mathcal{B}))^\bullet$ being $(\forall x \in N)(\forall y \in N)((y, x) \in \mathbb{Q}_{\triangleleft} \rightarrow (B(y))^\bullet \rightarrow (B(x))^\bullet)$, we get that FIT proves the following:

$$\left. \begin{aligned} (\text{Prog}_{\triangleleft}(\mathcal{B}))^\bullet &\leftrightarrow (\forall x \in N)(\forall y \in N)((y, x) \in \mathbb{Q}_{\triangleleft} \rightarrow ty \in \mathbb{F} \rightarrow tx \in \mathbb{F}) \\ &\leftrightarrow \text{Cl}_{N, \mathbb{Q}_{\triangleleft}}(\Lambda z.tz \in \mathbb{F}) \end{aligned} \right\} \tag{4}$$

This accumulates in the provability of A^\bullet . Namely, assume $(\text{Prog}_{\triangleleft}(\mathcal{B}))^\bullet$ and get $t \in (I_{N, \mathbb{Q}_{\triangleleft}} \rightarrow \mathbb{F})$ from (4) and (FT-ID), hence (1) yields that $\forall x (x \in I_{N, \mathbb{Q}_{\triangleleft}} \rightarrow tx \in \mathbb{F})$ is equivalent to $\forall x (x \in I_{N, \mathbb{Q}_{\triangleleft}} \rightarrow B^\bullet(x/a))$. The latter implies $(\forall x \in N)(x \in I_{N, \mathbb{Q}_{\triangleleft}} \rightarrow B^\bullet(x/a))$, so we are done because $(\forall x (P_{\triangleleft}x \rightarrow \mathcal{B}(x)))^\bullet$ is $(\forall x \in N)(x \in I_{N, \mathbb{Q}_{\triangleleft}} \rightarrow B^\bullet(x/a))$. □

Corollary 4.25 $\text{TID} \vdash A$ implies $\text{FIT} \vdash A_N^\bullet$ for each $A \in \mathcal{L}_{\text{TID}}$.

Proof The claim follows essentially from Theorems 4.18 and 4.24. In particular, we remark that for FIT the propositional logical rules and axioms and the quantificational logic for individual variables correspond (under the translation of Definition 4.19) to first-order predicate logic in the setting of TID. □

5 Ordinals

The lower bounds for the proof-theoretic ordinal of both FIT and TID are determined in Sect. 6 via wellordering proofs within TID and using an ordinal notation system $(\text{OT}, <)$. For carrying out these wellordering proofs in all details, we would need to provide a full definition of $(\text{OT}, <)$ and verify that all needed properties are provable

within TID. Due to constraints on the length of this article, though, we refer the interested reader to [18, Chapter 3] for further details and provide in the appendix hints on the definition and properties of $(OT, <)$.

The ordinal notation system $(OT, <)$ is based on a framework for representing ordinals that we shall introduce in this section. We shift to the literature most of the preparatory work that is needed to formulate the ordinals that are involved here and explain only as much as to make this article sufficiently self-contained and provide a structural impression on which properties we demand from $(OT, <)$. In particular, we use [21] and the treatment of *Klammersymbols* (that extends the concept of finitary Veblen functions to the transfinite) as the main source for the finitary Veblen functions because it is the most elaborated source for representation and recursion properties.

5.1 Ordinal notations

Working in the broad set-theoretic framework of Zermelo–Fraenkel set theory ZFC with the axiom of choice, let On denote the class of ordinals and use $\alpha, \beta, \gamma, \delta, \dots$ (i.e., small Greek letters in general) to denote elements of On . By Lim , we denote the class of *limit ordinals*, while ω denotes the first limit ordinal. Moreover, we write 0 for $\emptyset, \alpha <_{On} \beta$ (or just $\alpha < \beta$) for $\alpha \in \beta$, and $\alpha \leq_{On} \beta$ (or just $\alpha \leq \beta$) for $\alpha \subseteq \beta$. For $\alpha > 0$, we let Ω_α denote \aleph_α , i.e., $\{\Omega_\alpha : \alpha \in On\}$ is the class of all uncountable initial ordinals, and we write Ω for Ω_1 and Ω_0 for 0 . Over ZFC, we have that $\Omega_{\alpha+1}$ is regular. A *normal function* is a (with respect to $<$) strictly increasing continuous function $f : On \rightarrow On$. Demanding a knowledge about this broad set-theoretic framework, we use common notions and well-known properties of those tacitly, e.g.,

- the notion of *club* classes C with $C \subseteq On$ and its correspondence to normal functions,⁸
- the existence of the *derivative* $fix(f) := \{\alpha \in On : f(\alpha) = \alpha\}$ of a normal function f , being a club class itself,
- *basic ordinal arithmetic* for $\alpha, \beta \in On$ with (ordinal) addition $\alpha +_{On} \beta$ (or just $\alpha + \beta$), (ordinal) multiplication $\alpha \cdot_{On} \beta$ (or just $\alpha \cdot \beta$ or $\alpha\beta$), and (ordinal) exponentiation $exp_{On}(\alpha, \beta)$ (or just α^β),
- the usual representation of natural numbers as *von Neumann ordinals* $(0)_{On} := \emptyset$ and $(n + 1)_{On} := \{(n)_{On}\} \cup (n)_{On}$ for each $n \in \mathbb{N}$ within On , while we identify $(n)_{On}$ with n .

See [7] for more details on the relationship between different approaches to ordinal notations. It shall be clear from the context whether $<$ means $<_{\mathbb{N}}$ or $<_{On}$ (and similar for the other mentioned expressions).

Definition 5.1 Let $\mathbf{P} := \{\omega^\eta : \eta \in On\}$ denote the class of *additive principal numbers*.

Remark 5.2 For $\alpha \in \mathbf{P}$ and $\beta, \gamma < \alpha$, we have $\beta + \gamma < \alpha$ and $\beta + \alpha = \alpha$.

⁸ Each club class C induces a normal function $enum_C$ that enumerates the elements of C in increasing order.

5.1.1 The finitary Veblen functions

Definition 5.3 The $n + 1$ -ary Veblen function $\varphi_{n+1} : \text{On}^{n+1} \rightarrow \text{On}$ is obtained for each $n \in \mathbb{N}$ from the ω -exponential function and the binary Veblen function φ_2 by generalizing its definition principle, i.e., we let $\varphi_1(\gamma) := \omega^\gamma$ for each $\gamma \in \text{On}$ and define φ_{n+2} for $n \geq 0$ as follows:

- $\varphi_{n+2}(0, \bar{\alpha}^{(n)}, \gamma) := \varphi_{n+1}(\bar{\alpha}^{(n)}, \gamma)$.
- If $\alpha_1, \alpha_k > 0$ holds for some $1 \leq k \leq n + 1$ with $\alpha_{k+1} = \dots = \alpha_{n+1} = 0$, then $\varphi_{n+2}(\bar{\alpha}^{(n+1)}, \gamma)$ denotes the γ -th common fixed-point of the functions

$$\xi \mapsto \varphi_{n+2}(\bar{\alpha}^{(k-1)}, \beta, \xi, \bar{0}^{(n-k+1)})$$

that are defined on On and for each $\beta < \alpha_k$.

Definition 5.4 The *small Veblen ordinal* is the least ordinal $\alpha > 0$ not expressible from ordinals smaller than α and by means of ordinal addition and the finitary Veblen functions.

Notation 5.5 We often write φ instead of φ_{n+1} if the arity $n + 1$ is clear from the context.

Remark 5.6 We have $\Gamma_0 = \varphi(1, 0, 0)$ (the Feferman–Schütte ordinal) and $\varepsilon_0 = \varphi(0, 1, 0) = \varphi(1, 0)$.

Lemma 5.7 Let $k, l \in \mathbb{N}$ and $\alpha_1, \dots, \alpha_k \in \text{On}$ be given with $\alpha_1 \neq 0$ and $\alpha_k \neq 0$. Then $\beta < \alpha_k$ and $\xi = \varphi(\bar{\alpha}^{(k)}, \bar{0}^{(l+1)})$ imply $\varphi(\bar{\alpha}^{(k-1)}, \beta, \bar{0}^{(i)}, \xi, \bar{0}^{(j)}) = \xi$ for every β, ξ and $i, j \in \mathbb{N}$ with $i + j = l$.

Proof This follows easily from Definition 5.3. □

5.1.2 Klammersymbols

The following introduction to Klammersymbols and related results are included here because Klammersymbols provide the most elaborated source to derive the major properties of the finitary Veblen functions.

Definition 5.8 Following [21], we introduce Schütte’s Klammersymbols⁹ which are a generalization of the finitary Veblen functions to the transfinite by allowing arguments to be indexed by ordinals. A *Klammersymbol* K is an expression of the form

$$\left(\begin{array}{c} \alpha_1 \dots \alpha_{n+1} \\ \beta_1 \dots \beta_{n+1} \end{array} \right)$$

for $\alpha_1, \dots, \alpha_{n+1}, \beta_1, \dots, \beta_{n+1} \in \text{On}$ and with the condition

$$0 \leq \beta_1 < \beta_2 < \dots < \beta_{n+1} \tag{5}$$

⁹ Klammer is the German word for bracket, so Klammersymbols can be read as “bracket symbols”.

We use K as a syntactic variable to denote a Klammersymbol. Two Klammersymbols K_1 and K_2 are defined to be *equal*, denoted $K_1 = K_2$, in case K_1 and K_2 can be transformed into the same Klammersymbol by adding or dropping of columns of the form $\begin{smallmatrix} \gamma \\ \beta \end{smallmatrix}$. We write $K_1 \neq K_2$ in case that K_1 and K_2 are not equal.

Given a normal function $f : \text{On} \rightarrow \text{On}$ such that $f(0) > 0$, the *value* fK of a Klammersymbol K (under f) is defined as follows:

1. If K is $\begin{pmatrix} \gamma \\ 0 \end{pmatrix}$, then fK is $f(\gamma)$.
2. If K is $\begin{pmatrix} \gamma & \alpha_1 & \dots & \alpha_{n+1} \\ 0 & \beta_1 & \dots & \beta_{n+1} \end{pmatrix}$ and $\alpha_i = 0$ for some $i \in \{1, \dots, n + 1\}$, then fK is fK' where K' is obtained from K by deleting the column $\begin{smallmatrix} 0 \\ \beta_i \end{smallmatrix}$.
3. If K is $\begin{pmatrix} \gamma & \alpha_1 & \dots & \alpha_{n+1} \\ 0 & \beta_1 & \dots & \beta_{n+1} \end{pmatrix}$ and $\alpha_i \neq 0$ for all $i \in \{1, \dots, n + 1\}$, then fK is the γ -th common solution η for the following equations and for all $\alpha' < \alpha_1$ and $\beta' < \beta_1$:

$$f \begin{pmatrix} \eta & \alpha' & \alpha_2 & \dots & \alpha_{n+1} \\ \beta' & \beta_1 & \beta_2 & \dots & \beta_{n+1} \end{pmatrix} = \eta$$

In case of $n = 0$, we read $f \begin{pmatrix} \eta & \alpha' & \alpha_2 & \dots & \alpha_{n+1} \\ \beta' & \beta_1 & \beta_2 & \dots & \beta_{n+1} \end{pmatrix}$ as $f \begin{pmatrix} \eta & \alpha' \\ \beta' & \beta_1 \end{pmatrix}$.

Definition 5.9 The *large Veblen ordinal* is the least ordinal $\alpha > 0$ not expressible from ordinals smaller than α and by means of ordinal addition and the use of the value of Klammersymbols under the ω -exponential function φ_1 .¹⁰

Remark 5.10 For all Klammersymbols K_1 and K_2 , there are ordinals $\gamma_1, \dots, \gamma_{n+1}$ with $\gamma_1 < \dots < \gamma_{n+1}$ and ordinals $\alpha_1, \dots, \alpha_{n+1}, \beta_1, \dots, \beta_{n+1}$ such that $K_1 = \begin{pmatrix} \alpha_1 & \dots & \alpha_{n+1} \\ \gamma_1 & \dots & \gamma_{n+1} \end{pmatrix}$ and $K_2 = \begin{pmatrix} \beta_1 & \dots & \beta_{n+1} \\ \gamma_1 & \dots & \gamma_{n+1} \end{pmatrix}$, simply by adding or removing of columns of the form $\begin{smallmatrix} 0 \\ \gamma_i \end{smallmatrix}$ where necessary.

Definition 5.11 A *lexicographic order* $<$ on Klammersymbols is defined for Klammersymbols K_1 and K_2 with $K_1 \neq K_2$ as follows:

1. If $K_1 = \begin{pmatrix} \alpha_1 & \dots & \alpha_{n+1} \\ \gamma_1 & \dots & \gamma_{n+1} \end{pmatrix}$ and $K_2 = \begin{pmatrix} \beta_1 & \dots & \beta_{n+1} \\ \gamma_1 & \dots & \gamma_{n+1} \end{pmatrix}$ and $i \in \{1, \dots, n + 1\}$ is the *largest* index with $\alpha_i \neq \beta_i$, then we have $K_1 < K_2$ in case of $\alpha_i < \beta_i$ and $K_2 < K_1$ otherwise.
2. If $K_1 = K'_1, K_2 = K'_2$, and $K_1 < K_2$, then also $K'_1 < K'_2$.

Proposition 5.12 Let $f : \text{On} \rightarrow \text{On}, \xi \mapsto \omega^\xi$. Then $\xi \mapsto f \left(\frac{1}{\xi} \right)$ is a normal function. In particular, we have $f \left(\frac{1}{\lambda} \right) = \sup_{\xi < \lambda} f \left(\frac{1}{\xi} \right)$ for each $\lambda \in \text{Lim}$.

Proof See [21, (4.1)–(4.3)]. □

Proposition 5.13 Let $f : \text{On} \rightarrow \text{On}, \xi \mapsto \omega^\xi$ and $K := \begin{pmatrix} \alpha_1 & \dots & \alpha_{n+1} \\ \gamma_1 & \dots & \gamma_{n+1} \end{pmatrix}$ be a Klammersymbol. For each Klammersymbol K' with $K < K'$, the following holds:

- (a) $fK = fK'$ holds if and only if $k \in \{1, \dots, n + 1\}$ exists such that $\alpha_k = fK'$ and the following holds:
- $\alpha_i = 0$ for each i with $1 \leq i < k$, and

¹⁰ In Sect. 8, the conclusion of this article, we shall mention a theory TID_1^1 which has the large Veblen ordinal as its proof-theoretic ordinal.

- $\alpha_i < f K'$ for each i with $k < i \leq n + 1$.
- (b) $f K < f K'$ holds if $\alpha_i < f K'$ holds for all $i \in \{1, \dots, n + 1\}$.
- (c) $f K' < f K$ holds if
 - either $f K' < \alpha_k$ holds for some $k \in \{1, \dots, n + 1\}$, or
 - $n \geq 1$ and $j, k \in \{1, \dots, n + 1\}$ exist such that $j < k$, $\alpha_j \neq 0$, and $\alpha_k = f K'$.

Proof See (7.1)–(7.4) in [21]. Note that the negation of the condition given in (a) yields the conditions stated in (b) and (c). For this, note in particular that $\alpha_k = f K'$ implies $\alpha_k \neq 0$ and hence if $\alpha_i = 0$ holds for each i with $1 \leq i < k$ and the condition of (a) does not hold, then $k < n + 1$ holds and i exists with $\alpha_i \geq f K'$ and $k < i \leq n + 1$, leading to one of the conditions in (c). □

Lemma 5.14 *Let $f: \text{On} \rightarrow \text{On}$, $\xi \mapsto \omega^\xi$. Then we have*

$$\varphi(\alpha_1, \dots, \alpha_{n+1}) = f \left(\begin{matrix} \alpha_{n+1} & \alpha_n & \dots & \alpha_1 \\ 0 & 1 & \dots & n \end{matrix} \right)$$

where $0, 1, \dots, n$ in the Klammersymbol's second row denote finite ordinals.

Proof If $n = 0$ or $\alpha_1 = \dots = \alpha_{n+1} = 0$ holds, then the claim is clear. Otherwise, assume $n \neq 0$ and without loss of generality that $\alpha_1 \neq 0$ holds. Further, let $k \in \{1, \dots, n\}$ with $\alpha_k \neq 0$ and $\alpha_{k+1} = \dots = \alpha_n = 0$. The claim now follows by transfinite induction on α_k since $\varphi(\bar{\alpha}^{(n+1)})$ is the α_{n+1} -th common fixed point of the functions

$$\xi \mapsto \varphi \left(\bar{\alpha}^{(k-1)}, \beta, \xi, \bar{0}^{(n-k)} \right)$$

given for each $\beta < \alpha_k$. Now, we get

$$\begin{aligned} \varphi \left(\bar{\alpha}^{(k-1)}, \beta, \xi, \bar{0}^{(n-k)} \right) &= f \left(\begin{matrix} 0 & \dots & 0 & \xi & \beta & \alpha_{k-1} & \dots & \alpha_1 \\ 0 & \dots & n-(k+1) & n-k & n-(k-1) & n-(k-2) & \dots & n \end{matrix} \right) \\ &= f \left(\begin{matrix} \xi & \beta & \alpha_{k-1} & \dots & \alpha_1 \\ n-k & n-(k-1) & n-(k-2) & \dots & n \end{matrix} \right) \end{aligned}$$

from the induction hypothesis and for each $\xi \in \text{On}$. Hence the claim follows from Definition 5.8 and Lemma 5.7. □

Corollary 5.15 *Let $n \geq 1$ and ordinals $\alpha_1, \dots, \alpha_n$ be given, then we have the following:*

- (a) $\alpha_i \leq \varphi(\bar{\alpha}^{(n)})$ for all $i \in \{1, \dots, n\}$.
- (b) If $\alpha_k \neq 0$ for some $k \in \{1, \dots, n\}$, then $\alpha_i < \varphi(\bar{\alpha}^{(n)})$ holds for all $i \in \{1, \dots, k - 1\}$.

Proof This follows from (3.3) and (6.1) in [21] by making use of Lemma 5.14. □

Corollary 5.16 *Let $n \geq 1$ and ordinals $\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_n$ be given. Then $\varphi(\bar{\alpha}^{(n)}) < \varphi(\bar{\beta}^{(n)})$ holds if and only if some $r \in \{1, \dots, n\}$ exists such that $\alpha_r \neq \beta_r$ holds with $\alpha_i = \beta_i$ for all $i \in \{1, \dots, r - 1\}$, and such that one of the following holds:*

1. $\alpha_r < \beta_r$ and $\alpha_i < \varphi(\bar{\beta}^{(n)})$ for all $i \in \{r + 1, \dots, n\}$, or
2. $\beta_r < \alpha_r$ and
 - either $\varphi(\bar{\alpha}^{(n)}) < \beta_k$ holds for some $k \in \{1, \dots, n\}$, or
 - $\varphi(\bar{\alpha}^{(n)}) = \beta_k$ and $\beta_i \neq 0$ for some $i, k \in \{1, \dots, n\}$ with $k < i$.

Proof This follows immediately from Proposition 5.13 and Lemma 5.14. For the first case $\alpha_r < \beta_r$, note that $\alpha_i < \varphi(\bar{\beta}^{(n)})$ holds anyway for $i \in \{1, \dots, r\}$ by Corollary 5.15: On the one hand, we have $\beta_r \leq \varphi(\bar{\beta}^{(n)})$ and so $\alpha_r < \varphi(\bar{\beta}^{(n)})$, and on the other hand, $\alpha_r < \beta_r$ also implies $\beta_r \neq 0$ which by Corollary 5.15(b) gives $\alpha_i = \beta_i < \varphi(\bar{\beta}^{(n)})$ for $i \in \{1, \dots, r - 1\}$. □

Definition 5.17 Let $n \geq 1$.

- (a) $\text{NF}_n^\varphi(\bar{\alpha}^{(n)})$ holds if and only if we have $\alpha_i < \varphi(\bar{\alpha}^{(n)})$ for each $1 \leq i \leq n$. We read $\text{NF}_n^\varphi(\bar{\alpha}^{(n)})$ as “ $\alpha_1, \dots, \alpha_n$ is in normal form with respect to φ_n ”.
- (b) $\beta =_{\text{NF}} \varphi(\bar{\alpha}^{(n)})$ denotes $\beta = \varphi(\bar{\alpha}^{(n)})$ and $\text{NF}_n^\varphi(\bar{\alpha}^{(n)})$.

Remark 5.18 The following lemma describes an important recursion property that is needed for the definition of the ordinal notation system (OT, $<$) in [18, Chapter 3]. It explains how to obtain an expression $\varphi_n(\bar{\beta}^{(n)})$ with $\text{NF}_n^\varphi(\bar{\beta}^{(n)})$, i.e., such that $\beta_i \neq \varphi_n(\bar{\beta}^{(n)})$ holds for each $1 \leq i \leq n$.

Lemma 5.19 Let $n \geq 1, k \in \{1, \dots, n\}$, and ordinals $\beta_1, \dots, \beta_n, \alpha_1, \dots, \alpha_n$ be given with $\beta_k =_{\text{NF}} \varphi(\bar{\alpha}^{(n)})$ and $\beta_{k+1} = \dots = \beta_n = 0$. Then $\text{NF}_n^\varphi(\bar{\beta}^{(n)})$ holds if and only if $\alpha_r \neq \beta_r$ holds for some $r \in \{1, \dots, k\}$ with $\alpha_i = \beta_i$ for all $i \in \{1, \dots, r - 1\}$ and one of the following holds:

1. $\alpha_r < \beta_r$, or
2. $\beta_r < \alpha_r$ and for some $i \in \{1, \dots, k - 1\}$, we have $\beta_k \leq \beta_i$.
We then get $i \in \{r + 1, \dots, k - 1\}$.

Proof Note that we have $\beta_k \neq 0$ due to $\varphi(\bar{\alpha}^{(n)}) \neq 0$ and our assumption $\beta_k =_{\text{NF}} \varphi(\bar{\alpha}^{(n)})$, and hence by Corollary 5.15 and $\text{NF}_n^\varphi(\bar{\alpha}^{(n)})$, we have

$$\alpha_i < \beta_k \leq \varphi(\bar{\beta}^{(n)}) \tag{6}$$

for all $i \in \{1, \dots, n\}$ and also $\beta_i < \varphi(\bar{\beta}^{(n)})$ for all $i \in \{1, \dots, k - 1\}$. Furthermore, there has to be some $r \in \{1, \dots, k\}$ such that $\alpha_r \neq \beta_r$ and $\alpha_i = \beta_i$ holds for all $i \in \{1, \dots, r - 1\}$ since we have $\alpha_k < \beta_k$. Recalling the assumption $\beta_{k+1} = \dots = \beta_n = 0$, we thus have that $\text{NF}_n^\varphi(\bar{\beta}^{(n)})$ holds if and only if $\beta_k < \varphi(\bar{\beta}^{(n)})$ holds, i.e., $\varphi(\bar{\alpha}^{(n)}) < \varphi(\bar{\beta}^{(n)})$. Due to Corollary 5.16, it suffices to treat the following three situations:

1. Case $\alpha_r < \beta_r$: We have $\alpha_i < \varphi(\bar{\beta}^{(n)})$ for each $i \in \{r + 1, \dots, n\}$ due to (6) and can use Corollary 5.16 directly to finish the proof for this case.
2. Case $\beta_r < \alpha_r$ and $\varphi(\bar{\alpha}^{(n)}) < \beta_i$ for some $i \in \{1, \dots, n\}$: Since $\beta_l = 0$ holds for all $l > k$ and $\beta_k = \varphi(\bar{\alpha}^{(n)})$, this case is equivalent to $\beta_r < \alpha_r$ and $\beta_k < \beta_i$ for some $i \in \{1, \dots, k - 1\}$. Furthermore, we have $i \in \{r + 1, \dots, k - 1\}$ due to $\beta_i = \alpha_i < \beta_k$ for $i < r$ and $\beta_r < \alpha_r < \beta_k$.

3. Case $\beta_r < \alpha_r$ and $\varphi(\bar{\alpha}^{(n)}) = \beta_i$ holds with $\beta_j \neq 0$ for some $i, j \in \{1, \dots, n\}$ with $i < j$: Since $\beta_l = 0$ holds for all $l > k$, we have that $i < k$ holds. With this and since $\varphi(\bar{\alpha}^{(n)}) = \beta_k \neq 0$ holds, we can always choose $j = k$ and get that the current case is equivalent to $\beta_r < \alpha_r$ and $\beta_k = \beta_i$ for some $i \in \{1, \dots, k - 1\}$. Furthermore, we have $i \in \{r + 1, \dots, k - 1\}$ due to $\beta_i = \alpha_i < \beta_k$ for $i < r$ and $\beta_r < \alpha_r < \beta_k$. □

Remark 5.20 In order to simplify the formulation and proof of Lemma 5.19, we took the lists of ordinals $\alpha_1, \dots, \alpha_n$ and β_1, \dots, β_n to have the same length $n \geq 2$. Clearly, the lemma holds analogously for lists of ordinals with different length ≥ 2 (just add ordinals of the form 0 to the front of the shorter list to make them the same length).

5.1.3 The ϑ -function

See [7,20] for details on the definition and properties of the function $\vartheta : \text{On} \rightarrow \text{On}$. We state only the notation $\Omega(n, x)$ that is defined for each $x \in \text{On}$ and each $n \in \mathbb{N}$ by

$$\Omega(0, x) := \Omega \cdot x \quad \Omega(1, x) := \Omega^x \quad \Omega(n + 1, x) := \Omega^{\Omega(n, x)}$$

and cite the following properties in order to relate the ϑ -function to the setting of Klammersymbols when we present results from the literature in Sect. 7. By using the Buchholz ψ -functions from [5] or the Feferman–Aczel θ -functions from [4], we also have the following correspondence $\vartheta \Omega^\omega = \psi_0 \Omega^{\Omega^\omega} = \theta \Omega^\omega 0$. See also the last paragraph in [19].

Proposition 5.21 *Let $f : \text{On} \rightarrow \text{On}, \xi \mapsto \omega^\xi$.*

- (a) $f \left(\begin{smallmatrix} 1 \\ \omega \end{smallmatrix} \right)$ denotes the small Veblen ordinal.
- (b) $f \left(\begin{smallmatrix} 1 \\ \omega \end{smallmatrix} \right) = \vartheta \Omega^\omega$ and $\varphi(\omega, 0) = f \left(\begin{smallmatrix} \omega \\ 1 \end{smallmatrix} \right) = \vartheta(\Omega \cdot \omega)$.

Proof (a) follows essentially from Definition 5.4, Lemma 5.14, and Proposition 5.13. (b) is due to [22], see also [7] or [18, Chapter 2]. □

5.2 Fundamental sequences and ordinal notations

As described in the appendix of this article (see Appendix A) and carried out in [18, Chapter 3] with proofs and descriptions in detail, it is possible to define a primitive recursive ordinal notation system $(\text{OT}, <)$ for the small Veblen ordinal that is based on the finitary Veblen functions (e.g., by using codes of the form $\phi \bar{a}^{(n+1)}$) and whose basic properties (such as being a linear ordering) can be formalized and verified within TID. Moreover, it is possible to define a sensible semantics for ordinal notations $a \in \text{OT}$ by means of a function $\text{o} : \text{OT} \rightarrow \text{On}$ such that in particular $\text{o}(\phi \bar{a}^{(n+1)}) = \varphi(\text{o}(a_1), \dots, \text{o}(a_{n+1}))$ holds for $\phi \bar{a}^{(n+1)} \in \text{OT}$. See [18, Definition 3.20] for all details.

Given this interpretation, we can define primitive recursive functions on the natural numbers \mathbb{N} such as $\tilde{\varphi}, \tilde{\omega}, \tilde{\cdot}, \tilde{+}$ that act as proof-theoretic counterparts to the set-theoretic functions $\varphi, \omega, \cdot, +$ on the ordinals On . They simulate the ordinal arithmetic on the

codes based on OT. For instance for φ or to be more precise for φ_{n+1} and $n \in \mathbb{N}$, we have that

- $\varphi_{n+1} : \text{On}^{n+1} \rightarrow \text{On}$ is the $n + 1$ -ary ordinal-theoretic Veblen function.
- $\phi : \mathbb{N} \rightarrow \mathbb{N}$ is a unary primitive recursive function whose purpose is to produce Gödel numbers $\phi(x)$ for (codes of) lists of notations $x = \langle a_1, \dots, a_{n+1} \rangle$ with $a_1, \dots, a_{n+1} \in \text{OT}$. When formally defining OT, only certain codes $\phi(x)$ are allowed to be in OT. We write $\phi a_1 \dots a_{n+1}$ to denote $\phi(\langle a_1, \dots, a_{n+1} \rangle)$. See the appendix for more details and see [18] for a full definition and more explanations.
- $\tilde{\varphi}_{n+1} : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$ is a primitive recursive function that simulates certain behaviors of the order-theoretic function φ and where we have $o(\tilde{\varphi}(\bar{a}^{(n+1)})) = \varphi(o(a_1), \dots, o(a_{n+1}))$ for each $a_1, \dots, a_{n+1} \in \text{OT}$.

Given such a setup, fundamental sequences $a[x]$ for ordinal notations $a \in \text{OT}$ and natural numbers $x \in \mathbb{N}$ can be defined with the following properties (see also Definition A.1 in the appendix). The notions Suc and Lim in Proposition 5.22 below are primitive recursively defined from OT and consist of successor codes and limit codes, respectively.

More precisely, the fundamental sequences’ definition as given in Definition A.1 in the appendix can be motivated from the set-theoretic perspective of Sect. 5.1. As a consequence, the following interplay of the finitary Veblen functions with the given fundamental sequences can be intuitively understood without relying on the intrinsic properties of the ordinal notation system (OT, $<$).

Proposition 5.22 [18, Section 3.4]

- (a) $\text{TID} \vdash \forall d, x (d \in \text{Suc} \rightarrow d[x] < d)$.
- (b) $\text{TID} \vdash \forall d, x (d \in \text{Lim} \rightarrow (0 < d[x] \wedge d[x] < d[x +_{\mathbb{N}} 1] \wedge d[x] < d))$.
- (c) $\text{TID} \vdash \forall d, d_0 (d \in \text{Lim} \wedge d_0 < d \rightarrow \exists x (d_0 < d[x]))$.

Proposition 5.23 [18, Section 3.4] *Assume that $d_0 < \tilde{\varphi}(\bar{a}^{(m+1)}, \bar{0}^{(k)})$, b holds for given $\bar{a}^{(m)}, b, d_0 \in \text{OT}$ and $k, m \in \mathbb{N}$. Then TID proves the following:*

- (a) $b \in \text{Lim} \rightarrow \exists x (d_0 < \tilde{\varphi}(\bar{a}^{(m+1)}, \bar{0}^{(k)}, b[x]))$.
- (b) $(b \notin \text{Lim} \wedge a_1 = 0 \wedge \dots \wedge a_{m+1} = 0) \rightarrow \exists x (d_0 < \tilde{\omega}^{b[x]} \tilde{\cdot} (x +_{\mathbb{N}} 1))$.
- (c) $(b = 0 \wedge a_{m+1} \in \text{Lim}) \rightarrow \exists x (d_0 < \tilde{\varphi}(\bar{a}^{(m)}, a_{m+1}[x], \bar{0}^{(k+1)}))$.

Lemma 5.24 [18, Section 3.2] *For $a, b, a_1, \dots, a_{n+1}, c \in \text{OT}$ and $n, m \in \mathbb{N}$, we have within TID:*

- (a) *If $a \leq c$ and $c < a \tilde{+} b$ hold, then $c = a \tilde{+} d$ holds for some $d \in \text{OT}$ with $d < b$.*
- (b) *$\tilde{\varphi}(\bar{x}^{(n+1)}) \in \text{OT}$ and $x_1, \dots, x_{n+1} \in \text{OT}$ are equivalent for each $x_1, \dots, x_{n+1} \in \mathbb{N}$.*
- (c) *$x \tilde{+} y \in \text{OT}$ and $x, y \in \text{OT}$ are equivalent for each $x, y \in \mathbb{N}$.*
- (d) *If $a_1 \neq 0$ and $a_{m+1} \notin \{0\} \cup \{\phi \bar{a}^{(n+1)} : a_1 \neq 0 \ \& \ a_1, \dots, a_{n+1} \in \mathbb{N}\}$ hold, then $\tilde{\varphi}(\bar{a}^{(m+1)}, \bar{0}^{(k)}) = \phi \bar{a}^{(m+1)} \bar{0}^{(k)}$ holds.*

However, having a more accessible approach to the wellordering proofs by using fundamental sequences instead of working directly with the underlying ordinal notation system has the cost that one has to verify the fundamental sequences’ adequate behaviour in the background (compare [18, Sections 3.4 and 11.1.2]).

6 Lower Bound $\vartheta\Omega^\omega$ for FIT and TID

In this section, we obtain a lower bound for the proof-theoretic ordinal of the theory TID by means of wellordering proofs. Hence, together with the embedding of TID into FIT from Sect. 4.2, we automatically get a lower bound for FIT as well. We rely for the following proofs on the availability of fundamental sequences for the underlying ordinal notation system $(\text{OT}, <)$ as described in Sect. 5.2. This allows us to shift the dependence on the specific implementation of $(\text{OT}, <)$ to the properties of the fundamental sequences.

Alternatively, we could implement the following proofs directly within the setting of the ordinal notation system $(\text{OT}, <)$, avoiding the introduction of fundamental sequences but for the cost of then having the ordinal notation system $(\text{OT}, <)$ appear more prominently in the wellordering proofs. In this case, it would be technically more sensible to work with fixed-point free variants $\bar{\varphi}_{n+1}$ of the finitary Veblen functions and base $(\text{OT}, <)$ on those (see also [21, §3]). We did not choose this approach for the sake of a better motivation and understanding of the main methods of the wellordering proof. For this section, we fix the following conventions:

- (a) We work within TID and presume knowledge on the ordinal notation system $(\text{OT}, <)$ that is mentioned in the appendix and given in [18, Chapter 3].
- (b) The notion *ordinal* denotes terms that are given according to the ordinal notation system $(\text{OT}, <)$. Small Greek letters $\alpha, \beta, \gamma, \delta, \dots$ denote explicit terms for ordinal notations in the sense of OT and which are given externally in the meta-theory. Furthermore, we shall simplify the rigid notation that appears in the appendix and Sect. 5.2 by dropping the tilde in the expressions such as $\tilde{\varphi}, \tilde{\omega}, \tilde{\dagger}$ and writing instead φ, ω, \dagger , respectively.
- (c) $P, \text{Acc}, \text{Prog}$, and TI shall denote $P_{<}, \text{Acc}_{<}, \text{Prog}_{<}$, and $\text{TI}_{<}$, respectively.

Proposition 6.1

- (a) $\text{TID} \vdash \forall x (x \notin \text{OT} \rightarrow Px)$.
- (b) $\text{TID} \vdash \forall x (Px \rightarrow \text{TI}(\mathcal{A}, x))$ for all $\mathcal{A} \in \mathcal{L}_{\text{PA}}$.
- (c) $\text{TID} \vdash \text{TI}(\mathcal{A}, \alpha)$ holds for each $\alpha < \omega$ and $\mathcal{A} \in \mathcal{L}_{\text{TID}}$.

Proof (Sketch) (a) holds immediately by (Cl), using that $a \notin \text{OT}$ implies $b \not\prec a$ for all b . For (b), assume $Pa, \text{Prog}(\mathcal{A})$, and $b < a$. We get Pb by (FP) from Remark 4.16, and since $\mathcal{A} \in \text{Pos}_1$ holds, we then get $\mathcal{A}(b)$ by (TID). For (c), note that we can show $\text{TI}(\mathcal{A}, \mathfrak{n}_k)$ for all $k \in \mathbb{N}$ by (meta-)induction on k , while we set $\mathfrak{n}_0 := 0$ and $\mathfrak{n}_{m+1} := \mathfrak{n}_m \tilde{\dagger} \tilde{1}$ for each $m \in \mathbb{N}$. \square

Remark 6.2 Dropping the restriction on the induction formula used in (Ind) yields $\text{TID} \vdash \text{TI}(\mathcal{A}, \alpha)$ for each $\alpha < \varepsilon_0$ and $\mathcal{A} \in \mathcal{L}_{\text{TID}}$. This is since TID would extend PA in this case with complete induction for the full language \mathcal{L}_{TID} , and we could then follow the usual wellordering proofs for PA, adapting them to the representation of ordinals below ε_0 as given here.

Remark 6.3 Due to Proposition 6.1(a), we can assume from now on without loss of generality that $a \in \text{OT}$ holds whenever we try to show Pa for some a within TID. In

particular, if we aim to prove $P(a + b)$ or $P\varphi(\bar{a}^{(n)})$ for some a, b, a_1, \dots, a_n (with $n \geq 1$), we tacitly assume that $a + b \in \text{OT}$ and $\varphi(\bar{a}^{(n)}) \in \text{OT}$ hold, respectively. Due to Lemma 5.24, we also get $a, b, a_1, \dots, a_n \in \text{OT}$. Since $x < y$ implies $x, y \in \text{OT}$ for all x, y by the definition of $<$, statements such as $c < a + b$ or $c < \varphi(\bar{a}^{(n)})$ imply $a + b \in \text{OT}$ and $\varphi(\bar{a}^{(n)}) \in \text{OT}$, respectively.

Lemma 6.4 $\text{TID} \vdash \forall x, y (Px \wedge Py \rightarrow P(x + y))$.

Proof Assume a_1, a_2 with Pa_1 and Pa_2 , so we have to show $P(a_1 + a_2)$. By showing $\text{Prog}(\mathcal{B})$ for $\mathcal{B} := \Lambda b.P(a_1 + b)$, we can use (TID) together with Pa_2 to get the claim. Now, $\text{Prog}(\mathcal{B})$ is $\forall z(\text{Acc}(\mathcal{B}, z) \rightarrow \mathcal{B}(z))$, so assume c and $\text{Acc}(\mathcal{B}, c)$, i.e., $\forall z < c(P(a_1 + z))$. Due to (Cl), it suffices to show $(\forall z < a_1 + c)(Pz)$. Let now $d < a_1 + c$. We have either $d < a_1$ and can use (FP) on assumption Pa_1 to get Pd , or otherwise, we have $a_1 \leq d < a_1 + c$ by Lemma 5.24. In the latter case, we have $d = a_1 + c_0$ for some $c_0 < c$, so our assumptions yield the claim. \square

6.1 The simple case for the binary Veblen function

We treat the case for the binary Veblen function separately in order to give a more transparent proof that avoids the technicalities that appear in the treatment of the general case in Sect. 6.2 (such as the use of auxiliary class terms of the form Part_n^k for $1 \leq k \leq n$).

Lemma 6.5 $\text{TID} \vdash \forall x, y (Px \wedge Py \rightarrow P\varphi(x, y))$.

Proof Note that $P0$ and hence $P1$ hold due to (Cl). Now, we assume a_1, a_2 with Pa_1, Pa_2 . We use the class term $\mathcal{B} := \Lambda a.\forall y(Py \rightarrow P\varphi(a, y))$ with $\mathcal{B} \in \text{Pos}_1$ and show $\text{Prog}(\mathcal{B})$. Then we can use (TID) with Pa_1 and Pa_2 . Now, in order to bring the proof of this lemma closer to the proof of Theorem 6.12 that deals with the general case of a finitary Veblen function, we note that $\text{Prog}(\mathcal{B})$ is $\forall z(\forall x < z(\mathcal{B}(x)) \rightarrow \mathcal{B}(z))$ Now, using the class term $\mathcal{A}_2^1 := \Lambda a.\forall y(Py \rightarrow \forall x < a(P\varphi(x, y)))$ and that $\mathcal{A}_2^1(a)$ is logically equivalent to $\forall x < a(\mathcal{B}(x))$, we get that $\text{Prog}(\mathcal{B})$ is logically equivalent to $\forall z(\mathcal{A}_2^1(z) \rightarrow \mathcal{B}(z))$ So, in order to show the latter, assume a with

$$\mathcal{A}_2^1(a) \tag{7}$$

and show $\mathcal{B}(a)$, while for proving $\mathcal{B}(a)$, assume b with Pb and show $P\varphi(a, b)$. Once more, we can use (TID), namely with $\mathcal{A}_2^2 := \Lambda d.P\varphi(a, d)$ on assumption Pb since $\mathcal{A}_2^2 \in \text{Pos}_1$ holds, while we have to show $\text{Prog}(\mathcal{A}_2^2)$.¹¹ Now, for proving $\text{Prog}(\mathcal{A}_2^2)$, we

¹¹ Noting our current assumption (7) and our current goal, we remark that we actually show

$$\mathcal{A}_2^1(a) \rightarrow \text{Prog}(\mathcal{A}_2^2)$$

which is a special case of Theorem 6.10, and also note that $\mathcal{A}_2^1 \in \text{Pos}_1$ holds with $\mathcal{A}_2^1 \notin \text{Pos}_0$, while we have $\mathcal{A}_2^2 \in \text{Pos}_0$.

assume d and z with

$$\forall z_0 \prec d \left(\mathcal{A}_2^2(z_0) \right) \quad \left(\text{i.e., } \forall z_0 \prec d (P\varphi(a, z_0)) \right) \tag{8}$$

and $z \prec \varphi(a, d)$, having to show Pz . This yields $P\varphi(a, d)$ by (CI) because z is arbitrary. We consider now the following case distinction.

- 1.** If $d \in \text{Lim}$: We get that $z \prec \varphi(a, d[x])$ holds for some x by Proposition 5.23. Since we have $d[x] \prec d$ by Proposition 5.22, we get $P\varphi(a, d[x])$ by (8), implying Pz by (FP).
- 2.** If $d \notin \text{Lim}$:
 - 2.1.** If $a = 0$: Since $d \notin \text{Lim}$ holds, we get $z \prec t(x)$ for some x by Proposition 5.23, where we let $t(x) := \omega^{d[x]} \cdot (x +_{\mathbb{N}} 1)$. We show $\forall x (P(t(x)))$ by induction on x and note that (Ind) is applicable here because of $\Lambda x. P(t(x)) \in \text{Pos}_1$. For $x = 0$, we can argue as for the case $d \in \text{Lim}$ and get $P(\omega^{d[0]})$. For $x = x_0 +_{\mathbb{N}} 1$, the claim follows from $P(t(0))$, the induction hypothesis, and Lemma 6.4, noting that $d[0] = d[x_0]$ holds by definition and because of $d \notin \text{Lim}$.
 - 2.2.** If $a \in \text{Lim}$ and $d = 0$: We have by Proposition 5.23 that $z \prec \varphi(a[x], 0)$ holds for some x . Since we have $a[x] \prec a$ by Proposition 5.22, we get $P\varphi(a[x], 0)$ with (7).
 - 2.3.** Otherwise, i.e., either $d = 0$ with $a \in \text{Suc}$ or $d \in \text{Suc}$ with $a \neq 0$: Letting $t := \varphi(a, d)$, we have by Proposition 5.22 some x such that $z \prec t[x]$ holds. Proving $\forall x (P(t[x]))$ by induction on x suffices now. Note again that (Ind) is applicable due to $\Lambda x. P(t[x]) \in \text{Pos}_1$, and also note for the following computations of $t[x]$ according to Definition A.1 in the appendix that we have $\varphi(a, d) = \phi ad$ by Lemma 5.24, bearing in mind that ϕ is from the ordinal notation system (OT, \prec).
 - 2.3.1** If $x = 0$: If $d = 0$ holds, then we have $t[0] = 1$ and are done since we have $P1$. If $d \in \text{Suc}$ holds with $d = d_0 + 1$, then we have $t[0] = \varphi(a, d_0) + 1$, and since $d_0 \prec d$ holds, we get $P(t[0])$ from (8) and Lemma 6.4 by using $P1$.
 - 2.3.2** If $x = x_0 +_{\mathbb{N}} 1$: We have $t[x_0 +_{\mathbb{N}} 1] = \varphi(a[x_0], t[x_0])$, so the claim follows with $a[x_0] \prec a$ from Proposition 5.22, the induction hypothesis $P(t[x_0])$, and (7).

□

Corollary 6.6 $\text{TID} \vdash P\varphi(\alpha, 0)$ holds for each $\alpha \prec \omega$.

Proof The claim is a direct consequence of Lemma 6.5. Note that $P\alpha$ for $\alpha \prec \omega$ follows from Proposition 6.1(c): $\text{TI}(\Lambda a. Pa, \alpha)$ and (CI) imply $\forall x \prec \alpha (Px)$, hence $P\alpha$ by (FP). □

Remark 6.7 We proved Lemma 6.5 by applying (TID) to a class term \mathcal{B} in Pos_1 that is not in Pos_0 . Though, in order to show $\text{Prog}(\mathcal{B})$ in the proof of Lemma 6.5, we can work with a (weaker) subtheory TID_0 of TID that is obtained from TID by restricting (the instances of) the axiom schemes (TID) and (Ind) to class terms that are in Pos_0 (rather than Pos_1), see [18, Chapter 8] for details. This theory TID_0 is the restriction of the theory $\text{ID}_1^* \upharpoonright$ to accessible part positive operator forms, i.e., to the language \mathcal{L}_{TID} , while

$ID_1^* \upharpoonright$ is a subtheory of ID_1 for positive induction and with the same restriction for complete induction. The proof-theoretic ordinal of $ID_1^* \upharpoonright$ is $\varphi(\omega, 0)$, see for instance [16]. Note furthermore that the proof of Theorem 6.10 below does not invoke (TID), so this result holds also for the restriction TID_0 .

6.2 The general case for the finitary Veblen functions

Definition 6.8 For $k, n \in \mathbb{N}$ with $1 \leq k < n$, we define

$$\begin{aligned} \text{Part}_n^k &:= \Lambda \bar{a}^{(k)}. \forall y \left(P y \rightarrow \forall x \prec a_k \left(P \varphi \left(\bar{a}^{(k-1)}, x, y, \bar{0}^{(n-k-1)} \right) \right) \right) \\ \text{Hyp}_n^k &:= \Lambda \bar{a}^{(k)}. \text{Part}_n^1(a_1) \wedge \dots \wedge \text{Part}_n^k(\bar{a}^{(k)}) \\ \text{Hyp}_n^0 &:= (0 = 0) \end{aligned}$$

Lemma 6.9 For $k, n \in \mathbb{N}$ and variables a_1, \dots, a_{n-1} , the following holds:

- (a) $(\Lambda a. P(\varphi(\bar{a}^{(n-1)}, a))) \in \text{Pos}_0$ for $1 \leq n$.
- (b) $(\Lambda a. \text{Part}_n^k(\bar{a}^{(k-1)}, a)) \in \text{Pos}_1$ for $1 \leq k < n$.

Proof (a) is obvious. For (b), note in the definition of $\text{Part}_n^k(\bar{a}^{(k-1)}, a)$ that $P y$ and $\forall x(x \prec a \rightarrow P \varphi(\bar{a}^{(k-1)}, x, y, \bar{0}^{(n-k-1)}))$ are in Pos_0 . Furthermore, $P y$ does not contain a as a free variable, so we get indeed that $\forall y(P y \rightarrow \forall x \prec a(P \varphi(\bar{a}^{(k-1)}, x, y, \bar{0}^{(n-k-1)})))$ is in $\text{Pos}_1(a)$. \square

Theorem 6.10 $TID \vdash \forall \bar{a}^{(n-1)} (\text{Hyp}_n^{n-1}(\bar{a}^{(n-1)}) \rightarrow \text{Prog}(\Lambda a. P \varphi(\bar{a}^{(n-1)}, a)))$ holds for $n \geq 1$.

Proof Assume $n \geq 1$ and $\bar{a}^{(n-1)}$ with $\text{Hyp}_n^{n-1}(\bar{a}^{(n-1)})$. For showing $\text{Prog}(\Lambda a. P \varphi(\bar{a}^{(n-1)}, a))$, assume a and d such that $d \prec \varphi(\bar{a}^{(n-1)}, a)$ and

$$\forall x \prec a (P \varphi(\bar{a}^{(n-1)}, x)) \tag{9}$$

hold and show Pd . This would yield $P \varphi(\bar{a}^{(n-1)}, a)$ by (Cl) because d is arbitrary.

1. If $n = 1$ or $a_1 = \dots = a_{n-1} = 0$ hold: As in Lemma 6.5 due to $\varphi(\bar{a}^{(n-1)}, a) = \varphi(a) = \omega^a$.
2. Otherwise: We can assume now that some $1 \leq l \leq n - 1$ exists with $a_l \neq 0$ and $a_{l+1} = \dots = a_{n-1} = 0$, i.e., that we have $\varphi(\bar{a}^{(n-1)}, a) = \varphi(\bar{a}^{(l)}, \bar{0}^{(n-l-1)}, a)$ with $a_l \neq 0$. Furthermore, our assumption $\text{Hyp}_n^{n-1}(\bar{a}^{(n-1)})$ yields

$$\text{Part}_n^1(a_1) \wedge \dots \wedge \text{Part}_n^{n-1}(\bar{a}^{(n-1)}) \tag{10}$$

Consider now the following case distinction and note that $P0$ and hence $P1$ hold due to (Cl).

- 2.1. If $a \in \text{Lim}$: We get that $d \prec \varphi(\bar{a}^{(n-1)}, a[x])$ holds for some x by Proposition 5.23 and assumption $d \prec \varphi(\bar{a}^{(n-1)}, a)$. Since we have $a[x] \prec a$ by Proposition 5.22, we get $P \varphi(\bar{a}^{(n-1)}, a[x])$ by (9) which implies Pd by (FP).

- 2.2.** If $a \notin \text{Lim}$:
- 2.2.1.** If $a_l \in \text{Lim}$ and $a = 0$: By Proposition 5.23, we have $z < \varphi(\bar{a}^{(l-1)}, a_l[x], \bar{0}^{(n-l)})$ for some x . We get $P\varphi(\bar{a}^{(l-1)}, a_l[x], \bar{0}^{(n-l)})$ with $\text{Part}_n^l(\bar{a}^{(l)})$ from (10) because we have $a_l[x] < a_l$ by Proposition 5.22.
- 2.2.2.** Otherwise, i.e., either $a = 0$ with $a_l \in \text{Suc}$ or $a \in \text{Suc}$ with $a_l \neq 0$: In this situation, Lemma 5.24 implies $\varphi(\bar{a}^{(l)}, \bar{0}^{(n-l)}) = \phi a_p \dots a_l \bar{0}^{(n-l-1)} a$ for some $1 \leq p \leq l$ where $a_1, \dots, a_p = 0$ holds. For the computation of $t[x]$, see Definition A.1 in the appendix and bear in mind that ϕ is from the ordinal notation system $(\text{OT}, <)$. In order to simplify notation and without loss of generality, we shall assume $p = 1$, noting that the following argument works for the general case as well. Letting $t := \phi \bar{a}^{(l)} \bar{0}^{(n-l-1)} a$ we have by Proposition 5.22 some x such that $z < t[x]$ holds. Proving $\forall x (P(t[x]))$ by induction on x suffices now. (Ind) is applicable because $\Lambda x. P(t[x]) \in \text{Pos}_1$ holds.
 - 2.2.2.1.** If $x = 0$: For $a = 0$, we have $t[0] = 1$ and are done since we have $P1$. If $a \in \text{Suc}$ holds with $a = a_0 + 1$, then we have $t[0] = \varphi(\bar{a}^{(n-1)}, a_0) + 1$ due to the form of t and the definition of $t[0]$. Since $a_0 < a$ holds, we get $P(t[0])$ from (9) and Lemma 6.4 by using $P1$.
 - 2.2.2.2.** If $x = x_0 +_{\mathbb{N}} 1$: We get $t[x_0 +_{\mathbb{N}} 1] = \varphi(\bar{a}^{(l-1)}, a_l[x_0], t[x_0], \bar{0}^{(n-l)})$, so the claim follows with $a_l[x_0] < a_l$ from Proposition 5.22, the induction hypothesis $P(t[x_0])$, and $\text{Part}_n^l(\bar{a}^{(l)})$ from (10). □

Corollary 6.11 $\text{TID} \vdash \forall \bar{a}^{(n)} (\text{Hyp}_n^{n-1}(\bar{a}^{(n-1)}) \wedge Pa_n \rightarrow P\varphi(\bar{a}^{(n)}))$ for $n \in \mathbb{N}$ with $n \geq 1$.

Proof Immediate from Theorem 6.10 by using (TID) and Lemma 6.9(a). □

Theorem 6.12 $\text{TID} \vdash \forall \bar{a}^{(k-1)} (\text{Hyp}_n^{k-1}(\bar{a}^{(k-1)}) \rightarrow \text{Prog}(\Lambda a. \text{Part}_n^k(\bar{a}^{(k-1)}, a)))$ for $1 \leq k < n$.

Proof We fix $n \geq 1$ and argue by induction on $n - k$ for $1 \leq k < n$. Let $\bar{a}^{(k-1)}$ be given with $\text{Hyp}_n^{k-1}(\bar{a}^{(k-1)})$, while note that this just gives us the formula $0 = 0$ in case of $k = 1$. Assume now a, a_k, a_{k+1} with

$$\forall x \ x < a \left(\text{Part}_n^k(\bar{a}^{(k-1)}, x) \right) \wedge Pa_{k+1} \wedge a_k < a \tag{11}$$

in order to show $\text{Prog}(\Lambda a. \text{Part}_n^k(\bar{a}^{(k-1)}, a))$. In case we have $k \neq n - 1$, further let $a_j := 0$ for each $k + 1 < j \leq n$. We have to show $P(\varphi(\bar{a}^{(k+1)}, \bar{0}^{(n-k-1)}))$, i.e.,

$$P(\varphi(\bar{a}^{(n)})) \tag{*}$$

For this, we show by a side induction on i that $1 \leq i < n$ implies $\text{Hyp}_n^i(\bar{a}^{(i)})$. Then for $i := n - 1$, we get (*) by Corollary 6.11 since we have Pa_n by (11) or (if $k = n - 1$) by $P0$.

For the remaining claim, note that $\text{Part}_n^k(\bar{a}^{(k-1)}, a_k)$ holds by (11) and so $\text{Hyp}_n^k(\bar{a}^{(k)})$ by definition, using assumption $\text{Hyp}_n^{k-1}(\bar{a}^{(k-1)})$. We use this argument in case of

$1 \leq i \leq k$. If we have $k < i < n$, then we can use the side induction hypothesis and get $\text{Hyp}_n^{i-1}(\bar{a}^{(i-1)})$. So, the main induction hypothesis yields $\text{Prog}(\Lambda a.\text{Part}_n^i(\bar{a}^{(i-1)}, a))$ and hence we get $\forall a(Pa \rightarrow \text{Part}_n^i(\bar{a}^{(i-1)}, a))$ by (TID), while noting Lemma 6.9(b). Now, $\text{Part}_n^i(\bar{a}^{(i)})$ follows from Pa_{k+1} in (11) in case of $i = k + 1$ and otherwise from $P0$. Hence, we get $\text{Hyp}_n^i(\bar{a}^{(i)})$ by $\text{Hyp}_n^{i-1}(\bar{a}^{(i-1)})$. \square

Corollary 6.13 $\text{TID} \vdash \forall \bar{a}^{(k)}(\text{Hyp}_n^{k-1}(\bar{a}^{(k-1)}) \wedge Pa_k \rightarrow \text{Hyp}_n^k(\bar{a}^{(k)}))$ holds for all $k, n \in \mathbb{N}$ with $1 \leq k < n$.

Proof From $\text{Hyp}_n^{k-1}(\bar{a}^{(k-1)}) \wedge Pa_k$, we get $\text{Part}_n^k(\bar{a}^{(k)})$ by Theorem 6.12 and (TID), while noting Lemma 6.9(b). Hence, we get $\text{Hyp}_n^k(\bar{a}^{(k)})$. \square

Theorem 6.14 $\text{TID} \vdash \forall \bar{a}^{(n)}(\bigwedge_{i=1}^n Pa_i \rightarrow P\varphi(\bar{a}^{(n)}))$ holds for each $n \geq 1$.

Proof Let $n \geq 1$ and assume $\bar{a}^{(n)}$ with $\bigwedge_{i=1}^n Pa_i$. We now show by induction on $k \in \mathbb{N}$ that $0 \leq k < n$ implies $\text{Hyp}_n^k(\bar{a}^{(k)}) \wedge \bigwedge_{i=k+1}^n Pa_i$. Then the claim $P\varphi(\bar{a}^{(n)})$ follows from this and Corollary 6.11 with $k := n - 1$. So assume $0 \leq k < n$. If $k = 0$, we trivially get $\text{Hyp}_n^0(\bar{a}^{(0)}) \wedge \bigwedge_{i=1}^n Pa_i$ due to the definition of Hyp_n^0 and our assumption $\bigwedge_{i=1}^n Pa_i$. If $0 < k \leq n$, the induction hypothesis yields $\text{Hyp}_n^{k-1}(\bar{a}^{(k-1)}) \wedge \bigwedge_{i=k}^n Pa_i$ and hence the claim $\text{Hyp}_n^k(\bar{a}^{(k)}) \wedge \bigwedge_{i=k+1}^n Pa_i$ due to Corollary 6.13. \square

Corollary 6.15 (Lower bound of TID) $\text{TID} \vdash \text{TI}(\mathcal{A}, \alpha)$ holds for each $\mathcal{A} \in \mathcal{L}_{PA}$ and $\alpha \in \text{OT}$.

Proof By induction on the build-up of $\alpha \in \text{OT}$ as described in the appendix. We can use Lemma 6.4 and Theorem 6.14 together with Proposition 6.1(b). \square

7 Upper bound $\vartheta\Omega^\omega$ for FIT and TID

For determining the upper bound of FIT, we apply one result from [14] that relates over ACA_0 the scheme $(\Pi_3^1\text{-RFN})$ of ω -model reflection for Π_3^1 formulas to the scheme $(\Pi_2^1\text{-BI})$ of bar induction for Π_2^1 formulas, and one result of [20] that determines the proof-theoretic ordinal of $\Pi_2^1\text{-BI}_0$ to be the small Veblen ordinal $\vartheta\Omega^\omega$. Then an embedding of FIT into the second order theory $\Pi_3^1\text{-RFN}_0$ of ω -model reflection for Π_3^1 formulas suffices to get the desired upper bound result for FIT. Moreover and due to Sect. 4.2, this also provides an upper bound for TID. In particular, we shall exploit the Π_1^1 definability of a least fixed-point. A similar approach has been taken in [1, 16] for the treatment of the theories $\Pi_2^1\text{-RFN}_0$ and ID_1^* (a subsystem of ID_1 that allows only positive induction for the predicates $P_{\mathfrak{A}}$ that are assigned to each positive operator form \mathfrak{A}). Below, we shall provide an upper bound for FIT by embedding it directly into $\Pi_3^1\text{-RFN}_0$. We remark that if we were to investigate only the subtheory TID of ID_1 , we could have embedded it directly into $\Pi_3^1\text{-RFN}_0$ (rather than taking the detour via FIT as Fig. 1 from Sect. 1 suggests). Furthermore, we recall that D. Probst’s modular ordinal analysis from [17] determines the proof-theoretic ordinal of $\Pi_3^1\text{-RFN}_0$ to be the small Veblen ordinal by metapredicative methods.

7.1 Subsystems of second order arithmetic

We shall introduce here subsystems of second order arithmetic, and we formulate them in the language \mathcal{L}_{PA}^2 that we defined in Sect. 4. In particular, recall that \mathcal{L}_{PA}^2 formulas allow for quantification over set variables X . The following definitions are taken to some extent from [14] and [23], respectively, and we refer to these sources for more details on subsystems of second order arithmetic and in particular to the underlying two-sorted logic.

Definition 7.1 We use the following standard abbreviations

$$\begin{aligned} (X)_t &:= \Lambda a. \langle t, a \rangle \in X \\ (QY \dot{\in} X)A &:= (Qy)A((X)_y/Y) && \text{(where } Q \in \{\forall, \exists\}) \\ Y \dot{\in} X &:= (\exists Z \dot{\in} X)(Z = Y) && \text{(i.e., } Y \dot{\in} X \text{ is } \exists z((X)_z = Y)) \end{aligned}$$

By A^X or also $X \models A$, we denote the (canonical) relativization of a formula A to a set variable X , which is defined inductively as follows: If A is an atomic formula, then $A^X := A$. If A is $\neg A_0$, then $A^X := \neg(A_0^X)$. If A is $A_0 \circ A_1$ with $\circ \in \{\vee, \wedge, \rightarrow\}$, then $A^X := A_0^X \circ A_1^X$. If A is $(Qx)A_0$ with $Q \in \{\exists, \forall\}$, then $A^X := (Qx)A_0^X$. If A is $(QX)A_0$ with $Q \in \{\exists, \forall\}$, then $A^X := (QY \dot{\in} X)A_0^X$.

As usual, we assume tacitly a renaming of bound variables in order to avoid a clash of variables. Note that set variables occur at most free in A^X , i.e., A^X is arithmetical.

Definition 7.2 $A \in \Pi_0^1$ and $A \in \Sigma_0^1$ denotes that A is arithmetical, i.e., A is a Π_0^1 (or also Σ_0^1) formula, i.e., a \mathcal{L}_{PA}^2 formula without quantifications over set variables. $A \in \Pi_n^1$ denotes that A is a Π_{n+1}^1 formulas, i.e., a formula which is of the form $\forall X_1 \exists X_2 \dots (Q_{n+1} X_{n+1})B$ for some $B \in \Pi_0^1$, and where Q_{n+1} is \exists for even n and Q_{n+1} is \forall otherwise. $A \in \Sigma_n^1$ denotes that A is a Σ_{n+1}^1 formulas, i.e., of the form $\exists X B$ with $B \in \Pi_n^1$.

Definition 7.3 ACA_0 is the two-sorted theory based on \mathcal{L}_{PA}^2 and the axioms of PA without complete induction, where the equality axioms (for the first sort) hold for the language \mathcal{L}_{PA}^2 . Moreover, ACA_0 consists of *set induction*, i.e., $\forall X((0 \in X \wedge \forall x(x \in X \rightarrow \mathbf{S}x \in X)) \rightarrow \forall x(x \in X))$, and of *arithmetical comprehension (ACA)*, i.e., $\exists X \forall x(x \in X \leftrightarrow A)$ for each $A \in \Pi_0^1$ that does not contain X .

Proposition 7.4 ACA_0 is finitely axiomatizable by a Π_2^1 -sentence F_{ACA} .

Proof See for instance [23, Lemma VIII.1.5]. □

Definition 7.5 We define the following theories and principles:

- $\Sigma_1^1\text{-AC}_0$ is ACA_0 extended by the principle of Σ_1^1 axiom of choice ($\Sigma_1^1\text{-AC}$), i.e., $\forall x \exists X A \rightarrow \exists Y \forall x(A((Y)_x/X))$ for each $A \in \Sigma_1^1$.
- $\Pi_n^1\text{-RFN}_0$ is ACA_0 extended by the principle of Π_n^1 ω -model reflection for $n \in \mathbb{N}$, i.e. for each $A \in \Pi_n^1$ with at most U_1, \dots, U_k occurring as free set variables in A (and where F_{ACA} is taken from Proposition 7.4), the principle is the formula $\forall U_1, \dots, U_k(A \rightarrow \exists X(A^X \wedge F_{ACA}^X \wedge U_1 \dot{\in} X \wedge \dots \wedge U_k \dot{\in} X))$.

For the principles $(\Pi_n^1\text{-Bl}_0)$ of Π_n^1 bar induction and $(\Sigma_1^1\text{-DC})$ of Σ_1^1 axiom of dependent choice and their corresponding theories $\Pi_n^1\text{-Bl}_0$ and $\Sigma_1^1\text{-DC}_0$, see the references made in Sect. 7.2.

7.2 Upper bound results from the literature

Theorem 7.6 $|\Pi_{n+2}^1\text{-Bl}_0| = \vartheta\Omega(n + 1, \omega)$ holds for all $n \in \mathbb{N}$.

Proof See [20, Theorem 10.2.]. □

Theorem 7.7 $(\Pi_{n+1}^1\text{-Bl})$ and $(\Pi_{n+2}^1\text{-RFN})$ are equivalent over ACA_0 for all $n \in \mathbb{N}$.

Proof See [14, Main Theorem]. □

Theorem 7.8 Over ACA_0 , we have

- (a) $(\Pi_n^1\text{-RFN})$ implies $(\Pi_k^1\text{-RFN})$ for $k \leq n$.
- (b) $(\Pi_2^1\text{-RFN})$ is equivalent to $(\Sigma_1^1\text{-DC})$.
- (c) $(\Sigma_1^1\text{-DC})$ implies $(\Sigma_1^1\text{-AC})$.

Proof See [23, Theorem VIII.5.12, Lemma VII.6.6]. □

Corollary 7.9 $(\Pi_{n+2}^1\text{-RFN})$ implies $(\Sigma_1^1\text{-AC})$ over ACA_0 for all $n \in \mathbb{N}$.

Proof By Theorem 7.8. □

Theorem 7.10 $|\Sigma_1^1\text{-DC}_0| = \varphi(\omega, 0)$.

Proof See the last paragraph in [8]. □

Corollary 7.11 $|\Pi_{n+2}^1\text{-RFN}_0| = \vartheta\Omega(n, \omega)$ holds for all $n \in \mathbb{N}$.

Proof For $n \geq 1$, this is immediate from Theorems 7.6 and 7.7. For $n = 0$, use Theorems 7.10 and 7.8(b) and that $\vartheta\Omega(0, \omega) = \vartheta(\Omega \cdot \omega) = \varphi(\omega, 0)$ holds by Proposition 5.21(b). □

7.3 Some syntactical properties of $\mathcal{L}_{\text{PA}}^2$ formulas

Definition 7.12 Let T be some theory of $\mathcal{L}_{\text{PA}}^2$ as introduced in Sect. 7.1. $A \in \Pi_n^1(T)$ denotes that A is an Π_n^1 formula over T , i.e., $A \in \mathcal{L}_{\text{PA}}^2$ such that A is provably equivalent over T to some formula $A' \in \Pi_n^1$.

Remark 7.13 In case that $A \in \Pi_n^1(T)$ is given for some theory T of $\mathcal{L}_{\text{PA}}^2$ and we consider some $A' \in \Pi_n^1$ that is provably equivalent over T to A , then we can assume that A and A' have the same free variables, and we shall tacitly do so from now on. Moreover, if T_1, T_2 are theories of $\mathcal{L}_{\text{PA}}^2$ as introduced in Sect. 7.1 such that T_2 comprises T_1 , then obviously $A \in \Pi_n(T_1)$ implies $A \in \Pi_n(T_2)$.

Proposition 7.14 Let $k, n \in \mathbb{N}$ and $T \in \{\text{ACA}_0, \Sigma_1^1\text{-AC}_0\}$. Then we have the following.

- (a) If $A \in \Pi_k^1(\mathbb{T})$ holds with $k < n$, then we have $A \in \Pi_n^1(\mathbb{T})$ and $\neg A \in \Pi_n^1(\mathbb{T})$.
- (b) $\Pi_n^1(\mathbb{T})$ is closed under conjunction, disjunction, and universal quantification for number variables, i.e., we have that $A, B \in \Pi_n^1(\mathbb{T})$ implies $A \circ B \in \Pi_n^1(\mathbb{T})$ for $\circ \in \{\wedge, \vee\}$ and that $A \in \Pi_n^1(\mathbb{T})$ implies $\forall x A \in \Pi_n^1(\mathbb{T})$.
- (c) $\Pi_{n+1}^1(\mathbb{T})$ is closed under universal quantification for set variables, i.e., we have that $A \in \Pi_{n+1}^1(\mathbb{T})$ implies $\forall X A \in \Pi_{n+1}^1(\mathbb{T})$.

Proof Straightforward by using essentially (ACA), see [18, Proposition 7.36]. □

Corollary 7.15 Let $k, n \in \mathbb{N}$ and $\mathbb{T} \in \{\text{ACA}_0, \Sigma_1^1\text{-AC}_0\}$. Given $A_0, \dots, A_k \in \Pi_n^1(\mathbb{T})$ and $B \in \Pi_{n+1}^1(\mathbb{T})$, we have $\forall \vec{x}(A_0 \rightarrow \dots \rightarrow A_k \rightarrow B) \in \Pi_{n+1}^1(\mathbb{T})$.

Proof Immediate by Proposition 7.14 and induction on $k \in \mathbb{N}$, while noting that $A_k \in \Pi_n^1(\mathbb{T})$ implies $\neg A_k \in \Pi_{n+1}^1(\mathbb{T})$, and that $A_k \rightarrow B$ is equivalent to $\neg A_k \vee B$. □

7.4 Embedding FIT into $\Pi_3^1\text{-RFN}_0$

In order to interpret within $\Pi_3^1\text{-RFN}_0$ the applicative part of FIT, i.e., **I**, in Definition 3.10, we shall first implement the so-called *canonical model* $\mathfrak{B}\mathfrak{R}$ for this applicative part. It is built upon ordinary recursion theory and by using indices of *partial recursive* functions for interpreting the function symbol \cdot of \mathcal{L}_{FIT} . A thorough introduction to this construction and a more detailed treatment of the following (in a slightly different setting) will be elaborated in [11]. Without going into detail, we let **T** be the ternary, primitive recursive relation **T** according to Kleene’s Normal Form Theorem, and let **U** be the corresponding unary primitive recursive (result-extracting) function. Reading $\{e\}$ as the partial recursive function indexed by the number e , we write $\{e\}(n_1, \dots, n_k) \simeq m$ to denote $\exists x(\mathbf{T}(e, (n_1, \dots, n_k), x) \wedge \mathbf{U}(x) = m)$ for $e, k, m, n_1, \dots, n_k \in \mathbb{N}$. Furthermore, let $\Pi_1^1(x, y)$ with $x \neq y$ denote a universal Π_1^1 formula for Π_1^1 formulas that have one free variable, i.e., we have $\Pi_1^1(x, y) \in \Pi_1^1$ and for each $\mathcal{L}_{\text{PA}}^2$ formula $A \in \Pi_1^1$ with $\text{FV}(A) = \{y\}$, we have that $\exists x \forall y(\Pi_1^1(x, y) \leftrightarrow A)$ holds over ACA_0 .¹²

Definition 7.16 In the abovementioned setting, we let **T** and **U** also denote the corresponding relation and function symbols in \mathcal{L}_{PA} , and then we set $\{a\}(b) \simeq c := \exists x(\mathbf{T}(a, b, x) \wedge \mathbf{U}(x) = c)$. Next, we assume an assignment of the constants k, s of \mathcal{L}_{FIT} to numerals k^*, s^* that have corresponding properties over ACA_0 as described by the axiom group **I.1** in Definition 3.10. For the remaining constants $p, p_0, p_1, 0, s_N, p_N, d_N$ of \mathcal{L}_{FIT} , we set p^* to be the numeral of the (primitive recursive) function $(m, n) \mapsto \langle m, n \rangle$; p_i^* to be the numeral of $m \mapsto (m)_i$ for $i = 0, 1$; 0^* to be 0; s_N^* to be the numeral of $m \mapsto m + 1$; p_N^* to be the numeral of $m \mapsto m \dot{-} 1$ (i.e., the modified subtraction function such that $m \dot{-} \mathbb{N} n = 0$ holds if $m <_{\mathbb{N}} n$); and d_N^* to be the numeral of the case distinction function, mapping (k, l, m) to l if $k = 0$, otherwise to m .

The translation $V_t^*(x)$ of a \mathcal{L}_{FIT} term t into the language of \mathcal{L}_{PA} (expressing that the value of t equals x) is defined for variables $x \notin \text{FV}(t)$ as follows: $t = x$ if t is

¹² Bear in mind that this universal Π_1^1 formula shall include the unary relation symbol **U** of \mathcal{L}_{PA} as a parameter.

a variable, $t^* = x$ if t is a constant, and $\exists y, z(V_t^*(y) \wedge V_s^*(z) \wedge \{y\}(z) \simeq x)$ if t is rs . For each \mathcal{L}_{FT} formula A , we let the $\mathcal{L}_{\rho_A}^2$ formula A^* be defined recursively on the build-up of A as follows for every $x \notin \text{FV}(A)$:

$$\begin{aligned}
 (s = t)^* &:= \exists x(V_s^*(x) \wedge V_t^*(x)) & (t \in \mathbb{N})^* &:= \exists x(V_t^*(x)) \\
 (t \in \mathbb{U})^* &:= \exists x(V_t^*(x) \wedge x \in \mathbb{U}) & (t \in \overline{\mathbb{N}})^* &:= V_t^*(0) \wedge \neg V_t^*(0) \\
 (t \in X)^* &:= \exists x(V_t^*(x) \wedge x \in X) & (\{x : B_+\})^* &:= \exists x(V_t^*(x) \wedge B^*(z/x)) \\
 (\neg B)^* &:= \neg(B^*) & (B \circ C)^* &:= B^* \circ C^* \\
 (QzB)^* &:= QzB^* & (QXB)^* &:= Qz(B^*((\Delta a.\Pi_1^1(z, a))/X))
 \end{aligned}$$

for $B_+ \in \text{For}^+$, $\circ \in \{\wedge, \vee, \rightarrow\}$ and $Q \in \{\forall, \exists\}$. For the case that A is $t \in \mathbb{I}_{\mathbb{P}\mathbb{Q}}$, we introduce first the following positive operator form (for any $\mathbb{P}, \mathbb{Q} \in \text{Ty}|\}$)

$$\text{Acc}_{\mathbb{P},\mathbb{Q}}^* := \Lambda X \Lambda x.(x \in \mathbb{P})^* \wedge \forall y((y \in \mathbb{P})^* \rightarrow (\langle y, x \rangle \in \mathbb{Q})^* \rightarrow y \in X)$$

and note that \mathbb{P}, \mathbb{Q} do not contain expressions of the form $\mathbb{I}_{\mathbb{P}\mathbb{Q}}$. Eventually, we set

$$(t \in \mathbb{I}_{\mathbb{P}\mathbb{Q}})^* := \forall X(\forall x(\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(X, x) \rightarrow x \in X) \rightarrow (t \in X)^*) \tag{\star-\mathbb{I}_{\mathbb{P}\mathbb{Q}}}$$

Furthermore, we tacitly assume in the definition of the translation A^* as usual a renaming of bound variables in order to avoid a clash of variables. Note also that the translation is meant to interpret type variables as Π_1^1 definable sets and that $\mathbb{I}_{\mathbb{P}\mathbb{Q}} \in \text{Ty}$ implies that \mathbb{P}, \mathbb{Q} do not contain type variables (since $\mathbb{P}, \mathbb{Q} \in \text{Ty}|\}$).

Lemma 7.17 *Let $A \in \mathcal{L}_{\text{FT}}$, then A^* and A have the same free variables.*

Proof This is clear from the definition of A^* . We note that it is due to this lemma that we defined $(t \in \overline{\mathbb{N}})^*$ as $V_t^*(0) \wedge \neg V_t^*(0)$ instead as $\neg(0 = 0)$. \square

Remark 7.18 For any $\mathbb{F} \in \text{FT}$, consider the $\mathcal{L}_{\rho_A}^2$ class term $\mathcal{A} := \Lambda z.(tz \in \mathbb{F})^*$. In order to make later arguments more readable, we shall make the translation of the \mathcal{L}_{FT} formula $\text{Cl}_{\mathbb{P},\mathbb{Q}}(\Lambda z.tz \in \mathbb{F})$ more explicit (cf., Notation 3.8):

$$(\text{Cl}_{\mathbb{P},\mathbb{Q}}(\Lambda z.tz \in \mathbb{F}))^* \text{ is } \forall x(\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(\mathcal{A}, x) \rightarrow \mathcal{A}(x)) \tag{\star-\text{Cl}_{\mathbb{P},\mathbb{Q}}}$$

We defined $\text{Cl}_{\mathbb{P},\mathbb{Q}}(\Lambda z.tz \in \mathbb{F})$ in Sect. 3 in order to have the above representation that allows to use $\text{Acc}_{\mathbb{P},\mathbb{Q}}^*$ in a intuitive way.

Lemma 7.19 *Let $n \geq 0$. For each \mathcal{L}_{FT} term t and each \mathcal{L}_{ρ_A} term r , the following holds.*

- (a) $V_t^*(x) \in \Pi_0^1$.
- (b) $(t \in X)^* \in \Pi_0^1$ and $\mathbb{P} \in \text{Ty}|\}$ imply $(t \in \mathbb{P})^* \in \Pi_0^1$.
- (c) $B \in \Pi_n^1(\mathbb{T})$ implies $\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(B, r) \in \Pi_n^1(\mathbb{T})$ for $\mathbb{T} \in \{\text{ACA}_0, \Sigma_1^1\text{-AC}_0\}$, $B := \Lambda a.B(a)$ with $B \in \mathcal{L}_{\rho_A}^2$, and $\mathbb{P}, \mathbb{Q} \in \text{Ty}|\}$. In particular, we have $\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(B, r) \in \Pi_0^1$ in case of $B \in \Pi_0^1$.

- (d) From $\mathbb{P}, \mathbb{Q} \in \text{Ty} \uparrow$, we get that $\text{Cl}_{\mathbb{P}, \mathbb{Q}}(X)^* \in \Pi_0^1$ and $(t \in \text{I}_{\mathbb{P}\mathbb{Q}})^* \in \Pi_1^1$ hold.
- (e) $A \in \text{For}^+$ implies $A \in \Pi_1^1(\Sigma_1^1\text{-AC}_0)$.
- (f) $\mathbb{F} \in \text{FT}$ implies $(t \in \mathbb{F})^* \in \Pi_2^1(\Sigma_1^1\text{-AC}_0)$.

Proof For (a): This follows easily after inspecting the definition of $V_t^*(x)$. For (b): $(t \in X)^* \in \Pi_0^1$ follows from (a). Given $\mathbb{P} \in \text{Ty} \uparrow$, we first note that then by definition, it can only be the case that \mathbb{P} is $\mathbb{N}, \bar{\mathbb{N}}$, or $\{x : A\}$ for some $A \in \text{For}^+$ such that A does not contain any $\mathbb{P}'\mathbb{Q}'$ expression or type variable. By (a) and Definition 7.16, one can easily verify that $(t \in \mathbb{P})^* \in \Pi_0^1$ holds.

For (c): $\text{Acc}_{\mathbb{P}, \mathbb{Q}}^*(\mathcal{B}, r)$ translates to the formula $(r \in \mathbb{P})^* \wedge \forall y((y \in \mathbb{P})^* \rightarrow ((y, x) \in \mathbb{Q})^*(r/x) \rightarrow y \in \mathcal{B})$ and then the claim follows from Proposition 7.14 and (b), using the assumption $B \in \Pi_n^1(\mathbb{T})$ and that $\mathbb{P}, \mathbb{Q} \in \text{Ty} \uparrow$ holds.

For (d): We have $\text{Acc}_{\mathbb{P}, \mathbb{Q}}^*(X, x) \in \Pi_0^1$ by the second claim of (c), and further with (b) and after inspecting $(\star\text{-I}_{\mathbb{P}\mathbb{Q}})$ on page 30 and $(\star\text{-Cl}_{\mathbb{P}, \mathbb{Q}})$ on page 30, the claim becomes clear.

For (e): We prove here a more general statement

$$\left. \begin{aligned} A \in \text{For}^+ &\implies A^* \in \Pi_1^1(\Sigma_1^1\text{-AC}_0) \\ \neg A \in \text{For}^+ &\implies \neg A^* \in \Pi_1^1(\Sigma_1^1\text{-AC}_0) \end{aligned} \right\} (*)$$

and by induction on the build-up of the \mathcal{L}_{FIT} formula A . Now, let $A \in \text{For}^+$ or $\neg A \in \text{For}^+$ be given. Note that A cannot be of the form $\forall X A_0$ or $\exists X A_0$ because of the definition of For^+ .

1. Base case: If A is of the form $t \in U, t \downarrow$, or $s = t$, we have $A^* \in \Pi_0^1$ and are done.

2. Step case $t \in \mathbb{P}$: If A is $t \in \mathbb{P}$ with $\mathbb{P} \in \text{Ty}$, then $A \in \text{For}^+$ must hold. Because of (b), we also only need to consider the case where $\mathbb{P} \notin \text{Ty} \uparrow$ and \mathbb{P} is not a type variable. Hence, \mathbb{P} is either of the form $\text{I}_{\mathbb{P}', \mathbb{Q}'}$ with $\mathbb{P}', \mathbb{Q}' \in \text{Ty} \uparrow$ or \mathbb{P} is of the form $\{z : B\}$ for some $B \in \text{For}^+$.

In the first case, we get $A^* \in \Pi_1^1(\Sigma_1^1\text{-AC}_0)$ from (d). For the second case, recall that $(t \in \{z : B\})^*$ equals $\exists x(V_t^*(x) \wedge B^*(x/z))$ and note that by the induction hypothesis for $(*)$ with $B(x/z)$, we get $A_0(U, x) \in \Pi_0^1$ for some set variable U such that $B^*(x/z)$ is equivalent to $\forall X A_0(X, x)$ over $\Sigma_1^1\text{-AC}_0$. Hence $(t \in \{z : B\})^*$ is equivalent to $\exists x \forall X (V_t^*(x) \wedge A_0(X, x))$.

Now it remains only to show that $\exists x \forall X (V_t^*(x) \wedge A_0(X, x))$ is equivalent to $\forall X A'_0(X)$ over $\Sigma_1^1\text{-AC}_0$ for $A'_0(W) := \exists x (V_t^*(x) \wedge A_0((W)_x, x))$, while noting $\forall X A'_0(X) \in \Pi_1^1$. One direction of the equivalence holds already over ACA_0 : In order to show $A_0((X)_y, y)$ for some y for any given set X , take x that is given in $\exists x \forall X (V_t^*(x) \wedge A_0(X, x))$. Then use (ACA) to get Z such that $z \in Z \leftrightarrow z \in (X)_x$ holds, then we get $V_t^*(x) \wedge A_0(Z, x)$, i.e., $V_t^*(x) \wedge A_0((X)_x, x)$. For the converse direction, we can work with the contraposition and apply $(\Sigma_1^1\text{-AC})$.

3. Otherwise: The remaining cases follow from standard arguments involving $\Sigma_1^1\text{-AC}_0$, the induction hypothesis, and Proposition 7.14. We refer to [18, Lemma 7.41] for details.

For (f): We prove this for $\mathbb{F} \in \text{FT}$ with $\mathbb{F} = \mathbb{P}_0 \rightarrow \dots \rightarrow \mathbb{P}_n$ by induction on $n \in \mathbb{N}$: If $n = 0$ holds, then we have $\mathbb{F} \in \text{Ty}$ and $(t \in \mathbb{F}) \in \text{For}^+$, so we can use (e). If $n > 0$ holds, then let $\mathbb{F}' := \mathbb{P}_1 \rightarrow \dots \rightarrow \mathbb{P}_n$. Now, $(t \in \mathbb{F})^*$ translates to $\forall x((x \in \mathbb{P})^* \rightarrow (tx \in \mathbb{F}')^*)$. By (e) and the induction hypothesis, we get $(x \in \mathbb{P})^* \in \Pi_1^1(\Sigma_1^1\text{-AC}_0)$ and $(tx \in \mathbb{F}')^* \in \Pi_2^1(\Sigma_1^1\text{-AC}_0)$. By Corollary 7.15, we get $(t \in \mathbb{F})^* \in \Pi_2^1(\Sigma_1^1\text{-AC}_0)$. \square

Theorem 7.20 $\Pi_3^1\text{-RFN}_0$ proves A^* for every instance A of (FT-ID).

Proof Let A be an instance of (FT-ID), say $\text{Cl}_{\mathbb{P},\mathbb{Q}}(\Lambda z.tz \in \mathbb{F}) \rightarrow t \in (\mathbb{I}_{\mathbb{P},\mathbb{Q}} \rightarrow \mathbb{F})$ with $\mathbb{F} \in \text{FT}$. Similar as in $(\star\text{-Cl}_{\mathbb{P},\mathbb{Q}})$ on page 30, we have with $\mathcal{A} := \Lambda z.(tz \in \mathbb{F})^*$ that A^* translates to $\forall x(\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(\mathcal{A}, x) \rightarrow \mathcal{A}(x)) \rightarrow \forall x((x \in \mathbb{I}_{\mathbb{P},\mathbb{Q}})^* \rightarrow \mathcal{A}(x))$ and therefore we assume (with a slight renaming of bound variables to make the following more readable) that $\forall y(\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(\mathcal{A}, y) \rightarrow \mathcal{A}(y))$ holds. Due to Lemma 7.19(f), we know that a formula $B \in \Pi_2^1$ exists such that $\Sigma_1^1\text{-AC}_0 \vdash B \leftrightarrow \mathcal{A}(y)$ holds. For $\mathcal{B} := \Lambda y.B$, we get from Corollary 7.15 and Lemma 7.19(c) a formula $C \in \Pi_3^1$ such that

$$\text{ACA}_0 \vdash C \leftrightarrow \forall y(\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(\mathcal{B}, y) \rightarrow \mathcal{B}(y)) \tag{12}$$

holds. Note that this holds over ACA_0 since we work with $B \in \Pi_2^1$ instead of $(ty \in \mathbb{F})^*$. Moreover, we have over $\Sigma_1^1\text{-AC}_0$ that our assumption $\forall y(\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(\mathcal{A}, y) \rightarrow \mathcal{A}(y))$ is equivalent to $\forall y(\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(\mathcal{B}, y) \rightarrow \mathcal{B}(y))$ and we proceed now by assuming that the conclusion in A^* is false and will derive a contradiction from this. So, let a_0 be such that

$$(x \in \mathbb{I}_{\mathbb{P},\mathbb{Q}})^*(a_0/x) \wedge \neg \mathcal{A}(a_0) \tag{13}$$

holds and note that the formula $\neg \mathcal{A}(a_0)$ (which is $\neg(tz \in \mathbb{F})^*(a_0/z)$) is equivalent over $\Sigma_1^1\text{-AC}_0$ to $\neg B(a_0/y)$. Note that $\neg B(a_0/y)$ is equivalent to a Π_3^1 formula, and since we have $C \in \Pi_3^1$, there exists by Proposition 7.14 some $D \in \Pi_3^1$ that is provably equivalent over ACA_0 to $C \wedge \neg B(a_0/y)$. Then due to Corollary 7.9, we can work with $\Pi_3^1\text{-RFN}_0$ to apply $(\Pi_3^1\text{-RFN})$ to D and thus obtain an ω -model M of ACA_0 such that $\neg B^M(a_0/y)$ and $\forall y(\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(\mathcal{B}, y)^M \rightarrow \mathcal{B}(y)^M)$ hold. For the latter, the relativization to M holds essentially because of the equivalence in (12) being provable over ACA_0 . Further, this formula unfolds by Definition 7.1 and the build-up of $\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(\mathcal{B}, y)$ to

$$\forall y \left(\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(\Lambda y.B^M, y) \rightarrow B^M \right) \tag{14}$$

Since B^M is arithmetical, (ACA) provides a set X_0 such that we have

$$\forall y(y \in X_0 \leftrightarrow B^M) \wedge \forall y \left(\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(X_0, y) \rightarrow y \in X_0 \right) \tag{15}$$

Now, after recalling $(\star\text{-I}_{\mathbb{P},\mathbb{Q}})$ on page 30, we can instantiate $(x \in \mathbb{I}_{\mathbb{P},\mathbb{Q}})^*(a_0/x)$ from (13) with X_0 and then use (15) to obtain $(x \in X_0)^*(a_0/x)$, i.e., $\exists z(\mathbb{V}_x^*(z) \wedge z \in X_0)(a_0/x)$. This formula is equivalent to $a_0 \in X_0$ since $\mathbb{V}_x^*(z)$ is just $x = z$. But then we get

$B^M(a_0/y)$ by (15) which is a contradiction to $\neg B^M(a_0/y)$ that we previously obtained. So, we have proven the lemma. \square

Remark 7.21 In the previous proof, we consider (14) as the pivotal property for the used proof method because it allowed to internalize an argument within the ω -model M . In particular, we needed that the positive operator form $\text{Acc}_{\mathbb{P},\mathbb{Q}}^*$ has the property described by Lemma 7.19(c) with T being ACA_0 . A conceptually similar proof in the setting of $\Pi_2^1\text{-RFN}_0$ and using similar standard results from the area of subsystems of second order arithmetic can be found in [1], treating the embedding of the theory ID_1^* of positive induction into $\Pi_2^1\text{-RFN}_0$.

Now, turning to the question if our proof method would also work for arbitrary positive operator forms \mathfrak{A} , we point out that a direct embedding of TID into $\Pi_3^1\text{-RFN}_0$ can be carried out almost literally as the embedding of FIT into $\Pi_3^1\text{-RFN}_0$. More precisely, the previous lemmas can be reformulated in a very similar way so that they work for TID as well. The pivotal property to make the proof work would again correspond to (14) and essentially because Acc in the setting of TID has a similar bounded complexity as $\text{Acc}_{\mathbb{P},\mathbb{Q}}^*$ here. The latter means that for (12) in the proof of Theorem 7.20, we used that we had the property $\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(\Lambda y.B, x) \in \Pi_2^1(\text{ACA}_0)$ at hand for $B \in \Pi_2^1$, namely as provided by Lemma 7.19(c).

Continuing from the perspective of TID, we shall consider for a moment its natural generalization TID^f (where **f** stands for *full*) that allows for arbitrary arithmetical operator forms \mathfrak{A} . So, having \mathfrak{A} instead of $\text{Acc}_{\mathbb{P},\mathbb{Q}}^*$ or Acc in (12), it would not always be possible to obtain a property such as $\mathfrak{A}(B, x) \in \Pi_2^1(\text{ACA}_0)$, nor can we expect that $G' \in \Pi_2^1$ exists that is equivalent over $\Sigma_1^1\text{-AC}_0$ or $\Pi_3^1\text{-RFN}_0$ to $\mathfrak{A}(B, x)$. Comparing this with the mentioned embedding of ID_1^* into $\Pi_2^1\text{-RFN}_0$ from [1], we note that essentially only Π_1^1 formulas B needed to be considered there, and since a formula such as $\mathfrak{A}(\Lambda z.B, t)$ can be proven to be equivalent over $\Sigma_1^1\text{-AC}_0$ to a Π_1^1 formula G' , one can continue the proof with this G' .

For an embedding of TID^f into $\Pi_3^1\text{-RFN}_0$ where we cannot control anymore the syntactical complexity of the positive operator forms \mathfrak{A} , we apparently cannot directly apply the method of this section. As described in the conclusion of this article (see Sect. 8), we remark here that the desired upper bound for TID^f can be obtained by turning to the setting of set-theory.

Theorem 7.22 *Over $\Pi_3^1\text{-RFN}_0$, the following holds.*

- (a) A^* holds for every formula A from axiom group **I**. of FIT.
- (b) A^* holds for every instance A of the N-induction scheme (FT-Ind) of FIT.
- (c) A^* holds for every instance A of the comprehension scheme (CA^+) of FIT.
- (d) A^* holds for every instance A of the closure axiom (FT-CI) of FIT.

Proof For (a): Note that according to Definition 7.16, the type N has no special role in the translation A^* of any of the formulas A given in the axiom group **I**. of FIT. As mentioned in Definition 7.16, we assume a standard interpretation of the constants k and s with the properties that we need for such a translation to be adequate. It is well-known that the combinators are available as partial recursive functions in the sense given here. Moreover, it is also more or less obvious that the interpretation

of the remaining constants has the properties needed to make the translation of the remaining formulas in axiom group **I**. go through.

For (b): Over ACA_0 , we have that $(\Pi_3^1\text{-RFN})$ implies transfinite induction for Π_2^1 formulas, and thus complete induction along the natural numbers for Π_2^1 formulas. For this, see in [23] Theorem VIII.5.12 and in particular Exercise VIII.5.15, while noting there that $\Sigma_4^1\text{-RFN}_0$ is equivalent to $\Pi_3^1\text{-RFN}_0$. Now, let A be an instance $t0 \in \mathbb{F} \wedge (\forall x \in \mathbb{N})(tx \in \mathbb{F} \rightarrow tx' \in \mathbb{F}) \rightarrow t \in (\mathbb{N} \rightarrow \mathbb{F})$ of the N-induction scheme (FT-Ind) of FIT, where $\mathbb{F} \in \text{FT}$ holds. By setting $\mathcal{B} := \Lambda z.(tz \in \mathbb{F})^*$, we have that A^* is equivalent over ACA_0 to $\mathcal{B}(0) \wedge \forall x(\mathcal{B}(x) \rightarrow \mathcal{B}(x+1)) \rightarrow \forall x(\exists y(V_x^*(y)) \rightarrow \mathcal{B}(x))$ since $\mathcal{B}(x+1)$ is equivalent to $(t(s_{\mathbb{N}}x) \in \mathbb{F})^*$. For the latter, note that $(t(s_{\mathbb{N}}x) \in \mathbb{F})^*$ is $\exists y(V_{t(s_{\mathbb{N}}x)}^*(y) \wedge (y \in \mathbb{F})^*)$ and that this is equivalent to $\exists y, z_1, z_2(V_t^*(z_1) \wedge \{s_{\mathbb{N}}\}(x) \simeq z_2 \wedge \{z_1\}(z_2) \simeq y \wedge (y \in \mathbb{F})^*)$ which again simplifies to $\exists y, z_1(V_t^*(z_1) \wedge \{z_1\}(x+1) \simeq y \wedge (y \in \mathbb{F})^*)$ and this is equivalent to $\mathcal{B}(x+1)$. Now arguing over $\Pi_3^1\text{-RFN}_0$, we have that $\mathcal{B}(0) \wedge \forall x(\mathcal{B}(x) \rightarrow \mathcal{B}(x+1)) \rightarrow \forall x(\exists y(V_x^*(y)) \rightarrow \mathcal{B}(x))$ is equivalent to an instance of complete induction along the natural numbers for a Π_2^1 formula (use Lemma 7.19(f)) and hence we are done.

For (c): If A is an instance $y \in \{x: B\} \leftrightarrow B(y/x)$ of (CA^+) with $B \in \text{For}^+$, then A^* yields $\exists x(V_x^*(x) \wedge B^*) \leftrightarrow (B(y/x))^*$, being equivalent to the tautology $(B(y/x))^* \leftrightarrow (B(y/x))^*$.

For (d): Let $A := \text{Cl}_{\mathbb{P},\mathbb{Q}}(\Lambda z.z \in \text{I}_{\mathbb{P},\mathbb{Q}})$ be an instance of (FT-Cl). Following $(\star\text{-Cl}_{\mathbb{P},\mathbb{Q}})$ on page 30 and in order to show A^* , assume for $\mathcal{A} := \Lambda z.(z \in \text{I}_{\mathbb{P},\mathbb{Q}})^*$ that we have $\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(\mathcal{A}, z_0)$ for some z_0 , and we aim to prove $\mathcal{A}(z_0)$, i.e., $(\forall X(\forall x(\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(X, x) \rightarrow x \in X) \rightarrow (z \in X)^*))(z_0/z)$ and in order to prove this, we shall prove $z_0 \in X_0$, assuming X_0 such that $\forall x(\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(X_0, x) \rightarrow x \in X_0)$ holds. From the latter and the definition of $(z \in \text{I}_{\mathbb{P},\mathbb{Q}})^*$, we get $\forall z((z \in \text{I}_{\mathbb{P},\mathbb{Q}})^* \rightarrow z \in X_0)$, i.e., we have $\forall z(\mathcal{A}(z) \rightarrow z \in X_0)$. With this, we get $\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(\mathcal{A}, z_0) \rightarrow \text{Acc}_{\mathbb{P},\mathbb{Q}}^*(X_0, z_0)$ through the *positivity* of the operator form $\text{Acc}_{\mathbb{P},\mathbb{Q}}^*$. Now, $\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(X_0, z_0)$ follows from assumption $\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(\mathcal{A}, z_0)$ and then the remaining goal $z_0 \in X_0$ follows from assumption $\forall x(\text{Acc}_{\mathbb{P},\mathbb{Q}}^*(X_0, x) \rightarrow x \in X_0)$. \square

Corollary 7.23 $\text{FIT} \vdash A$ implies $\Pi_3^1\text{-RFN}_0 \vdash A^*$ for all $A \in \mathcal{L}_{\text{FIT}}$.

Proof By Theorems 7.20 and 7.22, one only has to show that the logical part of FIT embeds into $\Pi_3^1\text{-RFN}_0$, i.e., that $\text{LPT} \vdash A$ implies $\Pi_3^1\text{-RFN}_0 \vdash A^*$. The proof is by induction on the definition of the derivability notion $\text{LPT} \vdash A$ that may have been chosen in Definition 3.9 with respect to any sound Hilbert calculus. See [18, Corollary 7.45] for details. \square

8 Conclusion and further remarks

Recapitulating the results of this article, we obtained theories FIT and TID that both have the small Veblen ordinal $\vartheta\Omega^\omega$ as their proof-theoretic ordinal, while FIT is a natural extension of Feferman’s two-sorted theory $\text{QL}(F_0\text{-IR}_N)$ from [10] and TID becomes from this perspective a natural subsystem of ID_1 . Moreover, we used techniques from the realm of predicative proof-theory in order to obtain a wellordering proof for TID (and hence for FIT).

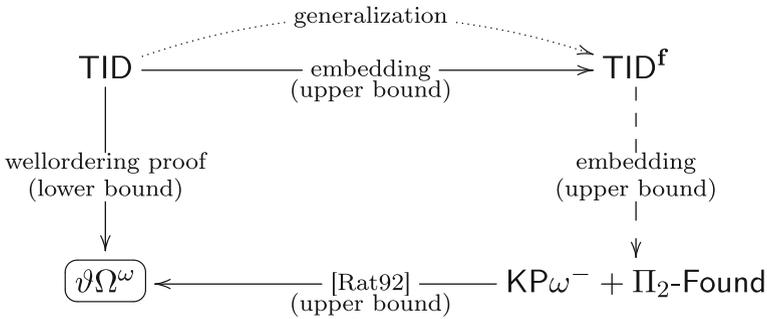


Fig. 2 Generalization of TID to the full theory TID^f

With regard to the upper-bound results, we embedded FIT into the subsystem $\Pi_3^1\text{-RFN}_0$ of second order arithmetic, while exploiting the Π_1^1 definability of a least fixed point in such a setting. This method can be used almost literally for embedding TID directly into $\Pi_3^1\text{-RFN}_0$ (see [18, Chapter 9] for this). This approach seems to fail though if we extend TID to a theory TID^f for general typed inductive definitions with the *full* range of positive arithmetical operator forms (as described in Remark 7.21). A way to avoid this obstacle is to shift the setting to set-theory rather than subsystems of second order arithmetic, namely by exploiting the Σ_1 definability of a least fixed point. Working then in $KP\omega^- + \Pi_2\text{-Found}$ from [19] (i.e., Kripke–Platek set-theory with a restricted axiom scheme for foundation) shall suffice to get an analog result as for FIT which we can apply to the theory TID^f . Summing up, what we gain from these embeddings is that extending TID to the theory TID^f retains the proof-theoretic upper bound $\vartheta\Omega^\omega$ because $KP\omega^- + \Pi_2\text{-Found}$ has by [19] the same proof-theoretic strength as $\Pi_2^1\text{-Bl}_0$. Since TID trivially embeds into TID^f and as depicted in Fig. 2, we get that TID^f has the same proof-theoretic strength as TID (in a similar way as ID_1 and its restriction $ID_1(\text{Acc})$ to accessible part arithmetical operator forms correspond, see also [3]).

We finish with some remarks and conjectures on how to extend the proof-methods from Sects. 6 and 7 in order to analyze stronger systems: First, the collections of formulas Pos_0 and Pos_1 suggest already a generalization to collections Pos_n for any $n \geq 2$ in the sense that the correspondence of Pos_1 to *function* types of the form $\mathbb{P}_1 \rightarrow \dots \rightarrow \mathbb{P}_k$ for each $k \in \mathbb{N}$ (i.e., “level-1-functional types”) passes over to Pos_2 being in correspondence to *level-2-functional* types $\mathbb{F}_1 \rightarrow \dots \rightarrow \mathbb{F}_k$ for each $k \in \mathbb{N}$, and similar for $n > 2$. Such theories are essentially considered in [18] and named TID_n . Moreover, intermediate systems TID_n^+ are considered that are located between TID_n and TID_{n+1} , and which arise essentially by adding a rule of inference (TID^+) of the form

$$(TID^+) \frac{P_{\triangleleft} t}{\text{Prog}_{\triangleleft}(\mathcal{B}) \rightarrow \mathcal{B}(t)}$$

for arbitrary $\mathcal{B} \in \mathcal{L}_{TID}$ and for each binary relation symbol $\triangleleft \in \mathcal{L}_{PA}$ and each term t . As shown in [18], TID_1^+ has the strength of the *large Veblen ordinal* $\vartheta\Omega^\Omega$ and (TID^+)

corresponds to a variant of a *bar rule*. This and the results of [17] suggest that by adding the usual bar rule (BR) to the theory $\Pi_3^1\text{-RFN}_0$ with strength of the small Veblen ordinal $\vartheta^{\Omega^\omega}$, we obtain a theory $\Pi_3^1\text{-RFN}_0 + (\text{BR})$ with strength of the large Veblen ordinal $\vartheta^{\Omega^\Omega}$.

Concerning the wellordering proof for TID_1^+ in [18, Chapter 11], we remark here that it is based on the methods from Sect. 6 by using *Klammersymbols* instead of finitary Veblen functions and by *internalizing* the arity of the finitary Veblen functions within the theory. More precisely and as an informal and intuitive explanation why Corollary 6.15 is the best we can expect from TID, we mention that the method used in the proof of Theorem 6.12 relied on an *external* representation of the finite list of arguments that the finitary Veblen function is applied to. In particular, induction in the meta-theory has been applied to cope with arbitrary but finite lists of arguments. The proof of Theorem 6.12 is designed for the theory TID, and in order to use it to get beyond the small Veblen ordinal, for instance by working with infinitary Veblen functions or Klammersymbols, the first step is to *internalize the proof* and deal with non-standard argument positions (for which we do not have a denotation in the meta-theory).

For more details and further remarks on the other theories, we refer to [18]. The use of a higher-type functional for iterating the fixed-point construction on Klammersymbols allows to extend the ordinal notation system. Endowed with stronger induction principles that become available in the theories TID_n and TID_n^+ , this may lead towards higher ordinals via wellordering proofs based on the accessible part of the primitive recursive ordering of the new ordinal notation system.

A Appendix: The ordinal notation system (OT, <)

For carrying out the wellordering proofs in TID, we fix a primitive recursive notation system (OT, <) for ordinals below the small Veblen ordinal, based on codes of natural numbers. For a full definition and proofs, see [18, Chapter 3]. (OT, <) is based on Lemma 5.19 (i.e., essentially on (7.1)–(7.4) in [21]) and its representation is inspired by [6].

We point out that the definition and properties of (OT, <) can be formalized and established within PA and hence within TID. Due to constraints on the length of this article, we hide these things and establish here only the minimal technical information needed in order to be able to carry out the wellordering proofs in Sect. 6 in a formally satisfying way. In particular, we provide Sect. 5 as the set-theoretic intuition for those proofs and we refer to [18] for the full definition of (OT, <), i.e., its formalization within \mathcal{L}_{PA} , the (technically cumbersome) encoding and verification within PA of crucial primitive recursive properties (e.g., that (OT, <) is a strict total order), and the set-theoretic interpretation of (OT, <) by means of a function $\circ: \text{OT} \rightarrow \text{On}$ as described in Sect. 5.2.

This appendix contains the formal definition and central properties of *fundamental sequences* for ordinal notations because the proofs of Sect. 6 are centrally based on these. In order to be able to parse this definition, we stress that OT consists of the following codes for ordinals:

- 0 for the zero ordinal;
- $\phi\bar{a}^{(n+1)}$ denoting a code for the intended application of the finitary Veblen function φ to given $a_1, \dots, a_{n+1} \in \text{OT}$ with $a_1 \neq 0$ and where a_1, \dots, a_{n+1} suffice certain (normal-form) conditions that are motivated by Lemma 5.19;
- $a \oplus b$ denoting a code for the intended sum of ordinals in Cantor normal form, where essentially a is of the form $\phi\bar{a}^{(n+1)}$ and $b \neq 0$.

Suc and Lim denote primitive recursive sets of codes for successor ordinals and limit ordinals, respectively. As already mentioned in Sect. 5.2, ordinal arithmetic is simulated on such ordinal notations by making use of primitive recursive operations: We have $\dot{+}: \text{OT} \times \text{OT} \rightarrow \text{OT}$ and $\tilde{\varphi}_{n+1}: \text{OT}^{n+1} \rightarrow \text{OT}$ that correspond to $+_{\text{On}}: \text{On} \times \text{On} \rightarrow \text{On}$ and $\varphi_{n+1}: \text{On}^{n+1} \rightarrow \text{On}$ from Sect. 5. We just write $\tilde{\varphi}$ instead of $\tilde{\varphi}_{n+1}$ if the arity is clear from the context. Further, let $\tilde{1} := \phi 0$ and $\tilde{\omega}^a := \tilde{\varphi}(a)$ for each $a \in \text{OT}$. Also, we have $\tilde{\cdot}: \text{OT} \times \mathbb{N} \rightarrow \text{OT}$ to denote finite addition (encoded via \oplus) of the same ordinal notation, i.e., $a \tilde{\cdot} 0 := 0$ and $a \tilde{\cdot} (x +_{\mathbb{N}} 1) := a \oplus (a \tilde{\cdot} x)$ for all $a \in \text{OT}$ and $x \in \mathbb{N}$.

Definition A.1 *Fundamental sequences* for ordinal notations $d \in \text{OT}$ are defined within PA by means of a binary primitive recursive function L whose defining equations are described as follows, where a, x range over natural numbers and where we write $a[x]$ to denote $L(a, x)$.

- If $d = 0$ or $d \notin \text{OT}$, then $d[x] := 0$
- If $d \in \text{Suc}$ with $d = d_0 \dot{+} \tilde{1}$, then $d[x] := d_0$
- If $d \in \text{Lim}$ and $d = a \oplus b$ with $a \in \text{OT}$ and $b \in \text{Lim}$, then $d[x] := a \dot{+} b[x]$
- If $d \in \text{Lim}$ and $d = \phi a$ with $a \neq 0$, then $d[x] := \begin{cases} \tilde{\omega}^{a_0} \tilde{\cdot} (x +_{\mathbb{N}} 1) & \text{if } a = a_0 \dot{+} \tilde{1} \\ \tilde{\omega}^{a[x]} & \text{otherwise} \end{cases}$
- If $d \in \text{Lim}$ with $d = \phi\bar{a}^{(m)}b\bar{0}^{(k)}c$ for some $\bar{a}^{(m)}, b, c \in \text{OT}$ with $b \neq 0$ and $m, k \in \mathbb{N}$, then

$$d[0] := \begin{cases} \tilde{\varphi}(\bar{a}^{(m)}, b, \bar{0}^{(k)}, c[0]) & \text{if } c \in \text{Lim} \\ \tilde{\varphi}(\bar{a}^{(m)}, b[0], \bar{0}^{(k+1)}) & \text{if } c = 0 \text{ and } b \in \text{Lim} \\ \tilde{1} & \text{if } c = 0 \text{ and } b \in \text{Suc} \\ \tilde{\varphi}(\bar{a}^{(m)}, b, \bar{0}^{(k)}, c[0]) \dot{+} \tilde{1} & \text{otherwise (if } c \in \text{Suc)} \end{cases}$$

$$d[x +_{\mathbb{N}} 1] := \begin{cases} \tilde{\varphi}(\bar{a}^{(m)}, b, \bar{0}^{(k)}, c[x +_{\mathbb{N}} 1]) & \text{if } c \in \text{Lim} \\ \tilde{\varphi}(\bar{a}^{(m)}, b[x +_{\mathbb{N}} 1], \bar{0}^{(k+1)}) & \text{if } c = 0 \text{ and } b \in \text{Lim} \\ \tilde{\varphi}(\bar{a}^{(m)}, b[x], d[x], \bar{0}^{(k)}) & \text{otherwise (if } c \in \text{Suc or } (c = 0 \text{ and } b \in \text{Suc})) \end{cases}$$

References

1. Afshari, B., Rathjen, M.: A note on the theory of positive induction, ID₁^{*}. Arch. Math. Logic **49**, 275–281 (2010). <https://doi.org/10.1007/s00153-009-0168-9>
2. Beeson, M.J.: Foundations of Constructive Mathematics: Metamathematical studies. Springer, Berlin (1985)
3. Buchholz, W., Feferman, S., Pohlers, W., Sieg, W.: Iterated Inductive Definitions and Subsystems of Analysis: Recent Proof-Theoretical Studies, volume 897 of Lecture Notes in Mathematics. Springer, Berlin (1981)

4. Bridge, J.: A simplification of the Bachmann method for generating large countable ordinals. *J. Symb. Logic* **40**(2), 171–185 (1975)
5. Buchholz, W., Schütte, K.: *Proof Theory of Impredicative Subsystems of Analysis*, volume 2 of *Studies in Proof Theory Monographs*. Bibliopolis, Napoli (1988)
6. Buchholz, W.: *Prädikative Beweistheorie (Predicative proof theory)*. Lecture notes, University of Munich (2004–2005)
7. Buchholz, W.: A survey on ordinal notations around the Bachmann–Howard ordinal. In: Kahle, R., Strahm, T., Studer, T. (eds.) *Advances in Proof Theory*, pp. 1–29. Birkhaeuser, Springer, Basel (2016)
8. Cantini, A.: On the relation between choice and comprehension principles in second order arithmetic. *J. Symb. Logic* **51**(2), 360–373 (1986)
9. Feferman, S.: Constructive theories of functions and classes. In: van Dalen, D., Boffa, M., Mcaloon, K. (eds.) *Logic Colloquium '78 Proceedings of the colloquium held in Mons*, volume 97 of *Studies in Logic and the Foundations of Mathematics*, pp. 159–224. Elsevier (1979)
10. Feferman, S.: Logics for termination and correctness of functional programs, II. Logics of strength PRA. In: Aczel, P., Simmons, H., Wainer, S.S. (eds.) *Proof Theory*, pp. 195–225. Cambridge University Press, Cambridge (1992)
11. Feferman, S., Jäger, G., Strahm, T.: *Foundations of Explicit Mathematics (in preparation)*
12. Jäger, G.: Metapredicative and explicit Mahlo: a proof-theoretic perspective. In: Cori, R., Razborov, A., Todorcevic, S., Wood, C. (eds.) *Proceedings of Logic Colloquium '00*, volume 19 of *Association of Symbolic Logic Lecture Notes in Logic*, pp. 272–293. AK Peters (2005)
13. Jäger, G., Kahle, R., Setzer, A., Strahm, T.: The proof-theoretic analysis of transfinitely iterated fixed point theories. *J. Symb. Logic* **64**(1), 53–67 (1999)
14. Jäger, G., Strahm, T.: Bar induction and ω model reflection. *Ann. Pure Appl. Logic* **97**(1–3), 221–230 (1999)
15. Jäger, G., Strahm, T.: Reflections on reflections in explicit mathematics. *Ann. Pure Appl. Logic* **136**(1–2), 116–133 (2005) (**Festschrift on the occasion of Wolfram Pohlers' 60th birthday**)
16. Probst, D.: The proof-theoretic analysis of transfinitely iterated quasi least fixed points. *J. Symb. Logic* **71**(3), 721–746 (2006)
17. Probst, D.: *Modular Ordinal Analysis of Subsystems of Second-Order Arithmetic of Strength up to the Bachmann–Howard Ordinal*. Habilitation, Universität Bern (2017)
18. Ranzi, F.: From a flexible type system to metapredicative wellordering proofs. PhD thesis, Universität Bern. <http://www.iam.unibe.ch/ltgpub/2015/ran15.pdf> (2015). <https://doi.org/10.7892/boris.75102>
19. Rathjen, M.: Fragments of Kripke–Platek set theory with infinity. In: Aczel, P., Simmons, H., Wainer, S.S. (eds.) *Proof Theory. A Selection of Papers from the Leeds Proof Theory Programme 1990*, pp. 251–273. Cambridge University Press, Cambridge (1992)
20. Rathjen, M., Weiermann, A.: Proof-theoretic investigations on Kruskal's theorem. *Ann. Pure Appl. Logic* **60**, 49–88 (1993)
21. Schütte, K.: Kennzeichnung von Ordnungszahlen durch rekursiv erklärte Funktionen. *Math. Ann.* **127**, 15–32 (1954). <https://doi.org/10.1007/BF01361109>
22. Schütte, K.: Beziehungen des Ordinalzahlensystems $OT(\vartheta)$ zur Veblen-Hierarchie. Unpublished notes (1992)
23. Simpson, S.G.: *Subsystems of Second Order Arithmetic*, 2nd edn. Cambridge University Press, Cambridge (2009) (**Cambridge Books Online**)
24. Strahm, T.: First steps into metapredicativity in explicit mathematics. In: Barry Cooper, S., Truss, J.K. (eds.) *Sets and Proofs*, volume 258 of *London Mathematical Society Lecture Notes*, pp. 383–402. Cambridge University Press, Cambridge (1999)
25. Van der Meeren, J., Rathjen, M., Weiermann, A.: An order-theoretic characterization of the Howard–Bachmann-hierarchy. *Arch. Math. Logic* **56**(1–2), 79–118 (2017)