universität bern
institut für informatik
und angewandte mathematik

Frege structures for partial applicative theories

Reinhard Kahle

Technischer Bericht IAM-96-013

September 1996

Adresse:
Institut für Informatik und angewandte Mathematik
Universität Bern
Neubrückstr. 10
CH - 3012 Bern

Frege structures for partial applicative theories

Reinhard Kahle

Abstract

Due to strictness problems, usually the syntactical definition of Frege structures is conceived as a truth theory for *total* applicative theories. To investigate Frege structures in a *partial* framework we can follow two ways. First, simply by ignoring undefinedness in the truth definition. Second, by introducing of a certain notion of pointer. Both approaches are compatible with the traditional formalizations of Frege structures and preserve the main results, namely abstraction and the proof-theoretic strength.

1 Introduction

Frege structures were introduced by Aczel in [Acz80] as a semantical concept to provide a new concept of sets by means of a *partial truth predicate*. This approach is closely related to prior work of Scott [Sco75] and was originally developed for questions around Martin-Löf's type theory. Later on, Beeson gave a formalization of Frege structures as a truth theory for applicative theories [Bee85]. Applicative theories go back to Feferman's systems of explicit mathematics [Fef75, Fef79]. The actual formalization of the underlying basic theory of operations and numbers BON is presented in [FJ93] as the classical version of Beeson's theory EON without induction.

Frege structures are of interest to research in truth theory as well as in applicative theories. The most comprehensive exposition about it can be found in the monograph *Logical Frameworks for Truth and Abstraction* [Can96] of Cantini where he presents also many extensions and applications.

All known axiomatizations start with a total version of applicative theories, where all objects are defined. In this paper we discuss the problem of defining Frege structures for the partial theory BON. A straightforward definition of truth for BON runs into problems with strictness which requires that all arguments of predicates are defined. We will present two possible solutions, first by ignoring undefinedness in the truth definition, second by introducing a notion of pointer which allows us to avoid the strictness problem.

The paper is organized as follows: First we introduce concisely the theory BON, then we recapitulate the Frege structures for total applicative theories. After discussing the strictness problem in the partial framework, we present the theory $\widehat{\mathsf{PFS}}$ where truth is defined only for the total part of BON. In the last section we use pointers to give a truth definition for the whole theory.

2 BON

As introduced in [FJ93] the basic theory of operations and numbers BON is formulated in $\mathcal{L}_p$, the first order language of partial operations and numbers.

$\mathcal{L}_p$ comprises individual variables $x, y, z, v, w, f, g, h, \ldots$, individual constants 0, k, s, p, p_0, p_1, $\mathsf{s_N}$, $\mathsf{p_N}$, $\mathsf{d_N}$, a binary function symbol $\cdot$ for term application, and the relation symbols $\downarrow, =$ and N.

Terms $(r, s, t, \ldots)$ are built up from individual variables and individual constants by term application. Formulae $(\varphi, \psi, \ldots)$ are defined by $\neg$, $\wedge$ and $\forall$ as usual, starting from the atomic formulae $t\downarrow$, $t = s$, and $\mathsf{N}(t)$.

In the following we write $s\,t$ for $(s \cdot t)$ with the convention of association to the left, $\vee$, $\rightarrow$ and $\exists$ are defined as usual and we use the following abbreviations.

$$\begin{aligned} t \simeq s &:= (t\downarrow \vee\, s\downarrow) \rightarrow t = s \\ t \neq s &:= t\downarrow \wedge\, s\downarrow \wedge\, \neg t = s \\ t' &:= \mathsf{s_N}\, t \end{aligned}$$

The logic of BON is the (classical) logic of partial terms (cf. [Bee85, Sec. VI.1]) from which we quote only the strictness condition which will be crucial in the following:

(S1) $$\varphi(t_1, \ldots, t_n) \rightarrow t_1\downarrow \wedge \ldots \wedge t_n\downarrow$$

where φ is an atomic formula.

The non-logical axioms of BON include:

I. **Partial combinatory algebra.**

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

(2) $\mathsf{s}\,x\,y\downarrow \wedge\, \mathsf{s}\,x\,y\,z \simeq x\,z\,(y\,z)$,

II. **Pairing and projection.**

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

III. **Natural numbers.**

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

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

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

IV. **Definition by cases on N.**

(7) $\mathsf{N}(v) \wedge \mathsf{N}(w) \wedge v = w \rightarrow \mathsf{d_N}\, x\, y\, v\, w = x$,

(8) $\mathsf{N}(v) \wedge \mathsf{N}(w) \wedge v \neq w \rightarrow \mathsf{d_N}\, x\, y\, v\, w = y$.

It is well-known that in BON a notion of λ-abstraction can be introduced, also the recursion theorem holds (cf. [Bee85]).

Proposition 1 For every variable x and every term t of $\mathcal{L}_p$ there exists a term $\lambda x.t$ of $\mathcal{L}_p$ whose free variables are those of t, excluding x, so that

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

Proposition 2 There is a term rec of $\mathcal{L}_p$ so that

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

Moreover we need the fact that there is a term which does not belong to the natural numbers (cf. [Kah95]).

Lemma 3 There is a term $\mathsf{not_N}$ of $\mathcal{L}_p$ so that

$$\mathsf{BON} \vdash \neg \mathsf{N}(\mathsf{not_N}).$$

Proof: Choose the term $\mathsf{not_N}$ as $\mathsf{rec}\, (\lambda x.\lambda y.\mathsf{d_N}\, 1\, 0\, (x\, y)\, 0)\, 0$. □

The total version of BON, called TON, is formally introduced in a weaker language where the predicate $\downarrow$ is dropped. The logic of partial terms is replaced by classical first order predicate logic with equality. Syntactically the non-logical axioms only differ in the s-combinator for which we replace I.(2) by

$$\mathsf{s}\, x\, y\, z = x\, z\, (y\, z).$$

TON is equivalent to BON plus the axiom of totality:

(Tot) $$\forall x, y.\, x\, y \downarrow.$$

The theory TON is extensively studied in the paper [JS95]. It shows that the axiom (Tot) does not change the proof-theoretic strength of BON and of several of its extensions in the presence of natural forms of induction.

3 Frege Structures for total applicative theories

As mentioned in the introduction, Frege structures can be considered as a truth theory for applicative theories. Before we introduce Frege structures for the partial theory BON, we recapitulate the version for the total case and sketch the central result. There are several, slightly different forms of axiomatizations, we will follow the theory NMT of [Can93] or MF^- of [Can96].

The language $\mathcal{L}_F$ of FON is the language of TON extended by the new relation symbol T and new individual constants $\dot{=}, \dot{\mathsf{N}}, \dot{\neg}, \dot{\wedge}, \dot{\forall}$ and $\dot{\mathsf{T}}$.

The axioms of FON are the axioms of TON extended to the new language plus the following axioms:

I. **Closure under prime formulae of** TON

(1) $x = y \leftrightarrow \mathsf{T}(\dot{=}\, x\, y)$

(2) $\neg x = y \leftrightarrow \mathsf{T}(\dot{\neg}\, (\dot{=}\, x\, y))$

(3) $\mathsf{N}(x) \leftrightarrow \mathsf{T}(\dot{\mathsf{N}}\, x)$

(4) $\neg \mathsf{N}(x) \leftrightarrow \mathsf{T}(\dot{\neg}\, (\dot{\mathsf{N}}\, x))$

II. **Closure under composed formulae**

(5) $\mathsf{T}(x) \leftrightarrow \mathsf{T}(\dot{\neg}\, (\dot{\neg}\, x))$

(6) $\mathsf{T}(x) \wedge \mathsf{T}(y) \leftrightarrow \mathsf{T}(\dot{\wedge}\, x\, y)$

(7) $\mathsf{T}(\dot{\neg}\, x) \vee \mathsf{T}(\dot{\neg}\, y) \leftrightarrow \mathsf{T}(\dot{\neg}\, (\dot{\wedge}\, x\, y))$

(8) $(\forall x.\mathsf{T}(f\, x)) \leftrightarrow \mathsf{T}(\dot{\forall}\, f)$

(9) $(\exists x.\mathsf{T}(\dot{\neg}\, (f\, x))) \leftrightarrow \mathsf{T}(\dot{\neg}\, (\dot{\forall}\, f))$

III. **Self-reference**

(10) $\mathsf{T}(x) \leftrightarrow \mathsf{T}(\dot{\mathsf{T}}\, x)$

(11) $\mathsf{T}(\dot{\neg}\, x) \leftrightarrow \mathsf{T}(\dot{\neg}\, (\dot{\mathsf{T}}\, x))$

IV. **Consistency**

(12) $\neg(\mathsf{T}(x) \wedge \mathsf{T}(\dot{\neg}\, x))$

The axioms I. – III. express that T is closed under building formulae by $=$, N, T, $\wedge$, $\forall$, and negation where T may be used only positively. The axioms III. can also be regarded as a form of self-reference for T, but they cannot be closed under negation, which would immediately yield a contradiction to consistency which is axiomatized by IV.

In straightforward manner we define for every $\mathcal{L}_F$-formula φ a term $\dot{\varphi}$ by replacing $=, \mathsf{N}, \mathsf{T}, \neg, \wedge$, and $\forall$ by the corresponding individual constants $\dot{=}, \dot{\mathsf{N}}, \dot{\mathsf{T}}, \dot{\neg}, \dot{\wedge}$, and $\dot{\forall}$, respectively (using prefix notation). This definition is compatible with term substitution, i.e. we have $\dot{\varphi}[t/x] \equiv \overbrace{\varphi[t/x]}^{\cdot}$.

Since we cannot put the negation outside the scope of T in the axioms of self-reference, the formulae which are T-positive become of special interest. We define T-positive and T-negative formulae simultaneously:

Definition 4

1. $t = s$, $\mathsf{N}(t)$, $\neg t = s$ and $\neg\mathsf{N}(t)$ are T-positive as well as T-negative.
2. $\mathsf{T}(t)$ is T-positive; $\neg\mathsf{T}(t)$ is T-negative.
3. If φ is T-positive (T-negative), so $\neg\varphi$ is T-negative (T-positive).
4. If φ and ψ are T-positive (T-negative), so also $\varphi \wedge \psi$.
5. If φ is T-positive (T-negative), so also $\forall x.\varphi$.

By induction on the formation of formulae we get the following main result for Frege structures [Can96, Th. 8.8.]:

Proposition 5 For T-positive φ we have

$$\mathsf{FON} \vdash \mathsf{T}(\dot{\varphi}) \leftrightarrow \varphi.$$

Now we introduce *propositions* and *propositional functions* (also called *sets* or *classes*) in the following way.

$$\mathsf{P}(x) :\Leftrightarrow \mathsf{T}(x) \vee \mathsf{T}(\dot{\neg}\, x) \quad \text{and} \quad \mathsf{C}(f) :\Leftrightarrow \forall x.\mathsf{P}(f\, x).$$

As an easy consequence we get by diagonalizing of $\dot{\neg}$:

Lemma 6 $\mathsf{FON} \vdash \neg\forall x.\mathsf{P}(x)$.

Remark 7 Aczel's original aim to define a notion of *sets* using Frege structures, can be done by identifying them with propositional functions (cf. [Acz80, Def. 3.4]) and using the suggestive notions

$$\{x|\varphi\} :\equiv \lambda x.\dot{\varphi}, \qquad\qquad t \in s :\Leftrightarrow \mathsf{T}(s\, t).$$

For this reason proposition 5 can be considered as an *abstraction principle* for T-positive formulae. Given such a formula we have:

$$\begin{aligned} t \in \{x|\varphi\} &\leftrightarrow \mathsf{T}(\{x|\varphi\}\, t) \\ &\leftrightarrow \mathsf{T}((\lambda x.\dot{\varphi})\, t) \\ &\leftrightarrow \mathsf{T}(\dot{\varphi}[t/x]) \\ &\leftrightarrow \varphi[t/x] \end{aligned}$$

We can add three natural forms of complete induction over natural numbers to FON. First induction for propositional functions, second so-called truth-induction, and third formulae induction for arbitrary $\mathcal{L}_F$-formulae φ:

$$(\mathsf{C}\text{-}\mathrm{I}_{\mathsf{N}}) \qquad \mathsf{C}(f) \wedge \mathsf{T}(f\,0) \wedge (\forall x.\mathsf{N}(x) \wedge \mathsf{T}(f\,x) \rightarrow \mathsf{T}(f\,(x'))) \rightarrow \forall x.\mathsf{N}(x) \rightarrow \mathsf{T}(f\,x)$$

$$(\mathsf{T}\text{-}\mathrm{I}_{\mathsf{N}}) \qquad \mathsf{T}(f\,0) \wedge (\forall x.\mathsf{N}(x) \wedge \mathsf{T}(f\,x) \rightarrow \mathsf{T}(f\,(x'))) \rightarrow \forall x.\mathsf{N}(x) \rightarrow \mathsf{T}(f\,x)$$

$$(\mathrm{F}\text{-}\mathrm{I}_{\mathsf{N}}) \qquad \varphi(0) \wedge (\forall x.\mathsf{N}(x) \wedge \varphi(x) \rightarrow \varphi(x')) \rightarrow \forall x.\mathsf{N}(x) \rightarrow \varphi(x)$$

The theories $\mathsf{FON} + (\mathsf{C}\text{-}\mathrm{I}_{\mathsf{N}})$, $\mathsf{FON} + (\mathsf{T}\text{-}\mathrm{I}_{\mathsf{N}})$, and $\mathsf{FON} + (\mathrm{F}\text{-}\mathrm{I}_{\mathsf{N}})$ are equivalent to the theories MF_c, MF_p, and MF, resp. of [Can96]. As proof-theoretic results we have:

Proposition 8 [Can96, Th. 58.1]

$$|\mathsf{FON} + (\mathsf{C}\text{-}\mathrm{I}_{\mathsf{N}})\,| = |\mathrm{MF}_c| = \varepsilon_0$$
$$|\mathsf{FON} + (\mathsf{T}\text{-}\mathrm{I}_{\mathsf{N}})\,| = |\mathrm{MF}_p| = \varphi\,\omega\,0$$
$$|\mathsf{FON} + (\mathrm{F}\text{-}\mathrm{I}_{\mathsf{N}})\,| = |\mathrm{MF}| = \varphi\,\varepsilon_0\,0$$

Proof: For the upper bounds we refer to [Can96]. For the lower bound we will give a sketch which illustrates that proposition 5 provides the crucial step. In the case of $(\mathsf{C}\text{-}\mathrm{I}_{\mathsf{N}})$ this proposition allows to represent $\mathcal{L}_p$-formulae by propositional functions and we get the lower bound from the equivalence of PA and $\mathsf{BON} + (\mathrm{F}\text{-}\mathrm{I}_{\mathsf{N}})$[1] (cf. [FJ93, Cor. 6]). For the two other forms of induction we embed theories of inductive definitions, as described for MF in [Can96, § 10A]. The well-known theory $\widehat{\mathrm{ID}}_1$ (cf. [Fef82]) comprises (non necessarily least) fixed points of P-positive arithmetical operator forms $\varphi(P, x)$. It can be embedded in $\mathsf{FON} + (\mathrm{F}\text{-}\mathrm{I}_{\mathsf{N}})$ by interpreting the fixed points as $\mathsf{T}(p_\varphi\, x)$ where p_φ is the recursion theoretic fixed point of $\lambda p, x.\dot{\varphi}$, with $\overbrace{P(t)}^{\cdot} := \overbrace{\mathsf{T}(p\,t)}^{\cdot}$. It follows that this formula is T-positive and we can apply proposition 5. The translation of the induction of $\widehat{\mathrm{ID}}_1$ is contained in $(\mathrm{F}\text{-}\mathrm{I}_{\mathsf{N}})$. For $(\mathsf{T}\text{-}\mathrm{I}_{\mathsf{N}})$ we have to consider the theory $\mathrm{ID}_1^{\#}$, introduced in [JS9x]. This theory is defined as $\widehat{\mathrm{ID}}_1$, but induction is restricted to formulae which are positive in the fixed point constants. It is easy to see that this induction is entailed by $(\mathsf{T}\text{-}\mathrm{I}_{\mathsf{N}})$. Corollary 8 of [JS9x] yields $\varphi\,\omega\,0$ as lower bound for $\mathrm{ID}_1^{\#}$ and so also for $\mathsf{FON} + (\mathsf{T}\text{-}\mathrm{I}_{\mathsf{N}})$. $\square$

4 Frege Structures for partial applicative theories

4.1 Strictness

If we want to define Frege structures for partial applicative theories, we get in trouble with possibly undefined terms. The first attempt would be to introduce in

[1] Of course, here $(\mathrm{F}\text{-}\mathrm{I}_{\mathrm{N}})$ means formulae induction for $\mathcal{L}_p$-formulae.

FON (now formulated over BON) a further constant $\dot{\downarrow}$ and to axiomatize truth for negated existence in the following way:

$$\neg(x\downarrow) \leftrightarrow \mathsf{T}(\dot{\neg}\,(\dot{\downarrow}\,x))$$

But this axiom would be senseless, since variables are always defined in the logic of partial terms.

The corresponding axiom scheme:

$$\neg(t\downarrow) \leftrightarrow \mathsf{T}(\dot{\neg}\,(\dot{\downarrow}\,t))$$

violates the strictness axiom **(S1)**, if there is a undefined term t, or in other words, it proves (Tot).

This problem was overlooked by Beeson in [Bee85] where he worked in a partial framework. His proposition XVII.9.1 about abstraction can be preserved in the presented form only for the total case, or by ignoring undefined terms (see below). So all other axiomatizations of Frege structures in the literature [FM87b, FM87a, Tur90, Can93, HK95, Can96] are all worked out over total theories.

4.2 Ignoring undefined terms

One possibility to overcome the strictness problems is to ignore undefined terms in the axiomatization of T. This seems not to be ideal with respect to an notion of truth for the whole theory BON, but we can handle the “defined” part of it. So abstraction has to be restricted to formulae containing defined terms only, what is enough to preserve the proof-theoretic strength hoped for. For this theory $\widehat{\mathsf{PFS}}$ we extend BON in the same way as TON.

The language of $\widehat{\mathsf{PFS}}$ is the language of BON extended by the new relation symbol T and new individual constants $\dot{=}, \dot{\mathsf{N}}, \dot{\neg}, \dot{\wedge}, \dot{\forall}$ and $\dot{\mathsf{T}}$.

The axioms of $\widehat{\mathsf{PFS}}$ are the axioms of BON extended to the new language plus the additional ones of FON (cf. p. 4).

Please note, that we have in $\widehat{\mathsf{PFS}}$ no term which represents the existence predicate. But in BON $t\downarrow$ is equivalent to $t = t$, so we can represent the existence predicate using the corresponding equality. Nevertheless, the variables in the axioms of FON are always defined in the logic of partial terms. So, for instance, axiom I.(2)

$$\neg x = y \leftrightarrow \mathsf{T}(\dot{\neg}\,(\dot{=}\,x\,y))$$

is not applicable for $\neg t = s$, if t or s is not defined, especially not for $\neg(t\downarrow)$.

Nevertheless, defining $\dot{\varphi}$ as the term representing φ as in FON, we get the central proposition for Frege structures, but only in a restricted version.

Proposition 9 Let φ be a T-positive formula whose subterms are exactly $t_1, \ldots, t_n$. Then we have

$$\mathsf{FON} \vdash t_1 \downarrow \wedge \ldots \wedge t_n \downarrow \rightarrow (\mathsf{T}(\dot{\varphi}) \leftrightarrow \varphi).$$

Of course, $\widehat{\mathsf{PFS}}$ plus the axiom of totality yields FON. Since in the proof of proposition 8 the interpretation of PA, $\widehat{\mathrm{ID}}_1$, and $\mathrm{ID}_1^{\#}$ can be worked out in $\widehat{\mathsf{PFS}}$ using defined terms only, proposition 9 yields the same proof-theoretic lower bounds.

Proposition 10

$$|\widehat{\mathsf{PFS}} + (\mathsf{C}\text{-}\mathrm{I}_{\mathsf{N}})\,| = \varepsilon_0$$
$$|\widehat{\mathsf{PFS}} + (\mathsf{T}\text{-}\mathrm{I}_{\mathsf{N}})\,| = \varphi\,\omega\,0$$
$$|\widehat{\mathsf{PFS}} + (\mathrm{F}\text{-}\mathrm{I}_{\mathsf{N}})\,| = \varphi\,\varepsilon_0\,0$$

There is one interesting property of $\widehat{\mathsf{PFS}}$ we should mention: The assumption $\forall x.\mathsf{P}(x)$ is consistent (in contrast to FON). This can be shown by a recursion theoretic interpretation of $\widehat{\mathsf{PFS}}$ in the theory BON extended by the recursion theoretic functional $E^{\#}$ (cf. [KM77, p. 695]).

Definition 11 The language of $\mathsf{BON}(E^{\#})$ is $\mathcal{L}_p$ extended by the individual constant $E^{\#}$. The axioms of $\mathsf{BON}(E^{\#})$ are those of BON extended to the new language plus the following axioms:

I. **$E^{\#}$ quantification operator**

(1) $E^{\#}\,f = 0 \leftrightarrow \exists x.\mathsf{N}(x) \wedge f\,x = 0$,

(2) $E^{\#}\,f = 1 \leftrightarrow \forall x.\mathsf{N}(x) \rightarrow \mathsf{N}(f\,x) \wedge f\,x \neq 0$,

(3) $\neg\mathsf{N}(E^{\#}\,f) \leftrightarrow (\forall x.\mathsf{N}(x) \rightarrow \neg f\,x = 0) \wedge \exists x.\mathsf{N}(x) \wedge \neg\mathsf{N}(f\,x)$.

Remark 12 The consistency of $\mathsf{BON}(E^{\#})$ can be shown by a slight modification of the treatment of the theory $\mathsf{BON}(\mu)$ in [FJ93]. Moreover this recursion theoretic interpretation satisfies $\forall x.\mathsf{N}(x)$. In the presence of the so-called set induction, operation induction[2], and formula induction we get as proof-theoretic ordinals ε_0, $\varphi\,\omega\,0$, and $\varphi\,\varepsilon_0\,0$, respectively [FJ93, JS9x].

Proposition 13 $\widehat{\mathsf{PFS}} + \forall x.\mathsf{P}(x) + \forall x.\mathsf{N}(x)$ can be embedded in $\mathsf{BON}(E^{\#}) + \forall x.\mathsf{N}(x)$.

Proof: We translate $\mathsf{T}(t)$ by $\mathsf{d_N}\,1\,0\,t\,0 = 0$. If we choose for the dotted constants the following interpretations

$$\begin{aligned}
\dot{=}^{\star} &:\equiv \lambda x, y.\mathsf{d_N}\,1\,0\,x\,y \\
\dot{\mathsf{N}}^{\star} &:\equiv \lambda x.1 \\
\dot{\neg}^{\star} &:\equiv \lambda x.\mathsf{d_N}\,1\,0\,x\,0 \\
\dot{\forall}^{\star} &:\equiv \lambda f.E^{\#}\,f \\
\dot{\mathsf{T}}^{\star} &:\equiv \lambda x.x
\end{aligned}$$

[2]For the definition of these forms of induction see [JS9x].

and extend $\cdot^\star$ in the straightforward way to all formulae, the verification of the axioms of $\widehat{\mathsf{PFS}}$ becomes a straightforward calculation. We present the case of the negated universal quantification.

$$\begin{array}{rcl}
(\exists x.\mathsf{T}(\dot\neg\,(f\,x)))^\star & \leftrightarrow & \exists x.\mathsf{d_N}\,1\,0\,(\dot\neg\,(f\,x))^\star\,0 = 0 \\
& \leftrightarrow & \exists x.\mathsf{d_N}\,1\,0\,(\mathsf{d_N}\,1\,0\,(f\,x)\,0)\,0 = 0 \\
& \leftrightarrow & \exists x.\mathsf{d_N}\,1\,0\,(f\,x)\,0 \neq 0 \\
& \leftrightarrow & \exists x.f\,x = 0 \\
& \leftrightarrow & E^{\#}\,f = 0 \\
& \leftrightarrow & \mathsf{d_N}\,1\,0\,(E^{\#}\,f)\,0 \neq 0 \\
& \leftrightarrow & \dot\neg^\star\,(E^{\#}\,f) \neq 0 \\
& \leftrightarrow & \mathsf{d_N}\,1\,0\,(\dot\neg\,(\dot\forall\,f))^\star\,0 = 0 \\
& \leftrightarrow & \mathsf{T}(\dot\neg\,(\dot\forall\,f))^\star
\end{array} \qquad \square$$

This interpretation also yields the same proof-theoretic upper bounds for $\widehat{\mathsf{PFS}}$ in the presence of $\forall x.\mathsf{P}(x)$ and $\forall x.\mathsf{N}(x)$, since by this translation (C-$\mathsf{I_N}$) and (T-$\mathsf{I_N}$) become instances of the set induction and operation induction, respectively.

Corollary 14

$$\begin{aligned}
|\widehat{\mathsf{PFS}} + \forall x.\mathsf{P}(x) + \forall x.\mathsf{N}(x) + (\mathsf{C}\text{-}\mathsf{I_N})\,| &= \varepsilon_0 \\
|\widehat{\mathsf{PFS}} + \forall x.\mathsf{P}(x) + \forall x.\mathsf{N}(x) + (\mathsf{T}\text{-}\mathsf{I_N})\,| &= \varphi\,\omega\,0 \\
|\widehat{\mathsf{PFS}} + \forall x.\mathsf{P}(x) + \forall x.\mathsf{N}(x) + (\mathrm{F}\text{-}\mathsf{I_N})\,| &= \varphi\,\varepsilon_0\,0
\end{aligned}$$

4.3 Using pointers

In a second approach we will solve the strictness problems by use of *pointers*. Therefore we associate an arbitrary, maybe undefined term t with an object $\bar{t}$ which is always defined and allows to reconstruct t itself. So $\bar{t}$ plays the role of a pointer to t. It is easy to find a proper candidate for $\bar{t}$ in BON: the constant function $\lambda y.t$ (with $y \notin \mathrm{FV}(t)$) is always defined and we can reconstruct t trivially by applying it to 0.

In contrast to the total case, where a map $t \mapsto \bar{t}$ is definable within the theory as $\lambda f.\mathsf{k}\,f$, in the partial setting, the definition is inductively on the formation of terms. This definition causes some difficulties, especially it is no longer compatible with substitution. A extensively discussion of this problem can be found in [Str96].

So the map $t \mapsto \bar{t}$ becomes a meta theoretical operation. Fortunately we do not need the operation at the object level to establish abstraction in our theory. Instead of "shifting" the terms t to the pointer $\bar{t}$ in the formalization we start with the pointers as constant functions and reevaluate the value by applying 0. So the truth condition for negated existence, which would by in a direct form only expressible by the schema $\neg(t\downarrow) \leftrightarrow \mathsf{T}(\dot{\neg}\,(\dot{\downarrow}\,\bar{t}))$ can be written as an axiom as $\mathsf{Cf}(x) \rightarrow (\neg(\mathsf{val}\,x\downarrow) \leftrightarrow \mathsf{T}(\dot{\neg}\,(\dot{\downarrow}\,x)))$ using the abbreviations

$$\mathsf{Cf}(t) :\Leftrightarrow \forall x, y. t\,x \simeq t\,y \quad \text{and} \quad \mathsf{val} := \lambda x.x\,0.$$

So we define the theory PFS of Frege structures for BON as follows.

The language $\mathcal{L}_F^p$ of PFS is the language $\mathcal{L}_p$ of BON extended by the new relation symbol T and new individual constants $\dot{\downarrow}, \dot{=}, \dot{\mathsf{N}}, \dot{\neg}, \dot{\wedge}, \dot{\forall}$ and $\dot{\mathsf{T}}$.

The axioms of PFS are the axioms of BON extended to the new language plus the additional axioms of FON where the first group of closure under prime formulae of TON is replaced by

I. **Closure under prime formulae of** BON

(1) $\mathsf{Cf}(x) \rightarrow (\mathsf{val}\,x\downarrow \leftrightarrow \mathsf{T}(\dot{\downarrow}\,x))$

(2) $\mathsf{Cf}(x) \rightarrow (\neg(\mathsf{val}\,x\downarrow) \leftrightarrow \mathsf{T}(\dot{\neg}\,(\dot{\downarrow}\,x)))$

(3) $\mathsf{Cf}(x) \wedge \mathsf{Cf}(y) \rightarrow (\mathsf{val}\,x = \mathsf{val}\,y \leftrightarrow \mathsf{T}(\dot{=}\,x\,y))$

(4) $\mathsf{Cf}(x) \wedge \mathsf{Cf}(y) \rightarrow (\neg\mathsf{val}\,x = \mathsf{val}\,y \leftrightarrow \mathsf{T}(\dot{\neg}\,(\dot{=}\,x\,y)))$

(5) $\mathsf{Cf}(x) \rightarrow (\mathsf{N}(\mathsf{val}\,x) \leftrightarrow \mathsf{T}(\dot{\mathsf{N}}\,x))$

(6) $\mathsf{Cf}(x) \rightarrow (\neg\mathsf{N}(\mathsf{val}\,x) \leftrightarrow \mathsf{T}(\dot{\neg}\,(\dot{\mathsf{N}}\,x)))$

Note that self-reference, inherited from FON, leaves negation inside the scope of the T-predicate. So there is no problem with strictness, and in contrast to the other prime formulae, we do not need to "shift" the argument.

4.3.1 PFS + (Tot)

First we will show that our approach is compatible with the total version, in the sense that PFS + (Tot) is equivalent FON.[3] Although the theories are not so trivially equivalent like BON + (Tot) and TON there are two interpretations into each other. (In the following it should be clear from the context, whether $\dot{=}$ and $\dot{\mathsf{N}}$ belongs to $\mathcal{L}_F^p$ or $\mathcal{L}_F$ resp.)

Proposition 15 Let be

$$\begin{aligned}
\dot{\downarrow}^* &:= (\lambda x.\dot{=}\,0\,0) \\
\dot{=}^* &:= (\lambda x, y.\dot{=}\,(\mathsf{val}\,x)\,(\mathsf{val}\,y)) \\
\dot{\mathsf{N}}^* &:= (\lambda x.\dot{\mathsf{N}}\,(\mathsf{val}\,x)) \\
c^* &:= c \qquad \text{for } c \text{ another individual constant}
\end{aligned}$$

[3] For the meaning of proof-theoretic equivalence we refer to [Fef88].

and extend this interpretation with $(t\downarrow)^* :\Leftrightarrow 0 = 0$ to all formulae φ of $\mathcal{L}_F^p$ in the straightforward way. So we have:

$$\mathsf{PFS} \vdash \varphi \Rightarrow \mathsf{FON} \vdash \varphi^*$$

Proposition 16 If we set

$$\begin{array}{rcll} \dot{=}^+ & := & \lambda x, y. \dot{=} (\mathsf{k}\, x)\, (\mathsf{k}\, y) & \\ \dot{\mathsf{N}}^+ & := & \lambda x. \dot{\mathsf{N}}\, (\mathsf{k}\, x) & \\ c^+ & := & c & \text{for } c \text{ another individual constant} \end{array}$$

and extend $\cdot^+$ in the standard way to all formulae ψ of $\mathcal{L}_F$, we get:

$$\mathsf{FON} \vdash \psi \Rightarrow \mathsf{PFS} + (\mathrm{Tot}) \vdash \psi^+$$

Proof: Both propositions are easy calculations, we present only the interpretation of the axiom for (positive) equality of FON interpreted in $\mathsf{PFS} + (\mathrm{Tot})$. Note that $\forall x. \mathsf{Cf}(\mathsf{k}\, x)$ holds in $\mathsf{PFS} + (\mathrm{Tot})$.

$$\begin{array}{rcl} \mathsf{T}(\dot{=}\, x\, y)^+ & \leftrightarrow & \mathsf{T}((\lambda x, y. \dot{=} (\mathsf{k}\, x)\, (\mathsf{k}\, y))\, x\, y) \\ & \leftrightarrow & \mathsf{T}(\dot{=} (\mathsf{k}\, x)\, (\mathsf{k}\, y)) \\ & \leftrightarrow & \mathsf{val}\, (\mathsf{k}\, x) = \mathsf{val}\, (\mathsf{k}\, y) \\ & \leftrightarrow & \mathsf{k}\, x\, 0 = \mathsf{k}\, y\, 0 \\ & \leftrightarrow & x = y \\ & \leftrightarrow & (x = y)^+ \end{array}$$

□

4.3.2 Abstraction

To get the required abstraction principle we have to use pointers in the definition of those terms representing formulae. Since an argument of a prime formulae can be an undefined object, we choose in the representation of $t\downarrow$, $\mathsf{N}(t)$, and $t = s$ the corresponding pointers, i.e. we "shift" t and s to the constant functions $\lambda y.t$ and $\lambda y.s$, respectively ($y \notin \mathrm{FV}(t, s)$).

Definition 17 Inductively we assign a term $\dot{\varphi}$ to every formula φ of $\mathcal{L}_F^p$:

$$\begin{array}{rcll} \dot{\overbrace{t\downarrow}} & :\equiv & \dot{\downarrow}\, (\lambda y.t) & y \notin \mathrm{FV}(t) \\ \dot{\overbrace{\mathsf{N}t}} & :\equiv & \dot{\mathsf{N}}\, (\lambda y.t) & y \notin \mathrm{FV}(t) \\ \dot{\overbrace{t = s}} & :\equiv & \dot{=}\, (\lambda y.t)\, (\lambda y.s) & y \notin \mathrm{FV}(t, s) \\ \dot{\overbrace{\mathsf{T}(t)}} & :\equiv & \dot{\mathsf{T}}\, t & \\ \dot{\overbrace{\neg\varphi}} & :\equiv & \dot{\neg}\, \dot{\varphi} & \\ \dot{\overbrace{\varphi \wedge \psi}} & :\equiv & \dot{\wedge}\, \dot{\varphi}\, \dot{\psi} & \\ \dot{\overbrace{\forall y.\varphi}} & :\equiv & \dot{\forall}\, (\lambda y.\dot{\varphi}) & \end{array}$$

Obviously $\mathrm{FV}(\varphi) = \mathrm{FV}(\dot{\varphi})$ holds. But note that the operation which maps φ to $\dot{\varphi}$ makes substantial use of λ-abstraction, so it does not commute with substitution, just as λ-abstraction. However, we do not need this property as long as this map is only used on the meta-level. In particular it is not necessary to prove the abstraction theorem.

Proposition 18 For T-positive φ we have

$$\mathsf{PFS} \vdash \mathsf{T}(\dot{\varphi}) \leftrightarrow \varphi.$$

Proof: The statement is provable by induction on φ, as in the total case and we present only the case of $\varphi \equiv t\downarrow$ to illustrate the behavior of the pointer.

$$\varphi \equiv t\downarrow \; (y \notin \mathrm{FV}(t)): \qquad \begin{aligned} \mathsf{T}(\dot{\varphi}) &\leftrightarrow \mathsf{T}(\dot{\downarrow}(\lambda y.t)) \\ &\leftrightarrow \mathsf{val}\,(\lambda y.t)\downarrow \\ &\leftrightarrow (\lambda y.t)\,0\downarrow \\ &\leftrightarrow t\downarrow \end{aligned}$$

□

4.3.3 Proof-theoretic results

Again proposition 18 is the only tool we need to prove the lower bounds and so we also get

Proposition 19

$$|\mathsf{PFS} + (\mathsf{C}\text{-}\mathrm{I}_{\mathsf{N}})| = \varepsilon_0$$
$$|\mathsf{PFS} + (\mathsf{T}\text{-}\mathrm{I}_{\mathsf{N}})| = \varphi\,\omega\,0$$
$$|\mathsf{PFS} + (\mathrm{F}\text{-}\mathrm{I}_{\mathsf{N}})| = \varphi\,\varepsilon_0\,0$$

4.3.4 $\forall x.\mathsf{P}(x)$

As for $\widehat{\mathsf{PFS}}$ the question arises whether $\forall x.\mathsf{P}(x)$ is consistent. This question is still open, but we can prove that a recursion theoretic interpretation as presented for $\widehat{\mathsf{PFS}}$ fails.

Proposition 20 For all closed $\mathcal{L}_F^p$ terms t and s the theory

$$\mathsf{PFS} + (\forall x.\mathsf{N}(x)) + (\forall x.\mathsf{T}(x) \leftrightarrow t\,x = s)$$

is inconsistent.

Proof: Since we can replace $\mathsf{T}(x)$ by an equation of natural numbers, the idea of the proof is to diagonalize negated existence using $\mathsf{d_N}$. Let f be the function satisfying the equation

$$f\,x \simeq \mathsf{d_N}\,(\lambda y.0)\,(\lambda y.\mathsf{not_N})\,(t\,(\dot{\neg}\,(\dot{\downarrow}\,f)))\,s\,0$$

That is: f satisfies

$$f\,x \simeq \begin{cases} 0 & \text{if } t\,(\dot{\neg}\,(\dot{\downarrow} f)) = s, \\ \mathsf{not_N} & \text{if } t\,(\dot{\neg}\,(\dot{\downarrow} f)) \neq s. \end{cases}$$

f is a constant function, since x does not occur on the right side of its definition. By case distinction on $\neg(f\,0\downarrow)$ and $f\,0\downarrow$ we get

$$\begin{array}{rcll} \neg(f\,0\downarrow) & \to & \neg(\mathsf{val}\,f\downarrow) & \\ & \to & \mathsf{T}(\dot{\neg}\,(\dot{\downarrow} f)) & \text{I.(2)} \\ & \to & t\,(\dot{\neg}\,(\dot{\downarrow} f)) = s & \\ & \to & f\,0 = 0 & \\ & \to & f\,0\downarrow & \\ f\,0\downarrow & \to & \mathsf{val}\,f\downarrow & \\ & \to & \mathsf{T}(\dot{\downarrow} f) & \text{I.(1)} \\ & \to & \neg t\,(\dot{\neg}\,(\dot{\downarrow} f)) = s & \end{array}$$

With $f\,0 = \mathsf{d_N}\,(\lambda y.0)\,(\lambda y.\mathsf{s_N}\,(f\,x))\,(t\,(\dot{\neg}\,(\dot{\downarrow} f)))\,s\,0$ and $f\,0\downarrow$ it follows from strictness that $t\,(\dot{\neg}\,(\dot{\downarrow} f))\downarrow$ and $s\downarrow$ hold. So we get $f\,0 = \mathsf{not_N}$ from the definition of f what contradicts $\forall x.\mathsf{N}(x)$. □

Since in the general case we have only very few functional structure on propositions we conjecture that $\forall x.\mathsf{P}(x)$ is consistent with PFS, but proposition 20 shows that it cannot have a smooth recursion theoretic interpretation.

4.3.5 `delay` and `force`

As a concluding remark we mention that our notion of pointer is closely related to the concept of `delay` and `force` for the functional programming language Scheme. `delay` is introduced (in most Scheme-dialects) to block the evaluation of an argument and so makes *laziness* possible. `force` is used to enforce the delayed evaluation. The `delay`-function is not definable as a pure functional construct in Scheme, since the evaluation of Scheme is strict and one would run into similar problems as those discussed here. So it is introduced as a kind of meta-function which operates on the syntax prior to functional evaluation. As described in [AS85, S.264] or [Dyb87, S.90], `delay` is conceptionally the same as the map $t \mapsto \bar{t}\ (\equiv \lambda y.t)$, and `force` coincides with val.

In [Stä9x] Stärk gives a connection between Scheme and applicative theories (based on Strahm's version with explicit substitution [Str96]). The analogy between `delay` and the map $t \mapsto \bar{t}$ gives a hint, how to extend this approach to *streams* which are based on `delay`. But as `delay` destroys the call-by-value evaluation of Scheme, $t \mapsto \bar{t}$ cannot be defined on the object-level while preserving partiality. So at this stage streams can be handled only at the meta-level.

References

[Acz80] Peter Aczel. Frege Structures and the Notion of Proposition, Truth and Set. In J. Barwise, H. Keisler, and K. Kunen, editors, *The Kleene Symposium*, pages 31–59. North-Holland, 1980.

[AS85] Harold Abelson and Gerald Sussmann. *Structure and Interpretation of Computer Programs*. MIT Press, 1985.

[Bee85] Michael Beeson. *Foundations of Constructive Mathematics*. Ergebnisse der Mathematik und ihrer Grenzgebiete; 3.Folge, Bd. 6. Springer, Berlin, 1985.

[Can93] Andrea Cantini. Extending the first-order Theory of Combinators with Self-Referential Truth. *Journal of Symbolic Logic*, 58(2), June 1993.

[Can96] Andrea Cantini. *Logical Frameworks for Truth and Abstraction*, volume 135 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1996.

[Dyb87] R. Kent Dybvig. *The SCHEME Programming Language*. Prentice Hall, 1987.

[Fef75] Solomon 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, 1975.

[Fef79] Solomon 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.

[Fef82] Solomon Feferman. Iterated Inductive Fixed-Point Theories: Application to Handcock's Conjecture. In G. Metakides, editor, *Patras Logic Symposion*, pages 171–196. North–Holland, Amsterdam, 1982.

[Fef88] Solomon Feferman. Hilbert's Program relativized: Proof-theoretical and foundational Reductions. *Journal of Symbolic Logic*, 53:364–384, June 1988.

[FJ93] Solomon Feferman and Gerhard Jäger. Systems of explicit mathematics with non-constructive μ-operator. Part I. *Annals of pure and applied logic*, 65(3):243–263, December 1993.

[FM87a] Robert Flagg and John Myhill. An Extension of Frege Structures. In D. Kueker, E. Lopez-Escobar, and C. Smith, editors, *Mathematical Logic and theoretical computer science*, volume 106 of *Lecture Notes in pure and applied mathematics*, pages 197–217. Dekker, New York and Basel, 1987.

[FM87b] Robert Flagg and John Myhill. Implication and Analysis in Classical Frege Structures. *Annals of pure and applied logic*, 34:33–85, 1987.

[HK95] Susumu Hayashi and Satoshi Kobayashi. A New Formulation of Feferman's System of Functions and Classes and its Relation to Frege Structures. *International Journal of Foundations of Computer Science*, 6(3):187–202, September 1995.

[JS95] Gerhard Jäger and Thomas Strahm. Totality in applicative theories. *Annals of Pure and Applied Logic*, 74:105–120, 1995.

[JS9x] Gerhard Jäger and Thomas Strahm. Some theories with positive induction of ordinal strength $\varphi\omega 0$. *to appear in: Journal of Symbolic Logic*, 199x.

[Kah95] Reinhard Kahle. Weak inductions in theories of operations and numbers. Technical Report 95-001, IAM, Universität Bern, Januar 1995.

[KM77] Alexander Kechris and Yiannis Moschovakis. Recursion in higher types. In Jon Barwise, editor, *Handbook of Mathematical Logic*, volume 90 of *Studies in Logic and the Foundations of Mathematics*, chapter C.6, pages 681–737. North-Holland, 1977.

[Sco75] Dana Scott. Combinators and Classes. In C. Böhm, editor, *λ-Calculus and Computer Science Theory*, volume 37 of *Lecture Notes in Computer Science*, pages 1–26. Springer, 1975.

[Stä9x] Robert Stärk. Applicative theories with explicit substitutions and call-by-value evaluation, 199x. Draft notes, January 23, 1995.

[Str96] Thomas Strahm. Partial applicative theories and explicit substitutions. *Journal of Logic and Computation*, 6(1):55–77, 1996.

[Tur90] Raymond Turner. *Truth and Modality for Knowledge Representation*. Pitman, 1990.