On applicative theories

Gerhard Jäger   Reinhard Kahle*   Thomas Strahm*

Abstract

These notes deal with some recent proof-theoretic results about applicative theories. We omit proofs, instead we refer to the original articles and related approaches.

1 Introduction

Systems of explicit mathematics were introduced in Feferman [7, 9] in order to give a logical account to Bishop-style constructive mathematics, and they soon turned out to be very important for the proof-theoretic analysis of subsystems of second order arithmetic and set theory. Moreover, systems of explicit mathematics provide a logical framework for functional programming languages.

In a typical formulation of explicit mathematics one has to deal with two sorts of objects, namely *operations* and *types*. While in the original research the emphasis was put on types and type existence axioms, it turned out only recently that already the applicative basis of these theories is of significant — especially proof-theoretic — interest. In contrast to traditional formalizations of mathematics which follow a set-theoretic paradigm and an extensional approach to functions, applicative theories and explicit mathematics focus on an intensional point of view. In applicative theories all objects may be regarded as operations (or rules) which can be freely applied to each other; selfapplication is meaningful but not necessarily total.

The purpose of these notes is to survey recent proof-theoretic results on applicative theories with classical logic and to present them in some structured form. For many intuitionistic aspects of this subject we refer to Beeson [2] and Troelstra and Van Dalen [49]. Since space does not permit to give any proofs, we confine ourselves to giving pointers to the original literature and mentioning approaches which deal with related topics.

The plan of this paper is as follows. In Section 2 we introduce some basic axioms and principles of applicative theories. In Sections 3 and 4 we study applicative theories with some typical functionals of higher types. Section 5 is dedicated to polynomial time applicative theories, while the rest of the paper addresses some extensions of the formalisms described here.

*Research supported by the Swiss National Science Foundation.

2 The basic theory of operations and numbers

The language of our applicative theories is a first order language $\mathcal{L}$ of partial terms with individual variables $x, y, z, u, v, w, f, g, h, \ldots$ (possibly with subscripts). $\mathcal{L}$ includes individual constants k, s (combinators), $\mathsf{p}, \mathsf{p_0}, \mathsf{p_1}$ (pairing and unpairing), $\mathsf{0}$ (zero), $\mathsf{s_N}$ (numerical successor), $\mathsf{p_N}$ (numerical predecessor), $\mathsf{d_N}$ (definition by numerical cases), and $\mathsf{r_N}$ (primitive recursion); later we will add further constants for dealing with specific functionals of higher types. Further, $\mathcal{L}$ has a binary function symbol $\cdot$ for (partial) term application, unary relation symbols $\downarrow$ (defined) and N (natural numbers), as well as a binary relation symbol $=$ (equality).

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

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

In the following we write (st) or just st instead of $(s \cdot t)$, and we adopt the convention of association to the left, i.e. $s_1 s_2 \ldots s_n$ stands for $(\ldots(s_1 s_2) \ldots s_n)$. Further we put $t' := \mathsf{s_N} t$ and $\mathsf{1} := \mathsf{0}'$.

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

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

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

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

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

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

Now we are going to recall the basic theory BON of operations and numbers which has been introduced in Feferman and Jäger [17]. Its underlying logic is the *classical logic of partial terms* due to Beeson [2]; it corresponds to E^+ logic with strictness and equality of Troelstra and Van Dalen [48], which is also described in Feferman [16]. The non-logical axioms of BON are divided into the following five groups.

I. Partial combinatory algebra.

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

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

II. Pairing and projection.

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

III. Natural numbers.

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

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

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

IV. Definition by numerical cases.

(7) $u \in \mathsf{N} \wedge v \in \mathsf{N} \;\wedge\; u = v \rightarrow \mathsf{d_N}xyuv = x$,

(8) $u \in \mathsf{N} \wedge v \in \mathsf{N} \;\wedge\; u \neq v \rightarrow \mathsf{d_N}xyuv = y$.

V. Primitive recursion on N.

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

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

As usual the axioms of a partial combinatory algebra allow one to define λ abstraction and to prove a recursion or fixed point theorem. For proofs of these standard results the reader is referred to [2, 7]. Some problems concerning substitutions in our partial setting are discussed in Strahm [45].

Let us recall the definition of a *subset of* N from [8, 17]. Sets of natural numbers are represented via their characteristic functions which are total on N. Accordingly, we define

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

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

In the following we are interested in four forms of complete induction on the natural numbers, namely set induction, operation induction, N induction, and full formula induction.

Set induction on N $(\mathsf{S\text{-}I_N})$.

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

Operation induction on N $(\mathsf{O\text{-}I_N})$.

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

N induction on N $(\mathsf{N\text{-}I_N})$.

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

Formula induction on N $(\mathsf{F\text{-}I_N})$. For all formulas $A(x)$ of $\mathcal{L}$:

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

Observe that it is trivial from these definitions that

$$(\mathsf{S\text{-}I_N}) \subset (\mathsf{O\text{-}I_N}) \subset (\mathsf{F\text{-}I_N}) \quad \text{and} \quad (\mathsf{N\text{-}I_N}) \subset (\mathsf{F\text{-}I_N}).$$

It is shown in Kahle [36] that there exists a term $\mathsf{not_N}$ so that BON proves that $\mathsf{not_N}$ does not belong to N. Making use of this term, he obtains the following result.

Proposition 1 $(\mathsf{N\text{-}I_N})$ *implies* $(\mathsf{S\text{-}I_N})$ *over* BON.

The exact relationship between operation induction and N induction over BON has not been settled yet; proof-theoretically, both forms of induction are equivalent over BON.

Proposition 2 *We have the following proof-theoretic equivalences:*

1. $\mathsf{BON} + (\mathsf{S\text{-}I_N}) \equiv \mathsf{BON} + (\mathsf{O\text{-}I_N}) \equiv \mathsf{BON} + (\mathsf{N\text{-}I_N}) \equiv \mathsf{PRA}$.
2. $\mathsf{BON} + (\mathsf{F\text{-}I_N}) \equiv \mathsf{PA}$.

The proof of these results is folklore, and we refer to Feferman [11] for an exact definition of the notion $\equiv$ of proof-theoretic equivalence.

Following Cantini [4] and Jäger and Strahm [32], the above equivalences still hold in the presence of totality of application, (Tot), and extensionality of operations, (Ext); for the definition of (Tot) and (Ext) cf. e.g. [32]. Here formalized term model constructions serve to determine proof-theoretic upper bounds.

The standard recursion-theoretic model of BON is obtained by taking as domain the natural numbers and interpreting application as partial recursive function application so that a applied to b is $\{a\}(b)$. Recently, Schlüter [42] came up with a modification of applicative theories which allows for an interpretation of application in terms of the primitive recursive indices.

3 The non-constructive μ and related operators

It has been convincingly argued in a series of articles, for example in Feferman [7, 8, 12], that strong operation existence axioms are needed for a smooth development of classical mathematics in applicative theories. One of these higher type operators is the non-constructive μ operator, which is characterized by the following two axioms.

The unbounded μ operator

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

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

In order to define the standard recursion-theoretic model of BON which satisfies the axioms of μ, Π^1_1 recursion theory is required. Sets in the sense of $\mathcal{P}(\mathsf{N})$ as defined above then correspond exactly to the hyperarithmetic sets.

From a recursion-theoretic point of view, recursion in μ is equivalent to recursion in the functional E_0 for quantification over the natural numbers. In our present context, the situation is slightly more delicate.

The quantification functional E_0

(E_0.1) $(f : \mathsf{N} \to \mathsf{N}) \leftrightarrow \mathsf{E}_0 f \in \mathsf{N}$,

(E_0.2) $(f : \mathsf{N} \to \mathsf{N}) \to ((\exists x \in \mathsf{N})(fx = 0) \leftrightarrow \mathsf{E}_0 f = 0)$.

In the following we write $\mathsf{BON}(\mu)$ for $\mathsf{BON} + (\mu.1, \mu.2)$ and $\mathsf{BON}(\mathsf{E}_0)$ for $\mathsf{BON} + (\mathsf{E}_0.1, \mathsf{E}_0.2)$. It is obvious that both, μ and E_0 provide for the elimination of quantifiers ranging over N.

By making use of some combinatorial properties, one can show that the axioms (E_0.1) and (E_0.2) of E_0 are derivable in $\mathsf{BON}(\mu)$. In order to deal with the axioms (μ.1) and (μ.2) of μ in $\mathsf{BON}(\mathsf{E}_0)$, it seems that some induction and a form of N strictness are required. A convenient side effect of E_0 is that it can be used to derive N induction from operation induction and vice versa. For details see Kahle [36, 39].

Proposition 3 (N-$\mathsf{I_N}$) *is equivalent to* (O-$\mathsf{I_N}$) *over* $\mathsf{BON}(\mathsf{E}_0)$.

In spite of its fairly strong standard model, which is also minimal in a suitable sense, $\mathsf{BON}(\mu)$ plus set induction and $\mathsf{BON}(\mathsf{E}_0)$ plus set induction are of the same proof-theoretic strength as Peano arithmetic. If we add stronger forms of complete induction then we reach larger but still predicative segments of the hyperarithmetic hierarchy.

Proposition 4 *We have the following proof-theoretic equivalences:*

1. $\mathsf{BON}(\mu) + (\mathsf{S}\text{-}\mathsf{I_N}) \equiv \mathsf{BON}(\mathsf{E}_0) + (\mathsf{S}\text{-}\mathsf{I_N}) \equiv \mathsf{PA}$.
2. $\mathsf{BON}(\mu) + (\mathsf{O}\text{-}\mathsf{I_N}) \equiv \mathsf{BON}(\mathsf{E}_0) + (\mathsf{O}\text{-}\mathsf{I_N}) \equiv (\Pi^0_1\text{-}\mathsf{CA})_{<\omega^\omega}$.
3. $\mathsf{BON}(\mu) + (\mathsf{N}\text{-}\mathsf{I_N}) \equiv \mathsf{BON}(\mathsf{E}_0) + (\mathsf{N}\text{-}\mathsf{I_N}) \equiv (\Pi^0_1\text{-}\mathsf{CA})_{<\omega^\omega}$.

4. $\mathsf{BON}(\mu) + (\mathsf{F\text{-}I_N}) \equiv \mathsf{BON}(\mathsf{E_0}) + (\mathsf{F\text{-}I_N}) \equiv (\Pi^0_1\text{-}\mathsf{CA})_{<\varepsilon_0}$.

The corresponding proof-theoretic ordinals are $\varepsilon_0, \varphi\omega 0$, and $\varphi\varepsilon_0 0$, respectively. All four results do not change if we add (Tot) *and* (Ext).

These results are proved in Feferman and Jäger [17] and in Jäger and Strahm [32, 33]. The theories $(\Pi^0_1\text{-}\mathsf{CA})_{<\alpha}$ are the usual theories for arithmetic comprehension iterated up to all ordinals less than α (along a given standard wellordering).

A further step consists in replacing $\mathsf{E_0}$ by the functional $\mathsf{E}_0^{\#}$ which acts on partial type 1 objects, cf. Hinman [24]. In Kahle [39] it is shown that (modulo N strictness) the proof theory of $\mathsf{E}_0^{\#}$ is the same as the one for $\mathsf{E_0}$ and μ.

4 The Suslin operator $\mathsf{E_1}$

The next (natural) step up in recursion-theoretic hierarchies is the well-known Suslin operator $\mathsf{E_1}$ which tests for wellfoundedness on total objects. The recursion theory of $\mathsf{E_1}$ is presented in detail for example in Hinman [24].

There exists a close relationship between recursion in $\mathsf{E_1}$, subsystems of the theory KPi of iterated admissible sets (cf. e.g. Jäger [26]) and Δ^1_2 comprehension in second order arithmetic. The least ordinal not recursive in $\mathsf{E_1}$ is the first recursively inaccessible ordinal and the sets recursive in $\mathsf{E_1}$ form a model of Δ^1_2 comprehension.

In our applicative context the Suslin operator $\mathsf{E_1}$ can be characterized by the following two axioms.

The wellfoundedness functional $\mathsf{E_1}$

$(\mathsf{E_1}.1)$ $\quad (f : \mathsf{N}^2 \to \mathsf{N}) \leftrightarrow \mathsf{E_1} f \in \mathsf{N}$,

$(\mathsf{E_1}.2)$ $\quad (f : \mathsf{N}^2 \to \mathsf{N}) \to ((\exists g : \mathsf{N} \to \mathsf{N})(\forall x \in \mathsf{N})(f(gx')(gx) = 0) \leftrightarrow \mathsf{E_1} f = 0)$.

In the following we write $\mathsf{BON}(\mu, \mathsf{E_1})$ for $\mathsf{BON}(\mu) + (\mathsf{E_1}.1, \mathsf{E_1}.2)$. It should be obvious that in $\mathsf{BON}(\mu, \mathsf{E_1})$ every Π^1_1 set of natural numbers can be represented by a set in the sense of $\mathcal{P}(\mathsf{N})$.

Proposition 5 *We have the following proof-theoretic equivalences:*

1. $\mathsf{BON}(\mu, \mathsf{E_1}) + (\mathsf{S\text{-}I_N}) \equiv (\Pi^1_1\text{-}\mathsf{CA})\!\upharpoonright$.
2. $\mathsf{BON}(\mu, \mathsf{E_1}) + (\mathsf{O\text{-}I_N}) \equiv \mathsf{BON}(\mu, \mathsf{E_1}) + (\mathsf{N\text{-}I_N}) \equiv (\Pi^1_1\text{-}\mathsf{CA})_{<\omega^\omega}$.
3. $\mathsf{BON}(\mu, \mathsf{E_1}) + (\mathsf{F\text{-}I_N}) \equiv (\Pi^1_1\text{-}\mathsf{CA})_{<\varepsilon_0}$.

The corresponding proof-theoretic ordinals are $\psi 0 \Omega_\omega, \psi 0 \Omega_{\omega^\omega}$, and $\psi 0 \Omega_{\varepsilon_0}$, respectively.

This proposition is proved in Jäger and Strahm [31]. By known proof-theoretic results it also exhibits the proof-theoretic relationship between $\mathsf{E_1}$ and theories for iterated admissible sets.

5 Polynomial time applicative theories

In the last decade, there have been many activities in the field of so-called *bounded arithmetic*, emerging from Buss' important work in [3]. A natural question — first posed in Feferman [14] — is whether a similar program can be carried through in the context of applicative theories or explicit mathematics. In this section we briefly sketch a solution which is presented in detail in the paper Strahm [46].

A first attempt in setting up a self-applicative framework of polynomial strength might consist in a direct mimicking of systems of bounded arithmetic, e.g. Buss' S^1_2. However, it has been shown in Strahm [43] that, due to the presence of unbounded recursion principles in the applicative language, this approach does not work, i.e., one immediately ends up with systems of strength PRA. As a consequence, a theory had to be developed that is better tailored for our applicative framework.

The theory of polynomial time operations PTO of [46] can be viewed as the polytime analogue of $\mathsf{BON} + (\mathsf{S}\text{-}\mathsf{I}_\mathsf{N})$. Instead of a predicate N for the natural numbers, PTO is based on a unary predicate W for binary words. Apart from the combinators k, s and the operations for pairing and unpairing, the language of PTO includes constants $\epsilon, \mathsf{0}, \mathsf{1}$ (empty word, zero, one), $*, \times, \mathsf{p_W}$ (word concatenation and multiplication, word predecessor), $\mathsf{c}_\subseteq$ (initial subword relation) and $\mathsf{r_W}$ (bounded primitive recursion on W). The axioms of PTO are based on a partial combinatory algebra plus pairing (axiom groups I. and II. of BON). Further, PTO includes defining axioms for the above mentioned constants; in particular, the bounded recursor $\mathsf{r_W}$ is axiomatized in order to provide an operation $\mathsf{r_W} fgb$ for primitive recursion from f and g with length bound b. Most crucially, PTO's induction principle is set induction on W, $(\mathsf{S}\text{-}\mathsf{I_W})$; sets of binary words are understood in the same way as sets of naturals, namely via their total characteristic functions on W.

PTO contains Ferreira's system of polynomial time computable arithmetic PTCA (cf. [20]), and it can be modeled in the theory $\mathsf{PTCA}^+ + (\Sigma\text{-}\mathsf{Ref})$, i.e., the extension of PTCA by NP induction and Σ reflection (equivalently: bounded collection). The feasibility of this latter theory w.r.t. Π^0_2 statements yields that closed terms of type $(\mathsf{W} \to \mathsf{W})$ describe exactly the polytime functions in PTO, cf. [46] for details.

Proposition 6 *We have the following proof-theoretic equivalences:*

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

Moreover, the provably total functions of PTO *are exactly the polytime functions.*

An interesting open problem concerns the status of the axiom of totality (Tot) in PTO. Although we conjecture that $\mathsf{PTO} + (\mathsf{Tot})$ is not stronger than PTO, the methods of [32, 46] do not immediately carry over to the presence of (Tot); this is related to the question whether the Church Rosser property can be derived in a feasible system. Cantini [5] has shown that the provably total functions of

PTO + (Tot) have *polynomial growth rate* only. Actually, he establishes this result for a substantial extension of PTO + (Tot).[1]

Let us finish this section by mentioning that based on the ideas for dealing with PTO, it is possible to come up with applicative theories capturing the levels of the Grzegorczyk hierarchy; for a more detailed discussion on this we refer to Strahm [44].

6 Extensions of applicative theories

As we have mentioned in the introduction, traditional systems of explicit mathematics allow the formation of classifications or types above the first order operational basis, thus providing rich and flexible type structures. Feferman [7] introduced two major frameworks of explicit mathematics, namely the theories T_0 and T_1, where T_1 is obtained from T_0 by adding the non-constructive μ operator. Accordingly, BON and BON(μ) form the applicative basis of T_0 and T_1, respectively.

While the proof theory of T_0 and its subsystems is well-known since the early eighties (cf. Feferman [7, 9], Feferman and Sieg [19], Jäger [25], Jäger and Pohlers [30]), the corresponding investigations for T_1 have been completed only recently by Feferman and Jäger [17, 18], and Glaß and Strahm [22]. Moreover, universes in the framework of T_0 and T_1 have been studied in Feferman [10], Marzetta [40], and Marzetta and Strahm [41]. A further valuable reference is Glaß' thesis [21].

Although types in explicit mathematics can form rather complicated collections of objects, they are represented by an operation of the underlying applicative universe. In this respect, Jäger [27] has provided a very perspicuous formulation using a naming relation between objects and types. Recently, Cantini and Minari [6], Jäger [29], and Jansen [34] have established interesting inseparability results with respect to the ontology of names in explicit mathematics.

Systems of explicit mathematics have also been used to develop a general logical framework for functional programming and type theory, where it is possible to derive such important properties of functional programs as termination and correctness. Relevant references are Feferman [13, 14, 15] and Jäger [28]. For investigations closer to actual programming languages, cf. e.g. Hayashi and Nakano [23], Kahle [37], and Talcott [47]. The first reference contains the description of an experimental implementation for extracting programs from constructive proofs in an explicit mathematics setting.

An alternative approach to introduce a notion of types (sets, classes) above applicative theories is provided by Frege structures. These were studied by Aczel [1] as a semantical concept and later formalized for example by Beeson [2] as a theory

[1] **Added in proof:** More recently, Andrea Cantini [Feasible operations and applicative theories based on $\lambda\eta$, Preprint, April 1998] was able to give a *feasible interpretation* of PTO + (Tot), thus confirming the above conjecture.

of partial truth over BON. The underlying applicative framework allows a very elegant representation of formulas by terms without referring to Gödelization. The so-obtained axiomatization is still first order, however, it makes sense in the presence of a total application operation only, cf. [35].

The most comprehensive exposition on Frege structures can be found in Cantini's monograph [4], where he also presents many extensions and applications. In analogy to explicit mathematics and Martin Löf type theory, it is possible to add a natural notion of universes to Frege structures, yielding systems of proof-theoretic strength Γ_0 and beyond, cf. Kahle [38, 39].

References

[1] Aczel, P. Frege structures and the notion of proposition, truth and set. In *The Kleene Symposium* (1980), J. Barwise, H. Keisler, and K. Kunen, Eds., North-Holland, pp. 31 – 59.

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

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

[4] Cantini, A. *Logical Frameworks for Truth and Abstraction*. North-Holland, Amsterdam, 1996.

[5] Cantini, A. On the computational content of weak applicative theories with totality I, 1996. Preliminary Draft.

[6] Cantini, A., and Minari, P. Uniform inspeparability in explicit mathematics. *Journal of Symbolic Logic 64*, 1 (1999), 313–326.

[7] 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.

[8] Feferman, S. A theory of variable types. *Revista Colombiana de Matemáticas 19* (1975), 95–105.

[9] 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.

[10] Feferman, S. Iterated inductive fixed-point theories: application to Hancock's conjecture. In *The Patras Symposium*, G. Metakides, Ed. North Holland, Amsterdam, 1982, pp. 171–196.

[11] Feferman, S. Hilbert's program relativized: proof-theoretical and foundational studies. *Journal of Symbolic Logic*, 53 (1988), 364–384.

[12] Feferman, S. Weyl vindicated: "Das Kontinuum" 70 years later. In *Temi e prospettive della logica e della filosofia della scienza contemporanee*. CLUEB, Bologna, 1988, pp. 59–93.

[13] Feferman, S. Polymorphic typed lambda-calculi in a type-free axiomatic framework. In *Logic and Computation*, W. Sieg, Ed., vol. 106 of *Contemporary Mathematics*. American Mathematical Society, Providence, Rhode Island, 1990, pp. 101–136.

[14] Feferman, S. Logics for termination and correctness of functional programs. In *Logic from Computer Science*, Y. N. Moschovakis, Ed., vol. 21 of *MSRI Publications*. Springer, Berlin, 1991, pp. 95–127.

[15] Feferman, S. Logics for termination and correctness of functional programs II: Logics of strength PRA. In *Proof Theory*, P. Aczel, H. Simmons, and S. S. Wainer, Eds. Cambridge University Press, Cambridge, 1992, pp. 195–225.

[16] Feferman, S. Definedness. *Erkenntnis 43* (1995), 295–320.

[17] Feferman, S., and Jäger, G. Systems of explicit mathematics with non-constructive μ-operator. Part I. *Annals of Pure and Applied Logic 65*, 3 (1993), 243–263.

[18] 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).

[19] Feferman, S., and Sieg, W. Proof-theoretic equivalences between classical and constructive theories for analysis. In *Iterated Inductive Definitions and Subsystems of Analysis: Recent Proof-Theoretical Studies*, W. Buchholz, S. Feferman, W. Pohlers, and W. Sieg, Eds., vol. 897 of *Lecture Notes in Mathematics*. Springer, Berlin, 1981, pp. 78–142.

[20] Ferreira, F. Polynomial time computable arithmetic. *Contemporary Mathematics 106* (1990), 137–156.

[21] Glass, T. *Standardstrukturen für Systeme Expliziter Mathematik*. PhD thesis, Westfälische Wilhelms-Universität Münster, 1993.

[22] Glass, T., and Strahm, T. Systems of explicit mathematics with non-constructive μ-operator and join. *Annals of Pure and Applied Logic 82* (1996), 193–219.

[23] Hayashi, S., and Nakano, H. *PX: A Computational Logic*. MIT Press, Cambridge, MA, 1988.

[24] Hinman, P. G. *Recursion-Theoretic Hierarchies.* Springer, Berlin, 1978.

[25] Jäger, G. A well-ordering proof for Feferman's theory T_0. *Archiv für mathematische Logik und Grundlagenforschung 23* (1983), 65–77.

[26] Jäger, G. *Theories for Admissible Sets: A Unifying Approach to Proof Theory.* Bibliopolis, Napoli, 1986.

[27] 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.

[28] Jäger, G. Type theory and explicit mathematics. In *Logic Colloquium '87*, H.-D. Ebbinghaus, J. Fernandez-Prida, M. Garrido, M. Lascar, and M. R. Artalejo, Eds. North Holland, Amsterdam, 1989, pp. 117–135.

[29] Jäger, G. Power types in explicit mathematics? *Journal of Symbolic Logic 62* (1997), 1142–1146.

[30] Jäger, G., and Pohlers, W. Eine beweistheoretische Untersuchung von $(\Delta^1_2\text{-CA}) + (\text{BI})$ und verwandter Systeme. In *Sitzungsberichte der Bayerischen Akademie der Wissenschaften, Mathematisch-naturwissenschaftliche Klasse.* 1982, pp. 1–28.

[31] Jäger, G., and Strahm, T. The proof-theoretic strength of the Suslin operator in applicative theories. In preparation.

[32] Jäger, G., and Strahm, T. Totality in applicative theories. *Annals of Pure and Applied Logic 74*, 2 (1995), 105–120.

[33] Jäger, G., and Strahm, T. Some theories with positive induction of ordinal strength $\varphi\omega 0$. *Journal of Symbolic Logic 61*, 3 (1996), 818–842.

[34] Jansen, D. Ontologische Aspekte expliziter Mathematik. Master's thesis, Mathematisches Institut, Universität Bern, 1997.

[35] Kahle, R. Frege structures for partial applicative theories. *Journal of Logic and Computation.* To appear.

[36] Kahle, R. N-strictness in applicative theories. *Archive for Mathematical Logic.* To appear.

[37] Kahle, R. Einbettung des Beweissystems LAMBDA in eine Theorie von Operationen und Zahlen. Master's thesis, Universität München, 1992.

[38] Kahle, R. Universes over Frege structures. Tech. Rep. IAM-96-010, Institut für Informatik und angewandte Mathematik, Universität Bern, May 1996.

[39] Kahle, R. *Applicative Theories and Frege Structures*. PhD thesis, Institut für Informatik und angewandte Mathematik, Universität Bern, 1997.

[40] Marzetta, M. *Predicative Theories of Types and Names*. PhD thesis, Institut für Informatik und angewandte Mathematik, Universität Bern, 1993.

[41] Marzetta, M., and Strahm, T. The μ quantification operator in explicit mathematics with universes and iterated fixed point theories with ordinals. *Archive for Mathematical Logic 37*, 5+6 (1998), 391–413.

[42] Schlüter, A. A theory of rules for enumerated classes of functions. *Archive for Mathematical Logic 34* (1995), 47–63.

[43] Strahm, T. Theories with self-application of strength PRA. Master's thesis, Institut für Informatik und angewandte Mathematik, Universität Bern, 1992.

[44] Strahm, T. *On the Proof Theory of Applicative Theories*. PhD thesis, Institut für Informatik und angewandte Mathematik, Universität Bern, 1996.

[45] Strahm, T. Partial applicative theories and explicit substitutions. *Journal of Logic and Computation 6*, 1 (1996).

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

[47] Talcott, C. A theory for program and data type specification. *Theoretical Computer Science 104* (1992), 129–159.

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

[49] Troelstra, A., and van Dalen, D. *Constructivism in Mathematics*, vol. II. North Holland, Amsterdam, 1988.

Address
Institut für Informatik und angewandte Mathematik, Universität Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland, {jaeger,kahle,strahm}@iam.unibe.ch