Realisability in weak systems of explicit mathematics

Daria Spescha* Thomas Strahm**

August 27, 2010

Abstract

This paper is a direct successor to Spescha and Strahm [12]. Its aim is to introduce a new realisability interpretation for weak systems of explicit mathematics and use it in order to analyze extensions of the theory PET in [12] by the so-called join axiom of explicit mathematics.

1 Introduction

This paper continues the research on weak systems of explicit mathematics in the sense of Feferman [6, 7, 8]. We are interested in a proof-theoretic approach to abstract computations and, in particular, in expressively rich Feferman-style systems which have a strong relationship to classes of computational complexity.

*Institut für Informatik und angewandte Mathematik, Universität Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland. Email: `spescha@iam.unibe.ch`. 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`. The second author is grateful to the Department of Computer Science of the University of Swansea for its hospitality while preparing the final version of this paper.

The operational or applicative core of explicit mathematics includes forms of combinatory logic and hence comprises a computationally complete functional language with the full defining power of the untyped lambda calculus. In this sense it is more expressive than standard arithmetical systems. Apart from operations or rules, the second basic entity in explicit mathematics are types, which can be thought of as successively generated collections of operations. In addition, and this is essential in the explicit approach, extensional types are represented (or named) by intensional operations, uniformly in their parameters. This interplay of operations and types on the level of representations makes explicit mathematics very powerful.

There are numerous previous contributions to weak first-order applicative theories which are relevant to the full systems of explicit mathematics studied in this article, cf. Cantini [4, 5, 3], Calamai [2] and Strahm [13, 14]. For a survey of results, see Strahm [15].

The extension of weak first-order applicative theories to the full language of explicit mathematics was initiated in Spescha and Strahm [12], where a natural extension of the first order applicative theory PT (cf. Strahm [13]) was proposed. The corresponding system PET features a weak explicit type system with restricted elementary comprehension; its provably total functions on binary words are the functions which are computable in polynomial time, FPTIME. The upper bound computations for PET in [12] have been obtained by a model-theoretic argument.

The present article is a direct successor to Spescha and Strahm [12]. Its aim is twofold. Firstly, we study a new realisability interpretation for weak systems of explicit mathematics. The realisability relation used in [13, 5] in the first-order context is extended to the full language of types and names. Using this new interpretation, one obtains a purely syntactical proof of the upper bound for PET. Secondly, we use the interpretation in order to study the extension of PET by the well-known join or disjoint union type constructor J of explicit mathematics, resulting in the system PETJ. The main result is that the provably terminating functions of the system PETJ^i, i.e. PETJ with intuitionistic logic, are still the polynomial time computable ones. The proof uses a combination of partial cut elimination and our new realisability interpretation.

The plan of this paper is as follows. In section 2, we will recapitulate the theory PT of [13] and the theory PET of [12], together with some of their extensions. In section 3 we introduce the join axioms, resulting in the theory PETJ. For later proof-theoretic analysis, we reformulate PETJ in a way which avoids type variables. Further, we discuss sequent-style reformulations of our systems and a preparatory partial cut elimination. In section 4, we define a standard model of PETJ. Section 5 constitutes the core of this article. It introduces our new realisability interpretation and contains a proof of the main realisability theorem, revealing that the provably total functions of PETJ^i are the ones computable in polynomial time. In section 6 we discuss the realisability of some extensions of PETJ^i. Finally, we conclude the paper by addressing the classical versions of our systems in section 7. In particular, we mention some very recent work of Probst [10] in this respect.

This paper is based on the second part of Spescha's PhD thesis [11].

2 Recapitulating the theories PT and PET

In this section we quickly sketch the (general) setting of explicit mathematics and recapitulate two theories that play a crucial role in this paper. The theory PT of polynomial time operations is an applicative theory introduced in Strahm [13] where it is proved that its provably total functions are the polynomial time computable ones. PT serves as the first order basis for the second order theory PET introduced in Spescha and Strahm [12]. The provably total functions of PET are still the polynomial time computable ones as we proved in [12].

The language $\mathcal{L}_{\mathsf{W}}$ of PT is a language of partial terms with *individual variables* $a, b, c, x, y, z, u, v, w, f, g, h, \ldots$ (possibly with subscripts). $\mathcal{L}_{\mathsf{W}}$ 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,

$\mathcal{L}_{\mathsf{W}}$ 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 $\mathcal{L}_{\mathsf{W}}$ (possibly with subscripts) are inductively generated from the variables and constants by means of application $\cdot$. In the following we usually 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 $\mathcal{L}_{\mathsf{W}}$ (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. We use the following conventions concerning substitutions: 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 context we often write $A[\vec{x}]$ instead of A and $A[\vec{s}]$ instead of $A[\vec{s}/\vec{x}]$.

Since our theories are based on the logic of partial terms LPT, term application is only partial and $t\downarrow$ signifies that the term t is defined or has a value. To improve readability, we use the following abbreviations:

$$s \simeq t := (s\downarrow \vee t\downarrow) \to (s = t) \qquad \mathsf{l_W} s := 1 \times s$$
$$(s,t) := \mathsf{p}st \qquad (t)_i := \mathsf{p}_i t \quad (i = 0,1)$$
$$0 := \mathsf{s}_0\epsilon \qquad 1 := \mathsf{s}_1\epsilon$$
$$s \subseteq t := \mathsf{c}_{\subseteq}st = 0 \qquad s \leq t := \mathsf{l_W}s \subseteq \mathsf{l_W}t$$

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

$$\begin{aligned} \vec{s} \in \mathsf{W} &:= \mathsf{W}(s_1) \wedge \cdots \wedge \mathsf{W}(s_n), \\ \mathsf{W}_a(s) &:= (\mathsf{W}(s) \wedge s \leq a), \\ (\exists x \in \mathsf{W})A &:= (\exists x)(x \in \mathsf{W} \wedge A), \\ (\forall x \in \mathsf{W})A &:= (\forall x)(x \in \mathsf{W} \to A), \end{aligned}$$

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

The underlying logic of PT is the classical logic of partial terms due to Beeson [1]. It is based on common Hilbert calculus with equality, but quantifiers range over defined objects only:

(Q1) $\forall x A[x] \wedge t{\downarrow} \to A[t]$

(Q2) $A[t] \wedge t{\downarrow} \to \exists x A[x]$

(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)

We are now ready to recapitulate the basic theory B which is used as the foundation for PT and PET. It consists of the following axiom groups defining the behavior of the built-in operators and predicates:

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

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

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

(8) $\quad \mathsf{p_W} : \mathsf{W} \mapsto \mathsf{W} \ \wedge \ \mathsf{p_W}\epsilon = \epsilon$,

(9) $\quad x \in \mathsf{W} \ \rightarrow \ \mathsf{p_W}(\mathsf{s}_0 x) = x \ \wedge \ \mathsf{p_W}(\mathsf{s}_1 x) = x$,

(10) $\quad x \in \mathsf{W} \wedge x \neq \epsilon \ \rightarrow \ \mathsf{s}_0(\mathsf{p_W} x) = x \vee \mathsf{s}_1(\mathsf{p_W} x) = x$.

IV. Initial Subword Relation

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

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

(13) $\quad x \in \mathsf{W} \wedge y \in \mathsf{W} \wedge y \neq \epsilon \ \rightarrow \ (x \subseteq y \leftrightarrow x \subseteq \mathsf{p_W} y \vee x = y)$.

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

As usual, we can simulate λ abstraction in B as a consequence of the partial combinatory algebra axioms (1) and (2).

Lemma 1 ($\boldsymbol{\lambda}$ Abstraction) *For each term t and all variables x, there is a term $(\lambda x.t)$ whose free variables are those of t except x and such that* B *proves*

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

We generalise λ abstraction to several arguments by writing $(\lambda x_1 \cdots x_n.t)$ for $(\lambda x_1. \cdots (\lambda x_n.t))$.

As a consequence of the enclosure of the partial combinatory algebra axioms, the existence of a recursion or fixed point operator is also derivable in B.

Lemma 2 (Fixed point operator) *There exists a closed term* fix *such that* B *proves*

$$\mathsf{fix}f\downarrow \wedge \mathsf{fix}fx \simeq f(\mathsf{fix}f)x$$

Strahm's theory PT is now defined as the theory B extended by induction for the formula class Σ^b_{W} of formulas of the form $A[x] \equiv (\exists y \leq fx)B[f,x,y]$ where B is positive and does not contain the relation symbol W. Formally, the induction schema is defined as follows:

$$(\Sigma^b_{\mathsf{W}}\text{-}\mathsf{I}_{\mathsf{W}}) \qquad f : \mathsf{W} \mapsto \mathsf{W} \wedge A[\epsilon] \wedge (\forall x \in \mathsf{W})(A[\mathsf{p}_{\mathsf{W}}x] \to A[x]) \\ \to (\forall x \in \mathsf{W})A[x]$$

In weak theories, the strength is usually measured in terms of the provably total functions. Thus, we first formally define this concept. Below, $\mathbb{W}$ denotes the set of finite binary words and for each $w \in \mathbb{W}$, we let $\overline{w}$ denote the canonical closed $\mathcal{L}_{\mathsf{W}}$ term designating w.

Definition 3 (Provably total function) A function $F : \mathbb{W}^n \to \mathbb{W}$ is called provably total in an $\mathcal{L}_{\mathsf{W}}$ theory T iff there exists a closed term t_F such that

1) $\mathsf{T} \vdash t_F : \mathsf{W}^n \mapsto \mathsf{W}$ and

2) $\mathsf{T} \vdash t_F\overline{w_1}\cdots\overline{w_n} = \overline{F(w_1 \cdots w_n)}$ for all $w_1, \ldots, w_n \in \mathbb{W}$

Strahm [13] proved that the provably total functions of PT are the polynomial time computable ones. In the proof of the upper bounds he employed a realisability relation which we will extend later.

We are now proceeding to recapitulate the theory PET from Spescha and Strahm [12]. PET stands for the theory of polynomial time operations with explicit types. It is formulated by a finite axiomatisation similar to the one given by Feferman and Jäger [9] for EET. It differs from the theory EET mainly by excluding the complement type constructor and replacing the type of natural numbers by initial segments of the binary words.

PET is formulated in the second order language $\mathcal{L}^2_{\mathsf{W}}$ which extends the language $\mathcal{L}_{\mathsf{W}}$ 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 *individual terms* $r, s, t, \ldots$ of $\mathcal{L}^2_{\mathsf{W}}$ are those of $\mathcal{L}_{\mathsf{W}}$, but taking into account the new constants, whereas the *type terms* consist of the type variables only. The *formulas* $A, B, C, \ldots$ of $\mathcal{L}^2_{\mathsf{W}}$ (possibly with subscripts) are built from the atomic formulas of $\mathcal{L}_{\mathsf{W}}$ 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 $\mathcal{L}^2_{\mathsf{W}}$ formula, we let $\mathrm{FV}_I(A)$ and $\mathrm{FV}_T(A)$ denote the set of its free individual and type variables, respectively. Finally, we write $\mathrm{FV}_I(t)$ for the set of individual variables occurring in the term t.

Types are extensional and their names are intensional in character. As we mostly refer to types by their names, we use the following abbreviations ($\vec{s} = s_1, \ldots, s_n$, $\overrightarrow{X} = X_1, \ldots, X_n$):

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

The logical axioms of PT are extended by the obvious strictness axioms for the new relation symbols of $\mathcal{L}^2_{\mathsf{W}}$. In addition, the logic of the types is just the usual predicate logic with equality. PET consists of the axioms of B plus the following axiom groups 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 and finally, we include the type induction axiom in group III.

I. Explicit representation and extensionality

(O.1) $\exists x \Re(x, X)$,

(O.2) $\Re(a, X) \wedge \Re(a, Y) \to X = Y$,

(O.3) $\forall z (z \in X \leftrightarrow z \in Y) \to X = Y$.

II. Type existence axioms

($\mathbf{w}_a$) $a \in \mathsf{W} \to \Re(\mathsf{w}(a)) \land \forall x(x \mathrel{\dot\in} \mathsf{w}(a) \leftrightarrow \mathsf{W}_a(x))$,

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

($\mathbf{un}$) $\Re(a) \land \Re(b) \to \Re(\mathsf{un}(a, b)) \land \forall x(x \mathrel{\dot\in} \mathsf{un}(a, b) \leftrightarrow (x \mathrel{\dot\in} a \lor x \mathrel{\dot\in} b))$,

($\mathbf{int}$) $\Re(a) \land \Re(b) \to \Re(\mathsf{int}(a, b)) \land \forall x(x \mathrel{\dot\in} \mathsf{int}(a, b) \leftrightarrow (x \mathrel{\dot\in} a \land x \mathrel{\dot\in} b))$,

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

($\mathbf{inv}$) $\Re(a) \to \Re(\mathsf{inv}(f, a)) \land \forall x(x \mathrel{\dot\in} \mathsf{inv}(f, a) \leftrightarrow fx \mathrel{\dot\in} a)$.

III. Type induction on W

(T-I_{W}) $\epsilon \in X \land (\forall x \in \mathsf{W})(\mathsf{p}_{\mathsf{W}} x \in X \to x \in X) \to (\forall x \in \mathsf{W})(x \in X)$

In [12], we established the proof-theoretic strength of PET as follows:

Theorem 4 (Strength of PET) *The provably total functions of* PET *are those computable in polynomial time.*

In the course of proving this theorem, we also stated a comprehension scheme for PET. Comprehension is available for the class of so-called $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formulas. As the name already suggests, they are not only a subset of the elementary formulas, but also closely related to the class of Σ^b_{W} formulas used for the induction scheme in PT. The relation symbol W is only allowed for bounded terms t in the form of $\mathsf{W}_a(t)$ (as defined on page 5). Furthermore, we must ensure that bounds of subformulas do not interfere with each other in composed formulas. Therefore, simultaneously with the notion of $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formulas, we have to define a set of designated free individual variables $\mathrm{FV}_{\mathsf{W}}(A)$ which shall be thought of as the 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 5 ($\mathbf{\Sigma^{b}_{T}}$ formulas) The class of $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formulas of $\mathcal{L}^2_{\mathsf{W}}$ and the set of variables $\mathrm{FV}_{\mathsf{W}}(A)$ are inductively defined as follows:

1) If A is an $\mathcal{L}^2_{\mathsf{W}}$ formula of the form $(s = t)$, $s{\downarrow}$ or $(s \in X)$, then A is a $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formula and $\mathrm{FV}_{\mathsf{W}}(A) := \emptyset$.

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

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,

$$(\mathrm{FV}_I(B) \setminus \mathrm{FV}_{\mathsf{W}}(B)) \cap \mathrm{FV}_{\mathsf{W}}(C) = \emptyset,$$
$$(\mathrm{FV}_I(C) \setminus \mathrm{FV}_{\mathsf{W}}(C)) \cap \mathrm{FV}_{\mathsf{W}}(B) = \emptyset,$$

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

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

In [12], we proved that the following comprehension scheme is available in the system PET:

Theorem 6 (Restricted elementary comprehension) *Assume that $A[x, \vec{v}, \vec{w}, \vec{X}]$ is a $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formula with the following free variables:*

$$\begin{aligned} \mathrm{FV}_T(A) &= \{X_1, \ldots, X_n\}, \\ \mathrm{FV}_{\mathsf{W}}(A) &= \{w_1, \ldots, w_m\}, \\ \mathrm{FV}_I(A) \setminus \mathrm{FV}_{\mathsf{W}}(A) &= \{x, v_1, \ldots, v_k\}. \end{aligned}$$

Then we can find a closed $\mathcal{L}^2_{\mathsf{W}}$ 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) $\mathsf{W}(\vec{w}) \wedge \Re(\vec{z}, \vec{X}) \rightarrow (\forall x)(x \,\dot{\in}\, c_A(\vec{v}, \vec{w}, \vec{z}) \leftrightarrow A[x, \vec{v}, \vec{w}, \vec{X}])$.

Together with (T-I_{W}), comprehension enables us to employ formula induction for $\Sigma^{\mathsf{b}}_{\mathrm{T}}$ formulas.

We now quickly mention some extensions for PET. Applicative theories are based on the logic of partial terms where operations are partial and terms may be undefined. However, sometimes systems with totality of application are better suited. For this purpose, we introduce the totality axiom:

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

In the presence of (**Tot**), every term is provably defined. Therefore, the underlying logic of partial terms can be replaced by ordinary predicate logic.

Although applicative theories focus on the intensional aspect of operations, extensionality for operations is sometimes desired. Therefore we can add an axiom stating that two operations are equal if they produce the same result for all arguments:

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

Cantini [5] adds a uniformity principle for positive $\mathcal{L}_{\mathsf{W}}$ formulas to PT and proves that this yields a theory whose provably total functions are still those computable in polynomial time. Cantini formulates the uniformity principle for a truth predicate defined for positive formulas. In our context, we can state Cantini's principle as follows. For each *positive* $\mathcal{L}_{\mathsf{W}}$ formula $A[x, y]$:

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

We exploit the fact that $(\exists y \leq t) A \equiv (\exists y \in \mathsf{W})(y \leq t \wedge A)$ which is obviously a positive formula. Therefore, we can specify the following form of *bounded uniformity* for positive $\mathcal{L}_{\mathsf{W}}$ formulas $A[x, y]$ which is readily entailed by (**UP**):

(**UP'**) $\forall x (\exists y \leq t) A[x, y] \to (\exists y \leq t) \forall x A[x, y]$

The principle (**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. We first add an additional constant all to $\mathcal{L}^2_{\mathsf{W}}$ and spell out the axiom as follows:

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

In [12] we proved that we can add this additional type constructor without strengthening the theory:

Theorem 7 *The provably total functions of* PET *augmented by any combinations of* (***all***), (***Tot***) *and* (***Ext***) *coincide with the polynomial time computable functions.*

3 Adding join

In this section we are discussing the addition of disjoint unions to the theory PET. We will first state the so-called join axioms and then reformulate the theory as a first order theory where types are only available in the form of names. Finally, we will give a sequent calculus reformulation of this theory.

In explicit mathematics, join is defined by the following two axioms, where j is supposed to be a new constant of the underlying language:

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

(J.2) $\Re(a) \wedge (\forall x \mathrel{\dot\in} a)\Re(fx) \rightarrow \forall x(x \mathrel{\dot\in} \mathsf{j}(a,f) \leftrightarrow$
$\exists y \exists z(x = (y,z) \wedge y \mathrel{\dot\in} a \wedge z \mathrel{\dot\in} fy))$

Let PETJ stand for the theory PET augmented by the join axioms.

For our subsequent proof-theoretic analysis, it will be more convenient to work with a first order version of PETJ. In this context, we only have names representing types instead of (second order) types. Therefore, $\Re$ is in this context a unary relation symbol denoting the collection of names. The elementhood relation is defined between two individuals where one of them is expected to be a name. Furthermore, the axioms about extensionality are dropped. This reformulation does not change the proof-theoretic strength as we will prove later.

PETJ_1 is formulated in the language $\mathcal{L}_{\mathsf{W}}^{T}$ which extends $\mathcal{L}_{\mathsf{W}}$ by a *unary* relation symbol $\Re$, binary relation symbol $\dot\in$ and the (individual) constants w, id, dom, un, int, inv, and j (disjoint union). The relation $\dot\in$ connects a name with the elements of its extension. In contrast to the previous section, it is not an abbreviation, but a relation symbol of the language. Of course, the semantics of this relation shall match those of the abbreviation.

Formally, the theory PETJ_1 consists of the axioms of B plus axioms about type construction and type induction. The axioms look the same as those of PETJ, since we used the abbreviation $\dot\in$ in the original formulation. In our present setting, we only have names and therefore loose the extensionality of types. Thus, we introduce a new abbreviation $\doteq$ for stating that two names are extensionally equal:

$$a \doteq b := \forall x(x \mathrel{\dot\in} a \leftrightarrow x \mathrel{\dot\in} b)$$

Since our main target is the analysis of the proof-theoretic strength of adding the join axioms to (the original formulation of) PET, we first have to translate formulas from the original version into the first order version preserving provability. In the following, $\mathcal{L}^2_\mathsf{W}$ is assumed to include the constant j.

Definition 8 (Translation from $\mathcal{L}^2_\mathsf{W}$ to $\mathcal{L}^T_\mathsf{W}$) $\cdot^\star$ translates any $\mathcal{L}^2_\mathsf{W}$ formula A into a formula $A^\star$ of $\mathcal{L}^T_\mathsf{W}$. First, we assume there is a new (individual) variable a_X for every type variable X. The translation is now defined by induction on the construction of A:

A atomic:

$$A \equiv s = t \mid s\downarrow \mid \mathsf{W}(s) \quad \Longrightarrow \quad A^\star \equiv A$$

$$A \equiv X = Y \quad \Longrightarrow \quad A^\star \equiv a_X \doteq a_Y$$

$$A \equiv \Re(s, X) \quad \Longrightarrow \quad A^\star \equiv \Re(s) \wedge s \doteq a_X$$

$$A \equiv s \in X \quad \Longrightarrow \quad A^\star \equiv s \mathrel{\dot\in} a_X$$

A composite formula:

$$A \equiv B \otimes C \ (\otimes = \wedge, \vee, \rightarrow) \quad \Longrightarrow \quad A^\star \equiv B^\star \otimes C^\star$$

$$A \equiv \neg B \quad \Longrightarrow \quad A^\star \equiv \neg B^\star$$

$$A \equiv QxB \ (Q = \forall, \exists) \quad \Longrightarrow \quad A^\star \equiv QxB^\star$$

$$A \equiv \forall XB \quad \Longrightarrow \quad A^\star \equiv \forall x(\Re(x) \rightarrow B^\star[x/a_X])$$

$$A \equiv \exists XB \quad \Longrightarrow \quad A^\star \equiv \exists x(\Re(x) \wedge B^\star[x/a_X])$$

We are ready to state the equivalence of the two theories:

Lemma 9 *For any $\mathcal{L}^2_\mathsf{W}$ formula $A[\vec{X}]$ where $\vec{X}$ is a conclusive enumeration of $\mathrm{FV}_T(A)$ we have:*

$$\mathsf{PETJ} \vdash A[\vec{X}] \quad \Longrightarrow \quad \mathsf{PETJ}_1 \vdash \Re(\vec{a}_X) \rightarrow A^\star[\vec{a}_X]$$

Proof The proof is by induction on the length of a proof of A in PETJ. It is routine and spelled out in detail in [11]. □

Below we will give a proof-theoretic analysis of PETJ on the basis of intuitionistic logic only. Accordingly, we let PETJ^i and PETJ^i_1 denote the intuitionistic version of PETJ and PETJ_1, respectively. In the following upper bound computations we implicitly assume that our applicative axioms satisfy the axioms of totality (**Tot**) and extensionality (**Ext**); in particular, we will work in the framework of ordinary predicate logic in the sequel, cf. [13] for a similar procedure.

We reformulate the theory PETJ_1^i in sequent calculus style where all the main formulas are positive. We are working in the same language $\mathcal{L}_{\mathsf{W}}^T$ as before and make use of the same abbreviations.

The theory PETJ_1^{iG} is the reformulation of the theory PETJ_1^i in Gentzen style. The axioms of the first order part are adopted from Strahm [13] and therefore omitted. In the following, we will write $\Gamma, \Delta, \dots$ for finite sequences of $\mathcal{L}_{\mathsf{W}}^T$ formulas.

PETJ_1^{iG} consists of the axioms and rules of intuitionistic sequent calculus, the reformulation of B as well as the following axioms and rules:

I. Type existence axioms

$(\mathbf{w}_a.1)$	$\Gamma, \mathsf{W}(s) \Rightarrow \Re(\mathsf{w}(s))$
$(\mathbf{w}_a.2)$	$\Gamma, \mathsf{W}(s), \mathsf{W}(t), t \leq s \Rightarrow t \mathrel{\dot\in} \mathsf{w}(s)$
$(\mathbf{w}_a.3)$	$\Gamma, \mathsf{W}(s), t \mathrel{\dot\in} \mathsf{w}(s) \Rightarrow \mathsf{W}(t) \wedge t \leq s$
(id.1)	$\Gamma \Rightarrow \Re(\mathsf{id})$
(id.2)	$\Gamma, \exists u(t = (u,u)) \Rightarrow t \mathrel{\dot\in} \mathsf{id}$
(id.3)	$\Gamma, t \mathrel{\dot\in} \mathsf{id} \Rightarrow \exists u(t = (u,u))$
(inv.1)	$\Gamma, \Re(s) \Rightarrow \Re(\mathsf{inv}(r,s))$
(inv.2)	$\Gamma, \Re(s), rt \mathrel{\dot\in} s \Rightarrow t \mathrel{\dot\in} \mathsf{inv}(r,s)$
(inv.3)	$\Gamma, \Re(s), t \mathrel{\dot\in} \mathsf{inv}(r,s) \Rightarrow rt \mathrel{\dot\in} s$
(dom.1)	$\Gamma, \Re(s) \Rightarrow \Re(\mathsf{dom}(s))$
(dom.2)	$\Gamma, \Re(s), \exists y((t,y) \mathrel{\dot\in} s) \Rightarrow t \mathrel{\dot\in} \mathsf{dom}(s)$
(dom.3)	$\Gamma, \Re(s), t \mathrel{\dot\in} \mathsf{dom}(s) \Rightarrow \exists y((t,y) \mathrel{\dot\in} s)$
(un.1)	$\Gamma, \Re(s_0), \Re(s_1) \Rightarrow \Re(\mathsf{un}(s_0,s_1))$
(un.2)	$\Gamma, \Re(s_0), \Re(s_1), t \mathrel{\dot\in} s_0 \Rightarrow t \mathrel{\dot\in} \mathsf{un}(s_0,s_1)$
(un.3)	$\Gamma, \Re(s_0), \Re(s_1), t \mathrel{\dot\in} s_1 \Rightarrow t \mathrel{\dot\in} \mathsf{un}(s_0,s_1)$
(un.4)	$\Gamma, \Re(s_0), \Re(s_1), t \mathrel{\dot\in} \mathsf{un}(s_0,s_1) \Rightarrow t \mathrel{\dot\in} s_0 \vee t \mathrel{\dot\in} s_1$

$$
\begin{array}{ll}
\textbf{(int.1)} & \Gamma, \Re(s_0), \Re(s_1) \Rightarrow \Re(\mathsf{int}(s_0, s_1)) \\
\textbf{(int.2)} & \Gamma, \Re(s_0), \Re(s_1), t \mathrel{\dot\in} s_0, t \mathrel{\dot\in} s_1 \Rightarrow t \mathrel{\dot\in} \mathsf{int}(s_0, s_1) \\
\textbf{(int.3)} & \Gamma, \Re(s_0), \Re(s_1), t \mathrel{\dot\in} \mathsf{int}(s_0, s_1) \Rightarrow t \mathrel{\dot\in} s_0 \\
\textbf{(int.4)} & \Gamma, \Re(s_0), \Re(s_1), t \mathrel{\dot\in} \mathsf{int}(s_0, s_1) \Rightarrow t \mathrel{\dot\in} s_1
\end{array}
$$

II. Join Rules

$$
(\mathbf{J.1})\ \frac{\Gamma, x \mathrel{\dot\in} s \Rightarrow \Re(rx) \qquad \Gamma \Rightarrow \Re(s)}{\Gamma \Rightarrow \Re(\mathsf{j}(s, r))}\ *
$$

$$
(\mathbf{J.2})\ \frac{\Gamma, x \mathrel{\dot\in} s \Rightarrow \Re(rx) \qquad \Gamma \Rightarrow \Re(s)}{\Gamma, t \mathrel{\dot\in} \mathsf{j}(s, r) \Rightarrow t = ((t)_0, (t)_1) \wedge (t)_0 \mathrel{\dot\in} s \wedge (t)_1 \mathrel{\dot\in} r(t)_0}\ *
$$

$$
(\mathbf{J.3})\ \frac{\Gamma, x \mathrel{\dot\in} s \Rightarrow \Re(rx) \qquad \Gamma \Rightarrow \Re(s)}{\Gamma, t = ((t)_0, (t)_1), (t)_0 \mathrel{\dot\in} s, (t)_1 \mathrel{\dot\in} r(t)_0 \Rightarrow t \mathrel{\dot\in} \mathsf{j}(s, r)}\ *
$$

III. Type Induction

$$
(\text{T-I}_{\mathsf{W}})\ \frac{\Gamma \Rightarrow \Re(s) \qquad \Gamma \Rightarrow \epsilon \mathrel{\dot\in} s \qquad \Gamma, \mathsf{W}(x), x \mathrel{\dot\in} s \Rightarrow \mathsf{s}_i x \mathrel{\dot\in} s}{\Gamma, \mathsf{W}(t) \Rightarrow t \mathrel{\dot\in} s}\ {}_{i=0,1}\ *
$$

*: x not free in Γ

In the following, we let $\bigwedge \Gamma$ abbreviate $A_0 \wedge \cdots \wedge A_n$ for $\Gamma = A_0, \ldots A_n$. We now state that this sequent-style reformulation is indeed adequate:

Lemma 10 (Equivalence of PETJ_1^i and PETJ_1^{iG}) *For all formulas C and all sequents $\Gamma \Rightarrow C$ we have*

1) $\mathsf{PETJ}_1^i \vdash C \implies \mathsf{PETJ}_1^{iG} \vdash \Rightarrow C$

2) $\mathsf{PETJ}_1^{iG} \vdash \Gamma \Rightarrow C \implies \mathsf{PETJ}_1^i \vdash \bigwedge \Gamma \to C$

Proof Proof by induction on the proof height. The routine proof is given in Spescha [11]. □

For the realisability interpretation, we depend on partial cut elimination, i.e. only cuts with positive formulas are allowed in our proofs. For non-positive formulas, the rank is defined as usual as the maximum of the ranks of its subformulas plus one and positive formulas have rank 0. Also $\mathsf{PETJ}_1^{iG} \mid\!\frac{}{r}\, \Gamma \Rightarrow C$ signifies as usual that PETJ_1^{iG} proves the sequent $\Gamma \Rightarrow C$ with a proof where all cut-formulas have rank less than r. We can now state partial cut reduction for non-positive formulas:

Lemma 11 (Partial Cut Reduction) *If* $\mathsf{PETJ}_1^{iG} \mid\!\frac{}{r}\, \Gamma \Rightarrow A$ *and* $\mathsf{PETJ}_1^{iG} \mid\!\frac{}{r}\, \Gamma', A \Rightarrow C$ *with* $\mathrm{rk}(A) = r > 0$*, then* $\mathsf{PETJ}_1^{iG} \mid\!\frac{}{r}\, \Gamma, \Gamma' \Rightarrow C$.

Thus, each proof in PETJ_1^{iG} can be transformed into a proof which has only positive cuts. By the subformula property, provable sequents of positive formulas have proofs consisting entirely of positive formulas.

4 A model for PETJ_1

In this section, we describe a model for PETJ_1 based on the well-known term model $\mathcal{M}(\lambda\eta)$, cf. [13, 12]. The construction is similar to the one presented in [12], but the stages for the construction of the interpretation of the names now run over all ordinals instead of the natural numbers as before. First, we give a definition of an $\mathcal{L}_{\mathsf{W}}^T$-structure.

Definition 12 ($\mathcal{L}_{\mathsf{W}}^T$-Structure) An $\mathcal{L}_{\mathsf{W}}^T$-structure $\mathcal{M}^\star$ is a tuple

$$(\mathcal{M}, \mathcal{R}, \mathcal{E}, \underline{\mathsf{w}}, \underline{\mathsf{id}}, \underline{\mathsf{un}}, \underline{\mathsf{int}}, \underline{\mathsf{dom}}, \underline{\mathsf{inv}}, \underline{\mathsf{j}})$$

meeting the following conditions:

(i) $\mathcal{M}$ is a $\mathcal{L}_{\mathsf{W}}$-structure,

(ii) $\mathcal{R}$ is a non-empty subset of $|\mathcal{M}|$,

(iii) $\mathcal{E}$ is a binary relation on $|\mathcal{M}| \times \mathcal{R}$, and

(iv) $\underline{\mathsf{w}}, \underline{\mathsf{id}}, \underline{\mathsf{un}}, \underline{\mathsf{int}}, \underline{\mathsf{dom}}, \underline{\mathsf{inv}}, \underline{\mathsf{j}}$ are elements of $|\mathcal{M}|$.

For the construction we take the model $\mathcal{M} = \mathcal{M}(\lambda\eta)$ of PT. We have usual $\beta\eta$-reduction adapted to fit our axioms and $|\mathcal{M}| = \{t : t\ \mathcal{L}_{\mathsf{W}}^T \text{ term}\}$.

We write $t \overset{\text{red}}{\Longrightarrow} s$ for reduction of terms and define the abbreviation $t_1 \overset{\beta\eta}{=} t_2 :\Longleftrightarrow t_1 \overset{\text{red}}{\Longrightarrow} s$ and $t_2 \overset{\text{red}}{\Longrightarrow} s$ for some term s.

Now we generate the model $\mathcal{M}^\star$ of PETJ_1 by adding interpretations $\mathcal{R}$ and $\mathcal{E}$ for $\Re$ and $\dot\in$ respectively while the constants are interpreted by themselves. For the construction of $\mathcal{R}$ we introduce sets $\mathcal{R}_\alpha \subseteq |\mathcal{M}^\star|$ by induction on the ordinal α and simultaneously establish a set $\mathcal{E}_\alpha \subseteq |\mathcal{M}| \times \mathcal{R}_\alpha$. For every ordinal number α, $\mathcal{R}_\alpha$ and $\mathcal{E}_\alpha$ are constructed as follows where $r, s, t \in |\mathcal{M}(\lambda\eta)|$.

$\alpha = 0$: $\mathcal{R}_0$ contains the names of the base types, i.e. formally $s \in \mathcal{R}_0$ iff

- $s \overset{\beta\eta}{=} \mathsf{id}$ and $(t, s) \in \mathcal{E}_0$ iff $t \overset{\beta\eta}{=} (m, m)$ for some $m \in |\mathcal{M}|$.
- $s \overset{\beta\eta}{=} \mathsf{w}a$ with $a \in \mathsf{W}^{\mathcal{M}}$ and $(t, s) \in \mathcal{E}_0$ iff $\mathcal{M} \models t \in \mathsf{W} \wedge t \leq a$.

$\alpha = \beta + 1$ *successor ordinal*: $\mathcal{R}_\beta \subseteq \mathcal{R}_\alpha$ and $\mathcal{E}_\beta \subseteq \mathcal{E}_\alpha$. In addition, for $s_0, s_1 \in \mathcal{R}_\beta$, $s \in \mathcal{R}_\alpha$ iff

- $s \overset{\beta\eta}{=} \mathsf{un}(s_0, s_1)$ and $(t, s) \in \mathcal{E}_\alpha$ iff $(t, s_0) \in \mathcal{E}_\beta$ or $(t, s_1) \in \mathcal{E}_\beta$.
- $s \overset{\beta\eta}{=} \mathsf{int}(s_0, s_1)$ and $(t, s) \in \mathcal{E}_\alpha$ iff $(t, s_0) \in \mathcal{E}_\beta$ and $(t, s_1) \in \mathcal{E}_\beta$.
- $s \overset{\beta\eta}{=} \mathsf{dom}(s_0)$ and $(t, s) \in \mathcal{E}_\alpha$ iff there is a $m \in |\mathcal{M}|$ such that $((t, m), s_0) \in \mathcal{E}_\beta$.
- $s \overset{\beta\eta}{=} \mathsf{inv}(r, s_0)$ and $(t, s) \in \mathcal{E}_\alpha$ iff $(rt, s_0) \in \mathcal{E}_\beta$
- $s \overset{\beta\eta}{=} \mathsf{j}(s_0, r)$ and $rt \in \mathcal{R}_\beta$ for all t such that $(t, s_0) \in \mathcal{E}_\beta$. Furthermore, $(t, s) \in \mathcal{E}_\alpha$ iff $t \overset{\beta\eta}{=} (m, n)$ such that $(m, s_0) \in \mathcal{E}_\beta$ and $(n, rm) \in \mathcal{E}_\beta$.

$\alpha =$ *limit ordinal*: $\mathcal{R}_\alpha = \bigcup_{\beta<\alpha} \mathcal{R}_\beta$ and $\mathcal{E}_\alpha = \bigcup_{\beta<\alpha} \mathcal{E}_\beta$.

Finally, we define $\mathcal{R} := \bigcup_{\alpha \in \Omega} \mathcal{R}_\alpha$ and $\mathcal{E} := \bigcup_{\alpha\in\Omega} \mathcal{E}_\alpha$ where Ω stands for the ordinals. Then our desired $\mathcal{L}^T_{\mathsf{W}}$ structure is given by

$$\mathcal{M}^\star \;:=\; (\mathcal{M}(\lambda\eta), \mathcal{R}, \mathcal{E}, \mathsf{w}, \mathsf{id}, \mathsf{un}, \mathsf{int}, \mathsf{dom}, \mathsf{inv}, \mathsf{j}).$$

For any $s \in \mathcal{R}_\alpha$, we define $ext(s) := \{t \in |\mathcal{M}^\star| : (t, s) \in \mathcal{E}_\alpha\}$. In the following, we use the abbreviation $t \; \varepsilon_\alpha \; s := (t, s) \in \mathcal{E}_\alpha$.

It immediately follows from our construction that $\mathcal{M}^\star$ indeed is a model for PETJ_1. When referencing $\mathcal{M}^\star$ from now on, we always refer to the specific model of PETJ_1 as constructed here, unless explicitly stated otherwise.

5 Realisability for positive formulas

In this section we define the notion of realisability for positive formulas of $\mathcal{L}_{\mathsf{W}}^{T}$. Realisers are binary words and shall contain some computational information. They can be seen as witnesses for the statement of the formula being realised. The definition in this paper is an extension of the one introduced in Strahm [13] for the first order language $\mathcal{L}_{\mathsf{W}}$. A similar relation in the first-order context of safe induction has been employed in Cantini [4].

In the sequel, we are mainly interested in statements concerning the predicate W, as the realisability is a means to prove that the provably total functions are the ones computable in polynomial time. In Theorem 17 we will show that the conclusion of every provable sequent can be realised by a polynomial time function from the realisers of the premise. The desired upper bounds immediately follow from this fact.

Recall that for realising provable sequents, we work in the theory PETJ_1^{iG} including the axioms for totality (**Tot**) and extensionality (**Ext**) as introduced before. In this context, the relation symbol $\downarrow$ becomes superfluous and can therefore be neglected.

Furthermore, the definition of the realisability depends on the model $\mathcal{M}^\star$ presented in the previous section. We first define the notion of realisability for formulas of the form $t \mathrel{\dot\in} s$. Below $\langle \cdot, \cdot \rangle$ abbreviates a polynomial time pairing function (cf. e.g. [11]).

Definition 13 (Realisability: $\boldsymbol{\rho}$ $\check{\mathrm{r}}$ $\boldsymbol{t \mathrel{\dot\in} s}$) For any $s \in \mathcal{R}$ and $\rho \in \mathbb{W}$, the notion $\rho \ \check{\mathrm{r}}_\alpha \ t \mathrel{\dot\in} s$ is defined by induction on the level α at which s was added to $\mathcal{R}$.

$\alpha = 0$:

$$\rho \ \check{\mathrm{r}}_0 \ t \mathrel{\dot\in} \mathsf{id} \quad \Longleftrightarrow \quad \rho = \epsilon \text{ and } \mathcal{M}^\star \models t = (t_0, t_0) \text{ for some term } t_0$$

$$\rho \ \check{\mathrm{r}}_0 \ t \mathrel{\dot\in} \mathsf{w}(s) \quad \Longleftrightarrow \quad \mathcal{M}^\star \models t = \overline{\rho} \wedge \overline{\rho} \leq s$$

$\alpha = \beta + 1$ *successor ordinal* where $s, s_0, s_1 \in \mathcal{R}_\beta$:

$$\rho \ \check{\mathrm{r}}_\alpha \ t \mathrel{\dot\in} s \quad \Longleftrightarrow \quad \rho \ \check{\mathrm{r}}_\beta \ t \mathrel{\dot\in} s$$

$$\rho \ \check{\mathrm{r}}_\alpha \ t \mathrel{\dot\in} \mathsf{dom}(s) \quad \Longleftrightarrow \quad \rho \ \check{\mathrm{r}}_\beta \ (t, t_0) \mathrel{\dot\in} s \text{ for a term } t_0$$

$$
\begin{array}{lcl}
\rho \ \check{\mathrm{r}}_\alpha \ t \mathrel{\dot\in} \mathsf{un}(s_0, s_1) & \iff & \rho = \langle i, \rho_0 \rangle \text{ and either} \\
 & & i = 0 \text{ and } \rho_0 \ \check{\mathrm{r}}_\beta \ t \mathrel{\dot\in} s_0 \text{ or} \\
 & & i = 1 \text{ and } \rho_0 \ \check{\mathrm{r}}_\beta \ t \mathrel{\dot\in} s_1 \\
\rho \ \check{\mathrm{r}}_\alpha \ t \mathrel{\dot\in} \mathsf{int}(s_0, s_1) & \iff & \rho = \langle \rho_0, \rho_1 \rangle \text{ and } \rho_0 \ \check{\mathrm{r}}_\beta \ t \mathrel{\dot\in} s_0 \text{ and } \rho_1 \ \check{\mathrm{r}}_\beta \ t \mathrel{\dot\in} s_1 \\
\rho \ \check{\mathrm{r}}_\alpha \ t \mathrel{\dot\in} \mathsf{inv}(r, s) & \iff & \rho \ \check{\mathrm{r}}_\beta \ rt \mathrel{\dot\in} s \\
\rho \ \check{\mathrm{r}}_\alpha \ t \mathrel{\dot\in} \mathsf{j}(s, r) & \iff & \rho = \langle \rho_0, \rho_1 \rangle \text{ and } \mathcal{M}^\star \models t = ((t)_0, (t)_1) \text{ and} \\
 & & \rho_0 \ \check{\mathrm{r}}_\beta \ (t)_0 \mathrel{\dot\in} s \text{ and } \rho_1 \ \check{\mathrm{r}}_\beta \ (t)_1 \mathrel{\dot\in} r(t)_0
\end{array}
$$

α *limit ordinal*:

$$
\rho \ \check{\mathrm{r}}_\alpha \ t \mathrel{\dot\in} s \iff \rho \ \check{\mathrm{r}}_\beta \ t \mathrel{\dot\in} s \text{ for some } \beta < \alpha
$$

To improve readability in this definition, *the name s is a placeholder* for all terms t such that $t \overset{\beta\eta}{=} s$.

We will also write $\rho \ \check{\mathrm{r}} \ t \mathrel{\dot\in} s$ if $\rho \ \check{\mathrm{r}}_\alpha \ t \mathrel{\dot\in} s$ for some ordinal α. We can now define the actual realisability for positive formulas of $\mathcal{L}^T_{\mathsf{W}}$.

Definition 14 (Realisability: $\rho \circledR A$) The realisability relation for positive formulas $\circledR \subseteq \mathbb{W} \times \mathsf{Pos}$ is defined by induction on the construction of the formula:

Atomic formulas:

$$
\begin{array}{lcl}
\rho \circledR \mathsf{W}(t) & \iff & \mathcal{M}^\star \models \overline{\rho} = t \\
\rho \circledR t_0 = t_1 & \iff & \rho = \epsilon \text{ and } \mathcal{M}^\star \models t_0 = t_1 \\
\rho \circledR t \mathrel{\dot\in} s & \iff & s \in \mathcal{R} \text{ and } \rho \ \check{\mathrm{r}}_\alpha \ t \mathrel{\dot\in} s \text{ for some } \alpha \\
\rho \circledR \Re(s) & \iff & s \in \mathcal{R} \text{ and } \forall \sigma, t : \sigma \circledR t \mathrel{\dot\in} s \implies \sigma \leq \rho
\end{array}
$$

Composite formulas:

$$
\begin{array}{lcl}
\rho \circledR A_0 \wedge A_1 & \iff & \rho = \langle \rho_0, \rho_1 \rangle \text{ and } \rho_0 \circledR A_0 \text{ and } \rho_1 \circledR A_1 \\
\rho \circledR A_0 \vee A_1 & \iff & \rho = \langle i, \rho_0 \rangle \ (i \in \{0, 1\}) \text{ and } \rho_0 \circledR A_i \\
\rho \circledR \forall x A[x] & \iff & \rho \circledR A[u] \text{ for a fresh variable } u \\
\rho \circledR \exists x A[x] & \iff & \rho \circledR A[t] \text{ for some term } t
\end{array}
$$

$\vec{\rho} \circledR \Gamma$ *for a sequence* $\Gamma = A_0, \ldots, A_n$*:*

$$
\vec{\rho} \circledR \Gamma \iff \vec{\rho} = \rho_0, \ldots, \rho_n \text{ and } \rho_i \circledR A_i
$$

As it will turn out in our realisability theorem below, the crucial clause in the above definition is the one for formulas of the form $\Re(s)$: A realiser ρ of $\Re(s)$ has the important property that it is a bound for any realisers of statements of the form $t \mathrel{\dot\in} s$. This property will be heavily used below in realising the type induction and join rules by polynomial time computable functions.

Before we can prove the main theorem of this section, we first need to state two important properties of this realisability. First, the realiser of a formula shall not be able to distinguish between two terms having a common reduct. Furthermore, whenever we have a realiser for a formula containing free variables, it realises all substitution instances.

Lemma 15 *For positive formulas A and terms t, s, we have*

1) *If $\rho \textcircled{r} A[t]$ and $t \stackrel{\beta\eta}{=} s$, then $\rho \textcircled{r} A[s]$*

2) *If $\rho \textcircled{r} A[u]$, then $\rho \textcircled{r} A[t]$.*

Proof This is obvious from the definition of realisability. □

Further, we also require that we have realisers for all statements of the form $t \mathrel{\dot\in} s$ modelled by $\mathcal{M}^\star$.

Lemma 16 *Assume that t and s are terms such that $\mathcal{M}^\star \models t \mathrel{\dot\in} s$. Then there is a $\rho \in \mathbb{W}$ such that ρ ř $t \mathrel{\dot\in} s$.*

Proof Assume that $\mathcal{M}^\star \models t \mathrel{\dot\in} s$. This implies that $s \in \Re^{\mathcal{M}^\star}$ and $t \in ext(s)$, i.e. $(t, s) \in \mathcal{E}$. Further, $s \in \Re^{\mathcal{M}^\star}$ iff $s \in \mathcal{R}_\alpha$ for some α. The proof is by induction on α. For details we refer to [11]. □

We are now ready to state the realisability theorem. It claims that whenever we can prove a (positive) sequent $\Gamma \Rightarrow C$ in PETJ_1^{iG}, there is a polynomial time computable function F constructing a realiser for C given realisers for Γ. F may depend on the proof of the sequent and therefore the same sequent can have various realising functions constructed from different deductions.

Theorem 17 (Realisability) *For every positive sequent* $\Gamma[\vec{x}] \Rightarrow C[\vec{x}]$ *(with* $\Gamma = A_0[\vec{x}], \dots, A_n[\vec{x}]$*) provable in* PETJ_1^{iG}*, where* $\vec{x}$ *is a conclusive enumeration of the free variables, there is a function* $F \in$ FPTIME *such that for all terms* $\vec{t}$*:*

$$\vec{\rho} \circledR \Gamma[\vec{t}] \implies F(\vec{\rho}) \circledR C[\vec{t}]$$

Proof This proof is by induction on a quasi-cutfree derivation of $\Gamma \Rightarrow C$ (where $\Gamma \equiv A_0, \dots, A_n$).
Derivation length 0, i.e. $\Gamma \Rightarrow C$ *is an axiom.* The proof for the axioms of B is already spelled out by Strahm in [13] and therefore omitted here.

(w$_a$.1) Assume $\rho_n \circledR \mathsf{W}(s)$, that is $s \overset{\beta\eta}{=} \overline{\rho_n}$. We now set F to be $(\lambda\vec{x}.x_n)$. Obviously, $\mathsf{w}(s) \in \mathcal{R}$ by model construction. Thus, only the last condition remains to be shown: As $\sigma \circledR t \mathrel{\dot\in} \mathsf{w}(s) \iff \mathcal{M}^\star \models \overline{\sigma} = t \wedge \overline{\sigma} \leq s$, we know that $\sigma \leq s$ if $\sigma \circledR t \mathrel{\dot\in} \mathsf{w}(s)$ for any t and therefore $\sigma \leq \overline{\rho_n} = F(\vec{\rho})$.

(w$_a$.2) Assume (i) $\rho_{n-2} \circledR \mathsf{W}(s)$, (ii) $\rho_{n-1} \circledR \mathsf{W}(t)$ and (iii) $\rho_n \circledR t \leq s$ where $t \leq s \equiv \mathsf{c}_\subseteq(\mathsf{l}_\mathsf{W} t)(\mathsf{l}_\mathsf{W} s) = 0$. Choose $F = (\lambda\vec{x}.x_{n-1})$. Then $F(\vec{\rho}) = \rho_{n-1}$ and $\rho_{n-1} \circledR t \mathrel{\dot\in} \mathsf{w}(s)$ since $\mathcal{M}^\star \models t = \overline{\rho}_{n-1}$ with (ii) and $\mathcal{M}^\star \models \overline{\rho}_{n-1} \leq s$ because of (iii).

(w$_a$.3) Set $F = (\lambda\vec{x}.\langle x_n, \epsilon\rangle)$. Then $F(\vec{\rho}) \circledR \mathsf{W}(t) \wedge t \leq s$: $\rho_n \circledR \mathsf{W}(t)$ as $\rho_n \circledR t \mathrel{\dot\in} \mathsf{w}(s)$ and $\epsilon \circledR \overline{\rho}_n \leq s$ by assumption.

(id.1) $F = (\lambda\vec{x}.\epsilon)$.

(id.2) $F = (\lambda\vec{x}.\epsilon)$.

(id.3) $F = (\lambda\vec{x}.\epsilon)$.

(inv.1) $F = (\lambda\vec{x}.x_n)$.

(inv.2) $F = (\lambda\vec{x}.x_n)$.

(inv.3) $F = (\lambda\vec{x}.x_n)$.

(dom.1) $F = (\lambda\vec{x}.x_n)$.

(dom.2) $F = (\lambda\vec{x}.x_n)$.

(dom.3) $F = (\lambda \vec{x}.x_n)$.

(un.1) $F = (\lambda \vec{x}.\langle 1, x_n * x_{n-1}\rangle)$.

(un.2) $F = (\lambda \vec{x}.\langle 0, x_n\rangle)$.

(un.3) $F = (\lambda \vec{x}.\langle 1, x_n\rangle)$.

(un.4) $F = (\lambda \vec{x}.x_n)$.

(int.1) $F = (\lambda \vec{x}.\langle x_{n-1}, x_n\rangle)$.

(int.2) $F = (\lambda \vec{x}.\langle x_{n-1}, x_n\rangle)$.

(int.3) $F = (\lambda \vec{x}.\langle x_n\rangle_0)$.

(int.4) $F = (\lambda \vec{x}.\langle x_n\rangle_1)$.

Induction step:

(T-$\mathsf{I_W}$) Assume there are derivations for $\Gamma \Rightarrow \Re(s)$, $\Gamma \Rightarrow \epsilon \mathrel{\dot{\in}} s$ as well as $\Gamma, \mathsf{W}(u), u \mathrel{\dot{\in}} s \Rightarrow \mathsf{s}_i u \mathrel{\dot{\in}} s$ $(i = 0, 1)$. By induction hypotheses there are functions $E, F, G_0, G_1 \in$ FPTIME such that for all terms t and $i = 0, 1$:

$$\vec{\rho} \circledR \Gamma \implies E(\vec{\rho}) \circledR \Re(s) \tag{1}$$

$$\vec{\rho} \circledR \Gamma \implies F(\vec{\rho}) \circledR \epsilon \mathrel{\dot{\in}} s \tag{2}$$

$$\vec{\rho} \circledR \Gamma; \sigma \circledR \mathsf{W}(t); \tau \circledR t \mathrel{\dot{\in}} s \implies G_i(\vec{\rho}, \sigma, \tau) \circledR \mathsf{s}_i t \mathrel{\dot{\in}} s \tag{3}$$

The required function H is now defined by recursion on notation:

$$H(\vec{\rho}, \epsilon) = F(\vec{\rho})$$
$$H(\vec{\rho}, \mathsf{s}_i\sigma) = G_i(\vec{\rho}, \sigma, H(\vec{\rho}, \sigma))$$

Now we need to show that $H(\vec{\rho}, \sigma) \circledR t \mathrel{\dot{\in}} s$. We prove this informally by induction on σ. If $\sigma = \epsilon$, F will construct a realiser for $\epsilon \mathrel{\dot{\in}} s$. Assuming that $H(\vec{\rho}, \sigma)$ is a realiser for $\overline{\sigma} \mathrel{\dot{\in}} s$, $G_i(\vec{\rho}, H(\vec{\rho}, \sigma))$ will construct a realiser for $\mathsf{s}_i\overline{\sigma} \mathrel{\dot{\in}} s$.

To prove that H is a polynomial time function, a bound is needed as H is constructed from functions in FPTIME. This bound is provided by E.

By induction hypothesis, $E(\vec{\rho}) \circledR \Re(s)$. Thus $s \in \Re^{\mathcal{M}^\star}$ and $\sigma \leq E(\vec{\rho})$ if $\sigma \circledR t \dot{\in} s$ for some t by Definition 14. Therefore, $H(\vec{\rho}, \sigma) \leq E(\vec{\rho})$. Thus indeed H is defined by *bounded* recursion on notation from F, G_0, G_1 and E. This shows that H is a polytime function.

(J.1) Assume we have derivations for $\Gamma, x \dot{\in} s \Rightarrow \Re(rx)$ and $\Gamma \Rightarrow \Re(s)$. By induction hypothesis, there are functions $F, G \in$ FPTIME such that

$$\vec{\rho}, \sigma \circledR \Gamma, t \dot{\in} s \implies F(\vec{\rho}, \sigma) \circledR \Re(rt) \tag{4}$$

$$\vec{\rho} \circledR \Gamma \implies G(\vec{\rho}) \circledR \Re(s) \tag{5}$$

We now define the requested function H as

$$H(\vec{\rho}) = \langle G(\vec{\rho}), F'(\vec{\rho}, G(\vec{\rho}))\rangle$$

where F' is a monotone polynomial function majorising F (the polynomial limiting the growth of F).

In general, we know that if $\tau \circledR \Re(s)$ and $\tau < \tau'$, then $\tau' \circledR \Re(s)$ by Definition 14. Thus, $F'(\vec{\rho}, \sigma) \circledR \Re(rt)$ since $F(\vec{\rho}, \sigma) \leq F'(\vec{\rho}, \sigma)$.

Now we need to show that $H(\vec{\rho}) \circledR \Re(\mathsf{j}(s, r))$, assuming $\vec{\rho} \circledR \Gamma$. To obtain this, we have to prove $\mathsf{j}(s, r) \in \mathcal{R}$ and $\tau \leq H(\vec{\rho})$ if $\tau \circledR t \dot{\in} \mathsf{j}(s, r)$ for some t.

We know that $F(\vec{\rho}, \sigma) \circledR \Re(rt)$ provided $\sigma \circledR t \dot{\in} s$. This just guarantees only for realisable elements t that rt indeed is a name. But because of Lemma 16, every $t \dot{\in} s$ has a realiser and (5) ensures that $s \in \mathcal{R}$. Thus F constructs a realiser for $\Re(rt)$ for every $t \in \mathit{ext}(s)$. Therefore $\mathsf{j}(s, r) \in \mathcal{R}$ by construction of $\mathcal{M}^\star$.

It remains to be proved that realisers of elements of $\mathsf{j}(s, r)$ actually are smaller than or equal to the candidate realiser produced by H. Assume $\tau \circledR t \dot{\in} \mathsf{j}(s, r)$. Then we have $\tau = \langle \tau_0, \tau_1 \rangle$ such that $\tau_0 \circledR (t)_0 \dot{\in} s$ and $\tau_1 \circledR (t)_1 \dot{\in} r(t)_0$ by Definition 14. Therefore, $\tau_0 \leq G(\vec{\rho})$ by (5). Further, by (4), $F(\vec{\rho}, \tau_0) \circledR \Re(r(t)_0)$. Thus we have $\tau_1 \leq F(\vec{\rho}, \tau_0) \leq F'(\vec{\rho}, G(\vec{\rho}))$ since F' is monotone. Because of the monotonicity of the pairing function (see e.g. [11]), $\tau \leq H(\vec{\rho})$ as required.

(J.2) Define H by $H(\vec{\rho}, \tau) = \langle \epsilon, \tau \rangle$.

(J.3) Define H by $H(\vec{\rho}, \sigma, \tau_0, \tau_1) = \langle \tau_0, \tau_1 \rangle$.

Logical rules are treated in [13]. Although Strahm's theory is based on classical logic, the proof still works with little alteration. Our functions need not choose which formula to realise in the absence of side formulas on the right side. Hence the adjustment is straightforward and the functions become simpler as we can omit most case distinctions. $\square$

The desired upper bounds for PETJ_1^i immediately follow from the previous theorem. Assume that we have a provably total function $G : \mathbb{W}^n \to \mathbb{W}$, i.e. for some suitable term t_G,

$$\mathsf{PETJ}_1^i \vdash t_G : \mathsf{W}^n \mapsto \mathsf{W} \tag{6}$$

$$\mathsf{PETJ}_1^i \vdash t_G \overline{w_1} \cdots \overline{w_n} = \overline{G(w_1 \cdots w_n)} \tag{7}$$

Hence, $\mathsf{PETJ}_1^{iG} \vdash \Rightarrow t_G : \mathsf{W}^n \mapsto \mathsf{W}$ with Lemma 10. Unfolding abbreviations gives $\mathsf{PETJ}_1^{iG} \vdash \Rightarrow (\forall x_1, \ldots, x_n \in \mathsf{W})(t_G x_1 \cdots x_n \in \mathsf{W})$. We get $\mathsf{PETJ}_1^{iG} \vdash \mathsf{W}(r_1), \ldots, \mathsf{W}(r_n) \Rightarrow \mathsf{W}(t_G r_1 \cdots r_n)$ by applying logical rules. With the realisability theorem, we know that there is a function $F \in$ FPTIME such that $F(\sigma_1, \ldots, \sigma_n) \ \textcircled{r} \ \mathsf{W}(t_G r_1 \cdots r_n)$ if $\sigma_i \ \textcircled{r} \ \mathsf{W}(r_i)\,(i = 1, \ldots, n)$. By Definition 14, $F(\vec{\sigma}) \ \textcircled{r} \ \mathsf{W}(t_G r_1 \cdots r_n)$ iff $\overline{F(\vec{\sigma})} \stackrel{\beta\eta}{=} t_G r_1 \cdots r_n$. Furthermore, $\sigma_i \ \textcircled{r} \ \mathsf{W}(r_i)$ iff $\overline{\sigma_i} \stackrel{\beta\eta}{=} r_i$. With (7) and equality we get $F(\vec{\sigma}) = G(\vec{\sigma})$.

Theorem 18 *The provably total functions of* PETJ_1^i *coincide with the functions computable in polynomial time.*

6 Realising some extensions

When we studied the theory PET in [12], we also considered several extensions not increasing its proof-theoretic strength. Of particular interest were Cantini's uniformity principle together with the type constructor for universal quantification recapitulated in section 2. We claim that we can also add those two principles to PETJ_1^i and keep the upper bounds. To establish this result, we extend the theorems proved in the previous section to include the uniformity principle and the all constructor. We will mainly spell out the extensions of the important proofs and definitions.

Definition 19 The theory $\mathsf{PETJ}_1{+}\forall^{iG}$ is defined as the theory PETJ_1^{iG} plus the following axioms:

(**all.1**) $\quad \Gamma, \Re(s) \Rightarrow \Re(\mathsf{all}(s))$

(**all.2**) $\quad \Gamma, \Re(s), \forall y((t,y) \mathrel{\dot\in} s) \Rightarrow t \mathrel{\dot\in} \mathsf{all}(s)$

(**all.3**) $\quad \Gamma, \Re(s), t \mathrel{\dot\in} \mathsf{all}(s) \Rightarrow \forall y((t,y) \mathrel{\dot\in} s)$

and the following rule for positive formulas A:

$$(\mathbf{UP})\,\frac{\Gamma \Rightarrow \forall x(\exists y \in \mathsf{W})A[x,y]}{\Gamma \Rightarrow (\exists y \in \mathsf{W})\forall x A[x,y]}$$

It is easy to extend the proof of Lemma 10 to include the additional axioms. Thus, the axioms and the rule as stated above are adequate reformulations of the axioms of section 2.

Before we can adjust the definition of the realisability, we add the following case to the construction of the model $\mathcal{M}^\star$ at successor stages $\alpha = \beta + 1$:

– $s \overset{\beta\eta}{=} \mathrm{all}(s_0)$ and $(s,t) \in \mathcal{E}_\alpha$ iff $((t,y), s_0) \in \mathcal{E}_\beta$ for all $y \in |\mathcal{M}^\star|$

Consequently, we also expand Definition 13 by the following case for the successor ordinal $\alpha = \beta + 1$:

$$\rho \; \check{\mathrm{r}}_\alpha \; t \mathrel{\dot\in} \mathsf{all}(s) \qquad \Longleftrightarrow \qquad \rho \; \check{\mathrm{r}}_\beta \; (t, t_0) \mathrel{\dot\in} s \text{ for all terms } t_0$$

The properties stated in Lemma 15 and Lemma 16 can easily be checked for the extended definitions. Thus, we can repeat Theorem 17 for $\mathsf{PETJ}_1{+}\forall^{iG}$:

Theorem 20 *For every positive sequent* $\Gamma[\vec{x}] \Rightarrow C[\vec{x}]$ *provable in* $\mathsf{PETJ}_1{+}\forall^{iG}$, *where* $\vec{x}$ *is a conclusive enumeration of the free variables, there is a function* $F \in \mathrm{FP}\textsc{time}$ *such that for all terms* $\vec{t}$*:*

$$\vec{\rho} \; ⓡ \; \Gamma[\vec{t}] \implies F(\vec{\rho}) \; ⓡ \; C[\vec{t}]$$

Proof Again, the proof is by induction on the length of the derivation. We only need to consider the axioms (**all.1-3**) for the base case and the rule (**UP**) in the induction step:

(all.1) $F = (\lambda\vec{x}.x_n)$.

(all.2) $F = (\lambda\vec{x}.x_n)$.

(all.3) $F = (\lambda\vec{x}.x_n)$.

(UP) By induction hypothesis, there is a function G such that $G(\vec{\rho}) \text{ ⓡ } \forall x(\exists y \in \mathsf{W})A[x, y]$ if $\vec{\rho} \text{ ⓡ } \Gamma$. By Definition 14, we know that $G(\vec{\rho}) = \langle\rho_0, \rho_1\rangle$ such that $\rho_1 \text{ ⓡ } A[u, \overline{\rho}_0]$ for a fresh variable u. Therefore $\rho_1 \text{ ⓡ } \forall x A[x, \overline{\rho}_0]$ which means $\langle\rho_0, \rho_1\rangle \text{ ⓡ } (\exists y \in \mathsf{W})\forall x A[x, y]$. Hence, we define $F(\vec{\rho}) = G(\vec{\rho})$. □

Again, the desired upper bounds are immediately derived from this theorem.

7 Conclusion and further work

As already mentioned before, treating the constructor for disjoint union based on weak theories is more delicate than for stronger theories. We are not able to prove the upper bounds of PETJ (based on classical logic) employing the same scheme as for the intuitionistic variant: the induction step fails for (**J.1**) in the proof of Theorem 17. Since we have side formulas on the right side for the classical sequent calculus reformulation, we cannot guarantee that F always generates a realiser for $\Re(rx)$, it could sometimes realise one of the side formulas, depending on σ. Therefore it seems impossible to decide whether to realise one of the side formulas or the main formula in the conclusion of the rule (**J.1**).

However, the proof concept can be applied to classical PET *without* join, thus providing a syntactical realisability argument for the model-theoretic proof in Spescha and Strahm [12]. In a first step one reformulates PET in a classical sequent calculus PET^G, thus allowing side formulas on the right hand side of sequents. Partial cut elimination works as before. In the definition of $\mathcal{M}^\star$

we can restrict ourselves to finite stages, due to the absence of join. The definition of realisability remains untouched.

As we have a sequence of formulas on the right side now, our realising function has to specify which formula to realise. Therefore, for $\Delta \equiv A_0, \ldots, A_n$ we specify

$$\rho \circledR \bigvee \Delta \iff \rho = \langle i, \rho_0 \rangle \text{ and } \rho_0 \circledR A_i.$$

The realisability theorem now reads as follows:

Theorem 21 *For every positive sequent $\Gamma[\vec{x}] \Rightarrow \Delta[\vec{x}]$ provable in PET^G, where $\vec{x}$ is a conclusive enumeration of the free variables, there is a function $F \in$ FPtime such that for all terms $\vec{t}$:*

$$\vec{\rho} \circledR \Gamma[\vec{t}] \implies F(\vec{\rho}) \circledR \bigvee \Delta[\vec{t}]$$

The proof follows the same procedure as in the intuitionistic case. Due to the presence of classical sequents, the treatment of the logical rules is slightly more complicated and requires the definition of realising functions by case distinction; similarly, in the treatment of the type induction rule, one has to make a case distinction whether the functions given by induction hypotheses realise the induction formula or one of the side formulas. Similar case distinction techniques have been used in the treatment of PT in Strahm [13].

Although it had been strongly conjectured in Spescha [11] that the provably total functions of the *classical* system PETJ are still the polynomial time computable ones, this problem was left open in [11]. Very recently, Probst [10] was able to supplement a proof of this conjecture. Indeed, using a variety of very advanced techniques he could determine the provably total operations of classical PETJ to be the polynomial time computable functions. Probst's arguments include the treatment of a special boundedness principle with respect to the predicate W, an extended realisability interpretation as well as subtle reasoning with non-standard models and non-monotone inductive definitions.

Let us conclude this paper by mentioning that it is possible to come up with suitable variants of PETJ which characterize various other complexity classes, including the functions computable in linear space and polynomial space, as well as the functions computable simultaneously in polynomial time and linear space. For details, see Spescha [11].

References

[1] M. J. Beeson. *Foundations of Constructive Mathematics: Metamathematical Studies*. Springer, Berlin, 1985.

[2] G. Calamai. *Proof-theoretic contributions to computational complexity*. PhD thesis, University of Siena, 2008.

[3] A. Cantini. A footnote on the Parsons-Mints-Takeuti theorem. Talk at *Recent Trends in Proof Theory*, Bern, July 2008.

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

[5] A. Cantini. Choice and uniformity in weak applicative theories. In M. Baaz, S. Friedman, and J. Krajíček, editors, *Logic Colloquium '01*, volume 20 of *Lecture Notes in Logic*. Association for Symbolic Logic, 2005.

[6] S. Feferman. A language and axioms for explicit mathematics. In J. Crossley, editor, *Algebra and Logic*, volume 450 of *Lecture Notes in Mathematics*, pages 87–139. Springer, Berlin, 1975.

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

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

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

[10] D. Probst. The provably terminating operations of the subsystem PETJ of explicit mathematics. Preprint, 2010.

[11] D. Spescha. *Weak systems of explicit mathematics*. PhD thesis, Universität Bern, 2009.

[12] D. Spescha and T. Strahm. Elementary explicit types and polynomial time operations. *Mathematical Logic Quarterly*, 55(3):245–258, 2009.

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

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

[15] T. Strahm. Weak theories of operations and types. In R. Schindler, editor, *Ways of Proof Theory*. Ontos Verlag, 2010.