Elementary explicit types and polynomial time operations

Daria Spescha* Thomas Strahm**

Version of May 2008

Abstract

This paper studies systems of explicit mathematics as introduced by Feferman [9, 11]. In particular, we propose weak explicit type systems with a restricted form of elementary comprehension whose provably terminating operations coincide with the functions on binary words that are computable in polynomial time. The systems considered are natural extensions of the first-order applicative theories introduced in Strahm [19, 20].

Keywords: Proof theory, Feferman's explicit mathematics, applicative theories, types and names, feasible operations

1 Introduction

Explicit mathematics was introduced by Feferman [9, 10, 11] in the early seventies. Beyond its original aim to provide a basis for Bishop-style constructivism, the explicit framework has gained considerable importance in proof theory in connection with the proof-theoretic analysis of subsystems of second order arithmetic and set theory as well as for studying the proof theory of abstract computations.

There are two basic kinds of objects which are present in explicit mathematics, namely *operations* and *types*. The former may be thought of as

*Institut für Informatik und angewandte Mathematik, Universität Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland. Email: `spescha@iam.unibe.ch`. Homepage: `http://www.iam.unibe.ch/~spescha`. Research supported by the Swiss National Science Foundation.

**Institut für Informatik und angewandte Mathematik, Universität Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland. Email: `strahm@iam.unibe.ch`. Homepage: `http://www.iam.unibe.ch/~strahm`.

mechanical rules of computation, which can freely be applied to each other: self-application is meaningful, though not necessarily total. The basic axioms concerning operations are those of a partial combinatory algebra, thus giving immediate rise to explicit definitions (lambda abstraction) and a form of the recursion theorem. The standard interpretation of the operations is the domain of the partial recursive functions.

Types, on the other hand, are collections of operations and must be thought of as being generated successively from preceding ones. In contrast to the restricted character of operations, types can have quite complicated defining properties. What is essential in the whole explicit mathematics approach, however, is the fact that types are again represented by operations or, as we will call them in this case, *names*. Thus each type U is named or represented by a name u; in general, U may have many different names or representations. It is exactly this interplay between operations and types on the level of names which makes explicit mathematics extremely powerful and, in fact, witnesses its explicit character.

In previous proof-theoretic investigations on full systems of explicit mathematics, the focus has mainly been on theories ranging in strength from primitive recursive arithmetic PRA to highly impredicative systems. Weaker systems have only been studied for the operational core of explicit mathematics, so-called applicative theories. A first system PTO related to polynomial time computability was introduced in Strahm [18] and extended in Cantini [4]. Later, an improved version of PTO called PT was proposed in Strahm [19, 20] together with additional theories for other complexity classes. Cantini [6] has considered extensions of PT which we will address below. Finally in [5], he has also examined tiered applicative frameworks in the spirit of implicit computational complexity.

In this paper, we start off from Strahm's PT, which we summarise in Section 2, and extend it with types and names. To this end, we will propose and analyse a weak form of the well-known schema of elementary comprehension which is present in most systems of explicit mathematics. In particular, we will use the neat formulation due to Jäger [15] employing a naming relation $\Re$ between individuals and types. In our approach we will design a theory PET which features a natural restriction of the finite axiomatisation of elementary comprehension provided by Feferman and Jäger [12] together with type induction on the binary words W. More precisely, our restriction excludes complement types and replaces the type W of binary words by types for initial segments $\{x \in \mathsf{W} : x \leq a\}$ for each $a \in \mathsf{W}$, where $\leq$ denotes the less-than-or-equal relation with respect to the length of words.

In order to establish that PET proves the totality of the polynomial time computable functions, we will embed a natural restriction PT^- of PT which is strong enough to represent bounded recursion on notation in the form of a type two functional. This embedding is not completely straightforward and will use a bootstrapping functional mapping each operation f on W to an operation f^* such that $f^*x = \max_{y \subseteq x} fy$.

For the proof of the upper bounds we will start off from a model of PT^- and extend it to a model of PET satisfying the same first-order sentences. The construction is carried out in stages by defining the set of names and their extensions successively. Then one can show that the so-obtained model enjoys type induction.

Finally, we will propose some natural extensions of PET. They mostly rely on Cantini [6], who studies various extensions of the first-order theory PT by means of axioms for self-referential truth, a uniformity principle and a positive axiom of choice. In particular, we will describe a nice application of Cantini's uniformity principle which allows us to add a type generator for universal quantification. The full system PT can then be embedded into PET extended by this type constructor.

2 Recapitulating the first-order theory PT

In this section we will recapitulate the theory PT that was introduced and analysed in Strahm [19].

The applicative theory PT is formulated in the language $\mathcal{L}$; it is a language of partial terms with *individual variables* $a, b, c, x, y, z, u, v, f, g, h, \ldots$ (possibly with subscripts). $\mathcal{L}$ includes *individual constants* k, s (combinators), $\mathsf{p}, \mathsf{p}_0, \mathsf{p}_1$ (pairing and unpairing), d_W (definition by cases on binary words), ϵ (empty word), $\mathsf{s}_0, \mathsf{s}_1$ (binary successors), p_W (binary predecessor), $\mathsf{c}_\subseteq$ (initial subword relation), as well as the two constants $*$ (word concatenation) and $\times$ (word multiplication). Finally, $\mathcal{L}$ has a binary function symbol $\cdot$ for (partial) term application, unary relation symbols $\downarrow$ (defined) and W (binary words) as well as a binary relation symbol $=$ (equality).

The *terms* $r, s, t, \ldots$ of $\mathcal{L}$ (possibly with subscripts) are inductively generated from the variables and constants by means of application $\cdot$. In the following we often abbreviate $\cdot(s, t)$ simply as (st), st or sometimes also $s(t)$; the context will always ensure that no confusion arises. Further, we follow the standard convention of association to the left when omitting brackets in applicative terms. Finally, we will write $s{*}t$ and $s{\times}t$ instead of $*st$ and

$\times st$, respectively.

The *formulas* $A, B, C, \ldots$ of $\mathcal{L}$ (possibly with subscripts) are built from the atomic formulas $(s = t)$, $s\downarrow$ and $\mathsf{W}(s)$ by closing under negation, disjunction, conjunction, implication, as well as existential and universal quantification for individual variables.

Our conventions concerning substitutions are as follows. As usual we write $t[\vec{s}/\vec{x}]$ and $A[\vec{s}/\vec{x}]$ for the substitution of the terms $\vec{s}$ for the variables $\vec{x}$ in the term t and the formula A, respectively. In this connection we often write $A[\vec{x}]$ instead of A and $A[\vec{s}]$ instead of $A[\vec{s}/\vec{x}]$.

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

$$s \simeq t \;:=\; (s\downarrow \vee t\downarrow) \rightarrow (s = t).$$

We use the following list of abbreviations $(i = 0, 1)$:

$$\begin{array}{rclcrcl}
0 & := & \mathsf{s}_0\epsilon & \quad & 1 & := & \mathsf{s}_1\epsilon \\
(s_1, s_2) & := & \mathsf{p}st & & (s)_i & := & \mathsf{p}_i s \\
s \subseteq t & := & \mathsf{c}_{\subseteq}st = 0 & & \mathsf{l}_{\mathsf{W}}s & := & 1\times s \\
s \leq t & := & \mathsf{l}_{\mathsf{W}}s \subseteq \mathsf{l}_{\mathsf{W}}t & & & &
\end{array}$$

Furthermore, the following shorthand notations are used with respect to the predicate W ($\vec{s} = s_1, \ldots, s_n$):

$$\begin{array}{rcl}
\vec{s} \in \mathsf{W} & := & \mathsf{W}(s_1) \wedge \cdots \wedge \mathsf{W}(s_n), \\
\mathsf{W}_a(s) & := & (\mathsf{W}(s) \wedge s \leq a), \\
\vec{s} \in \mathsf{W}_a & := & \mathsf{W}_a(s_1) \wedge \cdots \wedge \mathsf{W}_a(s_n), \\
(\exists x \in \mathsf{W})A & := & (\exists x)(x \in \mathsf{W} \wedge A), \\
(\forall x \in \mathsf{W})A & := & (\forall x)(x \in \mathsf{W} \rightarrow A), \\
(\exists x \leq t)A & := & (\exists x \in \mathsf{W})(x \leq t \wedge A), \\
(\forall x \leq t)A & := & (\forall x \in \mathsf{W})(x \leq t \rightarrow A), \\
(t : \mathsf{W} \mapsto \mathsf{W}) & := & (\forall x \in \mathsf{W})(tx \in \mathsf{W}), \\
(t : \mathsf{W}^{m+1} \mapsto \mathsf{W}) & := & (\forall x \in \mathsf{W})(tx : \mathsf{W}^m \mapsto \mathsf{W}).
\end{array}$$

The underlying logic of PT is the *classical* logic of partial terms due to Beeson [1, 2]; it corresponds to E^+ logic with strictness and equality of Troelstra and Van Dalen [21]. According to this logic, quantifiers range over defined objects only:

(Q1) $\qquad \forall x A \wedge t\downarrow \rightarrow A[t]$

(Q2)	$A[t] \wedge t\downarrow \to \exists x A$
(D1)	$r\downarrow$ (r variable or individual constant)
(D2)	$(s \cdot t)\downarrow \to (s\downarrow \wedge t\downarrow)$
(D3)	$(s = t) \to (s\downarrow \wedge t\downarrow)$
(D4)	$\mathsf{W}(t) \to t\downarrow$
(E1)	$r = r$ (r variable or constant)
(E2)	$(s = t) \wedge A[s] \to A[t]$ (A atomic formula)

The basic theory of operations and binary words B was introduced in Strahm [19]. The axioms are given in the following groups I.-IV.:

I. Partial combinatory algebra and pairing

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

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

(3) $\mathsf{p}_0(x, y) = x \wedge \mathsf{p}_1(x, y) = y.$

II. Definition by cases on W

(4) $a \in \mathsf{W} \wedge b \in \mathsf{W} \wedge a = b \to \mathsf{d}_\mathsf{W}xyab = x,$

(5) $a \in \mathsf{W} \wedge b \in \mathsf{W} \wedge a \neq b \to \mathsf{d}_\mathsf{W}xyab = y.$

III. Closure, binary successors and predecessor

(6) $\epsilon \in \mathsf{W} \wedge (\forall x \in \mathsf{W})(\mathsf{s}_0 x \in \mathsf{W} \wedge \mathsf{s}_1 x \in \mathsf{W}),$

(7) $\mathsf{s}_0 x \neq \mathsf{s}_1 y \wedge \mathsf{s}_0 x \neq \epsilon \wedge \mathsf{s}_1 x \neq \epsilon,$

(8) $\mathsf{p}_\mathsf{W} : \mathsf{W} \mapsto \mathsf{W} \wedge \mathsf{p}_\mathsf{W}\epsilon = \epsilon,$

(9) $x \in \mathsf{W} \to \mathsf{p}_\mathsf{W}(\mathsf{s}_0 x) = x \wedge \mathsf{p}_\mathsf{W}(\mathsf{s}_1 x) = x,$

(10) $x \in \mathsf{W} \wedge x \neq \epsilon \to \mathsf{s}_0(\mathsf{p}_\mathsf{W} x) = x \vee \mathsf{s}_1(\mathsf{p}_\mathsf{W} x) = x.$

IV. Initial subword relation.

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

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

(13) $x \in \mathsf{W} \wedge y \in \mathsf{W} \wedge y \neq \epsilon \to (x \subseteq y \leftrightarrow x \subseteq \mathsf{p}_\mathsf{W} y \vee x = y).$

In addition, we have the following axioms about word concatenation and multiplication:

V. Word concatenation.

(14) $\quad * : \mathsf{W}^2 \mapsto \mathsf{W}$,

(15) $\quad x \in \mathsf{W} \ \rightarrow \ x{*}\epsilon = x$,

(16) $\quad x \in \mathsf{W} \wedge y \in \mathsf{W} \ \rightarrow \ x{*}(\mathsf{s}_i y) = \mathsf{s}_i(x{*}y) \quad (i = 0, 1)$.

VI. Word multiplication.

(17) $\quad \times : \mathsf{W}^2 \mapsto \mathsf{W}$,

(18) $\quad x \in \mathsf{W} \ \rightarrow \ x{\times}\epsilon = \epsilon$,

(19) $\quad x \in \mathsf{W} \wedge y \in \mathsf{W} \ \rightarrow \ x{\times}(\mathsf{s}_i y) = (x{\times}y){*}x \quad (i = 0, 1)$.

In the following we write $\mathsf{B}(*, \times)$ for the theory B augmented by the axioms in groups V. and VI.

Let us immediately turn to two crucial consequences of the partial combinatory algebra axioms (1) and (2) of B, namely *abstraction* and *recursion.* These two central results appear in slightly different form than in the setting of a total combinatory algebra, the essential ingredients in the proofs, however, are the same. The relevant arguments are given, for example, in Beeson [1] or Feferman [9].

Lemma 1 ($\boldsymbol{\lambda}$-Abstraction) *For each term t in $\mathcal{L}$ and every variable x there is an $\mathcal{L}$ term $(\lambda x.t)$ with the same variables as t except x such that* B *proves:*

$$(\lambda x.t)\!\downarrow \wedge\, (\lambda x.t)x \simeq t$$

Lemma 2 (Recursion) *There exists a closed $\mathcal{L}$ term* rec *so that* B *proves*

$$\mathsf{rec} f\!\downarrow \wedge\, \mathsf{rec} f x \simeq f(\mathsf{rec} f)x.$$

The theory PT is an extension of $\mathsf{B}(*, \times)$ by a suitable induction schema on W, so-called $(\Sigma^{\mathsf{b}}_{\mathsf{W}}\text{-}\mathsf{I}_{\mathsf{W}})$. Towards defining this induction principle, we need some preparatory definitions.

We call an $\mathcal{L}$ formula *positive* if it is built from atomic formulas by means of disjunction, conjunction as well as existential and universal quantification over individuals; i.e., the positive formulas are exactly the implication and negation free $\mathcal{L}$ formulas. We let Pos stand for the collection of positive formulas. Further, an $\mathcal{L}$ formula is called W *free* if the relation symbol W does not occur in it. Finally, the class of bounded existential formulas, $\Sigma^{\mathsf{b}}_{\mathsf{W}}$, can be introduced: a formula $A[f, x]$ belongs to $\Sigma^{\mathsf{b}}_{\mathsf{W}}$ if it is of the form $(\exists y \leq fx)B[f, x, y]$ where $B[f, x, y] \in \mathsf{Pos}$ and W-free. If, in addition, B does not contain $\forall$ quantifiers, A is called a $\Sigma^{\mathsf{b}-}_{\mathsf{W}}$ formula.

The theory PT is now defined as the theory $\mathsf{B}(*, \times)$ plus $(\Sigma^{\mathsf{b}}_{\mathsf{W}}\text{-}\mathsf{I}_{\mathsf{W}})$:

$$
(\Sigma^{\mathsf{b}}_{\mathsf{W}}\text{-}\mathsf{I}_{\mathsf{W}}) \qquad
\begin{array}{l}
f : \mathsf{W} \mapsto \mathsf{W} \wedge A[\epsilon] \wedge (\forall x \in \mathsf{W})(A[x] \rightarrow A[\mathsf{s}_0 x] \wedge A[\mathsf{s}_1 x]) \\
\qquad \rightarrow (\forall x \in \mathsf{W}) A[x] \\
\qquad \text{where } A[x] \equiv (\exists y \leq fx) B[f, x, y] \text{ a } \Sigma^{\mathsf{b}}_{\mathsf{W}}\text{-formula.}
\end{array}
$$

Later, we will also be interested in a natural subsystem of PT, called PT^-, where induction is only allowed for $\Sigma^{\mathsf{b}-}_{\mathsf{W}}$ formulas, i.e. PT^- is $\mathsf{B}(*, \times)$ plus $(\Sigma^{\mathsf{b}-}_{\mathsf{W}}\text{-}\mathsf{I}_{\mathsf{W}})$.

Let us quickly review the formal definition of the notion of a *provably total function* of a given theory T formulated in a language that contains $\mathcal{L}$. For that purpose, let $\mathbb{W} = \{0, 1\}^*$ denote the set of finite binary words. First note that for each word $w \in \mathbb{W}$ we have a canonical closed term $\overline{w}$ of $\mathcal{L}$ which represents w; of course, $\overline{w}$ is constructed from the empty word ϵ by means of the successor operations s_0 and s_1. In the sequel we sometimes identify $\overline{w}$ with w when working in the language $\mathcal{L}$.

Definition 3 A function $F : \mathbb{W}^n \rightarrow \mathbb{W}$ is called *provably total in an $\mathcal{L}$ theory* T, if there exists a closed $\mathcal{L}$ term t_F such that

1. $\mathsf{T} \vdash t_F : \mathsf{W}^n \mapsto \mathsf{W}$ and, in addition,
2. $\mathsf{T} \vdash t_F \overline{w}_1 \cdots \overline{w}_n = \overline{F(w_1, \ldots, w_n)}$ for all $w_1, \ldots, w_n$ in $\mathbb{W}$.

Let us write FPTIME for the class of polynomial time computable functions on $\mathbb{W}$. The following theorem has been established in Strahm [19].

Theorem 4 *The provably total functions of* PT *coincide with* FPTIME.

The lower bound for PT is established by deriving a natural form of bounded recursion of notation within PT. For its formulation we need the cut-off operator $|$. Informally speaking, $t|s$ is t if $t \leq s$ and s else. More formally, we can make use of definition by cases d_{W} on W and the characteristic function $\mathsf{c}_{\subseteq}$ in order to define $|$; then $t \mid s$ simply is an abbreviation for the $\mathcal{L}$ term $\mathsf{d}_{\mathsf{W}} ts(\mathsf{c}_{\subseteq}(\mathsf{l}_{\mathsf{W}} t)(\mathsf{l}_{\mathsf{W}} s))0$.

The following lemma is proved in Strahm [19].

Lemma 5 (Bounded recursion on notation) *There is a closed term* r_{W} *in $\mathcal{L}$ such that* PT^- *proves*

$$
\begin{array}{l}
f : \mathsf{W} \mapsto \mathsf{W} \wedge g : \mathsf{W}^3 \mapsto \mathsf{W} \wedge b : \mathsf{W}^2 \mapsto \mathsf{W} \rightarrow \\
\qquad (\mathsf{r}_{\mathsf{W}} fgb : \mathsf{W}^2 \mapsto \mathsf{W} \wedge \\
\qquad\qquad [x \in \mathsf{W} \wedge y \in \mathsf{W} \wedge y \neq \epsilon \wedge h = \mathsf{r}_{\mathsf{W}} fgb] \rightarrow \\
\qquad\qquad hx\epsilon = fx \wedge hxy = gxy(hx(\mathsf{p}_{\mathsf{W}} y)) \mid bxy)
\end{array}
$$

We close this section by stating a lemma concerning the existence of a polynomial time pairing operation on binary words which is provably total in PT^- and, in addition, has certain natural properties. We will use this function in Section 5 of this paper. Observe that PT has a built-in pairing operation p with associated projections $\mathsf{p_0}$ and $\mathsf{p_1}$, but this form of pairing does not necessarily map binary words to binary words.

Lemma 6 (Polynomial time pairing operation) *There are closed $\mathcal{L}$ terms $t_{\mathsf{p}}, t_{\mathsf{p0}}$ and t_{p1} such that PT^- proves:*

(1) $t_{\mathsf{p}} : \mathsf{W}^2 \mapsto \mathsf{W} \;\wedge\; t_{\mathsf{p0}} : \mathsf{W} \mapsto \mathsf{W} \;\wedge\; t_{\mathsf{p1}} : \mathsf{W} \mapsto \mathsf{W}$,

(2) $x \in \mathsf{W} \wedge y \in \mathsf{W} \;\rightarrow\; (t_{\mathsf{p0}}(t_{\mathsf{p}}xy) = x \wedge t_{\mathsf{p1}}(t_{\mathsf{p}}xy) = y)$,

(3) $x \in \mathsf{W} \wedge y \in \mathsf{W} \wedge u \in \mathsf{W} \wedge v \in \mathsf{W} \wedge x \leq u \wedge y \leq v \;\rightarrow\; t_{\mathsf{p}}xy \leq t_{\mathsf{p}}uv$.

In the sequel we will write $\langle s, t\rangle$ for $t_{\mathsf{p}}st$ and $\langle s\rangle_{\mathsf{i}}$ for $t_{\mathsf{pi}}s$. The proof of the above lemma is standard but tedious and hence will be left as an exercise to the reader.

3 Introducing the explicit type theory PET

In this section, we will introduce PET, a theory of **p**olynomial time operations with **e**xplicit **t**ypes. The theory PET is an extension of the applicative base theory $\mathsf{B}(*, \times)$ by means of a natural restriction of elementary comprehension, which is one of the crucial principles of explicit mathematics, see Feferman [9, 11]. Below we will use the language of explicit mathematics due to Jäger [15] which is based on a so-called naming relation $\Re$. Our type existence axioms are very naturally presented by means of a finite axiomatisation in the spirit of Feferman and Jäger [12].

PET is formulated in the second order language $\mathcal{L}_{\mathsf{T}}$ which extends the language $\mathcal{L}$ of PT by type variables $U, V, W, X, Y, Z, \ldots$, binary relation symbols $\Re$ (naming) and $\in$ (elementhood), as well as (individual) constants w (initial segment of W), id (identity), dom (domain), un (union), int (intersection), and inv (inverse image).

The *formulas* $A, B, C, \ldots$ of $\mathcal{L}_{\mathsf{T}}$ (possibly with subscripts) are built from the atomic formulas of $\mathcal{L}$ as well as formulas of the form $(s \in X)$, $\Re(s, X)$ and $(X = Y)$, by closing under negation, disjunction, conjunction, implication, as well as existential and universal quantification over individuals and types. If A is an $\mathcal{L}_{\mathsf{T}}$ formula, we let $FV_I(A)$ and $FV_T(A)$ denote the set of its free individual and type variables, respectively. Finally, we write $FV_I(t)$ for the set of individual variables occurring in the term t.

Types are extensional and have (explicit) names which are intensional. The names are generated via uniform operations and the link to the types they are referring to is established by the naming relation $\Re$. The element relation $\in$ is also a relation between an individual and a type, expressing that the individual is a member of the type. As we will mostly refer to types by using their names, we introduce a few additional abbreviations ($\vec{s} = s_1, \ldots, s_n$, $\vec{X} = X_1, \ldots, X_n$):

$$\begin{aligned}
\Re(\vec{s}, \vec{X}) &:= \Re(s_1, X_1) \wedge \cdots \wedge \Re(s_n, X_n), \\
\Re(s) &:= \exists X \Re(s, X), \\
\Re(\vec{s}) &:= \Re(s_1) \wedge \cdots \wedge \Re(s_n), \\
s \mathrel{\dot{\in}} t &:= \exists X (\Re(t, X) \wedge s \in X).
\end{aligned}$$

We are now ready to spell out the axioms of PET in detail. We start by presenting PET by means of a finite axiomatization of a restricted form of elementary comprehension; later we will carefully investigate the corresponding class of elementary formulas.

3.1 Axioms of PET

The logical axioms of PT are extended by the following strictness axioms for the new relation symbols:

(D5) $\quad s \in X \to s\downarrow$

(D6) $\quad \Re(s, X) \to s\downarrow$

In addition, the logic of the types is just the usual predicate logic with equality.

PET consists of the axioms of $\mathsf{B}(*, \times)$ (i.e. PT without the Axiom ($\Sigma^{\mathsf{b}}_{\mathsf{W}}$-$\mathsf{I}_{\mathsf{W}}$)) plus the following axioms about types. The axioms in group I. are the so-called ontological axioms about the naming relation and extensionality; in group II. we state the axioms about type existence; finally, we include type induction in group III. We recall from Section 2 that $\mathsf{W}_a(x)$ abbreviates $(\mathsf{W}(x) \wedge x \leq a)$.

I. Explicit representation and extensionality

(O1) $\quad \exists x \Re(x, X)$

(O2) $\quad \Re(a, X) \wedge \Re(a, Y) \to X = Y$

(O3) $\quad \forall z (z \in X \leftrightarrow z \in Y) \to X = Y$

II. Type existence axioms

$(\mathbf{w}_a)$ $\quad a \in \mathsf{W} \rightarrow \Re(\mathsf{w}(a)) \wedge \forall x(x \mathrel{\dot{\in}} \mathsf{w}(a) \leftrightarrow \mathsf{W}_a(x))$

$(\mathbf{id})$ $\quad \Re(\mathsf{id}) \wedge \forall x(x \mathrel{\dot{\in}} \mathsf{id} \leftrightarrow \exists y(x = (y, y)))$

$(\mathbf{inv})$ $\quad \Re(a) \rightarrow \Re(\mathsf{inv}(f, a)) \wedge \forall x(x \mathrel{\dot{\in}} \mathsf{inv}(f, a) \leftrightarrow fx \mathrel{\dot{\in}} a)$

$(\mathbf{un})$ $\quad \Re(a) \wedge \Re(b) \rightarrow \Re(\mathsf{un}(a, b)) \wedge \forall x(x \mathrel{\dot{\in}} \mathsf{un}(a, b) \leftrightarrow (x \mathrel{\dot{\in}} a \vee x \mathrel{\dot{\in}} b))$

$(\mathbf{int})$ $\quad \Re(a) \wedge \Re(b) \rightarrow \Re(\mathsf{int}(a, b)) \wedge \forall x(x \mathrel{\dot{\in}} \mathsf{int}(a, b) \leftrightarrow (x \mathrel{\dot{\in}} a \wedge x \mathrel{\dot{\in}} b))$

$(\mathbf{dom})$ $\quad \Re(a) \rightarrow \Re(\mathsf{dom}(a)) \wedge \forall x(x \mathrel{\dot{\in}} \mathsf{dom}(a) \leftrightarrow \exists y((x, y) \mathrel{\dot{\in}} a))$

III. Type induction on W, $(\mathrm{T}\text{-}\mathsf{I}_{\mathsf{W}})$

$$\epsilon \in X \wedge (\forall x \in \mathsf{W})(x \in X \rightarrow \mathsf{s}_0 x \in X \wedge \mathsf{s}_1 x \in X) \rightarrow (\forall x \in \mathsf{W})(x \in X)$$

Let us now turn to the definition of a subclass of the so-called elementary formulas in explicit mathematics.

3.2 Restricted elementary comprehension

In this subsection we will show that the finite axiomatisation of type existence in PET gives rise to a natural restriction of the well-known schema of elementary comprehension. In this context, the notion of a $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formula is crucial. In addition, we will define a set of designated free individual variables $FV_{\mathsf{W}}(A)$ which can be thought of as binary words bounding existential quantifiers in a $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formula A. These variables act as parameters in the comprehension schema below.

Definition 7 ($\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formulas) The class of $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formulas of $\mathcal{L}_{\mathsf{T}}$ and the set of variables $FV_{\mathsf{W}}(A)$ are inductively defined as follows:

1. If A is an $\mathcal{L}_{\mathsf{T}}$ formula of the form $(s = t)$, $s\downarrow$ or $(s \in X)$, then A is a $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formula and $FV_{\mathsf{W}}(A) := \emptyset$.

2. If A is the formula $\mathsf{W}_a(t)$ with $a \notin FV_I(t)$, then A is a $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formula and $FV_{\mathsf{W}}(A) := \{a\}$.

3. If A is the formula $(B \wedge C)$ or $(B \vee C)$ with B and C in $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ and, in addition,

$$(FV_I(B) \setminus FV_{\mathsf{W}}(B)) \cap FV_{\mathsf{W}}(C) = \emptyset,$$
$$(FV_I(C) \setminus FV_{\mathsf{W}}(C)) \cap FV_{\mathsf{W}}(B) = \emptyset,$$

then A is a $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formula and $FV_{\mathsf{W}}(A) := FV_{\mathsf{W}}(B) \cup FV_{\mathsf{W}}(C)$.

4. If A is the formula $\exists x B$ with $B \in \Sigma^{\mathsf{b}}_{\mathrm{T}}$ and $x \notin FV_{\mathsf{W}}(B)$, then A is a $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formula and $FV_{\mathsf{W}}(A) := FV_{\mathsf{W}}(B)$.

Remark 8 We observe that the above definition also captures formulas starting with a bounded (with respect to W) existential quantifier; namely, if $B[x]$ is a $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formula, then $(\exists x \leq a)B[x]$ can be expressed by the $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formula $\exists x(\mathsf{W}_a(x) \wedge B[x])$.

In the sequel we will assume that for each $\mathcal{L}_{\mathsf{T}}$ formula A, we have a mapping μ_A which assigns to each free type variable X in A a fresh individual variable $\mu_A(X)$ that does not occur in A. We assume that μ_A is injective. The elegant notation in the following definition is adapted from Krähenbühl [17].

Definition 9 Assume that A is a $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formula. Then we define a term $\rho_A x.B$ by induction on the complexity of the formula B in $\Sigma^{\mathsf{b}}_{\mathrm{T}}$, where we assume that $x \notin FV_{\mathsf{W}}(B)$ and x not bound in B:

$$
\begin{aligned}
\rho_A x.(s = t) &:= \mathsf{inv}(\lambda x.(s,t), \mathsf{id}), \\
\rho_A x.(s\downarrow) &:= \mathsf{inv}(\lambda x.(s,s), \mathsf{id}), \\
\rho_A x.(s \in \mathsf{W}_a) &:= \mathsf{inv}(\lambda x.s, \mathsf{w}(a)), \\
\rho_A x.(s \in X) &:= \mathsf{inv}(\lambda x.s, \mu_A(X)), \\
\rho_A x.(C \wedge D) &:= \mathsf{int}(\rho_A x.C, \rho_A x.D) \\
\rho_A x.(C \vee D) &:= \mathsf{un}(\rho_A x.C, \rho_A x.D) \\
\rho_A x.(\exists y C) &:= \mathsf{dom}(\rho_A x.(C[(x)_0/x, (x)_1/y])).
\end{aligned}
$$

Let us write $\rho x.A$ instead of $\rho_A x.A$. The following theorem states that we can derive uniform comprehension for $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formulas in PET.

Theorem 10 (Restricted elementary comprehension in PET) *Assume that A is a $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formula with $FV_T(A) = \{X_1, \ldots, X_n\}$ and $FV_{\mathsf{W}}(A) = \{w_1, \ldots, w_m\}$. If we let $z_i := \mu_A(X_i)$ for $1 \leq i \leq n$, then we have:*

(1) $FV_I(\rho x.A) = (FV_I(A) \setminus \{x\}) \cup \{z_1, \ldots, z_n\}$,

(2) $\mathsf{PET} \vdash \mathsf{W}(\vec{w}) \wedge \Re(\vec{z}, \vec{X}) \rightarrow \Re(\rho x.A)$,

(3) $\mathsf{PET} \vdash \mathsf{W}(\vec{w}) \wedge \Re(\vec{z}, \vec{X}) \rightarrow (\forall x)(x \,\dot{\in}\, \rho x.A \leftrightarrow A)$.

Using λ abstraction and projections, we obtain the following immediate consequence of the above theorem.

Corollary 11 *Assume that $A[x, \vec{v}, \vec{w}, \vec{X}]$ is a $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formula with the following free variables:*

$$\begin{aligned} FV_T(A) &= \{X_1, \dots, X_n\}, \\ FV_{\mathsf{W}}(A) &= \{w_1, \dots, w_m\}, \\ FV_I(A) \setminus FV_{\mathsf{W}}(A) &= \{x, v_1, \dots, v_k\}. \end{aligned}$$

Then we can find a closed $\mathcal{L}_{\mathsf{T}}$ term c_A such that PET *proves:*

(1) $\mathsf{W}(\vec{w}) \wedge \Re(\vec{z}, \vec{X}) \rightarrow \Re(c_A(\vec{v}, \vec{w}, \vec{z}))$,

(2) $\mathsf{W}(\vec{w}) \wedge \Re(\vec{z}, \vec{X}) \rightarrow (\forall x)(x \,\dot{\in}\, c_A(\vec{v}, \vec{w}, \vec{z}) \leftrightarrow A[x, \vec{v}, \vec{w}, \vec{X}])$.

Let us close this section by mentioning that the schema of uniform $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ comprehension clearly entails the type existence axioms as given in the finite axiomatisation of PET.

4 Lower bounds

In this section, we will show that PT^- can be embedded into PET. PT^- is expressively weaker than PT, but proof-theoretically equivalent. Recall from Lemma 5 (Bounded Recursion on Notation) that the provably total functions of PT^- are still the polynomial time computable functions. We will first prove some auxiliary lemmas and then specify the embedding.

In Theorem 10 we have seen that any $\Sigma^{\mathsf{b}}_{\mathrm{T}}$-formula defines a type in PET. With Axiom (T-I_{W}) we have induction on any type which gives us induction for $\Sigma^{\mathsf{b}}_{\mathrm{T}}$-formulas. In the following we will make heavy use of this fact. For convenience we will also exploit the obvious equivalence

$$(\forall x \in \mathsf{W})(x \in X \rightarrow \mathsf{s}_0 x \in X \wedge \mathsf{s}_1 x \in X) \leftrightarrow (\forall x \in \mathsf{W})(\mathsf{p}_{\mathsf{W}} x \in X \rightarrow x \in X).$$

Later, we will need some additional properties of the subword relation:

Lemma 12 *The following statements are provable in* PET*:*

(1) $x \in \mathsf{W} \wedge z \in \mathsf{W} \wedge x \subseteq \mathsf{p}_{\mathsf{W}} z \rightarrow x \subseteq z$,

(2) $x \in \mathsf{W} \wedge y \in \mathsf{W} \wedge z \in \mathsf{W} \wedge x \subseteq y \wedge y \subseteq z \rightarrow x \subseteq z$ *(Transitivity),*

(3) $x \in \mathsf{W} \wedge y \in \mathsf{W} \wedge x \subseteq y \rightarrow x \leq y$.

Proof In the following we will work informally in PET and assume that $x, y, z \in \mathsf{W}$.

(1) Immediate with Axiom (13).

(2) The $\Sigma^{\mathsf{b}}_{\mathrm{T}}$-formula $(\mathsf{c}_{\subseteq}xy = 1 \vee \mathsf{c}_{\subseteq}yz = 1 \vee \mathsf{c}_{\subseteq}xz = 0)$ is a reformulation of transitivity. We will prove it by induction on z and assume $\mathsf{c}_{\subseteq}xy = 0$ and $\mathsf{c}_{\subseteq}yz = 0$.

   $z = \epsilon$: With Axiom (12) we know that also $y = \epsilon$ and thus $x = \epsilon$. With the same axiom we immediately get $x \subseteq z$.

   We assume that transitivity holds for $\mathsf{p_W}z$, now we will prove that it also holds for z. With Axiom (13) we know that $v \subseteq w$ iff either $v = w$ or $v \subseteq \mathsf{p_W}w$ for any words v, w. If $x = y$ or $y = z$, $x \subseteq z$ is immediate with the equality axioms. Otherwise we have $x \subseteq \mathsf{p_W}y\,(i)$ and $y \subseteq \mathsf{p_W}z\,(ii)$. With (i) and part (1) of this lemma we get $x \subseteq y$. Therefore also $x \subseteq \mathsf{p_W}z$ by induction hypothesis and (ii) and again with part (1) we get $x \subseteq z$.

(3) We can write this property as $\mathsf{c}_{\subseteq}xy = 1 \vee \mathsf{c}_{\subseteq}(1{\times}x)(1{\times}y) = 0$ which obviously is a $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formula. Again we will only look at the case $x \subseteq y$ in the induction on y.

   $y = \epsilon$: As above, this implies that also $x = \epsilon$. Therefore obviously $1{\times}x = 1{\times}y$.

   Assume the assertion holds for $\mathsf{p_W}y$. To prove that $x \leq y$ if $x \subseteq y$ we make the same case distinction as above: if $x = y$ then $x \leq y$ is again obvious. Otherwise $x \subseteq \mathsf{p_W}y$. By induction hypothesis we have $1{\times}x \subseteq 1{\times}(\mathsf{p_W}y)\,(i)$. Further, $1{\times}y = (1{\times}\mathsf{p_W}y){*}1 = \mathsf{s_1}(1{\times}\mathsf{p_W}y)\,(ii)$. With part (1) we get $1{\times}x \subseteq 1{\times}y$ from (i) and (ii).

This concludes the proof of our lemma. $\square$

Remark 13 The above lemma, part (2) and the definition of $\leq$ immediately imply that also $\leq$ is transitive, provably in PET.

In PET, type induction can also be stated differently. As this notion will be more convenient in the following proofs, we will prove that both formulations are equivalent:

$$(\text{T-}\mathsf{I}^{\mathsf{b}}_{\mathsf{W}}) \qquad a \in \mathsf{W} \wedge \epsilon \in X \wedge (\forall x \subseteq a)(\mathsf{p_W}x \in X \to x \in X) \to a \in X$$

Lemma 14 *We have that* $(\text{T-}\mathsf{I}_{\mathsf{W}})$ *and* $(\text{T-}\mathsf{I}^{\mathsf{b}}_{\mathsf{W}})$ *are provably equivalent in* PET *without* $(\text{T-}\mathsf{I}_{\mathsf{W}})$.

Proof The fact that (T-$\mathsf{I}^{\mathsf{b}}_{\mathsf{W}}$) entails (T-$\mathsf{I}_{\mathsf{W}}$) is trivial. For the converse implication, we take any type X and some $a \in \mathsf{W}$. Then we can build the type $Y := \{x : \mathsf{c}_{\subseteq}xa = 1 \vee x \in X\}$. Its membership condition is equivalent to $x \subseteq a \to x \in X$ if $x \in \mathsf{W}$.

Now we assume that $\epsilon \in X$ and $(\forall x \subseteq a)(\mathsf{p}_{\mathsf{W}}x \in X \to x \in X)$ $(\star)$. Obviously, $\epsilon \in Y$ from the definition of Y. Now we have to show that if $x \in Y$ then also $\mathsf{s}_0 x \in Y \wedge \mathsf{s}_1 x \in Y$. Hence, assume $x \in Y$. If $\mathsf{c}_{\subseteq}(\mathsf{s}_i x)a = 1$, then obviously $\mathsf{s}_i x \in Y$. Otherwise, i.e., $\mathsf{s}_i x \subseteq a$, transitivity of $\subseteq$ readily entails $x \subseteq a$, which implies $x \in X$. Now we can use $(\star)$ to derive $\mathsf{s}_i x \in X$ and thus $\mathsf{s}_i x \in Y$.

Now we proved the conditions to apply (common) type induction (T-I_{W}) and get $(\forall x \in \mathsf{W})(x \in Y)$. Therefore also $a \in Y$. As $a \subseteq a \equiv \mathsf{c}_{\subseteq}aa = 0$, a must be in X. □

In the following lemma we want to prove that every function f of type $\mathsf{W} \mapsto \mathsf{W}$ can be bound by a monotone function in the sense of the following lemma. We will construct this function f^* as the function taking the maximum of f applied to all subwords, i.e. $f^*x = \max\limits_{y \subseteq x} fy$. The functional mapping f to f^* is a well-known basic feasible functional, cf. e.g. Cook and Kapron [7, 16].

Lemma 15 *There is a closed term* max *such that* PET *proves:*

(1) $f : \mathsf{W} \mapsto \mathsf{W} \to \mathsf{max}f : \mathsf{W} \mapsto \mathsf{W}$,

(2) $f : \mathsf{W} \mapsto \mathsf{W} \wedge f^* = \mathsf{max}f \wedge x \in \mathsf{W} \wedge y \in \mathsf{W} \wedge x \subseteq y \to f^*x \leq f^*y)$,

(3) $f : \mathsf{W} \mapsto \mathsf{W} \wedge f^* = \mathsf{max}f \wedge x \in \mathsf{W} \to fx \leq f^*x)$,

(4) $f : \mathsf{W} \mapsto \mathsf{W} \wedge f^* = \mathsf{max}f \wedge x \in \mathsf{W} \wedge y \in \mathsf{W} \wedge x \subseteq y \to fx \leq f^*y$.

Proof We first define an auxiliary function $\mathsf{max}_{\mathrm{arg}}$ locating the subword where f is maximised and write $\widetilde{f} = \mathsf{max}_{\mathrm{arg}}f$:

$$
\begin{aligned}
\mathsf{max}_{\mathrm{arg}}f\epsilon &\simeq \epsilon \\
\mathsf{max}_{\mathrm{arg}}f(\mathsf{s}_i x) &\simeq \begin{cases} \mathsf{max}_{\mathrm{arg}}fx & \text{if } f(\mathsf{s}_i x) \leq f(\mathsf{max}_{\mathrm{arg}}fx) \\ \mathsf{s}_i x & \text{otherwise} \end{cases}
\end{aligned}
$$

We can construct this term with Lemma 2, λ-abstraction and definition by cases. Now we have to prove that $\widetilde{f} : \mathsf{W} \mapsto \mathsf{W}$, provided $f : \mathsf{W} \mapsto \mathsf{W}$.

We fix some $a \in \mathsf{W}$ and define the $\Sigma^{\mathsf{b}}_{\mathrm{T}}$-formula $A[x] \equiv (\exists y \leq a)(\widetilde{f}x = y)$. Now we will make use of (T-$\mathsf{I}^{\mathsf{b}}_{\mathsf{W}}$) in order to show $A[a]$:

$x = \epsilon$: $\widetilde{f}\epsilon = \epsilon \leq a$.

Assume $x \subseteq a$ and $A[\mathsf{p_W}x]$: As we know that $z = \widetilde{f}(\mathsf{p_W}x) \in \mathsf{W}$ we can decide whether $fx \leq fz$ or not. In the first case we have $\widetilde{f}x = \widetilde{f}(\mathsf{p_W}x) \leq a$ by induction hypothesis. In the latter case we have $\widetilde{f}x = x$ which gives $\widetilde{f}x \leq a$ with Lemma 12.(3).

Now we define $\mathsf{max} := \lambda f.\lambda x.f(\mathsf{max}_{\mathrm{arg}} fx)$.

(1) Obvious from totality of $\widetilde{f}$.

(2) Follows from the construction of f^* by induction on y.

(3) Follows from the construction of $\widetilde{f}$ and induction.

(4) Immediate with parts (2) and (3) and Lemma 12.

This concludes the proof of our lemma. □

Now we are in the position to state the main theorem of this section, which will immediately entail the desired lower bounds for PET.

Theorem 16 PT^- *is contained in* PET.

Proof Clearly, we only need to prove that $\mathsf{PET} \vdash (\Sigma^{\mathsf{b}-}_{\mathsf{W}}\text{-}\mathsf{I_W})$, i.e. that induction holds for any $\Sigma^{\mathsf{b}-}_{\mathsf{W}}$-formula $A[x] \equiv (\exists y \leq fx)B[f,x,y]$. Let us work informally in PET and assume, in addition,

$$(1) \qquad f : \mathsf{W} \to \mathsf{W} \wedge A[\epsilon] \wedge (\forall x \in \mathsf{W})(A[x] \to A[\mathsf{s_0}x] \wedge A[\mathsf{s_1}x]).$$

Now we fix $c \in \mathsf{W}$ and aim at showing $A[c]$. First, we will need Lemma 15 in order to show that for $c \in \mathsf{W}$ and $x \subseteq c$ we get

$$(2) \qquad (\exists y \leq fx)B[x,y] \leftrightarrow (\exists y \leq f^*c)(y \leq fx \wedge B[x,y])$$

With Lemma 15 we immediately get $fx \leq f^*c$ and then the equivalence is obvious by Lemma 12. As $f^*c \in \mathsf{W}$ for $c \in \mathsf{W}$ and B an $\Sigma^{\mathsf{b}}_{\mathsf{T}}$ formula by definition of $\Sigma^{\mathsf{b}-}_{\mathsf{W}}$, we can invoke Theorem 10 and construct a type X with the defining property

$$(3) \qquad (\forall x \subseteq c)(x \in X \leftrightarrow (\exists y \leq f^*c)(y \leq fx \wedge B[x,y]))$$

By (1), (2) and (3) we immediately obtain

$$(4) \qquad \epsilon \in X \wedge (\forall x \subseteq c)(\mathsf{p_W}x \in X \to x \in X)$$

Now we can apply $(\mathsf{T}\text{-}\mathsf{I}^{\mathsf{b}}_{\mathsf{W}})$ and derive $c \in X$ and hence $A[c]$ as desired. □

The above theorem together with Lemma 5 shows that bounded recursion on notation can be represented as a type two functional in PET. Hence, the following corollary is immediate.

Corollary 17 *The polynomial time computable functions are provably total in* PET.

Indeed, we have that Buss' S^1_2 [3] and Ferreira's PTCA^+ [13, 14] as well as Cook and Urquhart's PV^ω [8] are all contained in PET, see Strahm [19, 20]. Hence, the basic feasible functionals are provably total in PET.

In Section 6 we will propose a natural extension of PET allowing us to embed the full theory PT.

5 Upper bounds

In this section we want to show that PET is conservative over PT^- for $\mathcal{L}$ sentences. As the formula expressing the totality of a function on W is an $\mathcal{L}$ sentence, the desired upper bounds can immediately be derived so that the provably total functions of PET coincide with the polynomial time computable functions.

Our proof strategy is to show that each model of PT^- can be extended to a model of PET. For the reader's convenience, let us briefly recall the notion of a structure for the language $\mathcal{L}_\mathsf{T}$.

Definition 18 ($\mathcal{L}_\mathsf{T}$ structure) A $\mathcal{L}_\mathsf{T}$-structure $\mathcal{M}^\star$ is a tuple

$$(\mathcal{M}, \mathcal{T}, \mathcal{E}, \mathcal{R}, \mathrm{w}, \mathrm{id}, \mathrm{dom}, \mathrm{un}, \mathrm{int}, \mathrm{inv})$$

where (i) $\mathcal{M}$ is a $\mathcal{L}$-structure, (ii) $\mathcal{T}$ is a non-empty set of subsets of $|\mathcal{M}|$, (iii) $\mathcal{E}$ is the usual $\in$ relation on $|\mathcal{M}| \times \mathcal{T}$, (iv) $\mathcal{R}$ is a non-empty subset of $|\mathcal{M}| \times \mathcal{T}$, and (v) $\mathrm{w}, \mathrm{id}, \mathrm{dom}, \mathrm{un}, \mathrm{int}, \mathrm{inv}$ are elements of $|\mathcal{M}|$.

Model construction

Now we give a scheme for extending any model $\mathcal{M}$ of PT^- to a model $\mathcal{M}^\star$ of PET: First, we need to choose selected elements $\mathrm{w}, \mathrm{id}, \mathrm{dom}, \mathrm{un}, \mathrm{int}, \mathrm{inv}$ of $|\mathcal{M}|$ in order to interpret the corresponding constants of $\mathcal{L}_\mathsf{T}$. This can easily be done in such a way that $\mathrm{c}x \neq \mathrm{c}y$ and $\mathrm{c}u \neq \mathrm{d}v$ for all $\mathrm{c} \neq \mathrm{d} \in \{\mathrm{w}, \mathrm{id}, \mathrm{dom}, \mathrm{un}, \mathrm{int}, \mathrm{inv}\}$ and all $x \neq y, u, v \in |\mathcal{M}|$.

For the construction of $\mathcal{T}$ and $\mathcal{R}$ we introduce sets $R_k \subseteq |\mathcal{M}|$ by induction on the natural number k and simultaneously for each $m \in R_k$ establish a set $ext(m) \subseteq |\mathcal{M}|$. Then we set

$$\begin{array}{rcl} \mathcal{T}_k & := & \{ext(m) : m \in R_k\}, \\ \mathcal{R}_k & := & \{(m, ext(m)) : m \in R_k\}, \end{array}$$

$$\mathcal{M}_k^\star \;:=\; (\mathcal{M}, \mathcal{T}_k, \mathcal{R}_k, \mathrm{w}, \mathrm{id}, \mathrm{dom}, \mathrm{un}, \mathrm{int}, \mathrm{inv}).$$

$k = 0$: R_0 contains names of the base types as well as their obvious extensions, i.e.

- $\mathrm{id} \in R_0$ and $\mathit{ext}(\mathrm{id}) := \{(m, m) : m \in |\mathcal{M}|\}$
- $\mathrm{w}a \in R_0$ if $a \in \mathsf{W}^{\mathcal{M}}$ and $\mathit{ext}(\mathrm{w}a) := \{m \in |\mathcal{M}| : \mathcal{M} \models m \in \mathsf{W} \wedge m \leq a\}$

$k > 0$: R_k contains R_{k-1}. In addition, if $a, b \in R_{k-1}$ then

- $\mathrm{inv}(f, a) \in R_k$ and $\mathit{ext}(\mathrm{inv}(f, a)) := \{m \in |\mathcal{M}| : \mathcal{M}_{k-1}^\star \models fm \mathrel{\dot\in} a\}$
- $\mathrm{un}(a, b) \in R_k$ and $\mathit{ext}(\mathrm{un}(a, b)) := \{m \in |\mathcal{M}| : \mathcal{M}_{k-1}^\star \models m \mathrel{\dot\in} a \vee m \mathrel{\dot\in} b\}$
- $\mathrm{int}(a, b) \in R_k$ and $\mathit{ext}(\mathrm{int}(a, b)) := \{m \in |\mathcal{M}| : \mathcal{M}_{k-1}^\star \models m \mathrel{\dot\in} a \wedge m \mathrel{\dot\in} b\}$
- $\mathrm{dom}(a) \in R_k$ and $\mathit{ext}(\mathrm{dom}(a)) := \{m \in |\mathcal{M}| : \mathcal{M}_{k-1}^\star \models \exists y((m, y) \mathrel{\dot\in} a)\}$

Finally, we define $\mathcal{T} := \bigcup_{k\in\mathbb{N}} \mathcal{T}_k$ and $\mathcal{R} := \bigcup_{k\in\mathbb{N}} \mathcal{R}_k$. Then our desired $\mathcal{L}_\mathsf{T}$ structure is given by

$$\mathcal{M}^\star \;:=\; (\mathcal{M}, \mathcal{T}, \mathcal{R}, \mathrm{w}, \mathrm{id}, \mathrm{dom}, \mathrm{un}, \mathrm{int}, \mathrm{inv}).$$

We are now ready to state the crucial model extension theorem.

Theorem 19 (Model extension) *Any model $\mathcal{M}^\star$ constructed as described above from a model $\mathcal{M}$ of PT^- satisfies the following conditions:*

(1) $\mathcal{M} \models A \iff \mathcal{M}^\star \models A$ *for any $\mathcal{L}$ sentence A,*

(2) $\mathcal{M}^\star \models \mathrm{T}\text{-}\mathsf{I}_\mathsf{W}$,

(3) $\mathcal{M}^\star \models \mathsf{PET}$.

Proof

(1) As the first-order part of the model $\mathcal{M}$ remains untouched, the same (first-order) formulas are satisfied in the extended model.

(2) In order to prove that $\mathcal{M}^\star$ satisfies type induction, we will show that every type X of $\mathcal{T}$ is *weakly $\Sigma_\mathsf{W}^{\mathsf{b}-}$ definable*. This means that membership for this type can be expressed by a $\Sigma_\mathsf{W}^{\mathsf{b}-}$ formula of $\mathcal{L}$ with a fixed bound, namely

$$X = \{m \in |\mathcal{M}| : \mathcal{M} \models (\exists y \leq \mathsf{k}bm)B[m, y]\}$$

for some $b \in \mathsf{W}^{\mathcal{M}}$ and B W-free, positive and without $\forall$. B may possibly contain parameters in $|\mathcal{M}|$. We will now use the fact that types are added to $\mathcal{T}$ as the extension of a name and that every name is added at a certain level. Therefore, we will make induction on R_k to show that every name a can be weakly $\Sigma_\mathsf{W}^{\mathsf{b}-}$ defined by a formula A:

$\mathbf{a} \in \mathbf{R_0}$: Then $a = \mathrm{id}$ or $a = \mathrm{w}b$ for some $b \in \mathsf{W}^{\mathcal{M}}$. In the former case, we choose A to be $\exists z(x = (z, z))$ which is a $\Sigma^{\mathsf{b}-}_{\mathsf{W}}$ formula without bound. In the latter case, A is set to $(\exists y \leq b)(x = y)$.

$\mathbf{a} \in \mathbf{R_{n+1}} \setminus \mathbf{R_n}$: Suppose $b, c \in R_n$. By induction hypothesis, there are corresponding defining formulas B and C,

$$\begin{array}{rcl} B[x] & \equiv & (\exists y \leq u)B'[x, y], \\ C[x] & \equiv & (\exists z \leq v)C'[x, z], \end{array}$$

where $u, v \in \mathsf{W}^{\mathcal{M}}$. We distinguish the following cases:

– $a = \mathrm{un}(b, c)$: Then we set $d = \langle u, v \rangle$ and $A[x]$ to be

$$(\exists y \leq d)[(\langle y \rangle_0 \leq u \wedge B'[x, \langle y \rangle_0]) \vee (\langle y \rangle_1 \leq v \wedge C'[x, \langle y \rangle_1])],$$

where we use the properties of Lemma 6.

– $a = \mathrm{int}(b, c)$: This case is analogous to the previous one.

– $a = \mathrm{inv}(f, b)$: In this case we can choose $A[x]$ to be $B[fx]$.

– $a = \mathrm{dom}(b)$: We can choose the formula $A[x]$ to be

$$(\exists y \leq u)\exists z B'[(x, z), y].$$

This concludes our argument that each type X in $\mathcal{T}$ can be weakly $\Sigma^{\mathsf{b}-}_{\mathsf{W}}$ defined in $\mathcal{M}$. It is now immediate that $\mathcal{M}^\star$ satisfies $(\mathsf{T}\text{-}\mathsf{I}_{\mathsf{W}})$ since $\mathcal{M}$ validates $(\Sigma^{\mathsf{b}-}_{\mathsf{W}}\text{-}\mathsf{I}_{\mathsf{W}})$.

(3) The remaining axioms are obvious by construction.

This concludes the proof of the model extension theorem. □

The following corollary is now immediate by Gödel completeness and Theorem 16.

Corollary 20 PET *is a conservative extension of* PT^- *with respect to* $\mathcal{L}$ *formulas.*

Corollary 21 *The provably total functions of* PET *coincide with the functions computable in polynomial time.*

Further, it follows from Strahm [20] that the provably total type two functionals of PET coincide with the basic feasible functionals of type two.

6 Extensions of PET

In this section we will describe and treat some natural extensions of PET. They mostly rely on Cantini [6] who – among other things – studies various extensions of the first-order theory PT by means of axioms for self-referential truth, a uniformity principle and a axiom of positive choice. Below we will indicate how the principles studied by Cantini give rise to interesting extensions of PET.

6.1 Uniformity and universal quantification

Cantini showed in [6] that adding a uniformity principle for positive formulas of $\mathcal{L}$ yields an extension of PT whose provably total functions are still the functions computable in polynomial time. In our context, we can state Cantini's principle as follows. For each *positive* $\mathcal{L}$ formula $A[x, y]$:

$$(\mathbf{UP}) \qquad \forall x(\exists y \in \mathsf{W})A[x, y] \to (\exists y \in \mathsf{W})(\forall x)A[x, y]$$

It is easy to see that ($\mathbf{UP}$) readily entails the following form of *bounded uniformity* for positive $\mathcal{L}$ formulas $A[x, y]$:

$$(\mathbf{UP}') \qquad \forall x(\exists y \leq t)A[x, y] \to (\exists y \leq t)(\forall x)A[x, y]$$

The principle ($\mathbf{UP}$') leads to a very natural extension of PET by adding a type existence axiom for universal quantification; this axiom is the natural dual analogue of the domain type present in PET:

$$(\mathbf{all}) \qquad \Re(a) \to \Re(\mathsf{all}(a)) \wedge \forall x(x \,\dot{\in}\, \mathsf{all}(a) \leftrightarrow \forall y((x, y) \,\dot{\in}\, a))$$

The presence of the axiom ($\mathbf{all}$) makes the type existence axioms more symmetric, i.e. the types are generated from base types (initial segments of W and the identity type) by closing under domains, unions, intersections, existential quantification (inverse image) and universal quantification.

In order to see that ($\mathbf{all}$) does not increase the proof-theoretic strength of PET, let $\mathcal{M}$ be any model of PT+($\mathbf{UP}'$). We can extend $\mathcal{M}$ to a model $\mathcal{M}^\star$ of PET+($\mathbf{all}$) in the obvious way by adding a clause for universal quantification in the construction of Section 5. The crucial step in Theorem 19 is to show that $\mathcal{M}^\star$ satisfies type induction. Towards this aim, we follow the strategy in the proof of the above mentioned theorem and show that each type in $\mathcal{M}^\star$ is weakly $\Sigma^{\mathsf{b}}_{\mathsf{W}}$ definable in $\mathcal{M}$. The only new case occurs for $a = \mathrm{all}(b)$ and already knowing that b has a weak $\Sigma^{\mathsf{b}}_{\mathsf{W}}$ definition, say $B[u] = (\exists v \leq s)B'[u, v]$. Then we have that $\mathcal{M}^\star$ satisfies

$$\forall u[u \,\dot{\in}\, b \leftrightarrow (\exists v \leq s)B'[u, v]]$$

Thus, by construction of our model $\mathcal{M}^\star$ we obtain that

$$\forall x[x \dot{\in} a \leftrightarrow \forall y B[(x,y)] \leftrightarrow \forall y(\exists v \leq s)B'[(x,y),v]]$$

Now we are in a position to invoke (**UP**') in order to get the equivalence

$$\forall y(\exists v \leq s)B'[(x,y),v] \leftrightarrow (\exists v \leq s)\forall y B'[(x,y),v].$$

The formula $(\exists v \leq s)\forall y B'[\langle x,y\rangle,v]$ is clearly a weak $\Sigma^{\mathsf{b}}_{\mathsf{W}}$ formula and, hence, $a = \mathrm{all}(b)$ is weakly $\Sigma^{\mathsf{b}}_{\mathsf{W}}$ definable. This shows that type induction holds in our model $\mathcal{M}^\star$.

To summarise, we have shown that A can already be derived in $\mathsf{PT} + (\mathbf{UP})$ if A is an $\mathcal{L}$ formula which is provable in $\mathsf{PET} + (\mathbf{all})$. Moreover, by Cantini [6] the provably total functions of the latter theory coincide with the functions computable in polynomial time.

Furthermore, the full theory PT is contained in $\mathsf{PET} + (\mathbf{all})$: We can extend the definition of $\Sigma^{\mathsf{b}}_{\mathsf{T}}$-formulas such that they are also closed under $\forall$ quantification. For that purpose we have to extend Definition 7 and the term $\rho_A(A,x)$ of Definition 9. Theorem 10 also holds for this extended class of formulas. With these preparations, we can obviously expand the proof of Theorem 16 to induction for $\Sigma^{\mathsf{b}}_{\mathsf{W}}$-formulas.

6.2 Axiom of choice

In addition to the uniformity principle discussed above, Cantini [6] also considers a form of positive choice in the context of PT with a partial truth predicate and shows that this principle does not increase the proof-theoretic strength. Positive choice in the language $\mathcal{L}$ includes for each positive $\mathcal{L}$ formula $A[x,y]$ the statement

$$(\mathbf{AC}) \qquad (\forall x \in \mathsf{W})(\exists y \in \mathsf{W})A[x,y] \rightarrow (\exists f : \mathsf{W} \mapsto \mathsf{W})(\forall x \in \mathsf{W})A[x, fx]$$

If we extend this schema to the language $\mathcal{L}_{\mathsf{T}}$ then, in addition, A is allowed to contain positive occurrences of subformulas of the form $t \in X$. We will call this principle (**AC**) as well and agree that the context always indicates whether we refer to the first-order or second order form of the choice axiom.

Then it is easy to see by means of the model-theoretic argument of Section 5 that each (first-order) model of $\mathsf{PT} + (\mathbf{AC})$ can be extended to a model of PET which satisfies choice (**AC**) in the extended language of PET as described above. Hence, the provably total functions of $\mathsf{PET} + (\mathbf{AC})$ are still the polynomial time computable ones.

6.3 Totality and extensionality

The upper bound computations in Strahm [19] for PT and Cantini [6] for its various extensions actually validate stronger applicative axioms as those spelled out in PT. In particular, totality of application and extensionality of operations can be added to our theories without increasing proof-theoretic strength.

In the presence of the totality axiom (**Tot**), partial combinatory logic reduces to total combinatory logic and the logic of partial terms can be replaced by ordinary predicate logic with equality.

(**Tot**) $$\forall x \forall y (xy\downarrow)$$

If, in addition, extensionality of operations is assumed, then the applicative basis is equivalent to ordinary untyped extensional lambda calculus $\lambda\eta$.

(**Ext**) $$\forall f \forall g (\forall x (fx \simeq gx) \rightarrow f = g)$$

We are now in a position to summarise the results of this section in the following theorem.

Theorem 22 *The provably total functions of* PET *augmented by any combination of the principles* (**all**)*,* (**AC**)*,* (**Tot**)*, and* (**Ext**) *coincide with the polynomial time computable functions.*

7 Conclusions and further work

In this paper, we considered a natural restriction of elementary comprehension with type induction as an extension of the theory PT. The polynomial time computable functions were shown to be the provably total operations of the proposed system PET. We also studied various conservative extensions of PET.

The next natural step will be to add the so-called *Join axiom*, which constructs disjoint unions of types named by an operation; it has been widely studied for many systems of explicit mathematics. In order to formulate this axiom, we need a new constant j. Below we write $\Sigma[f, a, x]$ for the formula

$$\exists y \exists z (x = (y, z) \wedge y \mathrel{\dot{\in}} a \wedge z \mathrel{\dot{\in}} fy)$$

Now the Join axioms are given by the following assertions (**J.1**) and (**J.2**):

$$(\mathbf{J.1}) \qquad \Re(a) \wedge (\forall x \mathrel{\dot{\in}} a)\Re(fx) \rightarrow \Re(\mathsf{j}(a, f))$$

$$(\mathbf{J.2}) \qquad \Re(a) \wedge (\forall x \mathrel{\dot{\in}} a)\Re(fx) \rightarrow \forall x(x \mathrel{\dot{\in}} \mathsf{j}(a, f) \leftrightarrow \Sigma[f, a, x])$$

It can be shown that join allows the construction of interesting new types which presumably cannot be generated without it. We conjecture that join does not increase the proof-theoretic strength of PET. We hope to give a detailed proof of our conjecture in a future publication.

In Strahm [19], weak first-order applicative systems for various other complexity classes were proposed. It is expected that the techniques of the present paper readily generalize in order to set up natural systems of types and names for those complexity classes. Details are in preparation and will be included in a further publication.

Finally, we are also interested in weak theories of partial truth as studied by Cantini e.g. in [6]. A first candidate for such a theory is presented in Cantini [6] in form of a theory PTT. This is an extension of Strahm's PT with a partial truth predicate. The formulation of the theory is somehow restrictive as the truth predicate is not allowed in induction formulas. It would be interesting to find out whether this restriction can be relaxed and a more liberal induction principle can be admitted without increasing the proof-theoretic strength of the underlying system. Further, it is natural to ask about the exact correspondence between weak systems of explicit mathematics and weak partial truth theories.

References

[1] Beeson, M. J. *Foundations of constructive mathematics: metamathematical studies.* Springer, Berlin, 1985.

[2] Beeson, M. J. Proving programs and programming proofs. In *Logic, methodology and philosophy of science VII*, Barcan Marcus et. al., Ed. North Holland, Amsterdam, 1986, pp. 51–82.

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

[4] Cantini, A. Feasible operations and applicative theories based on $\lambda\eta$. *Mathematical Logic Quarterly 46*, 3 (2000), 291–312.

[5] Cantini, A. Polytime, combinatory logic and positive safe induction. *Archive for Mathematical Logic 41*, 2 (2002), 169–189.

[6] Cantini, A. Choice and uniformity in weak applicative theories. In *Logic colloquium '01*, M. Baaz, S. Friedman, and J. Krajíček, Eds., vol. 20 of *Lecture Notes in Logic*. Association for Symbolic Logic, 2005, pp. 108–138.

[7] Cook, S. A., and Kapron, B. M. Characterizations of the basic feasible functionals of finite type. In *Feasible Mathematics*, S. R. Buss and P. J. Scott, Eds. Birkhäuser, Basel, 1990, pp. 71–95.

[8] Cook, S. A., and Urquhart, A. Functional interpretations of feasibly constructive arithmetic. *Annals of Pure and Applied Logic 63*, 2 (1993), 103–200.

[9] Feferman, S. A language and axioms for explicit mathematics. In *Algebra and logic*, J. Crossley, Ed., vol. 450 of *Lecture Notes in Mathematics*. Springer, Berlin, 1975, pp. 87–139.

[10] Feferman, S. Recursion theory and set theory: a marriage of convenience. In *Generalized recursion theory II, Oslo 1977*, J. E. Fenstad, R. O. Gandy, and G. E. Sacks, Eds., vol. 94 of *Stud. Logic Found. Math.* North Holland, Amsterdam, 1978, pp. 55–98.

[11] Feferman, S. Constructive theories of functions and classes. In *Logic Colloquium '78*, M. Boffa, D. van Dalen, and K. McAloon, Eds. North Holland, Amsterdam, 1979, pp. 159–224.

[12] Feferman, S., and Jäger, G. Systems of explicit mathematics with non-constructive μ-operator. Part II. *Annals of Pure and Applied Logic 79*, 1 (1996), 37–52.

[13] Ferreira, F. *Polynomial Time Computable Arithmetic and Conservative Extensions.* PhD thesis, Pennsylvania State University, 1988.

[14] Ferreira, F. Polynomial time computable arithmetic. In *Logic and Computation, Proceedings of a Workshop held at Carnegie Mellon University, 1987*, W. Sieg, Ed., vol. 106 of *Contemporary Mathematics*. American Mathematical Society, Providence, Rhode Island, 1990, pp. 137–156.

[15] Jäger, G. Induction in the elementary theory of types and names. In *Computer Science Logic '87*, E. Börger, H. Kleine Büning, and M.M. Richter, Eds., vol. 329 of *Lecture Notes in Computer Science*. Springer, Berlin, 1988, pp. 118–128.

[16] Kapron, B., and Cook, S. A new characterization of type 2 feasibility. *SIAM Journal on Computing 25* (1996), 117–132.

[17] Krähenbühl, J. Explicit mathematics with positive existential comprehension and join. Master's thesis, Institut für Informatik und angewandte Mathematik, Universität Bern, 2006. Available at `http://www.iam.unibe.ch/til/publications`.

[18] Strahm, T. Polynomial time operations in explicit mathematics. *Journal of Symbolic Logic 62*, 2 (1997), 575–594.

[19] Strahm, T. Theories with self-application and computational complexity. *Information and Computation 185* (2003), 263–297.

[20] Strahm, T. A proof-theoretic characterization of the basic feasible functionals. *Theoretical Computer Science 329* (2004), 159–176.

[21] Troelstra, A., and van Dalen, D. *Constructivism in mathematics*, vol. I. North-Holland, Amsterdam, 1988.