CHAPTER V

Gödel's Functional ("Dialectica") Interpretation

Jeremy Avigad
Department of Philosophy, Carnegie Mellon University
Pittsburgh, PA 15213

Solomon Feferman
Departments of Mathematics and Philosophy, Stanford University
Stanford, CA 94305

Contents

HANDBOOK OF PROOF THEORY
Edited by S. R. Buss

1. Introduction

1.1. Functional interpretations

In 1958, Kurt Gödel published in the journal *Dialectica* an interpretation of intuitionistic arithmetic in a quantifier-free theory of functionals of finite type, an interpretation which has since come to be known as Gödel's functional or *Dialectica* interpretation. When combined with Gödel's *double-negation* interpretation, which reduces classical arithmetic to intuitionistic arithmetic, the Dialectica interpretation (also referred to below as the D-interpretation) yields a reduction of the classical theory as well. This approach has since been extended and adapted to other theories, but the pattern usually follows Gödel's original example:

- first, one reduces a classical theory C to a variant I based on intuitionistic logic;
- then one reduces the theory I to a quantifier-free functional theory F.

Functional interpretations of this form can be interesting for a number of reasons. To begin with, the work can be seen as a contribution to a modified form of Hilbert's program, since, from a foundational point of view, the consistency of C is thereby reduced to the consistency of F. Subsequent analyses of F often lead to further gains, yielding, for example, reductions of (*prima facie*)

- infinitary systems to finitary ones,
- non-constructive systems to constructive ones, and
- impredicative systems to predicative ones.

Secondly, functional interpretation provides a way of extracting (or "unwinding") constructive information from proofs in I or C. For example, as a direct consequence of the interpretation one usually obtains the result that any recursive function whose totality can be proven either in I or in C is represented by a term of F. Via an additional interpretation of F in I, this characterization is in fact usually shown to be exact. It often turns out that the terms of F represent a natural class of functions, such as the primitive recursive or polynomial-time computable functions. In other cases, the theory F embodies independently interesting computational constructs, such as bar-recursion or polymorphism, which are discussed in this chapter.

Finally, functional interpretations often provide a useful stepping-stone to other goals. For example, the analyses of Gödel's functional calculus T due to Tait and Howard provide an alternative means to the ordinal analysis of classical arithmetic, and non-constructive interpretations due to the second author yield consequences for various subsystems of second-order arithmetic. Many of these results can also be obtained using Herbrand-Gentzen methods of syntactic transformation, and in certain domains (for example, in the ordinal analysis of strong subsystems of analysis and set-theory) these latter methods are the only ones currently known.[1] Nonetheless,

[1]Whether there is a fundamental obstacle against the use of D-interpretations for such purposes is a matter of methodological interest.

functional interpretations have proven to be a relatively powerful and versatile tool, with distinct advantages that will be illustrated below.

1.2. Historical background

Although the Dialectica interpretation was not published until 1958, Gödel began to develop these ideas in the latter part of the 1930's as a possible modification of Hilbert's program (cf. Sieg and Parsons [1994]), and the D-interpretation itself was arrived at by 1941 and presented in a lecture to the Mathematics and Philosophy Clubs at Yale University. Full notes for that lecture are found in Gödel's *Nachlass* and have been made available in Volume III of his *Collected Works* as Gödel [1941].

The interpretation was first brought to the attention of the logic community at large in a lecture by Georg Kreisel at the Summer Institute in Symbolic Logic held at Cornell University in 1957 (cf. the notes Kreisel [1957]). The publication Gödel [1958], in German, was for an issue of *Dialectica* in honor of Paul Bernays' 70th birthday. Gödel worked on a translation and expansion of that article for another issue in honor of Bernays a decade later, but though it reached the stage of proof sheets it never appeared in print until it was retrieved from the *Nachlass* for publication in Volume II of his *Collected Works* as Gödel [1972].

Subsequent work on functional interpretations was carried out in the 1960s through the early 1980s, following the initial developments by Kreisel in the late 1950s. After a lapse in the latter part of the 1980s, there has been a resurgence of interest in these methods in the 1990s, yielding a number of new applications. Some of the different general directions of work may be indicated roughly as follows (more or less along the lines of Troelstra [1990,pp. 236–239], which should be consulted for publication information concerning items not found in the references to this chapter).

1. The functionals in Gödel's interpretation are defined by schemata for explicit definition and a natural extension of primitive recursion to finite types, and are therefore called *primitive recursive functionals of finite type.*[2] There have been a number of investigations of this class of functionals, which have set-theoretic, recursion-theoretic and term models. Prominent in the study of the latter are various methods of normalization and related assignments of ordinals. Among the contributors here that one should mention are S. Hinata, J. Diller, W. Tait and W. Howard.
2. Next, Gödel's interpretation has been adapted and extended both to stronger and weaker theories. Here, briefly, in semi-chronological order, are some of the kinds of systems to which the D-interpretation has been extended, either directly in the case of intuitionistic systems, or indirectly by combination with the negative translation in the case of classical systems. (We also indicate some of the main contributors to each):

[2]This will be distinguished below from the class of functionals introduced by Kleene [1959b] using a weaker predicative extension of primitive recursion to finite types.

   (a) Intuitionistic arithmetic with principles of transfinite induction (G. Kreisel).
   (b) Impredicative ("full") classical analysis formulated with function variables (G. Kreisel, C. Spector, W. Howard, H. Luckhardt).
   (c) Subsystems of classical arithmetic (C. Parsons).
   (d) Impredicative systems of classical analysis formulated with set variables (J.-Y. Girard).
   (e) Intuitionistic and classical theories of ordinals (W. Howard, S. Feferman).
   (f) Predicative systems of classical analysis (W. Maass, S. Feferman).
   (g) Classical analysis with a game quantifier (W. Friedrich).
   (h) Systems of feasible arithmetic (S.A. Cook and A. Urquhart).
   (i) Iterated arithmetical fixed point theories (J. Avigad).
3. Furthermore, the interpretations have been applied towards a number of interesting proof theoretic ends. These applications include:
   (a) The no-counterexample interpretation for Peano Arithmetic (G. Kreisel).
   (b) Closure and conservation results for intuitionistic systems (A.S. Troelstra).
   (c) Conservation results for classical systems and characterization of the provably recursive functionals (C. Parsons, S. Feferman, U. Kohlenbach).
4. Finally, useful variants of Gödel's original interpretation have also been developed, among them those due to J. Shoenfield, J. Diller and W. Nahm, M. Beeson, U. Kohlenbach.

These lists, as well as the treatment below, are not comprehensive. For more information we refer the reader to the surveys Troelstra [1990] and Feferman [1993], the encyclopedic treatment of Troelstra [1973], and the related articles Feferman [1977] and Troelstra [1977].[3]

1.3. An overview of this chapter

In this chapter we try to give a broad and self-contained survey of the D-interpretation and its applications. First we provide the details of the interpretation of arithmetic, explicitly presenting the relevant axioms and the functional theory T. In section 3 we present some of the useful information that can be gleaned from the interpretation, and take a broader look at the general form of the interpretation in order to understand better how it might be adapted to other contexts.

The presentation of a functional theory raises the issue of what its models look like. In the case of Gödel's T, that issue is addressed in section 4. In section 5 we

[3]Unfortunately, notation in literature is not uniform, and here we have struck some compromises. For example, though the use of PR^{ω} to denote Gödel's theory of the primitive recursive functionals of finite type in Feferman [1977,1990] is more descriptive, here we follow Gödel's original use of the name T.

show how to weaken T in order to obtain useful characterizations of the provably total recursive functions of certain fragments of arithmetic, namely $I\Sigma_1$ and S_2^1.

In section 6 we go in the opposite direction and consider a strengthening of T that suffices to interpret full second-order arithmetic ("analysis"). In the following three sections we then consider ways in which the D-interpretation can also be used to obtain information regarding a number of interesting subsystems of analysis.

Finally, in the last section we show how functional theories based on polymorphic types arise in a natural way from a functional interpretation of full analysis, formulated using predicate variables instead of function variables.

The authors are very much indebted to Ulrich Kohlenbach for numerous comments and suggestions, as well as corrections to a draft of this chapter.

2. The Dialectica interpretation of arithmetic

2.1. Theories of arithmetic and the double-negation interpretation

The first-order theory Peano arithmetic, or PA, has already been discussed in Chapter II. Peano arithmetic has its intuitionistic analogue in a theory known as Heyting arithmetic, or HA, which differs from the former only in that it uses intuitionistic axioms and rules as the underlying predicate logic. For concreteness, we take the following list of axioms and rules, which is that used by Gödel [1958] (cf. also Troelstra [1973,1977]):

1. From φ, $\varphi \to \psi$ conclude ψ
2. From $\varphi \to \psi$, $\psi \to \theta$ conclude $\varphi \to \theta$
3. $\varphi \lor \varphi \to \varphi$, $\varphi \to \varphi \land \varphi$
4. $\varphi \to \varphi \lor \psi$, $\varphi \land \psi \to \varphi$
5. $\varphi \lor \psi \to \psi \lor \varphi$, $\varphi \land \psi \to \psi \land \varphi$
6. From $\varphi \to \psi$ conclude $\theta \lor \varphi \to \theta \lor \psi$
7. From $\varphi \to (\psi \to \theta)$ conclude $\varphi \land \psi \to \theta$
8. $\bot \to \theta$
9. From $\varphi \to \psi$ conclude $\varphi \to \forall x\, \psi$, assuming x is not free in φ
10. $\forall x\, \varphi \to \varphi[t/x]$, assuming t is free for x in φ
11. $\varphi[t/x] \to \exists x\, \varphi$, assuming t is free for x in φ
12. From $\varphi \to \psi$ conclude $\exists x\, \varphi \to \psi$, assuming x is not free in ψ

Here $\varphi[t/x]$ denotes the result of replacing all free occurrences of the variable x by t in the formula φ. It is common in intuitionistic systems to define negation by

$$\neg A = A \to \bot$$

where $\bot$ is an identically false statement, or "contradiction"; $\bot$ may be taken to be a closed atomic formula, or identified with $0 = 1$. We take the equality axioms to be given by

1. $x = x$
2. $x = y \rightarrow (\varphi[x/z] \rightarrow \varphi[y/z])$, where φ is atomic.

Finally, classical logic is obtained by adding to this list *tertium non datur*, the law of excluded middle:

$$\varphi \vee \neg\varphi.$$

Classical predicate logic can be reduced in a simple way to intuitionistic predicate logic via the so-called *double-negation* (or *negative*) translation due (independently) to Gödel and Gentzen. This is defined as follows:

1. $\varphi^N = \neg\neg\varphi$, for φ atomic
2. $(\varphi \wedge \psi)^N = \varphi^N \wedge \psi^N$
3. $(\varphi \vee \psi)^N = \neg(\neg\varphi^N \wedge \neg\psi^N)$
4. $(\varphi \rightarrow \psi)^N = \varphi^N \rightarrow \psi^N$
5. $(\forall x\ \varphi(x))^N = \forall x\ \varphi(x)^N$
6. $(\exists x\ \varphi(x))^N = \neg\forall x\ \neg\varphi(x)^N$

The "double negation" appellation is due not only to clause 1, but also the fact that $(\varphi \vee \psi)^N \leftrightarrow \neg\neg(\varphi^N \vee \psi^N)$ and $(\exists x\ \varphi)^N \leftrightarrow \neg\neg\exists x\ \varphi^N$ are provable intuitionistically.

Clearly, from a classical point of view every formula is equivalent to its N-interpretation. Moreover, one has the following

2.1.1. Theorem. *Suppose a set of axioms S proves a formula φ using classical logic. Then S^N proves φ^N using intuitionistic logic.*

For a proof of this and more general results, see Troelstra [1973] or Troelstra [1977, section 3.8]. For the case at hand, the preceding theorem provides the following useful

2.1.2. Corollary. *Suppose PA proves a formula φ. Then HA proves φ^N.*

2.1.3. Proof. We only need to verify that *HA* proves the N-interpretation of each axiom and rule of *PA*. Since *HA* proves $x = y \vee \neg(x = y)$ (using a double induction on x and y), the N-interpretations of the quantifier-free axioms of *PA* follow from their *HA* counterparts. Finally, the N-interpretation of an instance of the induction scheme is again an instance of the induction scheme. □

2.2. The primitive recursive functionals of finite type

The Dialectica interpretation reduces *HA* to a theory T which axiomatizes a class of functionals that Gödel called the "primitive recursive functionals of finite type."[4] While T is quantifier-free, its language is many-sorted, in that each term is assigned a *type symbol*, or *type* for short. The set of types is generated inductively by the following rules:

[4]For a variant of the D-interpretation that applies directly to *PA*, see Shoenfield [1967].

1. 0 is a type.
2. If σ and τ are types then so is $\sigma \to \tau$[5]

The intended use is that objects of type 0 are considered to be natural numbers and objects of type $\sigma \to \tau$ are considered to be *functions* from objects of type σ to objects of type τ. The latter may be interpreted as constructive functions in some sense or other, or set-theoretically as all functions of the specified type. By convention we interpret

$$\tau_1 \to \tau_2 \to \ldots \to \tau_n$$

by associating parentheses to the right, i.e. as

$$\tau_1 \to (\tau_2 \to (\ldots \to \tau_n)\ldots).$$

Objects of a type $(\sigma \to \tau) \to \rho$ have function arguments and are usually called *functionals*. An interpretation $\langle M_\sigma : \sigma \text{ a type}\rangle$ of such a typed language is called a *functional type structure* or simply a *type structure*.

One might also wish to include on the preceding list the following additional closure condition:

3. If σ and τ are types then so is $\sigma \times \tau$.

Here $\sigma \times \tau$ denotes the set of ordered pairs of objects $\langle s, t\rangle$ with s an object of type σ and t an object of type τ. This closure condition is eliminable in favor of 2 by "currying," that is, interpreting the type $(\rho \times \sigma) \to \tau$ as the type $\rho \to (\sigma \to \tau)$ and adopting the term-reading conventions described below. (Always in such choices there are trade-offs: fewer closure conditions on type symbols simplifies description of models, normalization of terms, etc., but more closure conditions provide added flexibility and naturalness of formulation.)

To each type σ we can assign a natural number $\mathrm{lev}(\sigma)$ as its *type level*, by:

1. $\mathrm{lev}(0) = 0$
2. $\mathrm{lev}(\sigma \to \tau) = \max(\mathrm{lev}(\sigma) + 1, \mathrm{lev}(\tau))$.

The language of T is said to be of *finite type* since every type is assigned a finite level by this convention. The *pure types* (n) are defined by

1. $(0) = 0$
2. $(n + 1) = (n) \to 0$.

Where the context determines that we are dealing with type symbols, we drop the parentheses around symbols for pure types. Then $\mathrm{lev}(n) = n$ for each $n < \omega$.

We now define the set of terms of T, as well as the relation $t : \tau$ (read "term t has type τ"), inductively via the following rules:

1. There are infinitely many variables $x^\tau, y^\tau, z^\tau, \ldots$ of each type τ.
2. If s is a term of type σ and t a term is of type $\sigma \to \tau$ then $t(s)$ is a term of type τ.
3. 0 is a constant of type 0.

[5]Gödel [1958] used (τ, σ) for our $\sigma \to \tau$. There are many alternative notations in use in the literature such as $(\sigma)\tau$ or $\tau(\sigma)$ or τ^σ, etc.; *caveat emptor.*

4. Sc is a constant of type $0 \to 0$.
5. For each pair of types σ, τ, $\mathrm{K}_{\sigma,\tau}$ is a constant of type $\sigma \to \tau \to \sigma$.
6. For each triple of types ρ, σ, τ, $\mathrm{S}_{\rho,\sigma,\tau}$ is a constant of type $(\rho \to \sigma \to \tau) \to (\rho \to \sigma) \to (\rho \to \tau)$.
7. For each type σ, R_σ is a constant of type $\sigma \to (0 \to \sigma \to \sigma) \to 0 \to \sigma$.

The intended interpretation is that 0 denotes the constant zero, $\mathrm{Sc}(t)$ denotes the successor of t (which we will also write t'), and $t(s)$ denotes the result of applying the function(al) t to the argument s. The intuitive meanings of $\mathrm{K}_{\sigma,\tau}$, $\mathrm{S}_{\rho,\sigma,\tau}$, and R_σ will become clear when we present their defining equations below.

Where possible without ambiguity, we will suppress the type superscripts on the constants K, S, R and on variables, and write, for example, $x, y, z, \ldots$. We will sometimes use capital letters $X, Y, Z \ldots$ to denote variables of functional type. To improve readability, if t is a term of type $\rho \to (\sigma \to \tau)$, r is a term of type ρ, and s is a term of type σ, we will write $t(r, s)$ instead of $t(r)(s)$ (which we in turn interpret by associating to the left, yielding $(t(r))(s)$). Similarly $t(r_1, r_2, \ldots, r_n)$ denotes $t(r_1)(r_2)\ldots(r_n)$.[6]

At the risk of confusion, *sequences* of variables (possibly empty) will be indicated by the same font, e.g. $x = (x_1, \ldots, x_n)$ or $X = (X_1, \ldots, X_n)$. If x is such a sequence then the term $t(x)$ should be interpreted as $t(x_1, \ldots, x_n)$ and the prefix $\exists x$ should be interpreted as the quantifier string $\exists x_1\, \exists x_2\, \ldots \exists x_n$. In general, we will rely on context to determine whether we are dealing with a single variable or sequence of such. If x and y denote sequences of variables, then x, y denotes their concatenation, as in $t(x, y)$ or $\exists x, y$.

In the intended interpretation, of course, 0 and Sc satisfy

$$x' \neq 0$$

and

$$x' = y' \to x = y.$$

The constants K and S in (5) and (6) above are the usual *typed combinators* whose interpretation is given by

$$\mathrm{K}(s, t) = s \quad \text{for } s : \sigma \text{ and } t : \tau,$$

and

$$\mathrm{S}(r, s, t) = r(t)(s(t)) \quad \text{for } r : (\rho \to \sigma \to \tau),\ s : \rho \to \sigma, \text{ and } t : \rho.$$

The meaning of equality at higher types is a delicate matter which will be taken up in section 2.5 below. For the time being we read these equations naively.

If terms are generated from the variables and constants solely by the operation of application then one obtains *combinatory completeness* as usual, i.e. we can associate

[6] In the literature on functional interpretations, parentheses are often dropped altogether for the sake of brevity, so that one may encounter e.g. $Xxyz$ instead of $X(x, y, z)$. When this is the case, convention and the types of the associated terms and variables dictate the appropriate reading.

with each term t and variable x another term $\lambda x.t$ whose free variables are all those of t other than x, and which satisfies the equation

$$(\lambda x.t)(s) = t[s/x].$$

It follows that if t has free variables among $x_1, \ldots, x_n$, then $(\lambda x_1 \ldots \lambda x_n.t)$ is a closed term with

$$(\lambda x_1 \ldots \lambda x_n.t)(s_1, \ldots, s_n) = t[s_1/x_1, \ldots, s_n/x_n]$$

for all s_i of the same type as x_i for $i = 1, \ldots, n$. Alternatively, we could have taken the λ operation as a basic term-forming operation, where the terms thereby formed have the defining equation above.

Finally, we have the equations for the recursors R, which define a simple form of primitive recursion:

$$\begin{aligned} &\mathrm{R}(f, g, 0) = f \\ &\mathrm{R}(f, g, n') = g(n, \mathrm{R}(f, g, n)). \end{aligned}$$

That is, $R(f, g)$ is a function h of type $0 \to \sigma$, with defining equations

$$\begin{aligned} &h(0) = f \\ &h(n') = g(n, h(n)). \end{aligned}$$

There is no need to mention parameters to h explicitly, since these can be absorbed in the types of f and g. We note that this kind of higher-type iteration is clearly anticipated in Hilbert [1926], and even, to some extent, in Weyl [1918].

The atomic formulas of T consist of assertions of equality between terms of the same type,[7] and more complex formulas are obtained by combining these with the usual propositional connectives. The axioms of T consist of the defining equations of 0, Sc, K, S, and R described above, a rule allowing for the substitution of arbitrary terms for variables of the same type, equality axioms, the axioms of *classical* propositional logic, and the scheme of induction

$$\text{from } \varphi(0) \text{ and } \varphi(x) \to \varphi(x') \text{ conclude } \varphi(t)$$

for arbitrary formulas φ and terms t in the language. Note that if one has included products $\sigma \times \tau$ among the finite types, one also needs to add basic terms denoting pairing and projection operations of the appropriate types, together with their defining equations as well.

In later sections we will consider variants of T which allow for more elaborate types (e.g. "transfinite types") and functionals. For the time being, T is strong enough to interpret HA, as we now show.

[7]In one treatment of equality in T, these are only for terms of type 0; cf. section 2.5 below.

2.3. The Dialectica interpretation

To each formula φ in the language of arithmetic we associate its *Dialectica* (or D-) *interpretation* φ^D, which is a formula of the form

$$\varphi^D = \exists x \, \forall y \; \varphi_D$$

where φ_D is a (quantifier-free) formula in the language of T. Here the free variables of φ_D consist of those free in φ, together with the sequences of variables (possibly empty) x and y. However, the free variables of φ are generally suppressed and we also write $\varphi_D(x, y)$ for φ_D. If one or more free variables z of φ are exhibited, as $\varphi(z)$, then we write $\varphi_D(x, y, z)$ for φ_D.

The associations $(\quad)^D$ and $(\quad)_D$ are defined inductively as follows, where

$$\varphi^D = \exists x \, \forall y \; \varphi_D \quad \text{and} \quad \psi^D = \exists u \, \forall v \; \psi_D.$$

1. For φ an atomic formula, x and y are both empty and $\varphi^D = \varphi_D = \varphi$.
2. $(\varphi \wedge \psi)^D = \exists x, u \, \forall y, v \; (\varphi_D \wedge \psi_D)$.
3. $(\varphi \vee \psi)^D = \exists z, x, u \, \forall y, v \; ((z = 0 \wedge \varphi_D) \vee (z = 1 \wedge \psi_D))$.
4. $(\forall z \; \varphi(z))^D = \exists X \, \forall z, y \; \varphi_D(X(z), y, z)$.
5. $(\exists z \; \varphi(z))^D = \exists z, x \, \forall y \; \varphi_D(x, y, z)$.
6. $(\varphi \to \psi)^D = \exists U, Y \, \forall x, v \; (\varphi_D(x, Y(x, v)) \to \psi_D(U(x), v))$.

The case of $\to$ has been put last here because this requires special explanation below. Since we have defined $\neg\varphi$ to be $\varphi \to \perp$, from 6 we obtain

7. $(\neg\varphi)^D = \exists Y \, \forall x \; \neg\varphi_D(x, Y(x))$.

In clause 1 we assume the obvious identification of the symbols $+$ and $\times$ of *HA* with the terms that represent addition and multiplication in T. The definition of φ^D for atomic φ, of $(\varphi \wedge \psi)^D$ and of $(\exists z \; \varphi(z))^D$ needs no comment. The definition of $(\varphi \vee \psi)^D$ is also clear on a constructive reading: the new parameter z tells which disjunct is being established, according as to whether $z = 0$ or $z = 1$. The definition of $(\forall z \; \varphi(z))^D$ is obtained by prefixing a universal quantifier to φ^D to obtain

$$\forall z \, \exists x \, \forall y \; \varphi_D(x, y, z)$$

and then "skolemizing" the existentially quantified variable.

The motivation behind the definition of $(\varphi \to \psi)^D$ given by Gödel is as follows: from a witness x to the hypothesis φ^D one should be able to obtain a witness u to the conclusion ψ^D, such that from a counterexample v to the conclusion one should be able to find a counterexample y to the hypothesis. In short, one uses equivalences (i–iv) below to bring quantifiers to the front, and then skolemizes the existential variables:

$$\begin{array}{lll}
(\exists x \, \forall y \; \varphi_D(x, y) \to \exists u \, \forall v \; \psi_D(u, v)) & \leftrightarrow & \text{(i)} \\
\forall x \; (\forall y \; \varphi_D(x, y) \to \exists u \, \forall v \; \psi_D(u, v)) & \leftrightarrow & \text{(ii)} \\
\forall x \, \exists u \; (\forall y \; \varphi_D(x, y) \to \forall v \; \psi_D(u, v)) & \leftrightarrow & \text{(iii)} \\
\forall x \, \exists u \, \forall v \; (\forall y \; \varphi_D(x, y) \to \psi_D(u, v)) & \leftrightarrow & \text{(iv)} \\
\forall x \, \exists u \, \forall v \, \exists y \; (\varphi_D(x, y) \to \psi_D(u, v)) & \leftrightarrow & \text{(v)} \\
\forall x \, \exists u, Y_1 \, \forall v \; (\varphi_D(x, Y_1(v)) \to \psi_D(u, v)) & \leftrightarrow & \text{(vi)} \\
\exists U, Y \, \forall x, v \; (\varphi_D(x, Y(x, v)) \to \psi_D(U(x), v)). & &
\end{array}$$

The definitions of $(\varphi \wedge \psi)^D$, $(\varphi \vee \psi)^D$, and $(\exists z\ \varphi(z))^D$ are all justified from a constructive as well as classical point of view. The definition of $(\forall z\ \varphi(z))^D$ is justified by an application of the axiom of choice,

$$(AC) \qquad \forall x\ \exists y\ \varphi(x,y) \rightarrow \exists Y\ \forall x\ \varphi(x, Y(x))$$

for arbitrary formulas φ. (AC) is usually accepted classically in set theory; it is also accepted by many constructivists, since the reading of the hypothesis is that one has a constructive proof of $\forall x\ \exists y\ \varphi(x,y)$, and such a proof must provide a means Y of constructively associating with each x a solution $y = Y(x)$ of $\varphi(x,y)$.

The analysis of $(\varphi \rightarrow \psi)^D$ is more delicate. While equivalences (i–vi) above are all classically justified, only (i), (iii), (v) and (vi) are acceptable from a constructive point of view, the first two by logic and the latter two by (AC). Equivalence (ii), namely,

$$(IP') \qquad (\forall y\ \varphi \rightarrow \exists u\ \forall v\ \psi) \rightarrow \exists u\ (\forall y\ \varphi \rightarrow \forall v\ \psi)$$

is a special case of a principle called *independence of premise.* Though this is valid in classical logic, it is *not* generally accepted constructively, since the constructive reading of the hypothesis $(\theta \rightarrow \exists u\ \eta)$ is that we have a constructive means of turning any proof of the premise θ into a proof of η with a witness for the existential quantifier applied to η. In general, the choice of such a u will then depend on the proof of θ, while (IP') tells us that u can be chosen independently of any proof of that premise.

Equivalence (iv) can be justified by a generalization of Markov's principle, namely

$$(MP') \qquad \neg\forall y\ \theta \rightarrow \exists y\ \neg\theta$$

in which θ is assumed to be quantifier-free. (Assuming that the law of excluded middle holds for ψ_D, argue thus: if ψ_D is true then (iv) is justified, and if ψ_D is false apply (MP').) The problem with (MP') is that there is no evident way to choose constructively a witness y to $\neg\theta$ from a proof that $\forall y\ \theta$ leads to a contradiction. However if y ranges over the natural numbers one can search for such a y given that one accepts its existence. In this case (MP') boils down to the usual form of Markov's principle

$$(MP) \qquad \forall x\ (\neg\neg\exists y\ \varphi(x,y) \rightarrow \exists y\ \varphi(x,y))$$

which is accepted in the Russian school of constructivity for φ quantifier-free.

While the reasoning leading to the form of the D-interpretation is not fully constructive it can still be used as a tool in constructive metamathematics and to derive constructive information. In section 3.1 we'll see that the D-interpretation verifies the three principles (AC), (IP'), and (MP') just discussed, and hence allows one to use the D-interpretation to extract constructive information from non-constructive proofs.

2.4. Verifying the axioms of arithmetic

Gödel's main result is as follows.

2.4.1. Theorem. *Suppose φ is a formula in the language of arithmetic, and HA proves φ. Then there is a sequence of terms t such that T proves $\varphi_D(t, y)$.*

We express this by saying that *HA* is *D-interpreted* in T. Combining Theorem 2.4.1 with Corollary 2.1.2 we obtain

2.4.2. Corollary. *Suppose φ is a formula in the language of arithmetic, such that PA proves φ. Then there is a sequence of terms t such that T proves $(\varphi^N)_D(t, y)$.*

In short, *PA* is *ND-interpreted* in T.

One proves Theorem 2.4.1 by induction on the length of the proof in *HA*. One only has to verify that the claim holds true when φ is an axiom of *HA*, and that it is maintained under rules of inference.

We begin by considering the axioms of rules of intuitionistic logic, listed in section 2.1. For most of these the verification is routine, and we only address a few key examples. For example, consider the rule "from $\varphi \to \psi$ and φ conclude ψ." Given a term a such that T proves $\varphi_D(a, y)$ and terms b and c such that T proves $\varphi_D(x, b(x, v)) \to \psi_D(c(x), v)$, we want a term d such that T proves $\psi_D(d, v)$. By substituting $b(a, v)$ for y in the first hypothesis and a for x in the second, we see that taking $d = c(a)$ works; so, in a sense, modus ponens corresponds to functional application. The reader can verify that, similarly, the axiom "from $\varphi \to \psi$ and $\psi \to \theta$ conclude $\varphi \to \theta$" corresponds to the composition of functions.

Handling the axiom $\varphi \to \varphi \wedge \varphi$ requires a bit more work. If the interpretation of the hypothesis is given by $\exists x\, \forall y\, \varphi_D(x, y)$, the interpretation of the conclusion is

$$\exists x_1, x_2\, \forall y_1, y_2\, (\varphi_D(x_1, y_1) \wedge \varphi_D(x_2, y_2)).$$

According to the clause for implication, we need to provide terms $c_1(x)$ and $c_2(x)$ taking a witness x for the hypothesis to witnesses x_1 and x_2 for the conclusion; for this purpose, we can simply take $c_1(x)$ and $c_2(x)$ to be x. But we also need a functional $d(x, y_1, y_2)$ that will take y_1 and y_2 witnessing the failure of $\varphi_D(x, y_1) \wedge \varphi_D(x, y_2)$ to a value d representing the failure of $\varphi_D(x, d)$. This functional must effectively determine which of $\varphi_D(x, y_1)$ and $\varphi_D(x, y_2)$ is false. We need the following

2.4.3. Lemma. *If φ is a formula of arithmetic, there is a term t_φ such that T proves*

$$t_\varphi(x, y) = 0 \leftrightarrow \varphi_D(x, y).$$

The lemma is proved by induction on the size of φ_D; the key instance occurs when φ_D is simply an atomic formula $s_1 = s_2$, for which case the required t can be obtained from an application of primitive recursion.

Another instance of primitive recursion yields a functional Cond such that T proves

$$\mathrm{Cond}(w, u, v) = \begin{cases} u & \text{if } w = 0 \\ v & \text{otherwise.} \end{cases}$$

Taking $d = \text{Cond}(t_\varphi(x, y_1), y_1, y_2)$ above then suffices to complete the proof for $\varphi \to \varphi \wedge \varphi$.

Aside from the two instances just described, the recursors R are not otherwise used to verify the logical axioms. Their primary purpose is to interpret the induction rule, "from $\varphi(0)$ and $\varphi(u) \to \varphi(u')$ conclude $\varphi(u)$." Inductively we are given terms a, b, and c so that T proves $\varphi_D(0, a, y)$ and

$$\varphi_D(u, x, b(u, x, y_1)) \to \varphi_D(u', c(u, x), y_1).$$

We want a term d such that $\varphi_D(u, d(u), y)$. Using the recursors we can define d using primitive recursion, so that

$$\begin{aligned} d(0) &= a \\ d(u') &= c(u, d(u)). \end{aligned}$$

This yields

$$\varphi_D(0, d(0), y)$$

and

$$\varphi_D(u, d(u), b(u, d(u), y)) \to \varphi_D(u', d(u'), y).$$

The following lemma will then allow us to conclude $\varphi_D(u, d(u), y)$, as desired.

2.4.4. Lemma. *From $\psi(0, y)$ and $\psi(u, b(u, y)) \to \psi(u', y)$ one can prove, in T, $\psi(u, y)$.*

The idea is to work backwards: if "$\dot{-}$" denotes truncated (or "cut off") subtraction, note that $\psi(u, y)$ follows from

$$\psi(u \dot{-} 1, b(u \dot{-} 1, y)),$$

which in turn follows from

$$\psi(u \dot{-} 2, b(u \dot{-} 2, b(u \dot{-} 1, y))),$$

and so on, until the first argument is equal to 0. More formally, one defines in T a function $e(y, z)$ by $e(y, 0) = y$ and $e(y, z') = b(u \dot{-} z', e(y, z))$, and then uses induction to prove that $\psi(w, e(y, u \dot{-} w))$ holds for every w less than or equal to u. For details, see Spector [1962] or Troelstra [1973].

This leaves only the quantifier-free axioms regarding 0, Sc, $+$, and $\times$ in HA, which follow immediately from their counterparts in T.

Once more we emphasize that the recursors of T are only essentially needed for the nonlogical axioms. Roughly speaking, it is the combinatory completeness of T that allows us to verify the axioms of intuitionistic predicate logic, whereas the recursors are the functional analogue of induction. This observation allows one to generalize the D-interpretation to other theories, as discussed in section 3.3 below.

2.5. Equality at higher types

We return to the question as to how equality at higher types is to be treated in T. There are two basic choices, the *intensional* formulation taken in Gödel [1958] and the *(weakly) extensional* formulation of Spector [1962]. In Gödel's formulation, we have a decidable equality relation $=_\sigma$ at each type, i.e. all formulas $s =_\sigma t$ with s, t of type σ are taken to be atomic, and the law of excluded middle is accepted for these. Suppressing type subscripts where there is no ambiguity, the equality axioms for T in this version are, as usual,

1. $s = s$
2. $s = t \wedge \varphi[s/x] \rightarrow \varphi[t/x]$.

The axioms for K, S and R at each type may be read as they stand.

In Spector's formulation, the atomic formulas are equations between terms of type 0 only, and the law of excluded middle is accepted only for these. For $\sigma = (\sigma_1 \rightarrow \ldots \rightarrow \sigma_n \rightarrow 0)$, an equation $s =_\sigma t$ between terms of type σ is regarded as an abbreviation for

$$s(x_1, \ldots, x_n) = t(x_1, \ldots, x_n)$$

where the x_i are fresh variables of type σ_i for $i = 1, \ldots, n$. In particular, the axioms for K, S and R are to be read as such abbreviations. Now in this version, the axioms 1, 2 are taken only for terms s, t of type 0; denote these by 1_0 and 2_0 respectively. For s, t of type $\sigma \neq 0$ we must further adjoin a rule

2′. From $s(x_1, \ldots, x_n) = t(x_1, \ldots, x_n)$ and $\varphi[s/x]$ infer $\varphi[t/x]$.

An alternative suggested by Gödel in a footnote to his revision [1972] of [1958] and made explicit by Troelstra [1990] is to follow Spector in taking only equations at type 0 as basic but to assume as axioms, besides 1_0 and 2_0, just the following consequences of 2′ for K, S, and R:

$$s[\mathrm{K}(u, v)/x] = s[u/x], \quad s[\mathrm{S}(u, v, w)/x] = s[u(v, u(w))/x],$$

$$s[\mathrm{R}(u, v, 0)/x] = s[u/x], \quad s[\mathrm{R}(u, v, w')/x] = s[v(w, \mathrm{R}(u, v, w))/x],$$

where $s[x]$ is a term of type 0 and u, v, w are terms of appropriate type. Note that the resulting theory is contained in both Gödel's and Spector's versions.

Most of the results for the functional interpretation are insensitive to which of these three formulations of T (and of other systems like T to be considered below) is taken. When it is necessary to make a distinction, as for example in the next section when considering extensions of T by adjunction of quantifiers, and in section 4.1 when considering models of T, we shall use *WE-T* to denote Spector's version and T_0 to denote the version, just described, elicited by Troelstra, while reserving T itself for Gödel's version. However, one further system must be considered along with the latter. It is implicit in the idea of intensional equality at higher types that we have an effective procedure to decide whether any two objects of the same type are identical. This is made explicit by adjoining a constant E_σ for the characteristic function of $=_\sigma$ at each type σ, with axiom:

$$\mathrm{E}_\sigma(x, y) = 0 \leftrightarrow x =_\sigma y.$$

The system T with these additional constants and axioms is then denoted $I\text{-}T$.

3. Consequences and benefits of the interpretation

3.1. Higher type arithmetic

Before turning to some of the immediate consequences of the Dialectica interpretation, we pause to consider some easy generalizations. First of all, note that the D-translation applies equally well if the source language is also typed. In other words, if we define HA^ω to be a version of T with quantifiers ranging over each finite type, with the axioms and rules of intuitionistic predicate logic and induction for all formulas in the new language, then we can apply the D-translation to this theory as well.

A problem, however, arises in verifying the interpretation of the axiom $\varphi \to \varphi \wedge \varphi$, and this depends on how equality at higher types is treated. For if φ is an equation $s =_\sigma t$, in order to extend the argument for Lemma 2.4.3 we shall need to have a characteristic function E_σ for $=_\sigma$. Otherwise we shall have to limit ourselves to a formulation of HA^ω using only equations of type 0. Thus we are led to consider three versions of HA^ω corresponding to the three versions $I\text{-}T$, $WE\text{-}T$ and T_0 in section 2.5, denoted respectively $I\text{-}HA^\omega$, $WE\text{-}HA^\omega$, and HA^ω_0. To be more precise, in $WE\text{-}HA^\omega$, given the availability of quantification at higher types, we may take equality there to be introduced *by definition* in terms of equality at lower types so as to satisfy the equivalences

$$x =_{\sigma\to\tau} y \leftrightarrow \forall z^\sigma \ (x(z) =_\tau y(z))$$

for each σ, τ. What is a new option now is that a *fully extensional* version $E\text{-}HA^\omega$ can be formulated as follows. In this theory one takes the symbol $=_\sigma$ to be basic at each type, and adopts the full axioms 1, 2 of section 2.5 as in the intensional version. What makes the difference is that the preceding equivalences are now taken as axioms in the fully extensional theory.

Now, with little or no modification, the proof sketched in section 2.4 carries over to show that for H any one of the three systems $I\text{-}HA^\omega$, $WE\text{-}HA^\omega$, or HA^ω_0, the system H is D-interpreted in the corresponding quantifier-free subsystem, i.e. $I\text{-}T$, $WE\text{-}T$, or T_0, resp. This does *not* hold for the system $E\text{-}HA^\omega$, as shown by Howard [1973]. In particular, no functional of that system satisfies the D-interpretation of $\forall u, x, y \ (x =_1 y \to u(x) =_0 u(y))$ where x, y are of type 1 (i.e. $0 \to 0$) and u is of type 2 (i.e. $1 \to 0$). However, $E\text{-}HA^\omega$ may be formally interpreted in HA^ω_0 preserving all formulas all of whose variables are of type 0 or 1, by relativizing the quantifiers to the hereditarily extensional objects in each type; we may then apply the D-interpretation to HA^ω_0 as above. This is the route taken by Luckhardt [1973]; cf. also Feferman [1977,section 4.4.2] for a brief outline of the details involved. Clearly $HA^\omega_0 \subset I\text{-}HA^\omega$ and $HA^\omega_0 \subset WE\text{-}HA^\omega \subset E\text{-}HA^\omega$.

In the following we shall take it that HA^ω is any one of the three systems $I\text{-}HA^\omega$, $WE\text{-}HA^\omega$, and HA^ω_0 for which the D-interpretation works directly as described above, and T is now taken for the corresponding quantifier-free subsystem; however, the reader is free to settle on any one of these three pairs as the preferred one according to taste.[8]

Let us return to the principles (AC), (IP'), and (MP') which figured in the motivation for the definition of $(\varphi \to \psi)^D$ in section 2.3. These make use of the quantified variables of arbitrary type and are thus formulated in the language of HA^ω. Given the discussion in section 2.3 it shouldn't be surprising that, in fact, we have

3.1.1. Theorem. *Over intuitionistic logic, the schemata $(AC) + (IP') + (MP')$ and $\varphi \leftrightarrow \varphi^D$ are equivalent.*

Henceforth we'll use $HA^\#$ to denote the theory HA^ω together with either of these two schemata. The previous discussion tells us that

3.1.2. Theorem. *$HA^\#$ is D-interpreted in T.*

What, now, can be said about classical theories of higher types? Take PA^ω to be the classical extension of HA^ω, obtained simply by adjoining the law of excluded middle for all formulas. It is N-interpreted in HA^ω just as PA is N-interpreted in HA. By the preceding theorem, it is natural to consider the system $PA^\#$ which is defined to consist of PA^ω together with all instances of $\varphi \leftrightarrow \varphi^{ND}$ for φ in the language of PA^ω. It then follows immediately that

3.1.3. Theorem. *$PA^\#$ is ND-interpreted in T.*

Consider now the principles (AC), (IP'), and (MP') on the classical side. Of these, (IP') and (MP') follow from classical logic, but (AC) is not derivable from $PA^\#$. Moreover, (AC) is not in general preserved under the ND-interpretation. For, the N-interpretation of $\forall x\, \exists y\, \varphi(x, y) \to \exists Y\, \forall x\, \varphi(x, Y(x))$ is equivalent to

$$\forall x\, \neg\forall y\, \neg\varphi^N(x, y) \to \neg\forall Y\, \neg\forall x\, \varphi^N(x, Y(x))$$

or, equivalently,

$$\forall x\, \neg\neg\exists y\, \varphi^N(x, y) \to \neg\neg\exists Y\, \forall x\, \varphi^N(x, Y(x)),$$

which cannot in general be proved in $HA^\#$. It *is*, however, provable for the special case $(QF\text{-}AC)$ of the axiom of choice with quantifier-free matrix φ. For, in that case,

$$\forall x\, \neg\neg\exists y\, \varphi^N(x, y) \leftrightarrow \forall x\, \exists y\, \varphi^N(x, y)$$

[8] But note that Troelstra [1990,p. 351] says that "*WE-HA*$^\omega$ as an intermediate possibility [between $I\text{-}HA^\omega$ and HA_0^ω] is not very attractive: the deduction theorem does not hold for this theory." Yet another alternative for dealing with the problem of verifying the axioms $\varphi \to \varphi \wedge \varphi$ without restriction on atomic formulas, but without additional E_σ functionals, makes use of a variant of the D-interpretation due to Diller and Nahm [1974], described in Troelstra [1973,pp. 243–245]. We shall not go into that variant in this chapter.

is provable in $HA^{\#}$ using (MP'). Then from (AC) in that system we infer $\exists Y\ \forall x\ \varphi^N(x, Y(x))$, which implies intuitionistically its double negation. The conclusion is that $(QF\text{-}AC)$ is ND-interpreted in T. The following observation by Kreisel [1959,p. 120] then gives the proper analogue to Theorem 3.1.1:

3.1.4. Theorem. *Over classical logic, the schemata $(QF\text{-}AC)$ and $\varphi \leftrightarrow \varphi^{ND}$ are equivalent.*

3.1.5. Proof. In the forward direction, one need only consider negative formulas φ, i.e. those which do not contain disjunction or existence symbols. For such formulas it is easily proved by induction on φ that φ^{ND} has the form $\exists X\ \forall y\ \psi(X(y), y)$ where ψ is quantifier-free and where $\varphi \leftrightarrow \exists X\ \forall y\ \psi(X(y), y) \leftrightarrow \forall y\ \exists x\ \psi(x, y)$. The reverse direction is immediate. □

Thus $PA^{\#}$ can equally well be thought of as $PA^{\omega} + (QF\text{-}AC)$.

When analyzing classical or intuitionistic theories of first- or second-order arithmetic, it often turns to be useful to embed them in fragments or extensions of $PA^{\#}$ and $HA^{\#}$ respectively; cf. the discussion in section 3.3 below.

3.2. Some consequences of the D-interpretation

Since the D-interpretations of HA and $HA^{\#}$, as well as the ND-interpretations of PA and $PA^{\#}$, are purely syntactic, they can be formalized in a weak theory of arithmetic. This yields the following theorem, which is of foundational importance.

3.2.1. Theorem. *Let S be any of the theories HA, $HA^{\#}$, PA, or $PA^{\#}$. Then a weak base theory proves*

$$\mathrm{Con}(T) \to \mathrm{Con}(S).$$

Of course, the interpretations yield far more information than just the relative consistency of the theories involved. Recall that a formula $\theta(x_1, x_2, \ldots, x_n)$ in the language of arithmetic is Δ^0_0 if all its quantifiers are bounded, in which case it defines a primitive recursive relation on the natural numbers. The characteristic function of this relation can be represented by a term t in the language of T, such that $PA^{\#}$ or $HA^{\#}$ proves $\theta(\vec{x}) \leftrightarrow t(\vec{x}) = 0$. Though it is somewhat an abuse of notation, when we say below that "T proves $\theta(\vec{x})$" for such a θ, we mean that it proves $t(\vec{x}) = 0$.

3.2.2. Theorem. *Let S be any of the theories above, and suppose S proves the Π^0_2 formula*

$$\forall x\ \exists y\ \theta(x, y),$$

where θ is Δ^0_0. Then there is a term f such that T proves

$$\theta(x, f(x)).$$

3.2.3. Proof. By embedding HA and PA in $HA^{\#}$ and $PA^{\#}$ respectively and proving the equivalence just discussed, we can assume that θ is quantifier-free. In that case the D-interpretation of $\forall x\ \exists y\ \theta(x,y)$ is $\exists Y\ \forall x\ \theta(x, Y(x))$, and its ND-interpretation is $\exists Y\ \forall x\ \neg\neg\theta(x, Y(x))$. The conclusion then follows from Theorem 3.1.2 in the case of $HA^{\#}$, and Theorem 3.1.3 in the case of $PA^{\#}$. □

Now suppose h is a recursive function whose graph is defined by a Σ^0_1 formula $\varphi(x,y)$ in the standard model. We say that the theory S proves h to be total if it proves $\forall x\ \exists! y\ \varphi(x,y)$. In section 2.2 we axiomatized the primitive recursive functionals of finite type, without addressing the issue of what exactly the terms denote. Deferring this discussion to section 4 below, for now let us assume that at least the closed type 1 terms denote functions. As a corollary to Theorem 3.2.2 we have

3.2.4. Corollary. *Every provably total recursive function of HA, $HA^{\#}$, PA, or $PA^{\#}$ is denoted by a term of T.*

In fact, in section 4.1 we will see that there are models of T that can be formalized in the language of arithmetic, yielding an interpretation of T in HA. This yields the following result, which is interesting in that it makes no mention of T at all:

3.2.5. Corollary. *PA, and hence $HA + (MP)$, is conservative over HA for Π^0_2 sentences.*

When it comes to PA, Theorem 3.2.2 is sharp in the following sense. Consider the Π^0_3 sentence "for every x there exists a y, such that either y is a halting computation for the Turing machine with index x, or Turing machine x doesn't halt." Though this statement is provable in PA, any function returning such a y for every x cannot be recursive since it solves the halting problem. Later we'll see that, on the other hand, the functions represented by terms of T *are* recursive, so that the analogue of Theorem 3.2.2 does not hold for Π^0_3 formulas.

Nonetheless, one *can* extract a different kind of constructive information from PA-proofs of complex formulas. Suppose PA (or $PA^{\#}$) proves a formula φ given by

$$\forall x_1\ \exists y_1\ \forall x_2\ \exists y_2\ \ldots \forall x_n\ \exists y_n\ \theta(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n)$$

where θ is quantifier-free. The N-interpretation of this formula intuitionistically implies

$$\neg\exists x_1\ \forall y_1\ \exists x_2\ \forall y_2\ \ldots \exists x_n\ \forall y_n\ \neg\theta(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n),$$

whose D-interpretation is the same as that of

$$\begin{aligned}&\forall X_1, X_2, \ldots, X_n\ \exists y_1, y_2, \ldots, y_n\\ &\qquad \neg\neg\theta(X_1, X_2(y_1), \ldots, X_n(y_1, y_2, \ldots, y_{n-1}), y_1, y_2, \ldots, y_n).\end{aligned}$$

As a result we have

3.2.6. Corollary. *Suppose PA proves φ as above. Then there are terms*

$$\begin{aligned} t_1 &= F_1(X_1, X_2, \ldots, X_n) \\ &\vdots \\ t_n &= F_n(X_1, X_2, \ldots, X_n) \end{aligned}$$

such that T proves

$$\theta(X_1, X_2(t_1), \ldots, X_n(t_1, t_2, \ldots, t_{n-1}), t_1, t_2, \ldots, t_n).$$

The observation that the functionals $F_1, F_2, \ldots, F_n$ can be taken to be recursive in their arguments is known as Kreisel's "no-counterexample interpretation" (cf. Kreisel [1951,1959]). To make sense of the name, think of the functions $X_1, X_2, \ldots, X_n$ as trying to provide counterexamples to the truth of φ, by making $\theta(X_1, X_2(y_1), \ldots, X_n(y_1, \ldots, y_{n-1}), y_1, \ldots, y_n)$ false for any given values of $y_1, y_2, \ldots, y_n$; in which case $F_1, F_2, \ldots, F_n$ effectively provide witnesses that foil the purported counterexample.

3.3. Benefits of the D-interpretation

In general, the D-interpretation is a powerful tool when applied to the reduction of an intuitionistic theory I to a functional theory F. As we have seen, the very form of the D-interpretation automatically brings a number of benefits. Since little more than the combinatorial completeness of F is necessary to interpret the logical axioms of I, one only has to worry about interpreting the non-logical axioms of I and tailor the functionals of F accordingly. As an added bonus, I can often be embedded in a higher-type analogue I^ω to which we can add the schemata (AC), (IP'), (MP') at no extra cost. Taken together (AC), (IP'), and (MP') prove the scheme $\varphi \leftrightarrow \varphi^D$, a fact that is often useful in practice since it allows one to pull facts about the D-interpretation back to the theory being interpreted. This observation will be employed in sections 6 and 7 below.

If one is trying to analyze a classical theory C by reducing it to I via a double-negation interpretation, one has, of course, to ensure that I is strong enough to prove the doubly-negated axioms of C. Once again, though, the choice of a D-interpretation for I has some advantages: C can often also be embedded in a higher-type analogue C^ω in a natural way, and the fact that Markov's principle is verified in the interpretation guarantees that one ultimately obtains Skolem terms for provable Π^0_2 sentences. This in turn yields a characterization of C's provably total recursive functions.

These advantages of the D-interpretation are summed up in Feferman [1993]:

> Applied to intuitionistic systems it takes care of the underlying logic once and for all, verifies the Axiom of Choice AC in all types, and interprets various forms of induction by suitably related forms of recursion. This

> then leads for such systems to a perspicuous mathematical characterization of the provably recursive functions and functionals. For application to classical systems, one must first apply the negative translation (again taken care of once and for all). Since the D-interpretation verifies Markov's principle even at higher types ... at least the provably recursive functions and functionals are preserved, as well as QF-AC in all types and induction schemata. The main disadvantage, though, comes with the analysis of other statements whose negative translation may lead to a complicated D-interpretation; special tricks may have to be employed to handle these.

A further distinguishing feature of the D-interpretation is its nice behavior with respect to modus ponens. In contrast to cut-elimination, which entails a global (and computationally infeasible) transformation of proofs, the D-interpretation extracts constructive information through a purely local procedure: when proofs of φ and $\varphi \to \psi$ are combined to yield a proof of ψ, witnessing terms for the antecedents of this last inference are combined to yield a witnessing term for the conclusion. As a result of this modularity, the interpretation of a theorem can be readily obtained from the interpretations of the lemmata used in its proof.

The process of applying the D-interpretation to specific classical theorems can sometimes be used to obtain appropriate constructivizations thereof, or to uncover additional numerical information that is implicit in their classical proofs; cf., for example, Bishop [1970] or Kohlenbach [1993].

4. Models of T, type structures, and normalizability

In section 2.2 we presented the set of terms of the theory T without a discussion of what these terms denote. In this section we exhibit several kinds of functional and term models.

4.1. Functional models

The most obvious model of T considered in either the intensional or extensional sense is the *full (set-theoretic) hierarchy of functionals of finite type*, in which the objects of type 0 are the natural numbers, and each type $\sigma \to \tau$ represents the set of *all* functions from the objects of type σ to those of type τ. The denotations of 0, Sc, K, S, and R are then apparent, as well as the denotation of terms built up through the application of these constants. The equality relation in T is taken to denote true (extensional) equality in this model. By the primitive recursive functionals in the set-theoretic sense we mean those denoted by closed terms of T.

One can obtain a "smaller" type structure in which the elements of each type are indices for recursive functions, as follows. Let φ_e denote a standard enumeration of the recursive functions, say, using Kleene's universal predicate. Define $M_0 = \mathbb{N}$, and

$$M_{\sigma\to\tau} = \{e \mid \forall x \in M_\sigma \; \exists y \in M_\tau \; (\varphi_e(x) \downarrow= y)\}.$$

One can associate recursive indices to constants of T in a natural way, and interpret equality as equality of indices. The resulting model $\mathcal{M}$ of $I\text{-}T$ (and of $I\text{-}HA^\omega$) is known as the *hereditarily recursive operations* (though "hereditarily total recursive operations" might be more accurate) and is usually denoted by HRO. Equality is not extensional in this model, since many different indices can represent the same functional. If instead one wants a model of $WE\text{-}T$ (and of $WE\text{-}HA^\omega$ or even $E\text{-}HA^\omega$), one can consider the *hereditarily effective operations* $\mathcal{N}$, usually denoted by HEO. For this model, sets N_σ and the equality relation $=_\sigma$ are defined inductively as follows: set $N_0 = \mathbb{N}$ and $=_0$ the usual equality relation for natural numbers,

$$\begin{aligned} N_{\sigma\to\tau} = \{e \mid {} & \forall x \in N_\sigma \; \exists y \in N_\tau \; (\varphi_e(x) \downarrow = y) \wedge \\ & \forall x \in N_\sigma, y \in N_\sigma \; (x =_\sigma y \to \varphi_e(x) =_\tau \varphi_e(y))\}, \end{aligned}$$

and

$$e =_{\sigma\to\tau} f \equiv \forall z \in N_\sigma \; (\varphi_e(z) =_\tau \varphi_f(z)).$$

Both HEO and HRO can be formalized in HA in the sense that the sets M_σ (resp. N_σ) and the equality relations $=_\sigma$ are defined by formulas in the language of arithmetic, HA proves each axiom of T true in the interpretation, and the natural numbers in the model correspond to the natural numbers of HA. Notice, however, that the complexity of the formulas defining M_σ and N_σ grow with the level of σ, so that HA (or PA) cannot prove the consistency of T outright.

By generalizing the notion of a continuous function to higher types, Kleene and Kreisel independently obtained further models of T (cf. also Troelstra [1973]); these will be of significance in section 6.

All of the models described in this subsection contain more than just the primitive recursive functionals (or the indices for such), and hence model extensions of T as well.[9] In contrast, a "minimal" model of T, which only contains objects denoted by terms, is provided by the *term model*, which we now describe.

4.2. Normalization and the term model

The defining equations for the typed combinators K, S, and R in section 2.2 describe a symmetric equality relation. From a computational point of view, it is often more useful to think of these defining equations as describing a *directed* relation, in which terms on the left-hand sides of the equations are "reduced" to more basic ones on the right. For example, the defining equations of the theory T yield the following reduction rules:

1. $\mathrm{K}(s,t) \rhd s$
2. $\mathrm{S}(r,s,t) \rhd r(t)(s(t))$
3. $\mathrm{R}(s,t,0) \rhd s$

[9]Some, like the recursion theoretic models, have generalizations to transfinite types; see, e.g. Beeson [1982]. Category-theoretic methods have also been used to construct models of functional theories, though we will not discuss such models here.

4. $\mathrm{R}(s,t,x') \triangleright t(x, \mathrm{R}(s,t,x))$.

If one uses lambda terms instead of combinators, the first two clauses can be replaced by the reduction rule $(\lambda x.s)(t) \triangleright t[s/x]$. If one wants to include product types, corresponding reduction rules for the pairing and projection operations can be defined as well.

If s and t are terms, we say that s *reduces to* t *in one step*, written $s \rightarrow t$, if t can be obtained by replacing some subterm u of s by a v such that $u \triangleright v$. We say that s *reduces to* t if t can be obtained from s by a finite sequence of one-step reductions. In other words, the reducibility relation $\rightarrow^*$ is the reflexive-transitive closure of $\rightarrow$.

Such a reducibility relation is an example of a *rewrite system* (cf. Dershowitz and Jouannaud [1990]). The following terminology is standard.

4.2.1. Definition.

1. If s is a term and u is a subterm of s, then u is a *redex* of s if a reduction rule can be applied to u; i.e. there is some v such that $u \triangleright v$.
2. If s has a redex, then s is *reducible.* Otherwise, s is *irreducible*, or in *normal form.*
3. A term is *normalizable* if it can be reduced to one in normal form. A system of reduction rules is *normalizing* if every term is normalizible.
4. A term s is *strongly normalizable* if there are no infinite (one-step) reduction sequences beginning with s; that is, *every* such sequence eventually leads to a term in normal form. A system of reduction rules is *strongly normalizing* if every term is strongly normalizable.
5. A system of reduction rules is *confluent*, or has the *Church-Rosser property*, if whenever $s \rightarrow^* u$ and $s \rightarrow^* v$ then there is a term t such that $u \rightarrow^* t$ and $v \rightarrow^* t$.

If a system of rules is confluent and s is any term, then s has at most one normal form. Furthermore, if the system is also normalizing, then s has *exactly* one normal form. Identifying closed terms with their normal forms then provides a *term model* for the defining equations corresponding to the reduction rules. Under this very simple semantics, one can think of each closed term representing nothing more than the "program" it computes, when applied to other closed terms in normal form. Strong normalizability implies that the "programming language" is insensitive to its implementation, in the sense that every reduction sequence terminates regardless of the order in which the reductions are performed.

In all the functional theories we consider in this chapter (together with the associated reduction relations), the only closed irreducible terms of type 0 are in fact numerals. In that case, we can identify type 0 objects of the corresponding term model with natural numbers. Showing that the reduction relation associated with a functional theory is confluent and normalizing then has two benefits:

- it implies that the functional theory is consistent, that is, it cannot prove $0 = 1$; and

- it justifies the intuition that the functional theory describes "computable" entities.

4.3. Strong normalization for T

Given the reduction rules corresponding to the theory T described above, one can apply brute but judicious combinatorial force to verify the following

4.3.1. Lemma. *The reduction relation $\to^*$ associated with T is confluent. Furthermore, any closed irreducible term of type 0 is a numeral.*

The "shortest known proof" of this, due to Tait and Martin-Löf, can be found in Hindley and Seldin [1986,appendix 1].

Assuming we can prove that the relation $\to^*$ is also normalizing, the resulting term model will satisfy the axioms of T: the uniqueness of normal forms implies that terms on either side of an equality axiom have the same interpretation under any instantiation of the variables; and the fact that the objects of type 0 in the term model are numerals reduces induction in T to induction in the metatheory.

Proving that $\to^*$ is normalizing, however, is bound to be tricky. Because it implies the consistency of Peano arithmetic, the proof must somehow go beyond the capabilities of that theory. W. Tait developed an elegant and flexible technique for proving normalization, using appropriate "convertibility" predicates (cf. Tait [1967] and Tait [1971]).

4.3.2. Definition. For each type σ, we define the set of *reducible terms of type* σ, denoted by Red_σ:

1. If t is a term of type 0, then t is in Red_0 if and only if t is normalizing.
2. If t is a term of type $\sigma \to \tau$, then t is in $\mathrm{Red}_{\sigma\to\tau}$ if and only if whenever s in Red_σ, $t(s)$ is in Red_τ.

Writing the second clause symbolically, we have that t is in $\mathrm{Red}_{\sigma\to\tau}$ if and only if

$$\forall s\ (s \in \mathrm{Red}_\sigma \to t(s) \in \mathrm{Red}_\tau).$$

Notice that the quantifier complexity of the first-order formula expressing "$t \in \mathrm{Red}_\rho$" grows with the complexity of ρ.

The normalization proof proceeds in two steps:

1. One shows, by induction on terms, that every t of type σ is in Red_σ.
2. One shows, by induction on the type σ, that every t in Red_σ is normalizing.

The only aspect of the proof that cannot be carried out in a weak base theory is the verification of clause 2, when t is the recursor R_σ: at this point the argument requires induction on a formula involving the predicate Red_σ. As a result we have

4.3.3. Theorem. *T is normalizing. Moreover, for each term t of T, PA proves that t is normalizable.*

This does not mean that Peano arithmetic proves that "for every term t, t is normalizable," since for various t the corresponding PA-proof can grow increasingly complex. The latter result, however, follows from the soundness of PA.

Another approach to proving normalization involves assigning ordinals (or notations representing ordinals) to terms in such a way that each one-step reduction leads to a decrease in the associated ordinal. For terms of T, this task is carried out by Howard [1970] using notations below the ordinal ε_0. Via a formal treatment of the term model, this yields, as a by-product, an ordinal analysis: over a weak base theory the assertion that "there are no infinite descending sequences of ordinal notations beneath ε_0" implies the consistency of PA.

4.4. Infinitely long terms

Another term model for T which is of special interest was provided by Tait [1965]. This uses infinitely long terms to replace the recursors, and thus produces a system of terms which is closer in character to the ordinary typed λ-calculus. The closure conditions on terms are as follows:

1. There are infinitely many variables $x^\tau, y^\tau, z^\tau, \ldots$ of each type τ.
2. 0 is a constant of type 0.
3. Sc is a constant of type $(0 \to 0)$.
4. If s is a term of type σ and t is a term of type $(\sigma \to \tau)$ then $t(s)$ is a term of type τ.
5. If t is a term of type τ then $\lambda x^\sigma.t$ is a term of type $(\sigma \to \tau)$.
6. If t_n $(n = 0, 1, 2, \ldots)$ is a sequence of terms of type τ then $\langle t_n \rangle$ is a term of type $(0 \to \tau)$.

Write $\mathbf{n}$ for $Sc^n(0)$. Then we translate each term t of T into a term t^+ of this system of infinite terms by taking $t^+ = t$ for t a variable, 0 or Sc,

$$\mathrm{K}^+ = \lambda x \lambda y.x, \;\; \mathrm{S}^+ = \lambda x \lambda y \lambda z.x(z, y(z))$$

and

$$\mathrm{R}^+ = \lambda f \lambda g \lambda x.\langle t_n \rangle \;\text{ where } t_0 = f \text{ and } t_{n+1} = g(\mathbf{n}, t_n),$$

and by requiring $(\cdot)^+$ to preserve application.

Each term t of the infinite system is assigned an ordinal $|t|$ as length in a natural way, with $|t| = 1$ for t a variable or constant, $|\lambda x.t| = |t| + 1$, $|t(s)| = |s| + |t|$ and, finally, $|\langle t_n \rangle| = \sup_{n<\omega}(|t_n| + 1)$. Note that for each of the constants C of T, $|C^+| \le \omega$, so for each term t of T, $|t^+| < \omega \cdot 2$.

We have three immediate reduction rules for this system of infinite terms:

1. $(\lambda x.t[x])(s) \rhd t[s/x]$
2. $\langle t_n \rangle(\mathbf{m}) \rhd t_m$

3. $(\langle t_n \rangle(r))(s) \triangleright \langle t_n(s) \rangle(r)$, when r is not a numeral and $\langle t_n \rangle(r)$ is not of type 0.

The relation $\rightarrow^*$ is then the least reflexive and transitive relation which extends the $\triangleright$ relation and preserves application. As before, a term t is said to be in normal form if whenever $t \rightarrow^* u$ then t is identical with u.

4.4.1. Theorem. *For each term t of T we can find a term t° in normal form such that $t^+ \rightarrow^* t^\circ$ and $|t^\circ| < \varepsilon_0$.*

The idea of Tait's proof of Theorem 4.4.1 is very much the same as that for the cut-elimination theorem for the extension of Gentzen's classical propositional sequent calculus to that for logic with countably long conjunctions Π and disjunctions Σ. Derivations in PA are translated into derivations in this calculus, by first translating formulas φ into propositional formulas φ^+, using $(\forall x\ \varphi[x])^+ = \Pi_{n<\omega}\varphi^+[\mathbf{n}/x]$. Then each derivation d from PA is translated into an infinite propositional derivation d^+ with finite cut-rank and $|d^+| < \omega \cdot 2$. Each derivation whose cut-rank is $\leq m+1$ and ordinal length is $\leq \alpha$ is effectively reduced to a derivation of the same end-formula whose cut-rank is $\leq m$ and ordinal length is $\leq \omega^\alpha$. So for each derivation d of T we eventually reduce d^+ to a derivation d° of length $< \varepsilon_0$. Similarly, we can assign a "cut-rank" to reducible infinite terms, and lower cut-complexity at the same exponential cost of increasing ordinal bounds. See Tait [1968], Schwichtenberg [1977], or Chapters III IV in this volume for more details concerning cut-elimination for sequent calculi for infinitary languages, and Tait [1965] or Feferman [1977] for details concerning normalization for infinitary term calculi.

More information can be extracted from these procedures as follows. Schwichtenberg [1977,section 4.2.2] shows in detail how the infinitary derivations generated from those in PA, as described above, may be coded by indices for primitive recursive functions. For each derivation d' in the sequence of reductions from d^+ to d°, the code of d' both determines the structure of d' as a tree and contains a bound on its cut-rank ($< \omega$) and on its length ($< \varepsilon_0$, in a primitive recursive notation system for ε_0). Exactly the same kind of thing can be done for each infinite term t' in the reduction sequence from t^+ to t°. This allows one to define an effective valuation function on t° when the initial t is a closed term of type 1 (i.e. $0 \rightarrow 0$), and that leads one to

4.4.2. Theorem. *The functions of type 1 generated by the primitive recursive functionals may be defined by schemata of effective transfinite recursion on ordinals $< \varepsilon_0$.*

The schemata referred to define, for any given ordinal $\alpha < \varepsilon_0$, a function $F(x, y)$ of numbers x, y by $F(0, y) = G(y)$ and for $x \neq 0$, $F(x, y)$ in terms of $F(H_i(x, y), y)$ where the H_i are one or more functions with $H_i(x, y) <_\alpha x$. The collection of F definable using such schemata coupled with the usual schemata for primitive recursive definition is denoted REC($<\varepsilon_0$). In this way one obtains the following result due to Kreisel [1951,1952] from Corollary 3.2.4 and Theorem 4.4.2:

4.4.3. Theorem. *The provably recursive functions of PA (and hence also of HA) are exactly those in* $\mathrm{REC}(<\varepsilon_0)$.

A similar characterization can be obtained of the functionals which are asserted to exist in Kriesel's no-counterexample interpretation of *PA*, here via the ND-interpretation of *PA* in *T* (Corollary 3.2.6).

Returning to Theorem 4.4.2, for its proof one makes use of the following simple description of the normal terms $t[x_1, \ldots, x_n]$ of type 0 in which all the free variables x_i are of type 0: either

1. $t = 0$ or
2. $t = \mathrm{Sc}(s)$ where s is normal or
3. t is a variable of type 0 or
4. $t = \langle t_n \rangle(s)$ where each t_n and s is normal of type 0, but s is not a numeral.

It follows that the only closed terms of type 0 are numerals. The closed normal terms of type 1 are just those of the form $\lambda x^0.t[x]$ where t is normal of type 0.

5. The interpretation of fragments of arithmetic

5.1. $I\Sigma_1$ and the primitive recursive functions

In the interpretation of *PA* and *HA* the recursors R_σ were used to interpret the induction axioms, and it should not be surprising that weaker forms of recursion can be used to interpret weaker forms of induction. Let $I\Sigma_1$ be the fragment of *PA* in which induction is restricted to Σ^0_1 formulas. A nice application of the D-interpretation due to Parsons (cf. [1970,1972]) shows that any provably total recursive function of $I\Sigma_1$ is in fact primitive recursive.

The definition of the recursors R_σ in section 2.2 for $\sigma \neq 0$ is "impredicative," in that the evaluation of $\mathrm{R}_\sigma(f, g, n')$ at a given argument x presumes that $\mathrm{R}_\sigma(f, g, n)$ has already been defined for arbitrary arguments z. A "predicative" restriction of this scheme is given by the recursors $\hat{\mathrm{R}}_\sigma$ due to Kleene, which have the defining schemata

$$\hat{\mathrm{R}}_\sigma(f, g, 0, b) = f(b)$$
$$\hat{\mathrm{R}}_\sigma(f, g, n', b) = g(n, \hat{\mathrm{R}}_\sigma(f, g, n, b), b).$$

Note that each type σ is uniquely of the form $(\sigma_1, \ldots, \sigma_k) \to 0$ for some sequence $(\sigma_1, \ldots, \sigma_k)$. In the equations above, then, f is of type σ, g is of type $0 \to 0 \to \sigma$, and $b = (b_1^{\sigma_1}, \ldots, b_k^{\sigma_k})$ is a sequence of variables chosen so that $\hat{\mathrm{R}}_\sigma(f, g, n, b)$ is of type 0.[10] We let $\widehat{T}$ denote the restriction of T which only allows this type of recursion.[11]

[10] In contrast to Gödel's recursors, each $\hat{\mathrm{R}}_\sigma$ can in fact be defined from $\hat{\mathrm{R}}_0$ by the equation $\hat{\mathrm{R}}_\sigma = \lambda f, g, n, b\, \hat{\mathrm{R}}_0(f(b), \lambda k, l\; g(k, l, b), n)$.

[11] In the versions $\widehat{T_0}$ and I-$\widehat{T}$ we need to add, as in Parsons [1972], conditional functionals $\mathrm{Cond}_\rho : (0, \rho, \rho) \to 0$. These functionals have defining equations $\mathrm{Cond}_\rho(w, u, v) = u$ if $w = 0$ and $\mathrm{Cond}_\rho(w, u, v) = v$ otherwise; in these theories they are no longer definable using the $\hat{\mathrm{R}}$ recursors.

5.1.1. Theorem. *The closed level 1 terms of $\widehat{T}$ denote primitive recursive functions. Moreover, there is a natural translation of type 0 terms t of $\widehat{T}$ whose only free variables are of type 0 to terms t^{PRA} of PRA, such that if $\widehat{T}$ proves $t = s$, then PRA proves $t^{PRA} = s^{PRA}$.*

5.1.2. Proof. The proof of the first assertion is by adaptation of Kleene [1959b, sections 1.5–1.7] to the present framework.[12] The idea is to define a class KL of functionals $F(b)$ of lists of arguments b of arbitrary finite type but with values of type 0 only, such that each F in KL is defined by a term in $\widehat{T}$, while each term t in $\widehat{T}$ is represented by an F in KL either directly (if it is of type 0) or by abstraction on some of the variables of F (otherwise).

In defining KL and in the proof of these facts we modify the conventions of section 2.2 as follows. We will use $\underline{\sigma}$ to denote (possibly empty) sequences of types $(\sigma_1, \ldots, \sigma_k)$, and by $(\underline{\sigma} \to 0)$ we mean 0 if $k = 0$ and $(\sigma_1 \to \ldots \to \sigma_k \to 0)$ otherwise. We will use $u^{\underline{\sigma}}$ to denote a sequence of terms $(u_1^{\sigma_1}, \ldots, u_k^{\sigma_k})$, and $\lambda u^{\underline{\sigma}}.F(u, \ldots)$ means $F(\ldots)$ when $n = 0$ and $\lambda u_1^{\sigma_1} \ldots \lambda u_k^{\sigma_k}.F(u_1, \ldots, u_k, \ldots)$ otherwise. So each type τ is uniquely of the form $(\underline{\sigma} \to 0)$ for some $\underline{\sigma}$, and $lev(\tau) = \max_{1 \le i \le k}(lev(\sigma_i) + 1)$.

The functionals F in KL are generated by the following schemata, in which n is a variable of type 0 and $b = (b_1^{\tau_1}, \ldots, b_\ell^{\tau_\ell})$:

1. $F(b) = 0$
2. $F(n, b) = n$
3. $F(n, b) = n'$
4. $F(a^\tau, b) = a(t_1, \ldots, t_k)$ where $\tau = (\underline{\sigma} \to 0)$, $\tau \neq 0$, $\sigma_i = (\underline{\rho}_i \to 0)$ (possibly 0) and $t_i = \lambda w_i^{\underline{\rho}_i} G_i(w_i, a, b)$ for $i = 1, \ldots, k$
5. $F(b) = G(H(b), b)$ where the first argument of G is of type 0
6. $F(b) = G(b_\pi)$ where b_π is a permutation of b by π
7. $F(0, b) = G(b)$, $F(k', b) = H(k, b, F(k, b))$

Then the following facts are established:

1. The functionals F of KL are closed under substitution of one or more terms t of type τ for variables a^τ of F where $\tau = (\underline{\sigma} \to 0)$, $\tau \neq 0$ and $t = \lambda u^{\underline{\sigma}}.H(u, \ldots)$.
2. For each $F(b)$ in KL we can find a term t of $\widehat{T}$ with free variables b, which defines F.
3. If t is a term of $\widehat{T}$ with free variables b, and t is of type $(\underline{\sigma} \to 0)$ (possibly 0), then we can find a functional $F(u, b)$ of KL such that $t(u) = F(u, b)$ for all u, b.
4. If $F(b)$ has all its variables $b_1, \ldots, b_\ell$ of type 0, then F is primitive recursive.

Fact 1 corresponds to the Full Substitution Theorem of Kleene [1959b,section 1.6]. It is proved by induction on the maximum m of the levels τ of the variables a^τ being substituted for and, for given m, by induction on the generation of F in KL. Facts 2 and 3 are straightforward, using 1 for the application step in 3. Finally, fact 4

[12] Here we do not consider the version I-$\widehat{T}$, with its additional functionals E_ρ.

follows by observation that scheme 4 of the definition of KL can never be applied in the generation of functionals all of whose arguments are of type 0.

This shows that the type 1 terms of $\widehat{T}$ denote primitive recursive functions. The conservation result of Theorem 5.1.1 is obtained by using the argument above to transform proofs in $\widehat{T}$ to proofs from schemata 1–7, and then using fact 4 to transform this to a proof in PRA.

(Remark. One can think of the proof of fact 1 as a normalization argument for a system of terms generated from schemata 1–7 together with a scheme of full substitution in place of scheme 4. Another route to the proof of Theorem 5.1.1 should be possible by normalization of the terms of $\widehat{T}$, but that would apparently require more work. For a proof of an analogous result for a system of feasible functionals of finite type, see the reference after Theorem 5.2.1 below.) $\square$

Let $\widehat{HA}^{\#}$ and $\widehat{PA}^{\#}$ denote variants of $HA^{\#}$ and $PA^{\#}$ respectively, in which induction is restricted to existential formulas. The reader can verify that the recursors $\hat{\mathrm{R}}$ are sufficient to interpret induction for these formulas, yielding

5.1.3. Theorem. *$\widehat{HA}^{\#}$ is D-interpreted in $\widehat{T}$, and $\widehat{PA}^{\#}$ is ND-interpreted in $\widehat{T}$.*

Since $\widehat{PA}^{\#}$ proves that any Σ^0_1 formula is equivalent to an existential one, $I\Sigma_1$ can be embedded in this theory, and Theorems 5.1.1 and 5.1.3 yield

5.1.4. Theorem. *$I\Sigma_1$ is conservative over PRA for Π^0_2 formulas, in the sense that if $I\Sigma_1$ proves $\forall x\, \exists y\, \varphi(x,y)$ for a Δ^0_0 formula φ, then there is a term f such that PRA proves $\varphi(x, f(x))$. Every provably total recursive function of $I\Sigma_1$ is primitive recursive.*

Since $I\Sigma_1$ can prove each primitive recursive function to be total, this last characterization is exact.

5.2. S^1_2 and the polynomial-time computable functions

When it comes to bounded arithmetic (cf. Chapter II), $I\Sigma_1$ is analogous to Buss' theory S^1_2, and PRA is analogous to Cook's theory PV, which axiomatizes the polynomial-time computable functions as characterized in section 1.3.7 of Chapter II. In Buss [1986] it is shown that S^1_2 is conservative over PV in the sense of Theorem 5.1.4. This result has been reobtained using a D-interpretation in Cook and Urquhart [1993], and it is this presentation that we now sketch (cf. also Feferman [1990]).

Cook and Urquhart start by defining a higher-type version PV^{ω} of the theory PV. Aside from a careful choice of initial functions, which hinge on the fact that one is supposed to think of the natural numbers in terms of their binary representations, the computational strength of PV^{ω} comes from recursors $\tilde{\mathrm{R}}$, which allow for "higher-type limited recursion on notation." These are described by the equations

$$\tilde{\mathrm{R}}(f,g,h,0,b) = f(b)$$

$$\tilde{\mathrm{R}}(f,g,h,n',b) = \begin{cases} g(n,\tilde{\mathrm{R}}(f,g,h,\lfloor n/2\rfloor,b),b) & \text{if this has length less than} \\ & \text{that of } h(n,b) \\ h(n,b) & \text{otherwise.} \end{cases}$$

Here b once again denotes a sequence of variables chosen so that $\tilde{R}(f,g,h,0,b)$ is of type 0, h is a type 1 function which provides a bound on the growth of the function being defined, $\lfloor\cdot/2\rfloor$ is among the initial functions of PV^ω, and additional initial functions representing "length" subtraction and a conditional are used to express the second equation.[13] The following theorem is analogous to Theorem 5.1.1.

5.2.1. Theorem. *The level 1 terms of PV^ω denote polynomial-time computable functions. Moreover, there is a natural translation of type 0 terms t of PV^ω whose only free variables are of type 0 to terms t^{PV} of PV, so that if PV^ω proves $t = s$, then PV proves $t^{PV} = s^{PV}$.*

Full details are provided in Cook and Urquhart [1993,pp. 140–146].

Finally, IPV^ω and CPV^ω are defined to be quantifier versions of PV^ω based on intuitionistic and classical logic respectively, where only type 0 equality is allowed. In these theories, induction is allowed for "NP-predicates," that is, formulas of the form $\exists y \leq t\ (r = s)$ where all the free variables of t have type 0. Since the recursors $\tilde{\mathrm{R}}$ are sufficient to interpret this form of induction, we have, in analogy to Theorem 5.1.3,

5.2.2. Theorem. *$IPV^\omega + (MP)$ is D-interpreted in PV^ω, and CPV^ω is ND-interpreted in PV^ω.*

S^1_2 can be embedded in CPV^ω, since the latter theory can prove any Σ^b_1 formula equivalent to an NP-predicate as described above. Hence we have the following

5.2.3. Theorem. *S^1_2 is conservative over PV for $\forall\Sigma^b_1$ sentences, in the sense that if S^1_2 proves $\forall x\, \exists y\, \varphi(x,y)$ for φ a Σ^b_1 formula, then there is a term f such that PV proves $\varphi(x, f(x))$. Every Σ^b_1-definable function of S^1_2 is polynomial-time computable.*

By Theorem 1.3.4.1 of Chapter II this last characterization is exact.

6. The interpretation of analysis

6.1. Towards the interpretation of stronger theories

At the end of his 1958 paper, Gödel made the following suggestions regarding the functional interpretation of stronger theories.

[13]In Cook and Urquhart [1993,section 6] only the type 0 recursors are taken to be basic, with the others defined as in footnote 10.

> It is clear that, starting from the same basic idea, one can also construct systems that are much stronger than T, for example by admitting transfinite types or the sort of inference that Brouwer used in proving the "fan theorem."

In section 10 we will address the first proposal, expanding the notion of "type" to the transfinite. In this section we explore the second proposal. The fan theorem that Gödel refers to is, from a classical point of view, equivalent to weak König's lemma, to be dealt with in section 7. Brouwer was able to prove the fan theorem using a principle of "bar induction" which he felt was justified by a constructive interpretation of the terms involved (cf. Beeson [1985], Troelstra and van Dalen [1988]). In a posthumous paper, C. Spector [1962] used a generalization of this principle to justify a computational scheme which he dubbed "bar recursion." With this scheme Spector was able to provide a functional interpretation of full second-order arithmetic, that is, the theory $PA^2 + (CA)$ defined in section 6.2 below.[14]

While Spector used bar induction to justify bar recursion, he apparently intended to show that, conversely, bar recursion could be used to obtain a functional interpretation of bar induction. This task was in fact carried out by Howard [1968], and is the approach we outline here.

6.2. Analysis and higher type extensions

The language of second-order arithmetic extends the language of Peano arithmetic with variables that range over sets of numbers, and a binary membership relation $x \in Y$. In this language we only allow first-order equality, defining the assertion $Y = Z$ to mean

$$\forall x\ (x \in Y \leftrightarrow x \in Z).$$

The theory of second-order arithmetic, $PA^2 + (CA)$, extends Peano arithmetic with induction for formulas in the expanded language, and the comprehension scheme

$$(CA) \qquad \exists Y\ \forall x\ (x \in Y \leftrightarrow \varphi(x))$$

for arbitrary formulas φ. (As usual we denote the scheme in which φ is restricted to formulas in Γ by $(\Gamma\text{-}CA)$.) The theory $PA^2 + (CA)$ is often called "analysis" due to the observation that, via the coding of real numbers and continuous functions as sets of natural numbers, one can develop a good theory of the continuum from these axioms. (In fact, by the work of Weyl [1918,1994], first-order arithmetic comprehension with parameters suffices for most practical purposes.)

In order to embed $PA^2 + (CA)$ in an extension of $PA^{\#}$, we identify sets Y with their characteristic functions χ_Y, and read $t \in Y$ as $\chi_Y(t) \neq 0$. This provides a

[14] We should mention in this connection Kreisel [1959], in which it is shown that if second-order arithmetic proves a formula φ, the witnessing functions can be taken to be "recursively continuous," in a sense defined, independently, by Kreisel [op. cit.] and Kleene [1959a] (cf. also Feferman [1993]). For an interesting extension of bar recursion to transfinite types, see Friedrich [1985].

natural way of interpreting the set variables of $PA^2 + (CA)$ with function variables of type 1.

Recall the axiom of choice

$$(AC) \qquad \forall x\, \exists y\, \varphi(x, y) \to \exists f\, \forall x\, \varphi(x, f(x)).$$

We will denote the scheme in which the variables x and y are restricted to be of type σ and τ respectively by $(AC_{\sigma\tau})$. We have seen that adding the axiom of choice (AC) to HA^ω results in a theory that is no stronger than Heyting arithmetic. In contrast, the addition of even (AC_{00}) to PA^ω results in a theory that includes (CA), since one can apply this choice principle to the classically valid formula

$$\forall x\, \exists y\, ((y = 0 \land \neg\varphi(x)) \lor (y = 1 \land \varphi(x))).$$

More generally, assuming that Γ satisfies some basic closure conditions (including closure under negations), $(\Gamma\text{-}CA)$ follows classically from $(\Gamma\text{-}AC_{00})$. So, to interpret analysis, it clearly suffices to interpret the theory $PA^\omega + (AC_{00})$.

6.3. The principle of bar induction

In the following statement of the principle of bar induction used in the Spector-Howard interpretation, the variable c represents a finite sequence $\langle c_0, c_1, \ldots, c_{k-1}\rangle$ of objects of type σ (suitably coded), while the variable f ranges over infinite sequences of objects of type σ, which is to say that f is a functional of type $0 \to \sigma$. Given such an f, let $\overline{f}(k)$ to denote the initial segment of f of length k, i.e. the finite sequence

$$\langle f(0), f(1), \ldots, f(k-1)\rangle.$$

Finally, if u is an element of type σ, $c\hat{\ }u$ denotes the sequence obtained by appending u to c.

6.3.1. Principle of bar induction at type σ. *Suppose φ and ψ are predicates of finite sequences of objects of type σ, with the following four properties:*

1. $\forall f\, \exists k\, \psi(\overline{f}(k))$
2. $\forall c\, (\psi(c) \lor \neg\psi(c))$
3. $\forall c\, (\psi(c) \to \varphi(c))$
4. $\forall c\, (\forall u\, \varphi(c\hat{\ }u) \to \varphi(c))$

Then φ holds at the empty sequence, $\langle\rangle$.

To make sense of this principle, imagine the tree of all finite sequences of objects of type σ. Clauses (1–3) imply that every path through the tree passes through a node c where ψ holds, and hence φ holds as well. The first nodes c where $\psi(c)$ hold form, in Brouwer's terminology, a "bar." Clause (4) asserts that if φ holds at every child of a node, then it holds at the node as well, allowing the property φ to "percolate" towards the root.

The principle of bar induction may be read as a statement of induction over a well-founded relation. Classically, it follows from an appropriate axiom of dependent choices. Assuming Clause (4), the failure of φ to hold at the empty sequence would allow one to construct an infinite sequence

$$c_0, c_1, c_2, \ldots$$

with the property that φ does not hold at any initial segment $\langle c_0, \ldots c_{k-1}\rangle$. Defining the function f by

$$f(k) =_{\text{def}} c_k$$

would then provide a counterexample to (1) and (3).

In the special case in which σ is the type of natural numbers, Principle 6.3.1 is Kleene's exposition of Brouwer's "bar theorem" (cf. Kleene and Vesley [1965]). The generalization to higher types is due to Spector.

To express this principle in the language of $HA^{\#}$, note that we may identify each sequence $\langle c_0, c_1, \ldots, c_{k-1}\rangle$ with the pair C, k, where C is of type $0 \to \sigma$ and

$$C(x) = \begin{cases} c_x & \text{if } x < k \\ 0^\sigma & \text{otherwise.} \end{cases}$$

(Inductively one can define for each σ a constant "zero" functional denoted by 0^σ.) References to such c can then be taken as shorthand for references to such pairs C, k. For this representation we will use $\hat{c}$ and length(c) to denote C and k respectively.

Let (BI_σ) denote the principle of bar induction for sequences of objects of type σ, and let (BI) denote the same principle for arbitrary σ. We will see in section 6.5 that the theory $HA^{\#} + (BI)$ is in fact strong enough to interpret $PA^\omega + (AC_{00})$, and hence analysis. First, however, we show that bar induction has a computational analogue in a form of recursion, much in the way that arithmetic induction has its computational analogue in the recursors R of section 2.2.

6.4. The interpretation of bar induction using bar recursion

For each σ and τ, the principle of bar recursion uniformly associates with given functionals G, H, and Y a new functional F which maps finite sequences c of objects of type σ to objects of type τ, according to the defining equation

$$F(c) = \begin{cases} G(c) & \text{if } Y(\hat{c}) < \text{length}(c) \\ H(\lambda u.F(c\hat{\ }u), c) & \text{if } Y(\hat{c}) \geq \text{length}(c). \end{cases}$$

To make the uniformity clear, and in analogy to the recursions in section 2.2, one introduces a single functional $\mathrm{B}_{\sigma\tau}$ for each σ and τ and replaces F above by $\mathrm{B}_{\sigma\tau}(G, H, Y)$.

From the definition of F we see that if $Y(\hat{c}) <$ length(c) then $F(c)$ is defined outright, and otherwise $F(c)$ depends on the values $F(c\hat{\ }u)$, for arbitrary u. One should think of the values of F as being computed by recursion along a well-founded tree of sequences determined by Y, in a way that will be made more explicit below. Spector showed that the principle of bar induction can be used to justify bar recursion, when the argument Y is taken to be a continuous functional (in the Kleene-Kreisel sense referred to in section 4.1 and footnote 14).

The following informal argument gives a hint as to how bar induction comes into play. Given the functional F described above, we would like to see that the value $F(\langle\rangle)$ is "defined." Let $\psi(c)$ denote the property $Y(\hat{c}) < \text{length}(c)$, and let $\varphi(c)$ denote the property that $F(c)$ is defined. Clauses (3–4) of Principle 6.3.1 are clearly seen to hold, and in an appropriate formalization clause (2) can be justified intuitionistically as well.

Seeing that the well-foundedness condition (1) holds requires verifying that for every function f, there is a finite initial segment $c = \overline{f}(k)$ so that $Y(\hat{c}) < k$. Suppose $Y(f) = n$. The computation of Y on f can only depend on finitely many values of f, contained among the list $f(0), f(1), \ldots, f(m)$ for some m. In particular, if $k = \max(m, n+1)$ and $c = \overline{f}(k)$, then

$$Y(\hat{c}) = Y(f) = n < k = \text{length}(c),$$

as desired.

The following lemma shows that conversely, bar recursion can be used to justify bar induction. We use $HA^{\#} + (BR)$ to denote $HA^{\#}$ augmented by the principle of bar recursion for arbitrary types.

6.4.1. Lemma. *$HA^{\#} + (BR)$ proves the principle of bar induction, (BI).*

While the proof requires some effort, the underlying idea is straightforward. By Theorem 3.1.1, $HA^{\#}$ proves (BI) equivalent to its D-interpretation, which asserts the existence of various functionals. One shows how to define these functionals explicitly, using bar recursion in a key instance, and employing a trick due to Kreisel to verify that these functionals have the desired properties. We refer the reader to Howard [1968] for details.

On the other hand, by Theorem 3.1.2, $HA^{\#} + (BR)$ is clearly D-interpreted in $T + (BR)$. Combining this observation with the previous lemma yields

6.4.2. Theorem. *$HA^{\#} + (BI)$ is D-interpreted in $T + (BR)$.*

6.5. Interpreting $PA^{\omega} + (AC_{00})$

We have seen that full second-order arithmetic can be embedded in the theory $PA^{\omega} + (AC_{00})$. Spector's interpretation applies not only to this but more generally to $(AC_{0\sigma})$ and an even stronger axiom scheme, (DC_{σ}), which asserts the existence of sequences formed by making dependent choices:

$$\forall x, a\, \exists b\, \varphi(x, a, b) \rightarrow \exists f\, \forall x\, \varphi(x, f(x), f(x+1)),$$

where x is of type 0 and a and b are of type σ. Let (DC) denote the union of the (DC_σ). In section 6.3 we pointed out that, classically, (DC) can be used to justify bar induction. The following theorem represents a kind of converse, and shows the full strength of Spector's interpretation.

6.5.1. Theorem. *$PA^\omega + (DC)$ is N-interpreted in $HA^\# + (BI)$.*

By Theorem 3.1.1 this is tantamount to the assertion that the double-negation interpretation of (DC) is provable in the latter theory. Howard was able to obtain this result by first reducing (DC) to a special case of bar induction in an appropriate extension of PA^ω (which is plausible when one considers the contrapositive of (DC)), and then deriving the double negation of this special case in $HA^\# + (BI)$. The entire proof would take us too far afield, and so once again we refer the reader to Howard [1968] for details.

Together with Theorem 6.4.2 this yields

6.5.2. Corollary. *$PA^\omega + (DC)$ is ND-interpreted in $T + (BR)$.*

Since $PA^\omega + (AC_{00})$ is contained in $PA^\omega + (DC)$, we have

6.5.3. Corollary. *The consistency of $T + (BR)$ implies the consistency of $PA^2 + (CA)$. Moreover, every provably total recursive function of $PA^2 + (CA)$ is represented by a type 1 term of $T + (BR)$.*

We have both a functional and a term model for $T + (BR)$. The former is given by the continuous functionals of Kleene and Kreisel indicated in section 4.1 and footnote 14, with the constants interpreted by recursively continuous functionals. The latter is established by work of Tait [1971] and independently Luckhardt [1973], which shows that $T + (BR)$ is normalizing and confluent. Both models can be formalized in $PA^2 + (CA)$, thus proving

6.5.4. Theorem. *The provably total recursive functions of $PA^2 + (CA)$ are exactly the ones represented by bar-recursive terms.*

6.6. Evaluation of Spector's interpretation

Spector was careful not to claim that the generalization of bar induction to higher types, which he used to justify bar recursion for continuous functionals, should be accepted on intuitionistic grounds. In fact, he offers the following caveat:

> The author believes that the bar theorem is itself questionable, and that until the bar theorem can be given a suitable foundation, the question of whether bar induction is intuitionistic is premature.

The question of whether bar recursion can be justified on constructive grounds was taken up in a seminar on the foundations of analysis led by G. Kreisel at Stanford in the summer of 1963. The seminar's conclusion, summarized by Kreisel in an ensuing report, was that

> ... the answer is negative by a wide margin, since not even bar recursion of type 2 can be proved consistent [by constructively accepted principles].

Despite this disappointing assessment, we feel that Spector's reduction of all of classical analysis to the *prima facie* simple computational construct of bar recursion is, in the end, rather remarkable.

7. Conservation results for weak König's lemma

7.1. The theory WKL_0

In this section we study a subsystem of second-order arithmetic known as WKL_0. WKL_0 is an interesting theory because it is just strong enough to prove, among other things, the Heine-Borel theorem, and the completeness and compactness of first-order logic (cf. Simpson [1987]). These facts make it all the more surprising that from a proof-theoretic standpoint the theory is fairly weak.

Formally WKL_0 is the fragment of $PA^2 + (CA)$ in which induction is restricted to Σ^0_1 formulas (set parameters are allowed), and instead of full comprehension the only set existence principles are given by a recursive comprehension scheme, (RCA), which asserts the existence of Δ^0_1-definable sets, and a weak version of König's lemma. The latter, denoted (WKL), asserts that every infinite binary tree has a path.

In order to express (WKL), let $\{0,1\}^k$ (resp. $\{0,1\}^{<\omega}$, $\{0,1\}^\omega$) denote the set of length-k (resp. finite, infinite) binary sequences, and fix a reasonable encoding of finite binary sequences as natural numbers. If s and t are sequences so coded, let $\mathrm{length}(s)$ denote the length of s and let the primitive recursive predicate $t \subseteq s$ assert that t is an initial segment of s. If b is an element of $\{0,1\}^\omega$, let $\overline{b}$ denote the initial segment function

$$\overline{b}(x) =_{\mathrm{def}} \langle b(0), b(1), \ldots, b(x-1)\rangle.$$

Finally, define the predicates

$$\mathrm{BinFunc}(b) \equiv_{\mathrm{def}} \forall x\ (b(x) = 0 \vee b(x) = 1)$$

which asserts that b is an element of $\{0,1\}^\omega$,

$$\mathrm{BinTree}(f) \equiv_{\mathrm{def}} \forall s \in f\ (s \in \{0,1\}^{<\omega} \wedge \forall t \subseteq s\ (t \in f))$$

which asserts that f is a binary tree, that is, a set of binary sequences closed under initial segments, and

$$\mathrm{Bounded}(f,k) \equiv_{\mathrm{def}} \forall s \in \{0,1\}^k\ (s \notin f)$$

which asserts that the height of the binary tree f is less than or equal to k. Weak König's lemma can be now be written

$$\forall f\ (\mathrm{BinTree}(f) \wedge \forall k\ \neg\mathrm{Bounded}(f,k) \to \exists b\ (\mathrm{BinFunc}(b) \wedge \forall k\ \overline{b}(k) \in f)).$$

In words, if f is a binary tree with branches at every level, then there is a path b through f.

Although the predicate Bounded is primitive recursive (since there are only finitely many binary sequences of length k), both BinFunc and BinTree require a universal quantifier. To avoid these, it turns out to be useful to employ the following trick. Given a function b, define b^{bin} by

$$b^{\mathrm{bin}} =_{\mathrm{def}} \begin{cases} 0 & \text{if } b(x) = 0 \\ 1 & \text{otherwise,} \end{cases}$$

denoting the casting of b to a binary function. (Formally, bin is a functional of type $(0 \to 0) \to (0 \to 0)$.) Similarly, define f^{tree} so that for any s

$$s \in f^{\mathrm{tree}} \leftrightarrow (s \in \{0,1\}^{<\omega} \wedge \forall t \subseteq s\ (t \in f)),$$

so that f^{tree} is a binary tree obtained from f by pruning away extraneous sequences. The theory $\widehat{HA}^{\omega}$ can then prove

$$\mathrm{BinFunc}(b^{\mathrm{bin}}) \wedge (\mathrm{BinFunc}(b) \to b = b^{\mathrm{bin}})$$

and, similarly,

$$\mathrm{BinTree}(f^{\mathrm{tree}}) \wedge (\mathrm{BinTree}(f) \to f = f^{\mathrm{tree}}).$$

As a result, when we are working in the language of $\widehat{HA}^{\omega}$, we can just as well take (WKL) to be given by

$$\forall f\ (\forall k\ \neg \mathrm{Bounded}(f^{\mathrm{tree}}, k) \to \exists b\ \forall k\ (\overline{b^{\mathrm{bin}}}(k) \in f^{\mathrm{tree}})).$$

Since Δ^0_1 comprehension follows from $(QF\text{-}AC)$, we can take WKL_0 to be contained in

$$\widehat{PA}^{\#} + (WKL).$$

The fact that WKL_0 is proof-theoretically weak is shown by the following celebrated theorem of H. Friedman.

7.1.1. Theorem. *WKL_0 is conservative over PRA for Π^0_2 formulas.*

While Friedman's original proof of Theorem 7.1.1 was model-theoretic, Kohlenbach [1992] showed how to use a D-interpretation to obtain the same result. In fact, Kohlenbach's work was dedicated to somewhat more general results; the approach we present here is a simplification of his methods applied to the specific case at hand. The details are manageable enough so that we can present them here in full. (The realization that hereditary majorizability, defined in the next section, can be used to obtain Theorem 7.1.1 is due to Sieg [1985], who used it with Herbrand methods. For other proof-theoretic approaches, as well as Harrington's strengthening of this theorem, see Hájek [1993], Avigad [1996a].)[15]

[15] A number of other interesting applications of the technique of majorization in combination with functional interpretation have been made by Kohlenbach in a series of papers; cf. Kohlenbach [1996a,1996b] among others.

7.2. Hereditary majorizability

If one puts the discrete topology on the set $\{0,1\}$, weak König's lemma expresses the compactness of $\{0,1\}^\omega$ under the associated product topology. Recall that $\widehat{T}$ has models in which functionals F of type $(0 \to 0) \to 0$ are computable, and hence continuous, since, for any g, the computation of $F(g)$ depends on only finitely many values of g. As a result, the compactness of $\{0,1\}^\omega$ implies that any such F is necessarily bounded when restricted to functions in this space.

The notion of hereditary majorizability, due to Howard [1973], is an effective generalization of this observation.

7.2.1. Definition. For each type σ, we define a relation $a \leq_* b$ for terms a and b of type σ, as follows.

1. If $\sigma = 0$, $a \leq_* b$ is just $a \leq b$.
2. If $\sigma = (\tau \to \rho)$, then $a \leq_* b$ if and only if

$$\forall x, y\ (x \leq_* y \to a(x) \leq_* b(y))$$

where the variables x and y are of type τ.

The relation $a \leq_* b$ is read "a is hereditarily majorized by b."

Notice that

$$\forall f\ (\mathrm{BinFunc}(f) \leftrightarrow f \leq_* \lambda x.1).$$

It is not difficult to majorize type 1 functions:

7.2.2. Proposition. *Given a type 1 term f, define*

$$f^*(x) =_{\mathrm{def}} \max_{y \leq x} f(y).$$

Then $f \leq_* f^*$.

Though we cannot prove in $\widehat{T}$ that for every functional F there is another functional G that majorizes it, we *can* majorize *closed* terms.

7.2.3. Proposition. *For every closed term F in the language of $\widehat{T}$ there is another closed term G such that $\widehat{HA}^\omega$ proves $F \leq_* G$.*

7.2.4. Proof. Inductively, for each term $F[x_1, \ldots, x_n]$ with free variables shown, one constructs a term $G[y_1, \ldots, y_n]$ such that $\widehat{T}$ proves

$$x_1 \leq_* y_1 \wedge \ldots \wedge x_n \leq_* y_n \to F \leq_* G.$$

The main case is for $F = \hat{\mathrm{R}}$. Here we can take G defined by

$$\begin{aligned} G(f, g, 0, b) &= f(b) \\ G(f, g, n', b) &= \max(G(f, g, n, b), g(n, G(f, g, n, b), b)). \end{aligned}$$

See Howard [1973] for further details. □

Notice that if F is a closed term of type $(0 \to 0) \to 0$, $F \leq_* G$, and $k = G(\lambda x.1)$, then Definition 7.2.1 and the remark immediately following imply that for every b in $\{0,1\}^\omega$, we have that $F(b) < k$.

The following technical lemma will be needed in section 7.4.

7.2.5. Lemma. *If the term B is of type $(0 \to 0) \to (0 \to 0)$, then $\widehat{T}$ proves*

$$\forall f \ \mathrm{BinFunc}(B(f)) \leftrightarrow B \leq_* \lambda f \lambda x.1.$$

7.2.6. Proof. $B \leq_* \lambda f \lambda x.1$ is equivalent to

$$\forall f, g \ (f \leq_* g \to B(f) \leq_* \lambda x.1),$$

that is,

$$\forall f, g \ (f \leq_* g \to \mathrm{BinFunc}(B(f))). \tag{1}$$

Taking $g = f^*$ from Proposition 7.2.2 we see that (1) implies $\forall f \ \mathrm{BinFunc}(B(f))$, and the converse is trivial. □

7.3. Reducing WKL_0 to $\widehat{HA}^{\#} + (WKL')$

Since WKL_0 is contained in $\widehat{PA}^{\#} + (WKL)$, to obtain Theorem 7.1.1 it is sufficient, by Theorems 5.1.3 and 5.1.1, to show that the latter theory is conservative over $\widehat{HA}^{\#}$ for Π^0_2 sentences. Our first step is to reduce it to an intuitionistic variant.

7.3.1. Lemma. *The theory $\widehat{PA}^{\#} + (WKL)$ is N-interpreted in $\widehat{HA}^{\#} + (WKL)$.*

7.3.2. Proof. $\widehat{PA}^{\#}$ is N-interpreted in $\widehat{HA}^{\#}$, and the double-negation interpretation of (WKL), given by

$$\forall f \ (\forall k \ \neg \mathrm{Bounded}(f^{\mathrm{tree}}, k) \to \neg\neg \exists b \ \forall k \ (\overline{b^{\mathrm{bin}}}(k) \in f^{\mathrm{tree}})),$$

follows from (WKL) since the double-negation only weakens the conclusion. □

We define a convenient variant of (WKL) as follows:

(WKL') $\qquad \forall f \ \exists b \ \forall k \ (\neg \mathrm{Bounded}(f^{\mathrm{tree}}, k) \to \overline{b^{\mathrm{bin}}}(k) \in f^{\mathrm{tree}}).$

Proving the following lemma is a simple exercise in intuitionistic logic.

7.3.3. Lemma. *Over intuitionistic logic, (WKL) is implied by (WKL').*

In fact, the two principles are equivalent over $\widehat{HA}^{\#}$, but the converse direction requires more work and is not needed below.

7.4. Reducing $\widehat{HA}^{\#} + (WKL')$ to $\widehat{HA}^{\#}$

Though $\widehat{HA}^{\#}$ doesn't prove (WKL), we can show that adding (WKL) doesn't allow $\widehat{HA}^{\#}$ to prove any new Π^0_2 sentences. The main avenue to this result is abstracted in the following

7.4.1. Lemma. *Suppose $\widehat{HA}^{\#}$ proves*

$$\exists a\ \forall b, c\ S(a,b,c) \to \forall x\ \exists y\ R(x,y). \tag{2}$$

Then there is a specific term $\tilde{c}(a,x)$ (whose other free variables are among those of R and S) such that $\widehat{HA}^{\#}$ proves

$$\forall x\ \exists a\ \forall b\ S(a,b,\tilde{c}(a,x)) \to \forall x\ \exists y\ R(x,y).$$

The proof is straightforward, using the scheme $\varphi \leftrightarrow \varphi^D$ of $\widehat{HA}^{\#}$ to convert (2) with "$\forall x$" deleted to its D-translation, applying Theorem 5.1.3 to extract a term $\tilde{c}$ of $\widehat{T}$, and then manipulating quantifiers. The upshot is that to eliminate the assumption $\exists a\ \forall b, c\ S(a,b,c)$ from a proof of $\forall x\ \exists y\ R(x,y)$ in $\widehat{HA}^{\#}$, it suffices to show that one can prove

$$\exists a\ \forall b\ S(a,b,\tilde{c}(a,x))$$

for any specific term $\tilde{c}$.

We now apply this to the situation at hand.

7.4.2. Lemma. *$\widehat{HA}^{\#} + (WKL)$ is conservative over $\widehat{HA}^{\#}$ for Π^0_2 sentences.*

7.4.3. Proof. By Lemma 7.3.3 and the deduction theorem, if $\widehat{HA}^{\#} + (WKL)$ proves

$$\forall x\ \exists y\ R(x,y),$$

then $\widehat{HA}^{\#}$ proves

$$\forall f\ \exists b\ \forall k\ (\neg\text{Bounded}(f^{\text{tree}},k) \to \overline{b^{\text{bin}}}(k) \in f^{\text{tree}}) \to \forall x\ \exists y\ R(x,y).^{16}$$

Applying (AC) to the hypothesis, we obtain

$$\exists B\ \forall f,k\ (\neg\text{Bounded}(f^{\text{tree}},k) \to \overline{B(f)^{\text{bin}}}(k) \in f^{\text{tree}}) \to \forall x\ \exists y\ R(x,y).$$

Applying Lemma 7.4.1 with k in place of c, we are reduced to showing that $HA^{\#}$ proves

$$\exists B\ \forall f\ (\neg\text{Bounded}(f^{\text{tree}},\tilde{k}(B,x)) \to \overline{B(f)^{\text{bin}}}(\tilde{k}(B,x)) \in f^{\text{tree}}) \tag{3}$$

[16] Here we need to use $\widehat{HA}_0^{\#}$ or $I\text{-}\widehat{HA}^{\#}$ instead of $WE\text{-}\widehat{HA}^{\#}$, since the deduction theorem fails for the latter theory; cf. 3.1. However, for a way around this, see Kohlenbach [1992], p. 1246.

for any closed term $\tilde{k}$.

We now bring in the notion of hereditary majorizability to bound the value of $\tilde{k}(B,x)$. By Proposition 7.2.3, we can find a term k^* that hereditarily majorizes $\tilde{k}$. Define

$$k'(x) =_{\text{def}} k^*(\lambda x \lambda f.1, x).$$

Since $x \leq_* x$, from Lemma 7.2.5 we see that $\widehat{HA}^{\#}$ proves

$$\forall f \ \text{BinFunc}(B(f)) \rightarrow \tilde{k}(B,x) \leq k'(x). \tag{4}$$

To verify (3), we only need construct an appropriate B. Define $B'(x,f)$ as follows. Let l be the maximum value less than or equal to $k'(x)$ such that f^{tree} has an element of length l, and let s be any (e.g. the leftmost) such element. Let

$$B'(x,f)(y) =_{\text{def}} \begin{cases} s_y & \text{if } y < l \\ 0 & \text{otherwise,} \end{cases}$$

so that $B'(x,f)$, as an element of $\{0,1\}^\omega$, consists of s followed by a string of zeros. In particular, $\widehat{HA}^{\#}$ proves

$$l \leq k'(x) \wedge \neg\text{Bounded}(f^{\text{tree}}, l) \rightarrow \overline{B'(x,f)}(l) \in f^{\text{tree}}. \tag{5}$$

Finally, let $B(x) =_{\text{def}} \lambda f.B'(x,f)$. Then $\widehat{HA}^{\#}$ proves

$$\forall f \ \text{BinFunc}(B(x,f))$$

and so, using (4),

$$\forall x \ (\tilde{k}(B(x),x) \leq k'(x)).$$

By (5) we then have

$$\forall f \ (\neg\text{Bounded}(f^{\text{tree}}, \tilde{k}(B(x),x)) \rightarrow \overline{B(x,f)^{\text{bin}}}(\tilde{k}(B(x),x)) \in f^{\text{tree}}).$$

So $B(x)$ witnesses the existential quantifier in (3). □

Friedman's Theorem 7.1.1 now follows from Lemmas 7.3.1 and 7.4.2, together with Theorems 5.1.3 and 5.1.1.

Harrington's theorem states that WKL_0 is conservative over its subtheory RCA_0 (which omits (WKL)) for Π^1_1 sentences. Since RCA_0 is easily interpretable in $I\Sigma_1$, this, together with Theorem 5.1.4, yields another proof of Friedman's result. However, the question as to whether a functional interpretation can be used to obtain Harrington's strengthening is still open.

The proof above can be adapted to yield similar conservation results for weaker fragments of second-order arithmetic, such as "elementary analysis," whose provably total recursive functions are all elementary recursive (cf. also Kohlenbach [1996b]). It is reasonable to conjecture that functional interpretations can be used to obtain similar results for systems of polynomial-time-computable arithmetic, especially in view of the conservation result of Ferreira [1988,1994], which showed that a suitable version of (WKL) is conservative over S^1_2 for Π^0_2 formulas (cf. also Cantini [1996]). However, the functional B' and the relation $\mathrm{Bounded}(f,k)$ from the proof above cannot be defined in, say, PV^ω, so new considerations seem to be necessary. (If the reader is concerned with polynomial *bounds* rather than a polynomial-time-computable Skolem function, he or she should consult Corollary 4.28 of Kohlenbach [1996b].)

8. Non-constructive interpretations and applications

8.1. Overview of the section; general pattern.

This section shows how the D-interpretation can be extended by inclusion of certain non-constructive functional operators in order to obtain conservation results of various finite type systems contained in $PA^\omega + (AC)$ over related second-order systems. As we have noted in section 3.1, $(QF\text{-}AC)$ is automatically preserved under the ND-interpretation. This can be extended to $(\Gamma\text{-}AC)$ for some larger classes of formulas Γ by adjunction of suitable (necessarily non-constructive) Skolem functionals with associated axioms which imply that each formula in Γ is equivalent to a QF- (i.e. quantifier-free) formula.

The paradigm case is given by adjunction of the non-constructive minimum operator μ, which allows us to reduce every arithmetical formula to QF form. Then axioms (μ) for this with $PA^\omega + (QF\text{-}AC)$ imply the second-order system $\Sigma^1_1\text{-}DC$, and it is shown that the system $PA^\omega + (\mu) + (QF\text{-}AC)$ is ND-interpreted in $T + (\mu)$. Normalization of a system of infinite terms for the latter, just as described in section 4.4 above for T by the methods of Tait [1965], shows that the type 1 functions thus generated lie in the hierarchy of hyperarithmetic functions H_α for $\alpha < \varepsilon_0$. Then formalization of this model can be carried out in a theory $(\Pi^0_1\text{-}CA)_{<\varepsilon_0}$ of relative arithmetical comprehension iterated up to ε_0, yielding a number of conservation results, including a well-known one due to Friedman [1970]. The arguments for this are detailed in sections 8.2 and 8.3. It is also shown that when the Bar Rule is adjoined to the above finite type system, we obtain conservation over the full predicative system $(\Pi^0_1\text{-}CA)_{<\Gamma_0}$.

Analogous results to those in sections 8.2–8.3 are sketched in section 8.4 for adjunction of the Kleene basis operator which transforms every Σ^1_1 formula into an equivalent Π^0_1 formula and thence a QF-formula using μ. This allows us to lift the preceding results throughout one step up in the analytical hierarchy. The results of sections 8.2–8.4 are due to Feferman [1971,1977,1979].

8.2. Finite type systems with non-constructive μ-operator, and related second-order systems

8.2.1. Subsystems of PA^ω

Besides the classical finite type theory PA^ω which includes the constants K, S and R in all types and full induction for all formulas, we shall consider also its subsystems $\widehat{PA}^\omega$ in which R is replaced by the predicative recursion operator $\widehat{\mathrm{R}}$ introduced in section 5.1, and subsystems of both, obtained by restricting induction as follows:

(*Res-Ind*) $$0 \in X \wedge \forall x\, (x \in X \to x' \in X) \to \forall x\, (x \in X).$$

Here X ranges over sets of type 1 (identified with characteristic functions as in section 6.3). When we add principles which imply (Γ-CA) for various classes of formulas Γ, we can infer from (*Res-Ind*) all substitution instances for X by formulas φ in Γ, but not full induction in general. If S is any system of second or higher order which includes the scheme of full induction, we shall denote by *Res-S* the system obtained by replacing that scheme by the axiom (*Res-Ind*) above. (In the literature this is also denoted $S\restriction$ or S_0.) The higher type systems which will figure prominently in the following are PA^ω, *Res-*PA^ω, $\widehat{PA}^\omega$ and *Res-*$\widehat{PA}^\omega$, together with (QF-AC) and some special axioms for non-constructive functionals.

8.2.2. Functionals for numerical quantification and choice

The functional E_0 of type 2 with values 0 and 1 only is defined by

(E_0) $$\mathrm{E}_0(f) = 0 \leftrightarrow \exists x\, (f(x) = 0).$$

(This functional is also denoted ${}^2E$ in the recursion-theoretic literature.) Using this axiom, every arithmetical formula is equivalent to a QF-formula. If we are to use it with an ND-interpretation, we should consider the two implications:

$$f(x) = 0 \to \mathrm{E}_0(f) = 0 \quad \text{and} \quad \mathrm{E}_0(f) = 0 \to \exists x\, (f(x) = 0).$$

The ND-interpretation preserves the first of these but requires for the second a functional X such that $\mathrm{E}_0(f) = 0 \to f(X(f)) = 0$. Combining this with the first implication gives $f(x) = 0 \to f(X(f)) = 0$. Such an X is then a choice or Skolem functional for type 0 quantification. We shall use the new symbol μ for such a functional and take as its axiom

(μ) $$f(x) = 0 \to f(\mu(f)) = 0.$$

Then E_0 may be defined simply in terms of μ by $\mathrm{E}_0(f) = 0$ if $f(\mu(f)) = 0$, otherwise $\mathrm{E}_0(f) = 1$. The advantage of using μ in place of E_0 is that its axiom is preserved directly under the ND-interpretation without requiring any supplementary functionals. Note that the non-constructive minimum operator in its usual sense, defined as the least x such that $f(x) = 0$ if $\exists x\, (f(x) = 0)$ and 0 otherwise, is also definable in terms of μ and the primitive recursive bounded minimum operator.

8.2.3. Second-order forms of (CA), (AC) and (DC)

We shall consider various restricted forms of these principles for the usual arithmetical (Π^0_n and Σ^0_n) and analytical (Π^1_n and Σ^1_n) hierarchies. That is, for Γ one of these classes of formulas, we shall consider the schemata

$$(\mathit{\Gamma}\text{-}CA) \qquad \exists Y\, \forall x\, (x \in Y \leftrightarrow \varphi(x))$$

$$(\mathit{\Gamma}\text{-}AC_{01}) \qquad \forall x\, \exists f\, \varphi(x, f) \to \exists g\, \forall x\, \varphi(x, g_x)$$

$$(\mathit{\Gamma}\text{-}DC_1) \qquad \forall x, f\, \exists g\, \varphi(x, f, g) \to \forall f\, \exists g\, (g_0 = f \wedge \forall x\, \varphi(x, g_x, g_{x+1}))$$

where in each case φ ranges over formulas in Γ, and in the two choice principles g_x is written for $\lambda y.g(x, y)$. In addition, we shall consider the scheme

$$(\mathit{\Delta}_{\mathit{\Gamma}}\text{-}CA) \qquad \forall x\, (\varphi(x) \leftrightarrow \neg\psi(x)) \to \exists Y\, \forall x\, (x \in Y \leftrightarrow \varphi(x))$$

where φ and ψ again range over formulas in Γ. In the context of second-order systems, we omit the subscripts from the (AC) and (DC) principles. Also we use these principles to name the second-order system S obtained by adjoining them to PA together with full induction for second-order formulas. Then, in accordance with 8.2.1, we use *Res*-S to name the same system with restricted induction. The systems of particular concern to us in this section and the next are: $\mathit{\Pi}^0_1$-CA, $\mathit{\Delta}^1_1$-CA, $\mathit{\Sigma}^1_1$-AC, $\mathit{\Sigma}^1_1$-DC, and their restricted versions, while in section 8.4 we shall take up the corresponding systems one level up in the analytic hierarchy. Note that the system $\mathit{\Pi}^0_1$-CA proves the same theorems as the system ACA based on the arithmetical comprehension axiom, i.e. $(\mathit{\Gamma}\text{-}CA)$ where Γ is the class of arithmetical formulas. For, we can derive closure under complement from $(\mathit{\Pi}^0_1\text{-}CA)$ and hence the principle $(\mathit{\Sigma}^0_1\text{-}CA)$, and then by iterating these we obtain (ACA) in general. We note this for the record in the following; the further relationships below are established just as easily.

8.2.4. Theorem.

1. $\mathit{\Pi}^0_1\text{-}CA = ACA$
2. $\mathit{\Pi}^0_1\text{-}AC = \mathit{\Sigma}^1_1\text{-}AC$
3. $\mathit{\Pi}^0_1\text{-}DC = \mathit{\Sigma}^1_1\text{-}DC$
4. $\mathit{\Pi}^0_1\text{-}CA \subseteq \mathit{\Delta}^1_1\text{-}CA \subseteq \mathit{\Sigma}^1_1\text{-}AC \subseteq \mathit{\Sigma}^1_1\text{-}DC$
5. *The same results hold for the corresponding theories with* (*Res-Ind*).

Apropos of this last, it is a familiar result that *Res*-$\mathit{\Pi}^0_1$-CA or ACA_0 is a conservative extension of PA; a model-theoretic argument gives the quickest proof of that.

8.2.5. Transfinite induction below ε_0.

In the following we shall use variables $\alpha, \beta, \gamma, \ldots$ to range both over ordinals and over terms in a natural recursive ordinal representation system for ordinals up to ε_0. We write $<_\alpha$ for the initial segment determined in this ordering by α. The scheme of transfinite induction up to α for a formula $\varphi(x)$ is given by

$$(TI_\alpha(\varphi)) \qquad \forall x\ (\forall y\ (y <_\alpha x \to \varphi(y)) \to \varphi(x)) \to \forall x\ \varphi(x).$$

It was proved by Gentzen that this scheme is derivable in PA for each formula φ of arithmetic and each $\alpha < \varepsilon_0$. Analysis of his argument actually shows more:

8.2.6. Theorem. *Suppose S is any system whose language contains that of PA and which contains full induction. Then for each formula φ of S and each $\alpha < \varepsilon_0$ we can prove $(TI_\alpha)(\varphi)$ in S.*

The argument for this proceeds by showing that transfinite induction up to ω^α with respect to φ can be reduced to transfinite induction up to α with respect to a formula φ^* which is built by propositional operations and numerical (first-order) quantification from φ. (For a simple choice of φ^* due to Schütte, cf. Feferman [1977,p. 946].)

8.2.7. Transfinitely iterated arithmetical comprehension

The hyperarithmetic hierarchy up to any α and relative to any initial set X is the sequence of sets $\langle H^X_\beta \rangle_{\beta<\alpha}$ for which $H^X_0 = X$, $(H^X_{\beta+1}) = (H^X_\beta)'$, and for limit β, H^X_β is the join of the sequence H^X_γ for $\gamma < \beta$. (Here Y' represents the set obtained by applying the Turing jump operator to Y.) Such a sequence may be represented by its join, which is a single set $H^X(\alpha)$ consisting of the pairs $\langle \beta, x\rangle$ such that $\beta < \alpha$ and $x \in H^X_\beta$. The assertion that $H^X(\alpha)$ exists for each X is denoted $(\Pi^0_1\text{-}CA)_\alpha$, since the successor step from β to $\beta + 1$ may be considered as being obtained by one application of $(\Pi^0_1\text{-}CA)$. We use $(\Pi^0_1\text{-}CA)_{<\varepsilon_0}$ to denote the collection of all $(\Pi^0_1\text{-}CA)_\alpha$ for $\alpha < \varepsilon_0$, as well as the system based on this collection. In these terms we can strengthen the first inclusion in part 4 of the Theorem 8.2.4 to the following

8.2.8. Theorem. $(\Pi^0_1\text{-}CA)_{<\varepsilon_0} \subseteq \Delta^1_1\text{-}CA$.

The proof consists in showing, for each $\alpha < \varepsilon_0$, the existence of $H^X(\beta)$ for $\beta < \alpha$ by the scheme of transfinite induction up to α. When passing to the limit β one makes use of the fact that if $H^X(\gamma)$ exists then it is uniquely determined by its recursive defining conditions, so the set $H^X(\beta)$ can be defined in Δ^1_1 form (cf. Feferman [1977,pp. 944–947] for further details).

Note that what is essential for this proof is availability of transfinite induction for Σ^1_1 formulas in the system $\Delta^1_1\text{-}CA$, which we do not have in its restricted version. Our next step is to put these second-order systems together with the finite-type systems of 8.2.1. This leads to two chains of interest:

8.2.9. Theorem.

1. $(\Pi^0_1\text{-}CA)_{<\varepsilon_0} \subseteq \Delta^1_1\text{-}CA \subseteq \Sigma^1_1\text{-}AC \subseteq \Sigma^1_1\text{-}DC \subseteq PA^\omega + (\mu) + (QF\text{-}AC)$
2. $PA \subseteq Res\text{-}\Pi^0_1\text{-}CA \subseteq Res\text{-}\Delta^1_1\text{-}CA \subseteq Res\text{-}\Sigma^1_1\text{-}AC$
$\subseteq Res\text{-}\widehat{PA}^\omega + (\mu) + (QF\text{-}AC)$.

What's added here to the information from above are the final inclusions. In 1, from $\forall x, f\ \exists g\ \varphi(x,f,g)$ with φ arithmetical, one uses (μ) and $(QF\text{-}AC)$ to infer

$$\exists G\ \forall x, f\ \varphi(x, f, G(f)).$$

Then, given any initial f, a sequence $\langle g_x\rangle$ with $g_0 = f \wedge \forall x\ \varphi(x, g_x, g_{x+1})$ is defined by the recursion $g_{x+1} = G(g_x)$ using the recursor R with values at type 1. The reason the same sort of argument cannot be carried out in 2 is that there we only have availability of the recursor $\widehat{\mathrm{R}}$ whose values are confined to type 0. Recursion is not needed though to infer the principle $(\Sigma^1_1\text{-}AC)$ from (μ) and $(QF\text{-}AC)$.

The next section is devoted to showing how these chains of inclusions can be closed proof-theoretically.

8.3. Functional interpretations using the (μ) operator

As noted above, the (μ) axiom is preserved under the ND-interpretation; in combination with the Theorems 3.1.3 and 3.1.4, this immediately yields:

8.3.1. Theorem. *$PA^\omega + (\mu) + (QF\text{-}AC)$ is ND-interpreted in $T + (\mu)$.*

The next step is to show how $T + (\mu)$ can be modeled in $(\Pi^0_1\text{-}CA)_{<\varepsilon_0}$. This is by an extension to μ, as a type 2 constant symbol, of Tait's infinite term model of T from section 4.4. No new arguments are needed for normalization and so, just as before, every term of $T + (\mu)$ translates into a (possibly) infinitely long term which, in turn, reduces effectively to a normal term t with $|t| < \varepsilon_0$. Only the description of normal infinitely long terms t at type 0 needs to be modified. For this purpose, one examines such t which may contain free variables x of type 0 and/or f of type 1, and arrives at the following possibilities (cf. the end of 4.4): either

1. $t = 0$ or
2. $t = Sc(s)$ where s is normal or
3. t is a variable of type 0 or
4. $t = \langle t_n\rangle(s)$ where each t_n is normal and s is normal or
5. $t = f(s)$ where f is a type 1 variable and s is normal or
6. $t = \mu(\lambda x.s)$ where s is normal or
7. $t = \mu(\langle t_n\rangle)$ where each t_n is normal.

Every closed term of type 1 in $T + (\mu)$ then reduces effectively to a normal term of the form $\lambda x.t$ where t is normal of type 0 or of the form $\langle t_n\rangle$ where each t_n is normal. What these represent as functions is the effective transfinite iteration of the μ operator, or equivalently the jump (E_0) operator, up to ordinals less than ε_0. Thus each closed term t of type 1 in $T + (\mu)$ represents a function which is recursive in H_α for some $\alpha < \varepsilon_0$. More generally, if t is a term with a free variable f of type 1, then t represents a function recursive uniformly in H^f_α for some $\alpha < \varepsilon_0$. Formalization of this argument leads finally to the following:

8.3.2. Theorem. *$PA^\omega + (\mu) + (QF\text{-}AC)$ is a conservative extension of $(\Pi^0_1\text{-}CA)_{<\varepsilon_0}$ for Π^1_2 sentences.*

For, given a provable Π^1_2 sentence $\forall f\ \exists g\ \varphi(f, g)$ with φ arithmetical, we use 8.3.1 to obtain a term $t[f]$ of type 1 such that $T + (\mu)$ proves $\varphi'(f, t[f])$, where φ' is quantifier- free and equivalent to φ under (μ), and then apply the preceding modeling argument.

8.3.3. Corollary. *$\Sigma^1_1\text{-}DC$ is a conservative extension of $(\Pi^0_1\text{-}CA)_{<\varepsilon_0}$ for Π^1_2 sentences.*

This result is due originally to Friedman [1970], who established it by model-theoretic techniques. Subsequently to the appearance of the above approach in the publications referred to in section 8.1, Feferman and Sieg [1981] used Herbrand-Gentzen methods with the μ operator to the same end.

Now turning to the restricted systems, the main result is

8.3.4. Theorem. *$Res\text{-}\widehat{PA}^\omega + (\mu) + (QF\text{-}AC)$ is a conservative extension of $Res\text{-}\Pi^0_1\text{-}CA$ for Π^1_2 sentences, and hence of PA for arithmetical sentences.*

Here, for the proof, one makes use of the ND-interpretation of our higher type system in $\widehat{T} + (\mu)$. The recursor $\hat{\mathrm{R}}$ can then be eliminated from the picture, since it can be defined arithmetically, hence by means simply of $+$, $\cdot$, and μ. Terms of the resulting system then normalize without passing to infinite terms, and so the functions represented by closed terms of type 1 are also arithmetical. The same holds for terms $t[f]$ of type 1 uniformly in a type 1 variable f.

8.3.5. Corollary. *$Res\text{-}\Sigma^1_1\text{-}AC$ is a conservative extension of $Res\text{-}\Pi^0_1\text{-}CA$ for Π^1_2 sentences, and hence of PA for arithmetical sentences.*

This was first established by model-theoretic methods by Barwise and Schlipf [1975] and independently Friedman [1976]. Again, Feferman and Sieg [1981] used Herbrand-Gentzen methods with μ to obtain the same result. Note that $Res\text{-}\Sigma^1_1\text{-}DC$ is stronger than $Res\text{-}\Pi^0_1\text{-}CA$ because it proves the existence of H_ω.

8.3.6. Predicatively provable ordinals

By formalizing work of Kleene, the systems $(\Pi^0_1\text{-}CA)_{<\alpha}$ are another form of ramified analysis RA_α when $\alpha = \omega \cdot \alpha$. The proof-theoretic ordinals[17] of these were established by Schütte [1960] (cf. also Schütte [1977]) in terms of the Veblen hierarchy of normal functions φ_a of ordinals, defined by:

1. $\varphi_0(\xi) = \omega^\xi$
2. For $\alpha \neq 0$, φ_α enumerates $\{\xi : \varphi_\beta(\xi) = \xi \text{ for all } \beta < \alpha\}$.

Each $\varphi_{\alpha+1}$ is then the critical function of φ_α, i.e. it enumerates the set of fixed points $\{\xi : \varphi_\alpha(\xi) = \xi\}$. In particular, $\varphi_1(\xi) = \varepsilon_\xi$, etc. The diagonal function $\lambda\xi.\varphi_\xi(0)$ is also normal and its least fixed point is denoted Γ_0. There is a natural notation system for ordinals up to Γ_0 (Feferman [1968b]), and we shall now use $\alpha, \beta, \gamma, \ldots$ to range over that system as well as over ordinals in the usual sense. It follows from Schütte's work referenced above that

8.3.7. Theorem. *For $\alpha = \omega \cdot \alpha$ (α not 0), the proof-theoretic ordinal of RA_α or, equivalently, of $(\Pi^0_1\text{-}CA)_{<\alpha}$, is $\varphi_\alpha(0)$.*

Now, the so-called *autonomous ordinals* of ramified analysis were defined to be those generated by the following "boot-strap" procedure:

1. 0 is an autonomous ordinal.
2. If β is an autonomous ordinal and α is a provably recursive ordinal of RA_β then α is an autonomous ordinal.

The systems RA_α for α autonomous were proposed by Kreisel as a characterization of predicative analysis, and so the autonomous ordinals are identified with the predicatively provable ordinals. Using 8.3.7, it was established independently by Schütte and Feferman in the mid 1960s that Γ_0 is the least non-autonomous ordinal, hence the least impredicative ordinal. In order to investigate the extent of predicativity in ordinary, unramified, analysis, Feferman [1964] described an unramified second-order system with finitely many schemata whose proof-theoretic ordinal is Γ_0. This was strengthened later in Feferman [1979] to the following:

8.3.8. Theorem. *The system $\Sigma^1_1\text{-}DC$ + (Bar-Rule) has proof-theoretic ordinal Γ_0; it is conservative over $(\Pi^0_1\text{-}CA)_{<\Gamma_0}$ for Π^1_2 sentences.*

The Bar-Rule, which is related to the principle of Bar Induction described in section 6.3 above, allows one to infer the scheme of transfinite induction on a recursive ordering when its well-foundedness has been established. The idea of the proof of Theorem 8.3.8 sketched op. cit. is to apply a functional interpretation to the still larger system $PA^\omega + (\mu) + (QF\text{-}AC) + (\mathit{Bar\text{-}Rule})$, carrying it into $T + (\mu) + (\mathit{Bar\text{-}Rule})$,

[17] The proof-theoretic ordinal of a system S is defined in several ways, e.g. as the least ordinal α which is not the order-type of a provably recursive well-ordering of the system, or as the least α (in a standard notation system) for which the consistency of S can be proved by transfinite induction up to α over a weak base system. These are equivalent in practice.

and then to model the latter in a boot-strap fashion in a system of infinitely long terms of length less than Γ_0. The final result is then obtained by formalization of the type 1 functions of that model as for Theorem 8.3.2 above. The details of this are rather complicated, and in its place a more digestible proof of Theorem 8.3.8 was subsequently obtained by Feferman and Jäger [1983] by Herbrand-Gentzen methods with the μ operator. But the functional interpretation served an important intermediate, partly heuristic role in arriving at this result.

8.4. Functional interpretations using the Kleene basis operator

8.4.1. Testing for (non-)well-foundedness as a functional

As in sections 6 and 7, we write $\overline{g}(x)$ for the sequence number coding the sequence $\langle g(0), \ldots, g(x-1)\rangle$ when g is of type 1. Then a type 1 function f represents a well-founded tree if $\forall g\ \exists x\ \ f(\overline{g}(x)) \neq 0$, where the tree consists of all sequence numbers s with $f(s) = 0$. Kleene's basis theorem for Σ^1_1 predicates provides a functional μ_1 which chooses a descending branch in this tree if it is not well-founded, recursive in the Suslin quantifier E_1 with values 0 and 1, given by

$$(\mathrm{E}_1) \qquad E_1(f) = 0 \leftrightarrow \exists g\ \forall x\ f(\overline{g}(x)) = 0.$$

Specifically, when $\mathrm{E}_1(f) = 0$, μ_1 is the left-most descending branch in this tree, with the successive values of $\mu_1(f) = g$ defined recursively by taking $g(x)$ to be the least y such that there is an infinite descending branch in the f-tree extending $\overline{g}(x)\hat{\ }\langle y\rangle$. Thus μ_1 satisfies the axiom

$$(\mu_1) \qquad \forall x\ f(\overline{g}(x)) = 0 \rightarrow \forall x\ f(\overline{\mu_1(f)}(x)) = 0.$$

Of course then E_1 is definable in terms of μ_1. This axiom is preserved under the N-interpretation, and in the presence of the axiom (μ) is equivalent to a quantifier-free statement,which is further preserved under the D-interpretation. So this leads us to consider the systems of 8.2.1 augmented by the axioms for both μ and μ_1. Then we obtain conservation results for these over second-order systems involving Π^1_1-CA, Σ^1_2-AC and Σ^1_2-DC. In addition, we have to consider the following.

8.4.2. Transfinitely iterated (Π^1_1-CA)

The explanation of the systems $(\Pi^1_1\text{-}CA)_\alpha$ and $(\Pi^1_1\text{-}CA)_{<\alpha}$ is analogous to that for the systems of iterated arithmetical comprehension. These formalize the existence of relative hierarchies based on iteration of the hyperjump operator, definable by E_1. Then in analogy to Theorem 8.2.9, we easily obtain the following chains of inclusions:

8.4.3. Theorem.

1. $(\Pi^1_1\text{-}CA)_{<\varepsilon_0} \subseteq \Delta^1_2\text{-}CA \subseteq \Sigma^1_2\text{-}AC \subseteq \Sigma^1_2\text{-}DC \subseteq PA^\omega + (\mu) + (\mu_1) + (QF\text{-}AC)$

2. $\mathit{Res}\text{-}\Pi^1_1\text{-}CA \subseteq \mathit{Res}\text{-}\Delta^1_2\text{-}CA \subseteq \mathit{Res}\text{-}\Sigma^1_2\text{-}AC$
$\subseteq \mathit{Res}\text{-}\widehat{PA}^\omega + (\mu) + (\mu_1) + (QF\text{-}AC)$[18]

In analogy to 8.3.2, using the ND-interpretation of $PA^\omega + (\mu) + (\mu_1) + (QF\text{-}AC)$ in $T + (\mu) + (\mu_1)$ we can also obtain:

8.4.4. Theorem. *$PA^\omega + (\mu) + (\mu_1) + (QF\text{-}AC)$ is a conservative extension of $(\Pi^1_1\text{-}CA)_{<\varepsilon_0}$ for Π^1_3 statements.*

8.4.5. Corollary. *$\Sigma^1_2\text{-}DC$ is a conservative extension of $(\Pi^1_1\text{-}CA)_{<\varepsilon_0}$ for Π^1_3 statements.*

Similarly we obtain a conservation result for $\mathit{Res}\text{-}\Pi^1_2\text{-}AC$ over $\mathit{Res}\text{-}\Pi^1_1\text{-}CA$. Corollary 8.4.5 is again a result first obtained by Friedman [1970]. For more details of the arguments using the approach sketched here through functional interpretations, see Feferman [1977,section 8].

8.4.6. Autonomously iterated Π^1_1-CA

Just as for the case of autonomously iterated ramified analysis or $\Pi^0_1\text{-}CA$, we may explain what are the autonomous ordinals for iteration of $(\Pi^1_1\text{-}CA)$. A much larger recursive ordinal notation system is needed to determine these, using the so-called Bachmann hierarchies of ordinal functions (subsequently simplified through work of Feferman, Aczel and Buchholz — cf. Schütte [1977,Chapter IX]). We shall not try to describe these here, except to take up the first ordinal of proof-theoretical interest from that hierarchy, the so-called Howard ordinal, in section 9.7 below. We content ourselves instead with the statement of an analogue of Theorem 8.3.8 in the following form.

8.4.7. Theorem. *$\Sigma^1_2\text{-}DC$ + (Bar-Rule) is a conservative extension of autonomously iterated $\Pi^1_1\text{-}CA$ for Π^1_3 statements.*

This can be proved by an extension of the functional interpretation methods indicated above, thus also giving conservation for the finite type system $PA^\omega + (\mu) + (\mu_1) + (QF\text{-}AC) + (\mathit{Bar\text{-}Rule})$ over autonomously iterated $\Pi^1_1\text{-}CA$. Theorem 8.4.7 has also been proved by Herbrand-Gentzen methods with the functionals μ and μ_1 in Feferman and Jäger [1983].

8.4.8. Further results and methodological discussion

Friedman [1970] obtained further analogues to Corollaries 8.3.5 and 8.4.5, of the following form:

[18] Actually, it is known from the Kondô-Addison theorem that we can strengthen two of the inclusions in 1 by $\Delta^1_2\text{-}CA = \Sigma^1_2\text{-}AC = \Sigma^1_2\text{-}DC$.

8.4.9. Theorem. *For $n \geq 2$, Σ^1_{n+1}-DC is a conservative extension of $(\Pi^1_n\text{-}CA)_{<\varepsilon_0}$ for Π^1_4 statements.*

This can be improved to conservation results for finite type extensions of Σ^1_{n+1}-*DC* by means of additional axioms for suitable Skolem functionals. However, those functionals do not occur naturally in practice, unlike the non-constructive minimum operator and the Kleene basis operator, so such results would be somewhat artificial to state. Friedman's own proof of this theorem was by model-theoretic methods, and later Feferman and Sieg [1981] gave a proof by Herbrand-Gentzen methods. Subsequently Feferman and Jäger [1983], using the same methods, obtained the related analogues of Theorem 8.4.7:

8.4.10. Theorem. *For $n \geq 2$, Σ^1_{n+1}-DC + (Bar-Rule) is a conservative extension of autonomously iterated Π^1_n-CA for Π^1_4 statements.*

9. The interpretation of theories of ordinals

9.1. A classical theory of countable tree ordinals

We now extend the methods of the previous section to a finite type theory OR^ω_1 of countable (tree) ordinals including the μ operator, in order to obtain a conservation result over a classical theory ID_1 of one arithmetical inductive definition.[19] This work is compared with that of Howard [1972] on analogous constructive theories; it is an interesting open question how these results may be combined. The work described here is based on an unpublished paper, Feferman [1968a].

The system OR^ω_1, to which the ND-interpretation is to be applied, is a variant of that used in Feferman [1968a]. It extends $PA^\omega + (\mu)$ as follows. We expand the type structure by an additional ground type, denoted Ω. Then types $\sigma, \tau, \ldots$ are generated from the ground types 0 and Ω by closing under $(\sigma \to \tau)$ as before. We have infinitely many variables of each type; we use the letters $\alpha, \beta, \gamma, \ldots, \xi, \eta, \zeta$ for variables of the new type Ω. These are informally understood to range over countable tree ordinals, which are closed under formation of suprema in the sense that if f is of type $0 \to \Omega$ then $\mathrm{Sup}(f) \in \Omega$ and $\mathrm{Sup}(f)$ represents the tree whose immediate subtrees are given by the sequence of values $f(x)$ for x a natural number. Formally, the constants of OR^ω_1 besides those of $PA^\omega + (\mu)$ are as follows:

1. 0_Ω is a constant of type Ω
2. Sup is a constant of type $(0 \to \Omega) \to \Omega$
3. Sup^{-1} is a constant of type $\Omega \to (0 \to \Omega)$
4. For each σ, $\mathrm{R}_{\Omega,\sigma}$ is a constant of type $(\Omega, (0 \to \sigma) \to \sigma), \sigma, \Omega \to \sigma$.

[19]Recent work of Burr and Hartung [n.d.] and Burr [1997] extends these results with an interpretation of $KP\omega$ (essentially a set-theoretic analogue of ID_1) in a theory of primitive recursive *set* functionals of finite type, instead of the ordinal functionals of OR^ω.

We shall omit the subscript 'σ' from the ordinal recursor whenever there is no ambiguity. In the following we shall write α_x for $(\mathrm{Sup}^{-1}(\alpha))(x)$ for x of type 0; for α non-zero, this represents the immediate subtree at position x. Just as we do not assume extensionality for functions, we do not assume extensionality for ordinals, i.e. it does *not* follow from $\forall x\ (\alpha_x = \beta_x)$ that $\alpha = \beta$. Nevertheless we shall assume there is a one-one correspondence between non-zero ordinals and functions of which they are the suprema, so that Sup^{-1} will indeed be the inverse of the Sup operation. This is not necessary, but simplifies some points.

The logic taken for OR_1^ω is full classical quantificational logic in its finite type language. The axioms of OR_1^ω consist of:

1. The axioms of $PA^\omega + (\mu)$ with the induction scheme extended to all formulas of the language
2. $\mathrm{Sup}(f) \neq 0_\Omega \wedge \mathrm{Sup}^{-1}(\mathrm{Sup}(f)) = f$, for f of type $(0 \to \Omega)$
3. $\alpha \neq 0_\Omega \to \mathrm{Sup}(\mathrm{Sup}^{-1}(\alpha)) = \alpha$
4. $(0_\Omega)_x = 0_\Omega$
5. (a) $\mathrm{R}_\Omega(f, a, 0_\Omega) = a$
   (b) $\alpha \neq 0_\Omega \to \mathrm{R}_\Omega(f, a, \alpha) = f(\alpha, \lambda x.\mathrm{R}_\Omega(f, a, \alpha_x))$ for each type σ, where the variable a is of type σ, and the variable f is of type $\Omega, (0 \to \sigma) \to \sigma$
6. $\varphi(0_\Omega) \wedge \forall\alpha\ (\alpha \neq 0_\Omega \wedge \forall x\ \varphi(\alpha_x) \to \varphi(\alpha)) \to \forall\alpha\ \varphi(\alpha)$
7. $(QF\text{-}AC)$

The quantifier-free subsystem of OR_1^ω may be axiomatized as follows.

1′. $T + (\mu)$, with induction rule extended to all quantifier- free formulas of OR_1^ω

2′.-5′. The same as 2-5 in OR_1^ω

6′. The rule of induction on ordinals, i.e. from $\varphi(0_\Omega)$ and $(\alpha \neq 0_\Omega \wedge \forall x\ \varphi(\alpha_x) \to \varphi(\alpha))$ infer $\varphi(\alpha)$ for all quantifier-free φ.

(This last is to be expressed in quantifier-free form using the μ operator.) We denote this system by $T_\Omega + (\mu)$. Just as for Theorem 8.3.1 we readily obtain:

9.1.1. Theorem. *OR_1^ω is ND-interpreted in $T_\Omega + (\mu)$.*

9.2. The classical systems ID_1 of one arithmetical inductive definition

We remind the reader briefly of these relatively familiar systems. Here $\theta(P^+, x)$ denotes a formula in the language of arithmetic with one additional predicate or set symbol P that has only positive occurrences in θ. This determines a monotonic operator from sets P to sets $\{x : \theta(P, x)\}$, which thus has a least fixed point. The theory $ID_1(\theta)$ associated with θ is given by the following axioms expressing that P is such a least fixed point, to the extent possible within the given first-order language:

1. The axioms of PA with induction expanded to include all formulas containing the symbol P.
2. $\theta(P, x) \to P(x)$
3. $\forall x\ (\theta(\psi/P, x) \to \psi(x)) \to \forall x\ (P(x) \to \psi(x))$, for each formula ψ.

Here $\theta(\psi/P, x)$ denotes the result of substituting any occurrence of the form $P(t)$ in θ by $\psi(t/x)$. The logic of $ID_1(\theta)$ is, of course, classical. When referring to any system described in this way, we simply write ID_1.

We will use ID_1^i to denote the corresponding theories based on intuitionistic logic. Here, however, we need to specify how the positivity requirement on θ is to be interpreted. We will say that θ is *weakly positive* if it is positive in the classical sense, and *strongly* (or *strictly*) *positive* if there are no occurrences of P in the antecedent of an implication, where the basic logical connectives are taken to be those of section 2.1. An even more restrictive requirement on θ is that it be an *accessibility* inductive definition, as described in section 9.6; cf. section 9.8 for a discussion.

9.3. Translation of ID_1 into OR_1^ω

The obvious way to carry out the translation of $ID_1(\theta)$ into OR_1^ω is to define the approximations to the least fixed point from below. For this purpose, we need a form of ordinal recursion which defines a function at an ordinal α in terms of its values at all smaller ordinals β. The appropriate less-than relation for tree ordinals is introduced as follows. We use the letter 's' to range over numbers of finite sequences of natural numbers. The number 0 is chosen to code the empty sequence, and the extension of a sequence s by one new term x is denoted $s\hat{}\langle x\rangle$.
For an ordinal α, the predecessor α_s of α "down along s" is defined recursively by:

1. $\alpha_0 = \alpha$
2. $\alpha_{s\hat{}\langle x\rangle} = (\alpha_s)_x$.

Then we define

3. $\beta < \alpha \leftrightarrow \alpha \neq 0_\Omega \wedge \exists s\, (s \neq 0 \wedge \beta = \alpha_s)$.

9.3.1. Lemma.

1. *The $<$ relation between ordinals is transitive.*
2. $\forall \alpha, \beta\, \exists \gamma\, (\alpha < \gamma \wedge \beta < \gamma)$.
3. *("Ω-upper bounds", or Quantifier-free collection) For every quantifier-free formula φ, we have $\forall x\, \exists \beta\, \varphi(x, \beta) \to \exists \alpha\, \forall x\, \exists \beta < \alpha\, \varphi(x, \beta)$.*

9.3.2. Proof. Claim 1 is immediate by definition. For claim 2, define f with $f(0) = \alpha$, $f(x') = \beta$, and take $\gamma = \mathrm{Sup}(f)$. Finally, under the hypothesis of claim 3, we have by $(QF\text{-}AC)$, existence of an f such that $\forall x\ \varphi(x, f(x))$; then take $\alpha = \mathrm{Sup}(f)$. □

Given a function f of type $\Omega \to \sigma$, the restriction of f to α for $\alpha \neq 0_\Omega$ is definable as $\lambda s, x.\, f(\alpha_{s\hat{}\langle x\rangle})$, which we also write both as $\lambda s \neq 0.\, f(\alpha_s)$ and as $\lambda\beta < \alpha.\, f(\beta)$. Then one can derive by use of our recursor R_Ω the following more general form of recursion with values in any type σ, given any a of type σ and G of type $\Omega, (0 \to \sigma) \to \sigma$:

$(<\text{-}Rec_\Omega)$ $\qquad F(0_\Omega) = a, \text{ and for } \alpha \neq 0_\Omega,\ F(\alpha) = G(\alpha, \lambda\beta < \alpha.\, F(\beta)).$

Correspondingly we can derive the following more general form of induction on Ω:

$$(<\text{-}Ind_\Omega) \qquad \varphi(0_\Omega) \wedge \forall\alpha\,(\alpha \neq 0_\Omega \wedge \forall\beta < \alpha\,\varphi(\beta) \to \varphi(\alpha)) \to \forall\alpha\,\varphi(\alpha)$$

9.3.3. Lemma. *Given arithmetical $\theta(P,x)$, we can define a function F of type $\Omega, 0 \to 0$ in OR_1^ω satisfying: $F(\alpha,x) = 0 \leftrightarrow \theta(\{y \mid \exists\beta < \alpha\; F(\beta,y) = 0\}, x)$ for all α.*

The proof of this depends essentially on the use of μ to eliminate all the quantifiers in θ and to eliminate the quantifier '$\exists\beta < \alpha$', which is just a quantifier over non-zero sequence numbers, so that we can then apply the principle ($<$-Rec_Ω) above.

Now suppose that $\theta(P,x)$ has just positive occurrences of P. Using the function F from the preceding lemma, define

$$P(x) \leftrightarrow \exists\alpha\; P(\alpha,x), \text{ where } P(\alpha,x) \leftrightarrow F(\alpha,x) = 0, \tag{6}$$

so that

$$P(\alpha,x) \leftrightarrow \theta(\{y \mid \exists\beta < \alpha\; P(\beta,y)\}, x), \text{ for all } \alpha. \tag{7}$$

9.3.4. Theorem. *The predicate P thus defined provably satisfies the axioms of $ID_1(\theta)$ in OR_1^ω.*

9.3.5. Proof. First, to show $\theta(P,x) \to P(x)$, we must show

$$\theta(\{y \mid \exists\beta\; P(\beta,y)\}, x) \to \exists\alpha\; P(\alpha,x).$$

Using the positivity of P in θ, we may bring the hypothesis of this implication to prenex normal form

$$Q_1 z_1 \ldots Q_n z_n\, \exists\beta_1\; \ldots \exists\beta_m\; \theta_0(x, z_1, \ldots, z_n, \beta_1, \ldots, \beta_m),$$

where Q_i is $\forall$ or $\exists$ and θ_0 is quantifier-free. Then by successive application of parts 2 and 3 of Lemma 9.3.1 we obtain

$$\exists\alpha\; Q_1 z_1 \ldots Q_n z_n\, \exists\beta_1 < \alpha\; \ldots \exists\beta_m < \alpha\; \theta_0(x, z_1, \ldots, z_n, \beta_1, \ldots, \beta_m),$$

which is equivalent back to

$$\exists\alpha\; \theta(\{y \mid \exists\beta < \alpha\; P(\beta,y)\}, x).$$

Hence by (6) and (7) we may conclude $\exists\alpha\; P(\alpha,x)$, i.e. $P(x)$.

Next, to show

$$\forall x\,(\theta(\psi/P, x) \to \psi(x)) \to \forall x\,(P(x) \to \psi(x)),$$

we simply prove by use of the principle ($<$-Ind_Ω) that, under the hypothesis, we have $\forall\alpha\,\forall x\,(P(\alpha,x) \to \psi(x))$. This proceeds as usual. □

9.4. Models of $T_\Omega + (\mu)$

We can build models of this system which parallel those of T in section 4.1. First of all, a full, extensional, set-theoretical model $\langle M_\sigma \rangle$ is definable as follows:

1. $M_0 = \mathbb{N}$
2. M_Ω is the smallest set X which contains 0 and which is such that whenever $f : N \to X$, then $\langle 1, f \rangle \in X$.
3. $M_{\sigma\to\tau} = \{f \mid f \text{ maps } M_\sigma \text{ into } M_\tau \}$.

Then we define

4. $0_\Omega = 0$, $\mathrm{Sup}(f) = \langle 1, f \rangle$ for $f : N \to M_\Omega$, and $\mathrm{Sup}^{-1}(\langle 1, f \rangle) = f$.

So for $\alpha = \mathrm{Sup}(f)$ we have $\alpha_x = f(x)$. With each $\alpha \in M_\Omega$ is associated an ordinal $|\alpha|$, in the usual set-theoretical sense, by

5. $|0| = 0$ and $|\mathrm{Sup}(f)| = \sup\{|f(x)| + 1 \mid x \in N\}$.

Then $|\alpha_x| < |\alpha|$ whenever $\alpha \neq 0$, and thus R_Ω is definable by classical ordinal recursion so as to satisfy axiom 5 of $T_\Omega + (\mu)$.

For an intensional recursion-theoretic model analogous to HRO we make use of recursion in the μ operator as a type 2 functional, which is a special case of the Kleene [1959b] development of recursion in finite type objects. In the following we shall use $f, g, h, \ldots$ to range over $\mathbb{N}$ and write $f(x_1, \ldots, x_n)$ for Kleene's $\{f\}(\mu, x_1, \ldots, x_n)$ whenever it is defined. Now the model $\langle N_\sigma \rangle$ is defined by

1. $N_0 = \mathbb{N}$
2. N_Ω = the smallest set $X \subseteq \mathbb{N}$ such that $0 \in X$ and whenever $f : N \to X$ then $\langle 1, f \rangle \in X$ (where the pairing function $\langle x, y \rangle$ is assumed to be from $\mathbb{N}^2$ to $\mathbb{N} - \{0\}$).
3. $N_{\sigma\to\tau} = \{f \mid \forall x\, (x \in N_\sigma \to f(x) \in N_\tau)\}$.

Then, just as above, we take

4. $0_\Omega = 0$, $\mathrm{Sup}(f) = \langle 1, f \rangle$ for $f \in N_{0\to\Omega}$, and $\mathrm{Sup}^{-1}(\langle 1, f \rangle) = f$.

Thus, again, $\alpha_x = f(x)$ for $\alpha = \mathrm{Sup}(f)$, and we may assign set-theoretical ordinals $|\alpha|$ to members α of N_Ω by the same definition as in 5 above. Finally, we may define the interpretations of the ordinal recursors $\mathrm{R}_{\Omega,\sigma}$ as functionals partial recursive in μ for each type σ, by means of the recursion theorem, which is used to produce numbers r satisfying

$$r(f, a, 0) = a, \quad \text{and} \quad r(f, a, (1, g)) \simeq f(g, \lambda x.r(f, a, g(x))).$$

Then induction on $|\alpha|$ shows that $r(f, a, \alpha)$ is defined on the objects f, a of appropriate type, for all $\alpha \in N_\Omega$.

Finally, one can define an analogue of the hereditarily extensional recursion-theoretic HRE model by first defining the notion $\alpha =_\Omega \beta$ inductively, and then defining $=_{\sigma\to\tau}$ in terms of $=_\sigma$ and $=_\tau$ as for HRE, with the objects at each type being those which preserve the defined equality relations at each type.

9.5. Interpreting OR_1^ω in ID_1

We can interpret OR_1^ω in $ID_1(O)$, where O is the set of Church-Kleene constructive ordinal notations, by first applying the ND-interpretation of OR_1^ω in its quantifier-free subtheory $T_\Omega + (\mu)$ and then formalizing the HRO model of the latter in ID_1. For this, though, instead of making use of Kleene's definition of recursion in μ as a special case of recursion in finite type objects, one takes a more concrete version which is possible from the fact that the partial-recursive-in-μ functions are exactly the Π_1^1 partial functions; those can be enumerated by uniformizing the Π_1^1 relations in a standard enumeration. Since O is a complete Π_1^1 predicate, this generalized recursion theory can be formalized in $ID_1(O)$ and, further, the definitions of the N_σ can be given for each σ in that theory. The resulting interpretation preserves arithmetical statements, and is such that with each closed term t of type Ω is associated a number n_t for which $ID_1(O)$ proves $n_t \in O$, and such that $|t| \leq |n_t|$. It follows that the provable ordinals of OR_1^ω are the same as those of $ID_1(O)$, and that is the same as its proof-theoretic ordinal. The latter will be described in section 9.8.

9.6. Functional interpretation of a constructive theory of countable tree ordinals

Howard [1972] introduced a first-order theory U that he called a system of abstract constructive ordinals, with just two sorts of variables, numbers and ordinals. In place of $(QF\text{-}AC)$ this made use of the principle of ω-upper bounds (Lemma 9.3.1 above, clause 3). Howard's system can be translated directly into a finite type theory U^ω which is the same as OR_1^ω, except that we omit the μ operator, and base it on intuitionistic logic in place of classical logic. Then U^ω has a D-interpretation in the system T_Ω of section 9.1, again without the μ operator; that system is just another version of Howard's quantifier-free finite-type theory V of abstract constructive (tree) ordinals op. cit. Howard gave a term model of V which is an extension of Tait's term model of T using infinitely long terms (cf. section 4.4), and used that to obtain an upper bound to the proof-theoretic ordinal of U as the least ordinal greater than $|t|$ for t a normal term of type Ω. The so-called *(Bachmann-)Howard ordinal* thus obtained will be described in the next section. Howard also showed how to translate intuitionistic $ID_1^i(\theta)$ for θ of a certain special form into his system U. This special form includes "accessibility" inductive definitions, where $\theta(P, x)$ is of the form $\forall y\ (y <_t x \to P(y))$ and t is a closed term which determines $y <_t x$ by definition as $t(x, y) = 0$. In the final part of Howard [1972] it was shown how to use work of Gerber [1970] to establish transfinite induction up to each ordinal smaller than the Howard ordinal in such a theory of an accessibility inductive definition. Thus the Howard ordinal is the proof-theoretic ordinal of U and is a least upper bound for the ordinals of all systems $ID_1^i(\theta)$ of the special form considered by Howard.

9.7. The Howard ordinal

The original description of this ordinal made use of an extension of the Veblen hierarchy of ordinal functions in a form due to Bachmann [1950], which gives sense to φ_α for suitable uncountable α. For the specific purposes of 9.6, it is sufficient to tell how this is to be done for α up to the first epsilon number greater than the least uncountable ordinal ω_1, namely ε_{ω_1+1}.[20] Roughly speaking one first assigns to each term α for a limit ordinal in a notation system for ordinals up to ε_{ω_1+1} a fundamental sequence of order type $\leq \omega_1$ in a reasonably canonical way. If the cofinality type of this sequence is countable, one proceeds to define φ_α in terms of the simultaneous fixed points of the φ_β for the terms $\beta = \alpha_\nu$ in that fundamental sequence as with the Veblen hierarchy; if its fundamental sequence is of length ω_1 then one diagonalizes, i.e. takes $\varphi_\alpha(\nu)$ to be $\varphi_{\alpha_\nu}(0)$. The Howard ordinal is then defined to be $\varphi_\alpha(0)$ for $\alpha = \varepsilon_{\omega_1+1}$.[21]

9.8. Discussion

From a foundational standpoint, it is desirable to have a reduction of classical ID_1 to a constructive theory. Although a slight variant of the double-negation interpretation serves to reduce ID_1 to its formally intuitionistic counterpart ID_1^i (cf. Buchholz et al. [1981,p. 56]), the latter theory is not evidently constructive, in the sense that there is no direct constructive justification of axioms 2 and 3 of section 9.2 for θ in which the predicate symbol P occurs only in a weakly positive way. (An indirect justification is provided by the intuitionistic theory of species — i.e. the formal counterpart of second-order analysis — whose constructivity is, however, a matter of dispute; cf. the discussion in Feferman [1982b,pp. 77–78].) What we really desire is a reduction of ID_1 to an intuitionistic theory of accessibility inductive definitions, whose very form provides a clear picture of how the corresponding sets are generated from the "bottom up"; or, alternatively, a reduction of ID_1 to the constructive theory U^ω without the μ operator.

In fact, the first type of reduction has been given by work of Pohlers and Buchholz in a series of steps beginning in the mid-1970s using interesting (prima-facie) uncountably infinitary extensions of Gentzen-Schütte style proof theory; cf. the reports in Buchholz et al. [1981]. Indeed, they have succeeded in determining the proof-theoretic ordinals of classical theories of iterated inductive definitions ID_α and in reducing them to corresponding intuitionistic theories of iterated accessibility inductive definitions, thus showing the proof-theoretic ordinals to be the same. In particular, in the special case of ID_1, the result of their work yields the Howard ordinal as the proof-theoretic ordinal of the system whether taken in its classical or restricted intuitionistic form.

[20] Here, ordinals are treated set-theoretically.

[21] The Bachmann approach has since been superseded by the Feferman-Aczel-Buchholz approach described in Schütte [1977,Chapter IX]: the latter is simpler in not requiring a prior assignment of fundamental sequences.

We do not, however, know how to achieve this same result via the method of functional interpretation, nor do we have a direct reduction of ID_1 to U^ω. In addition, no one has yet extended the method of functional interpretation to iterated ID_α, either classical or intuitionistic, in an informative way specific to those systems.[22] It would be of interest to know whether there is some fundamental methodological obstacle for doing so, or if it is simply for lack of a new idea — or simply lack of trying hard enough.

10. Interpretations based on polymorphism

10.1. Transfinite types and polymorphism

In this section we address strengthenings of T that provide mechanisms for defining "transfinite" types. For example, recall the types (n) from section 2.2, defined by $(0) = 0$ and $(n+1) = (n) \to 0$. One might want to define a function f that, for each natural number n, returns an object of type (n). Such a function f is an element of the product type

$$\Pi_{n \in 0}\ (n),$$

and clearly goes beyond the finite-type capabilities of T. The function f is also "polymorphic" in the sense that, for each n, the type of $f(n)$ depends on its argument.

A down-to-earth example of polymorphism arises in the context of writing a sorting algorithm. Instead of writing separate routines that sort lists of integers, real numbers, strings, and so on, one would prefer to write a general routine that, given a type X and a comparison function in $X \times X \to 0$, sorts lists of objects of type X. Assuming such lists are represented by the type $\mathrm{List}(X)$, for each type X we want $\mathrm{Sort}(X)$ to have the type

$$(X \times X \to 0) \times \mathrm{List}(X) \to \mathrm{List}(X).$$

The type of the function Sort itself can then be written

$$\Pi_X\ ((X \times X \to 0) \times \mathrm{List}(X) \to \mathrm{List}(X))$$

where the product Π_X now ranges over (some collection of) types.

In this section we will consider two different kinds of polymorphism. In the first, the variable X in the preceding example is allowed to range over *all* types, include the type of Sort itself. This scheme is known as *impredicative polymorphism* and was discovered independently by J.-Y. Girard, who was looking for a D-interpretation for second-order arithmetic (cf. Girard [1971]), and J. Reynolds, who was exploring

[22] For α a provably recursive ordinal of analysis we can, in principle, treat ID_α as a subsystem of analysis and then apply Spector's interpretation (section 6) or Girard's interpretation (section 10). But it would appear very difficult to extract meaningful descriptions of the associated proof-theoretic ordinals and reductions to the corresponding intuitionistic systems via those interpretations. At any rate this does not look at all like a practical possibility.

type-theoretic constructs from a computational point of view (cf. Reynolds [1974]). Given these independent motivations, Girard's main theorem is quite satisfying: the provably total recursive functions of second-order arithmetic are exactly the ones computable in the Girard-Reynolds framework.

The circularity of allowing the variable X in a type $\Pi_X\ T(X)$ to range over all types, including $\Pi_X\ T(X)$ itself, might seem disconcerting. *Predicative polymorphism* is more benign in that regard, since in this framework the variable X is restricted to range over a pre-set universe of "smaller" types. For example, in the type

$$\Pi_{X \in U_0}\ T(X)$$

the variable X takes values in the fixed universe U_0. This kind of polymorphism was developed by Martin-Löf as a framework for constructive mathematics (cf. Martin-Löf [1973]), and has been implemented in the underlying language of the Nuprl proof-development system (cf. Chapter X or Constable et al. [1986]). Once again, there is a result that nicely characterizes the axiomatic strength of this kind of polymorphism: the provably total recursive functions of predicative analysis (cf. section 8.3.6) are exactly the ones that can be defined using nested universes of types.

Here we will focus on the interpretative strength of polymorphism. For a detailed discussion of the various computational aspects of the subject, we refer the reader to Mitchell [1990] and Gallier [1990].

10.2. The second-order polymorphic lambda calculus, F

We now define Girard's theory F, a polymorphic extension of T strong enough to interpret second-order arithmetic. This allows for terms which can be applied to (terms representing) types, and abstraction across types. It is simpler here to take abstraction as basic operators rather than defined in terms of combinators. The types of F are defined inductively, as follows:

1. There are infinitely many type variables X, Y, Z, ...
2. 0 is a type.
3. If σ and τ are types, so are $\sigma \to \tau$ and $\sigma \times \tau$.
4. If σ is a type and X is a type variable, then $\Pi_X\ \sigma$ and $\Sigma_X\ \sigma$ are also types.

A term of type $\Pi_X\ \sigma$ denotes a polymorphic function that, for each type τ, returns an object of type $\sigma[\tau/X]$. A term of type $\Sigma_X\ \sigma$ denotes a pair $\langle \tau, t \rangle$, where τ is a type and t is an object of type $\sigma[\tau/X]$.

The terms of F are also defined inductively, as follows:

1. For every type σ there are infinitely many variables, x^σ, y^σ, z^σ, ...
2. 0, Sc, the recursors R_σ, pairing operators $\langle \cdot, \cdot \rangle$, and projection operators π_0, π_1 are terms of appropriate type.
3. If t is a term of type τ and x is a variable of type σ, then $\lambda x.t$ is a term of type $\sigma \to \tau$.
4. If t is a term of type $\sigma \to \tau$ and s is a term of type σ, then $t(s)$ is of type τ.

5. If t is a term of type τ, and the type variable X is not free in the type of a free variable of t, then $\Lambda X.t$ is a term of type $\Pi_X\ \tau$.
6. If t is a term of type $\Pi_X\ \tau$ and σ is a type, then $t(\sigma)$ is a term of type $\tau[\sigma/X]$.
7. If σ is a type and t is a term of type $\tau[\sigma/X]$, then $\langle\sigma, t\rangle$ is a term of type $\Sigma_X\ \tau$.
8. If $t[x^\tau]$ is a term of type σ and the type variable X is only free in the type of x^τ, then
$$\nabla\langle X, x^\tau\rangle.t$$
is a term of type $\Sigma_X\ \tau \to \sigma$.

While clauses (1–4) are essentially imported from T, clauses (5–8) provide the new polymorphic capabilities. The choice of lambda terms instead of combinators in clause (3) conforms with the majority of the literature on this subject.

The term $\Lambda X.t$ in clause (6) denotes a function that associates to every type σ an object of type $\tau[\sigma/X]$, with defining equation

$$(\Lambda X.t)(\sigma) = t[\sigma/X].$$

The requirement that X is not free in the type of any free variable of t precludes terms like
$$\Lambda X.x^X,$$
in which the free variable x cannot be assigned any reasonable type. On the other hand, it does allow constructions such as the polymorphic identity function

$$\Lambda X.\lambda x^X.x^X$$

and the Sort function defined in section 10.1. Strictly speaking, the application operation $(\Lambda X.t)(\sigma)$ in clause (6) is different from the application operation $(\lambda x.t)(s)$ in clause (4), though we will use the same notation.

The function $\nabla\langle X, x^\tau\rangle.t$ in clause (8) is a bit trickier: it takes a pair $\langle\sigma, s\rangle$, for which s is of type $\tau[\sigma/X]$, and returns the value of t with X replaced by σ and $x^{\tau[\sigma/X]}$ replaced by s. In other words, this term has the defining equation

$$(\nabla\langle X, x\rangle.t)\langle\sigma, s\rangle = t[\sigma/X][s/x^{\tau[\sigma/X]}].$$

As usual, the variable X becomes bound in clause (6), and both variables X and x become bound in clause (8).

For technical reasons the interpretation in the next section also requires a constant zero functional 0_σ of each type. We omit the rules defining its behavior, though they can be found in Girard [1971].

10.3. The interpretation of analysis

Whereas in Spector's interpretation of analysis the second-order objects of $PA^2 + (CA)$ become identified with function variables, in Girard's interpretation it is more natural to treat them as predicates. (Moreover, the interpretation extends to a higher-type version of this system, with predicates instead of functions at each level.) As in the interpretation of PA, we first apply the double-negation interpretation to reduce $PA^2 + (CA)$ to its intuitionistic variant, $HA^2 + (CA)$. Unlike the case of (AC) in Spector's interpretation, $(CA)^N$ follows directly from (CA) in HA^2.

The D-interpretation we are about to define translates formulas φ in the language of second-order arithmetic to formulas φ^D of the form

$$\exists x^\sigma \; \forall y^\tau \; F_\varphi(x,y) = 0 \tag{8}$$

where φ_D is quantifier-free with only type 0 equality. Adding dummy quantifiers as necessary and using the pairing and projection operations to combine quantified variables, we can assume that there is always exactly one variable after each quantifier.

The first new step needed in defining φ^D beyond first-order formulas is to find a suitable translation for formulas of the form $t \in Z$. Since we are aiming to interpret the comprehension axioms, we want the variable Z to "range" over arbitrary formulas φ. We define $(t \in Z)^D$ to be

$$\exists x^X \; \forall y^Y \; f(x,y,t) = 0 \tag{9}$$

where now X and Y are type variables and f is a function variable of type

$$X \times Y \times 0 \to 0.$$

Intuitively, $(t \in Z)^D$ represents an "arbitrary" formula of the form (8).

The translation is extended to the logical connectives and first-order quantifiers as before. To define $(\exists Z \; \varphi(Z))^D$, suppose $\varphi(Z)^D$ is given by

$$\exists x^{\sigma[X,Y]} \; \forall y^{\tau[X,Y]} \; F(x,y,f) = 0$$

where X, Y, and f are the variables we have associated with Z in the preceding paragraph. The formula $\exists Z \; \varphi(Z)$ should then translate to

$$\exists X, Y, f, x^{\sigma[X,Y]} \; \forall y^{\tau[X,Y]} \; F(x,y,f) = 0.$$

But now the type of y depends on the existentially quantified types X and Y, which is problematic. We remedy this by replacing y with an element of the appropriate product type to obtain

$$\exists X, Y, f, x^{\sigma[X,Y]} \; \forall y^{\Pi_{X,Y} \; \tau[X,Y]} \; F(x, y(X,Y), f) = 0.$$

Finally, we replace the existentially quantified variables X, Y, and f by a single variable u of type $\Sigma_X \; \Sigma_Y \; (X \times Y \times 0 \to 0)$. Then u may be paired with x to put $(\exists Z \; \varphi(Z))^D$ again in the form (8).

Similar manipulations are used to define the translation of the formula $\forall Z \; \varphi(Z)$. When this is done, the interpretation of HA^2 is verified just as in section 2.4. The comprehension axioms (CA) are also easily dealt with, as a consequence of the translation we have chosen for $x \in Z$. Details can be found in Girard [1971].[23]

This yields

[23] There, and in the literature, F is often used more specifically to denote the set of *reduction*

10.3.1. Theorem. *$PA^2 + (CA)$ is ND-interpreted in F.*

10.4. Strong normalization for F

In his doctoral dissertation Girard [1972] presented a powerful generalization of Tait's convertibility methods (cf. section 4.3), and used it to show that F is strongly normalizing.

Since the normalization of F implies the consistency of second-order arithmetic, the full argument cannot be formalized in that theory. To appreciate the difficulties that arise in trying to adapt the argument of section 4.3, consider the problem of extending the reducibility predicate defined there to a term t of type $\Pi_X\ \tau$. We would like to say that t is reducible if for every type σ the term $t(\sigma)$ is reducible. But σ is arbitrary, and could very well be the type $\Pi_X\ \tau$ itself. In that case, determining the reducibility of $t(\sigma)$ will require some knowledge of what it means for terms of type $\Pi_X\ \tau$ to be reducible— which is exactly the notion we are trying to define.

Girard's clever dodge is to define the notion of a "reducibility candidate," which is a predicate of terms which satisfies certain closure conditions. The reducibility predicate of section 4.3 is an example of a reducibility candidate, and, in fact, it is just these closure conditions that allow one to carry out the necessary induction on terms. Girard then declared $t \in \Pi_X\ \tau$ to be reducible if for *every* reducibility candidate C and every type σ, $t(\sigma)$ is reducible of type $\tau(\sigma)$, where now reducibility for $t(\sigma)$ is defined in terms of C.

Since the definition of reducibility for terms of type $\Pi_X\ \tau$ involves prefixing a second-order quantifier to a formula involving the definition of reducibility for terms of type τ, for arbitrary polymorphic types the definition requires second-order formulas of arbitrary complexity. The net result is the following

10.4.1. Theorem. *For each term t of F, $PA^2 + (CA)$ proves that t is strongly normalizing. In addition, $PA^2 + (CA)$ proves the confluence of F.*

Thus F can be interpreted in $PA^2 + (CA)$ via its model in the normal terms. This yields the following

10.4.2. Theorem. *The provably total recursive functions of $PA^2 + (CA)$ are exactly the ones that are represented by terms of F.*

The method of reducibility candidates extends to stronger functional theories, including a typed extension of F also introduced in Girard's dissertation, and Co-

---

rules corresponding to the defining equations of the theory described above; we have chosen to blur this distinction. A more minimal version of F which omits the base type 0 and the sum type — essentially the system $\lambda^{\to,\forall}$ of Mitchell [1990], Gallier [1990] — is discussed in Girard, Lafont and Taylor [1989].

The theory Y of Girard [1971] is essentially a logic-free version of our theory F. More precisely, if F proves a quantifier-free formula φ involving only type 0 equality, then Y proves the logic-free translation of φ when free variables are instantiated by any closed terms.

quand's "Calculus of Constructions." In fact, by identifying natural deductions with terms of an appropriate functional calculus, Girard [1971] was also able to use these methods to give a new proof of "Takeuti's conjecture," an extension of Gentzen's Hauptsatz to higher-order deductive sytems; this was realized independently by Martin-Löf [1971] and Prawitz [1971]. See Gallier [1990], Coquand [1990], Girard, Lafont and Taylor [1989] for more details.

10.5. Theories based on predicative polymorphism

A more restrictive, "predicative," method of extending the theory T polymorphically is represented by a sequence of theories P_n, based on Martin-Löf's theories ML_n. The theory P_0 is equivalent to T, while the theory P_{n+1} adds $n+1$ "universes" of types, U_0 through U_n, each of which will itself be a type.

Like the theory F, the theories P_n have product and sum types, but rather than taking products over *types*, one takes products over *terms* of a given type. That is, one has the following type formation rules:

1. 0 is a type.
2. If σ and $\tau[x^\sigma]$ are types, then so are $\Pi_{x\in\sigma}\,\tau$ and $\Sigma_{x\in\sigma}\,\tau$.

Terms of type $\Pi_{x\in\sigma}\,\tau$ denote functions f which take elements a of type σ to elements $f(a)$ of type $\tau[a/x]$, and terms of type $\Sigma_{x\in\sigma}\,\tau$ denote pairs $\langle a, b\rangle$, where a is an object of type σ and b is an object of type $\tau[a/x]$. The $\to$ and $\times$ constructions can be seen as special cases of the Π and Σ constructions respectively, in which the type τ doesn't depend on the variable x^σ.

One has to modify the usual formation rules for terms to accommodate these new dependent types. For example, the new rule of explicit definition takes the form

$$\text{If } t[x^\sigma] \text{ is a term of type } \tau[x^\sigma]\text{, then } \lambda x.t \text{ is a term of type } \Pi_{x\in\sigma}\tau.$$

Similar rules take the place of those of T regarding application, pairing, projection, and the recursors.

We haven't yet provided a mechanism for defining types that depend on variables. What makes the systems P_n polymorphic is that *types can be represented by terms*, since elements of type U_i are themselves taken to denote types. For example, P_1 has the following rules:

1. U_0 is a type
2. 0 (the type of the natural numbers) is an element of U_0
3. If σ and $\tau[x^\sigma]$ are elements of U_0, then so are $\Pi_{x\in\sigma}\tau$ and $\Sigma_{x\in\sigma}\tau$.
4. If t is a term of type U_0, then t is a type.

The last clause places the theory P_1 in sharp distinction to T, where there is no interplay between terms and types. In P_1, one can define a term t of type U_0, conclude that t is a type, and then define another term s of type t. For example, the types (n) of section 10.1 can be defined by a simple instance of primitive recursion with range U_0, whereupon the term $\Pi_{n\in 0}(n)$ is an element of U_0 and hence a type. In general, each theory P_n has n universes $U_0, \ldots, U_{n-1}$, each of which contains terms denoting all "smaller" universes.

10.6. The interpretation of predicative theories of analysis

The theory $\widehat{ID}_1$ is a weakening of the theory ID_1 defined in section 9.2. Here, rather than assert that P is a *least* fixed point of the positive arithmetic operator given by θ, one simply asserts that P is *some* fixed point; that is, one replaces 2 and 3 from section 9.2 by the single axiom

$$\forall x\ (\theta(P,x) \leftrightarrow P(x)).$$

In general, each theory $\widehat{ID}_n$ allows n nested inductive definitions, by allowing one to use any predicate of $\widehat{ID}_i$ to define a fixed-point in $\widehat{ID}_{i+1}$. Also we set

$$\widehat{ID}_{<\omega} = \bigcup_{n<\omega} \widehat{ID}_n.$$

It is known (cf. Feferman [1982a], Friedman, McAloon and Simpson [1982], Avigad [1996b]) that $\widehat{ID}_{<\omega}$ has strength Γ_0, and hence proves the same arithmetic formulas as predicative analysis (cf. the discussion preceding Theorem 8.3.8) as well as an important second-order theory known as ATR_0.[24] The following theorem is due to Avigad [n.d.].

10.6.1. Theorem. *Each theory $\widehat{ID}_n$ is ND-interpreted in P_n.*[25]

The interpretation is somewhat tortuous, so we only provide a rough outline here. The first reduction relies on the following lemma, due to P. Aczel (cf. Feferman [1982a]).

10.6.2. Lemma. *$\widehat{ID}_1$ is interpretable in Σ^1_1-AC. More generally, each theory $\widehat{ID}_{n+1}$ is interpretable in Σ^1_1-$AC(\widehat{ID}_n)$.*

Here the theory Σ^1_1-$AC(\widehat{ID}_n)$ is a second-order version of $\widehat{ID}_n$ together with a choice scheme for Σ^1_1 formulas in the expanded language. Given any arithmetic (or even Σ^1_1) formula $\varphi(x,Y)$ in which the predicate Y occurs positively, one can obtain a Σ^1_1 formula ψ defining a fixed-point of the corresponding set operation. (The proof is similar to that of Gödel's fixed-point lemma, while the scheme (Σ^1_1-AC) is needed to show that that formula ψ works as advertised.) One then uses these formulas ψ to interpret the fixed-point constants of $\widehat{ID}_{n+1}$.

By Theorem 8.3.1, one can interpret Σ^1_1-AC in the theory $T + (\mu)$, in which formulas in the language of Peano arithmetic translate to formulas that are quantifier-free. We can instead interpret Σ^1_1-AC in P_1 if we interpret formulas $t \in Z$ by equation (9) of section 10.3, where now the type variables range over the universe U_0 instead of all types. Iterating this idea leads to Theorem 10.6.1.

One can show (cf. Coquand [n.d.], Martin-Löf [1973]) that terms of $P_{<\omega}$ $(= \bigcup P_n)$ are normalizing. Since each $\widehat{ID}_\omega$ can define a recursion-theoretic model for P_n, we obtain

[24] The first functional interpretation of predicative ramified analysis was given by Maass [1976] via a specialization of Girard's interpretation described in section 10.3 above.

[25] The theories P_n of Avigad [n.d.] are logic-free subsystems of the theories P_n defined here.

10.6.3. Corollary. *The provably total recursive functions of each $\widehat{ID}_n$ are exactly the ones represented by terms of P_n. The provably total recursive functions of $\widehat{ID}_{<\omega}$, and hence predicative analysis and ATR_0, are exactly the ones represented by terms of $P_{<\omega}$.*

10.7. Final comments and questions

The functional interpretation of a classical theory provides a nice interplay between logic and theoretical computer science. Given a set of classical axioms, the corresponding computational schemata provide a constructive understanding of their strength, and normalization proofs provide evidence that these abstract axioms have interesting computational consequences. Conversely, given some computational schemata, the calibration of their classical axiomatic strength helps round out our understanding of their capabilities.

Martin-Löf has described extensions of the theories ML_n with "elimination rules" for the universes, yielding the same proof-theoretic strength as the impredicative theories ID_n (cf. Griffor and Rathjen [1994,Theorem 4.14]). Can such elimination rules be used to give the theories ID_n a direct functional interpretation?

The interpretations of Spector and Girard show that bar-recursion and impredicative polymorphism each exactly capture the provably total recursive functions of second-order arithmetic. Are there natural characterizations for interesting fragments and extensions of the latter theory, other than the ones we have already described in this chapter?

References

J. Avigad

[n.d.] *Predicative functionals and an interpretation of $\widehat{ID}_{<\omega}$*. To appear in the *Annals of Pure and Applied Logic*.

[1996a] Formalizing forcing arguments in subsystems of second-order arithmetic, *Annals of Pure and Applied Logic*, 82, pp. 165–191.

[1996b] On the relationship between ATR_0 and $\widehat{ID}_{<\omega}$, *Journal of Symbolic Logic*, 61, pp. 768–779.

H. Bachmann

[1950] Die Normalfunktionen und das Problem der ausgezeichneten Folgen von Ordnungzahlen, *Vierteljahresschr. Nat. Ges., Zürich*, 95, pp. 5–37.

J. Barwise

[1977] ed., *The Handbook of Mathematical Logic*, North-Holland, Amsterdam.

J. Barwise and J. S. Schlipf

[1975] On recursively saturated models of arithmetic, in: *Model theory and algebra: a memorial tribute to Abraham Robinson*, D. Saracino and V. B. Weispfennig, eds., Lecture Notes in Mathematics #498, Springer-Verlag, Berlin, pp. 42–55.

M. J. Beeson

[1978] A type-free Gödel interpretation, *Journal of Symbolic Logic*, 43, pp. 213–227.

[1982] Recursive models for constructive set theories, *Annals of Mathematical Logic*, 23, pp. 127–178.

[1985] *Foundations of Constructive Mathematics*, Springer-Verlag, Berlin.

E. Bishop

[1970] Mathematics as a numerical language, in: Kino, Myhill and Vesley [1970], pp. 53–71.

W. Buchholz, S. Feferman, W. Pohlers, and W. Sieg

[1981] *Iterated Inductive Definitions and Subsystems of Analysis: Recent Proof-Theoretical Studies*, Lecture Notes in Mathematics #897, Springer-Verlag, Berlin.

W. Burr

[1997] *A Diller-Nahm-style functional interpretation of $KP\omega$.* Preliminary draft.

W. Burr and V. Hartung

[n.d.] *A characterization of the Σ_1 definable functions of $KP\omega +$ (uniform AC).* To appear in the *Archive for Mathematical Logic.*

S. R. Buss

[1986] *Bounded Arithmetic*, Bibliopolis, Naples. A reprinting of the author's 1985 Princeton University dissertation.

A. Cantini

[1996] Asymmetric interpretation for bounded theories, *Mathematical Logic Quarterly*, 42, pp. 270–288.

R. L. Constable et al.

[1986] *Implementing Mathematics with the Nuprl Proof Development System*, vol. 37 of Graduate Texts in Mathematics, Prentice-Hall, Englewood Cliffs, NJ.

S. A. Cook and A. Urquhart

[1993] Functional interpretations of feasibly constructive arithmetic, *Annals of Pure and Applied Logic*, 63, pp. 103–200.

C. Coquand

[n.d.] A normalization proof for Martin-Löf's type theory, in: *25 Years of Constructive Type Theory*, G. Sambin and J. M. Smith, eds., Oxford University Press. To appear.

T. Coquand

[1990] Metamathematical investigations of a calculus of constructions, in: Odifreddi [1990], pp. 91–122.

N. Dershowitz and J.-P. Jouannaud

[1990] Rewrite systems, in: van Leeuwen [1990], pp. 243–320.

J. Diller and W. Nahm

[1974] Eine Variante zur Dialectica Interpretation der Heyting Arithmetik endlicher Typen, *Archive für Mathematische Logik und Grundlagenforschung*, 16, pp. 49–66.

S. Feferman

[1964] Systems of predicative analysis, *Journal of Symbolic Logic*, 29, pp. 1–30.

[1968a] *Ordinals associated with theories for one inductively defined set.* Unpublished paper.

[1968b] Systems of predicative analysis II: representations of ordinals, *Journal of Symbolic Logic*, 33, pp. 193–220.

[1971] Ordinals and functionals in proof theory, in: *Proceedings of the International Congress of Mathematicians, Nice*, vol. 1, Gauthier-Villars, Paris, pp. 229–233.

[1977] Theories of finite type related to mathematical practice, in: Barwise [1977], pp. 913–971.

[1979] A more perspicuous formal system for predicativity, in: *Konstruktionen versus Positionen I*, K. Lorenz, ed., de Gruyter, Berlin, pp. 87–139.

[1982a] Iterated inductive fixed-point theories: application to Hancock's conjecture, in: Metakides [1982].

[1982b] Monotone inductive definitions, in: *The L. E. J. Brouwer Centenary Symposium*, A. S. Troelstra and D. van Dalen, eds., North-Holland, Amsterdam, pp. 77–89.

[1990] *Milking the Dialectica interpretation.* Unpublished notes.

[1993] Gödel's Dialectica interpretation and its two-way stretch, in: *Computational Logic and Proof Theory*, G. Gottlob, A. Leitsch, and D. Mundici, eds., Lecture Notes in Computer Science #713, Springer-Verlag, Berlin, pp. 23–40.

S. Feferman and G. Jäger

[1983] Choice principles, the bar rule, and autonomously iterated comprehension schemes in analysis, *Journal of Symbolic Logic*, 48, pp. 63–70.

S. Feferman and W. Sieg

[1981] Proof-theoretic equivalences between classical and constructive theories for analysis, in: Buchholz et al. [1981], pp. 78–142.

J. E. Fenstad

[1971] ed., *Proceedings of the Second Scandinavian Logic Symposium*, North-Holland, Amsterdam.

F. Ferreira

[1988] *Polynomial time computable arithmetic and conservative extensions*, PhD thesis, Pennsylvania State University.

[1994] A feasible theory for analysis, *Journal of Symbolic Logic*, 59, pp. 1001–1011.

H. M. Friedman

[1970] Iterated inductive definitions and $\Sigma^1_2{-}AC$, in: Kino, Myhill and Vesley [1970], pp. 435–442.

[1976] Systems of second-order arithmetic with restricted induction I, II (abstract), *Journal of Symbolic Logic*, 41, pp. 557–559.

H. M. Friedman, K. McAloon, and S. G. Simpson

[1982] A finite combinatorial principle which is equivalent to the 1-consistency of predicative analysis, in: Metakides [1982], pp. 197–230.

W. Friedrich

[1985] Gödelsche Funktionalinterpretation für eine Erweiterung der Klassischen Analysis, *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 31, pp. 3–29.

J. H. Gallier

[1990] On Girard's 'Candidats de Reductibilité', in: Odifreddi [1990], pp. 123–203.

H. Gerber

[1970] Brouwer's bar theorem and a system of ordinal notations, in: Kino, Myhill and Vesley [1970], pp. 327–338.

J.-Y. Girard

[1971] Une extension de l'interpretation de Gödel à l'analyse, et son application à l'élimination des coupures dans l'analyse et dans la théorie des types, in: Fenstad [1971], pp. 63–92.

[1972] *Interprétation fonctionelle et élimination de coupures de l'arithmétique d'ordre supérieur*, PhD thesis, Université de Paris VII.

J.-Y. Girard, Y. Lafont, and P. Taylor

[1989] *Proofs and Types*, Cambridge University Press.

K. Gödel

[1941] In what sense is intuitionistic logic constructive?, in: Gödel [1994], pp. 189–200.

[1958] Über eine bisher noch nicht benüzte Erweiterung des finiten Standpunktes, *Dialectica*, 12, pp. 280–287. Reproduced with English translation in Gödel [1990], pp. 241–251.

[1972] On an extension of finitary methods which has not yet been used, in: Gödel [1994], pp. 271–280.

[1990] *Collected Works*, vol. II, Oxford University Press, New York. Solomon Feferman et al. eds.
[1994] *Collected Works*, vol. III, Oxford University Press, New York. Solomon Feferman et al. eds.

E. Griffor and M. Rathjen
[1994] The strength of some Martin-Löf type theories, *Archive for Mathematical Logic*, 33, pp. 347–385.

P. Hájek
[1993] Interpretability and fragments of arithmetic, in: *Arithmetic, Proof Theory, and Computational Complexity*, P. Clote and J. Krajicek, eds., Oxford University Press, Oxford.

J. van Heijenoort
[1967] *From Frege to Gödel: A sourcebook in mathematical logic, 1879-1931*, Harvard University Press.

A. Heyting
[1959] ed., *Constructivity in Mathematics*, North-Holland, Amsterdam.

D. Hilbert
[1926] Über das Unendliche, *Mathematische Annalen*, 95, pp. 161–190. English translation in van Heijenoort [1967], pp.367–392.

J. R. Hindley and J. P. Seldin
[1986] *Introduction to Combinators and λ-calculus*, Cambridge University Press, Cambridge.

W. A. Howard
[1968] Functional interpretation of bar induction by bar recursion, *Compositio Mathematica*, 20, pp. 107–124.
[1970] Assignment of ordinals to terms for primitive recursive functionals of finite type, in: Kino, Myhill and Vesley [1970], pp. 453–468.
[1972] A system of abstract constructive ordinals, *Journal of Symbolic Logic*, 37, pp. 355–374.
[1973] Hereditarily majorizable functionals of finite type, in: Troelstra [1973], pp. 454–461.

A. Kino, J. Myhill, and R. E. Vesley
[1970] eds., *Intuitionism and Proof Theory*, North-Holland, Amsterdam.

S. C. Kleene
[1959a] Countable functionals, in: Heyting [1959], pp. 81–100.
[1959b] Recursive functionals and quantifiers of finite types, I, *Transactions of the American Mathematics Society*, 91, pp. 1–52.

S. C. Kleene and R. E. Vesley
[1965] *Foundations of Intuitionistic Mathematics*, North-Holland, Amsterdam.

U. W. Kohlenbach
[1992] Effective bounds from ineffective proofs in analysis: an application of functional interpretation and majorization, *Journal of Symbolic Logic*, 57, pp. 1239–1273.
[1993] Effective moduli from ineffective uniqueness proofs: an unwinding of de La Vallée Poussin's proof for Chebycheff approximation, *Annals of Pure and Applied Logic*, 64, pp. 27–94.
[1996a] Analyzing proofs in analysis, in: *Logic: From Foundations to Applications: European Logic Colloquium '93*, W. Hodges et al., eds., Clarendon Press, Oxford, pp. 225–260.
[1996b] Mathematically strong subsystems of analysis with low rate of growth of provably recursive functionals, *Archive for Mathematical Logic*, 36, pp. 31–71.

G. Kreisel
[1951] On the interpretation of non-finitist proofs, part I, *Journal of Symbolic Logic*, 16, pp. 241–267.

[1952] On the interpretation of non-finitist proofs, part II: Interpretation of number theory, applications, *Journal of Symbolic Logic*, 17, pp. 43–58.

[1957] Gödel's interpretation of Heyting's arithmetic, in: *Summaries of talks, Summer Institute for Symbolic Logic, Cornell University*, Institute for Defense Analyses.

[1959] Interpretation of analysis by means of constructive functionals of finite type, in: Heyting [1959], pp. 101–128.

J. van Leeuwen

[1990] ed., *The Handbook of Theoretical Computer Science*, vol. B, MIT Press (Cambridge, MA) / Elsevier (Amsterdam).

H. Luckhardt

[1973] *Extensional Gödel Functional Interpretation*, Lecture Notes in Mathematics #306, Springer-Verlag, Berlin.

W. Maass

[1976] Eine Funktionalinterpretation der predikativen Analysis, *Archiv für Mathematische Logik und Grundlagenforschung*, 18, pp. 27–46.

P. Martin-Löf

[1971] Hauptsatz for the theory of species, in: Fenstad [1971], pp. 217–233.

[1973] An intuitionistic theory of types: predicative part, in: *Logic Colloqium '73*, H. E. Rose and J. C. Shepherdson, eds., North-Holland, Amsterdam.

G. Metakides

[1982] ed., *Patras Logic Symposion*, North-Holland, Amsterdam.

J. C. Mitchell

[1990] Type systems for programming languages, in: van Leeuwen [1990], pp. 365–458.

P. Odifreddi

[1990] ed., *Logic and Computer Science*, Academic Press, London.

C. Parsons

[1970] On a number-theoretic choice scheme and its relation to induction, in: Kino, Myhill and Vesley [1970], pp. 459–473.

[1972] On n-quantifier induction, *Journal of Symbolic Logic*, 37, pp. 466–482.

D. Prawitz

[1971] Ideas and results in proof theory, in: Fenstad [1971], pp. 235–307.

J. C. Reynolds

[1974] Towards a theory of type structure, in: *Programming Symposium*, B. Robinet, ed., Lecture Notes in Computer Science #19, Springer-Verlag, Berlin, pp. 408–425.

K. Schütte

[1960] *Beweistheorie*, Springer-Verlag, Berlin.

[1977] *Proof Theory*, Springer-Verlag, Berlin.

H. Schwichtenberg

[1977] Proof theory: Some aspects of cut-elimination, in: Barwise [1977], pp. 867–895.

J. R. Shoenfield

[1967] *Mathematical Logic*, Addison Wesley, Reading, MA.

W. Sieg

[1985] Fragments of arithmetic, *Annals of Pure and Applied Logic*, 28, pp. 33–72.

W. Sieg and C. Parsons

[1994] Introductory note to 1938a, in: Gödel [1994], pp. 62–85.

S. G. Simpson

[1987] Subsystems of Z_2 and reverse mathematics, in: appendix to Takeuti [1987], pp. 432–446.

C. Spector

[1962] Provably recursive functionals of analysis: a consistency proof of analysis by an extension of principles formulated in current intuitionistic mathematics, in: *Recursive Function Theory: Proc. Symposia in Pure Mathematics*, J. C. E. Dekker, ed., vol. 5, American Mathematical Society, Providence, Rhode Island, pp. 1–27.

W. W. Tait

[1965] Infinitely long terms of transfinite type, in: *Formal Systems and Recursive Functions*, J. N. Crossley and M. A. E. Dummett, eds., North-Holland, Amsterdam, pp. 176–185.

[1967] Intensional interpretations of functionals of finite type, I, *Journal of Symbolic Logic*, 32, pp. 198–212.

[1968] Normal derivability in classical logic, in: *The Syntax and Semantics of Infinitary Logic*, J. Barwise, ed., Lecture Notes in Mathematics #72, Springer-Verlag, Berlin, pp. 204–236.

[1971] Normal form theorem for bar recursive functions of finite type, in: Fenstad [1971], pp. 353–367.

G. Takeuti

[1987] *Proof Theory*, North-Holland, Amsterdam, second ed.

A. S. Troelstra

[1973] *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, Lecture Notes in Mathematics #344, Springer-Verlag, Berlin.

[1977] Aspects of constructive mathematics, in: Barwise [1977], pp. 973–1052.

[1990] Introductory note to 1958 and 1972, in: Gödel [1990], pp. 217–241.

A. S. Troelstra and D. van Dalen

[1988] *Constructivism in Mathematics: An Introduction*, vol. 1, North-Holland, Amsterdam.

H. Weyl

[1918] *Das Kontinuum. Kritische Untersuchungen über die Grundlagen der Analysis*, Veit, Leipzig. Second edition (1932).

[1994] *The Continuum. A Critical Examination of the Foundation of Analysis*, Dover, New York. English translation of Weyl [1918].

Name Index

Subject Index

Discard this page.