Finitary inductively presented logics

by

Solomon Feferman[1]
Departments of Mathematics and Philosophy
Stanford University

Abstract

A notion of *finitary inductively presented (f.i.p.) logic* is proposed here, which includes all syntactically described logics (formal systems) met in practice. A f.i.p. theory FS_0 is set up which is universal for all f.i.p. logics; though formulated as a theory of functions and classes of expressions, FS_0 is a conservative extension of PRA. The aims of this work are (i) conceptual, (ii) pedagogical and (iii) practical. The system FS_0 serves under (i) and (ii) as a theoretical framework for the formalization of metamathematics. The general approach may be used under (iii) for the computer implementation of logics. In all cases, the work aims to make the details manageable in a natural and direct way.

What is a logic? The question here is not "What is logic?", which (tendentiously) seeks to canonize some *one* distinguished system of reasoning as being the only true one. But also, here, we are not after *any* logic—only those that are *syntactically described,* or *formal,* as distinguished from those that are *semantically described.* For the latter, a reasonable general basic notion has evolved, that of *model-theoretic logic*; cf. e.g. Barwise-Feferman [1985]. Curiously (for a subject so devoted to foundational matters), there is no corresponding generally accepted basic notion for the formal logics. Such should cover as special cases propositional and predicate calculi of various kinds (classical, intuitionistic, many-valued, modal, temporal, deontic, relevance, etc.) and styles (Hilbert, Gentzen-natural deduction, Gentzen-sequential, linear, etc.), as well as equational calculi, lambda calculi, combinatory calculi (typed and untyped), and various applied logics (theories of arithmetic, algebraic systems, analysis, types, sets, etc.) and logics of programs.[2]

The first answer usually given is that by a syntactically described logic we mean a *formal system,* i.e. a triple consisting of a *language, axioms* and *rules of inference,* all of these specified by their syntactic form. But what does this last

[1] Reprinted, with some minor corrections and additions, from *Logic Colloquium '88* (R. Ferro, C. Bonotto, S. Valentini, and A. Zanardo, eds.), North-Holland Publishing Co., Amsterdam (1989), pp. 191–220. Permission to reprint granted by Elsevier Science Publishers, B.V.

[2] Thus the approach here is neutral as to the reasons for choosing any particular logic for study, or for choosing one logic in preference to another.

mean? And what is a language anyhow? In practice, languages are themselves systems of interrelated syntactic categories including such notions as sorts, variables, terms, propositional operators, quantifiers, abstraction operators, atomic formulas, formulas, etc. So what do we mean by a *formal language system*?

Some answers to these questions have been proposed, but none so far has gained wide acceptance. Either they are too general, so that lots of things we would not ordinarily call logics are lumped in together and nothing interesting about them can be proved, or they are too specific, so that many logics met in practice are excluded, or they are too abstract, so that many significant particularities of logics in practice are simply ignored, or they are too coercive, so that everything is forced into one uncomfortable mould.

The aim here is to propose a comprehensive definition of a *finitary inductive system* which includes both languages and logics and which covers all examples met in practice, in a reasonably natural and direct way. As is to be expected for work of this nature, many of the ideas will have already been employed in the literature, in one way or another, for this and related purposes. The novelty here lies mainly in the particular combination of ideas, although there are some definitely new concepts which are introduced, the main ones being that of a *presentation* of a finitary inductive system and thence of a *finitary inductively presented logic.*

The most closely related work by others is, on the one hand, that of Smullyan [1961] (following Post [1943]—cf. also Fitting [1987]) and, on another hand, that of Moschovakis [1984], where the question dealt with—"What is an algorithm?"—is answered in terms of a more general class of inductive systems. The direct predecessor of this paper is Feferman [1982], "Inductively presented systems and the formalization of meta-mathematics", which I refer to in the following as [IPS]. What is done below corrects, extends and improves [IPS] in certain respects, but is based on the same leading ideas; I shall refer to it frequently in the following.

There are three main reasons for pursuing our leading question:

(i) *Conceptual.* The topologists have their topological spaces, the algebraists have their algebraic structures (the *very* abstract algebraists have their categories), the analysts have their normed spaces, the probabilists have their measure spaces, the model-theorists have their model-theoretic logics, so the formal logicans ought to have their formal logics.

But the matter on the conceptual level goes deeper. Many subjects have been transformed by the search for the "right way" to provide a manageable systematic development of their part of mathematics, even though some of the basic ideas and results are very intuitive. Examples are the exterior differential calculus as the framework for the ordinary and higher dimensional versions of Stokes' and Gauss' theorems, or the categorical theory of homology as the framework for combinatorial topology.

Where an exact notion of formal system is particularly needed in our subject is in the formalization of metamathematics, beginning with *Gödel's Second Incompleteness Theorem.* As we know, this is very sensitive to *how* the formal systems dealt with are *presented* (cf. Feferman [1960], Kreisel [1965], 153–154). While the two-line informal argument for Gödel's theorem is very convincing[3], there is no really satisfactory presentation which is both detailed and sufficiently general. But, as anyone knows who has gone into proof theory at all, this is just the tip of the iceberg. For a number of other examples, see the concluding section of this paper.

(ii) *Pedagogical.* This is simply an extension of (i). A good conceptual framework is necessary to explain the formalization of metamathematics in a reasonable and convincing way without excessive hand-waving. Students gain confidence when they see that various steps can be worked out systematically and in detail. This does not mean that the subject must be presented entirely at such a level, just that the mechanics of the details are at hand when confidence falters. As intuition and experience take over, the need for such recedes, but the existence of a manageable underlying framework is always reassuring.

(iii) *Practical.* There has been much talk in recent years of implementing various kinds of logics on computers, especially for the purposes of proof-checking. Examples for relatively specific logical systems are provided by projects of de Bruijn, Boyer and Moore, Constable, Ketonen, and others. The ELF (Edinburgh Logical Framework) project led by Plotkin (see Harper, *et al* [1987] and Avron, *et al* [1987]) aims to handle an enormous variety of formal logics, by a kind of reduction in each case to the typed lambda-calculus. However, in the ELF approach much preparatory work must be done by hand for each logic so that it can then be implemented on a computer. I believe the step from a logic, as it is presented in usual (humanly understandable) terms to its computer-ready presentation should be as natural and direct as possible. The use of some notion such as that of (presented) finitary inductive system developed here, seems to me to be essential for this goal.

Because there is much ground to be covered, the work in the following concentrates on statements of notions and results; by necessity, proofs are omitted or only sketched.

[3] See p.137 of Kleene's introductory note to *Gödel 1931* in Gödel [1986].

1. The universe of tree expressions. The collection V of individuals dealt with, is taken to be built up from a class U of *urelements* by closure under the operation P of pairing. We write P_1, P_2 for the left and right inverses of P, resp., so that $P_i(P(x_1, x_2)) = x_i$, and x is a pair just in case $x = P(P_1x, P_2x)$. Thus U is the class of x such that $x \neq P(P_1x, P_2x)$. U has a distinguished element 0, and V_0 denotes the subclass of V generated from 0 by closure under pairing. Elements of V are called *(binary) tree expressions over* U, and elements of V_0 are called the *pure tree expressions*.

The standard alternative approach to syntax takes the universe of expressions to be the class U^* of *finite strings*, or *words*, $u_0 \dots u_{n-1}$ with $u_i \in U$, thinking of U as a set of *basic symbols* or *alphabet* (cf. e.g. Smullyan [1961], Fitting [1987]). These are represented here instead by finite sequences, defined below in terms of iterated pairing. The present approach builds in the tree structure of expressions met in practice, with x_1, x_2 being considered the immediate *subexpressions* of $P(x_1, x_2)$. The main advantage is that it provides for a more natural and direct introduction of syntactic functions, typically introduced by recursion (from subexpressions to expressions). The same approach is followed in the familiar list-processing programming language LISP.[4,5]

For simplicity in the following we shall work entirely with pure tree expressions. There is no loss of generality for syntactic applications over any finite alphabet U, by representation of such U within V_0. The more general situation (V over any U) is useful if we want to develop recursion theory over an arbitrary structure $\langle U, \dots \rangle$, as done e.g. in Moschovakis [1969], as well as treat general model theory (as initiated in [IPS]). All of our work extends directly to the more general situation.

Abbreviations. $(x, y) := P(x, y)$

$$(x_1) := x_1 \ , \quad (x_1, \dots, x_n, x_{n+1}) := ((x_1, \dots, x_n), x_{n+1}) \ .$$

Note that this representation makes each n-tuple $(x_1, \dots, x_n)$ an m-tuple for each $m \leq n$, e.g. $(x_1, x_2, x_3, x_4, x_5) = ((x_1, x_2, x_3), x_4, x_5)$. Finite sequences will be represented below in a modified form in such a way that each finite sequence has a definite length.

---

[4] A third alternative (to the theory of trees as here, or to concatenation theory) is to work in some form of hereditarily finite set theory. However, sets must still be represented by trees or lists when it comes to effective implementation.

[5] Another reason for working with trees rather than sequences is that the former have a natural generalization to syntactically described infinitary logics; cf. also the concluding remarks to this paper.

As further abbreviations, we take

$$x' := (x, 0) \ , \quad 1 := 0' \ , \quad 2 := 1', \text{ etc.}$$

Note $x' \neq 0$ (by $(x, y) \neq 0$), and $x' = y' \Rightarrow x = y$.

2. Functions, classes, relations.

Functions. We use f, g, h (with or without subscripts) to range over arbitrary unary functions from V_0 to V_0. Each unary f determines an n-ary $f^{(n)}$ by $f^{(n)}(x_1, \ldots, x_n) = fx$ for $x = (x_1, \ldots, x_n)$, i.e. $f^{(n)}$ is the restriction of f to the class of n-tuples. We do not distinguish $f^{(n)}$ from f; thus each f is simultaneously an n-ary function for all n. (This is an immediate advantage of the pairing structure of the universe.) Capital letters F, G, H are also used for specific functions.

Classes. We use A, B, C, X, Y, Z to range over arbitrary subclasses of V_0. The characteristic function $c_A : V_0 \to V_0$ of $A \subseteq V_0$ is given by $c_A x = \begin{cases} 0 & x \in A \\ 1 & x \notin A \end{cases}$. Here, and in the following, '0' represents 'True' and '1', 'False'.

Relations. As with functions, each class A determines an n-ary relation $A^{(n)}$ given by:

$$(x_1, \ldots, x_n) \in A^{(n)} \text{ iff } x \in A \ , \quad \text{for } x = (x_1, \ldots, x_n) \ .$$

Again we shall not distinguish $A^{(n)}$ from A; thus each A acts simultaneously as an n-ary relation for each n. Following usual relational notation, we shall also write $A(x_1, \ldots, x_n)$ for $(x_1, \ldots, x_n) \in A$.

Functionals and collections. At the next level, script letters like $\mathcal{G}$, $\mathcal{H}$ will be used for certain specific functionals on and to functions or classes, as e.g. $\mathcal{G}(f, g) = h$, $\mathcal{H}(A, B) = C$, etc., and script letters like $\mathcal{F}(\mathcal{K})$ will be used for certain collections of functions (classes).

3. The explicit functions.

The *basic functions* (on V_0) are I, K_0, D, P_1, P_2, where

$$Ix = x \ , \quad K_0 x = 0 \qquad \text{all } x,$$

$$D(x_1, x_2, y_1, y_2) = \begin{cases} y_1 & \text{if } x_1 = x_2 \\ y_2 & \text{otherwise} \end{cases} \ , \quad (Du = 0 \text{ if } u \text{ is not a 4-tuple}) \ ,$$

and P_1, P_2 are the inverses to pairing from §1.

The *explicit compounding functionals* are $\mathcal{P}$ (for pairing) and $\mathcal{C}$ (for composition), given by

$$\mathcal{P}(f,g)x = (fx, gx) \ ,$$
$$\mathcal{C}(f,g)x = f(gx) \ .$$

We also write (f,g) for $\mathcal{P}(f,g)$ and $f \circ g$ for $\mathcal{C}(f,g)$.

A class $\mathcal{F}$ of functions is said to be closed under *explicit definition,* or $\mathcal{E}$*-closed*, if it contains I, K_0, D, P_1, P_2 and is closed under $\mathcal{P}$ and $\mathcal{C}$. We denote by $\mathcal{E}$ the least $\mathcal{E}$-closed class.

Throughout the following $\mathcal{F}$ is any $\mathcal{E}$-closed class.

(i) $P_{n,i} \in \mathcal{E}$ for each i, $1 \leq i \leq n$, with $P_{n,i}(x_1, \ldots, x_n) = x_i$.

(ii) $K_a \in \mathcal{E}$ for each $a \in V_0$, where $K_a x = a$.

(iii) $\mathcal{F}$ is closed under n-tupling of functions: $(g_1, \ldots, g_n)x = (g_1 x, \ldots, g_n x)$.

(iv) $\mathcal{F}$ is closed under general composition:

$$(f \circ (g_1, \ldots, g_n))x = f(g_1 x, \ldots, g_n x) \ .$$

(Note then $(f \circ (g_1, \ldots, g_n))(x_1, \ldots, x_m) = f(g_1(x_1, \ldots, x_m), \ldots, g_n(x_1, \ldots x_m))$.)

(v) $E \in \mathcal{E}$, where $E(x_1, x_2) = \begin{cases} 0 & x_1 = x_2 \\ 1 & \text{otherwise} \end{cases}$ (and $E0 = 0$)

(vi) $E_a \in \mathcal{E}$ for each $a \in V_0$, where

$$E_a x = \begin{cases} 0 & x = a \\ 1 & \text{otherwise} \end{cases}$$

(vii) $Q_a \in \mathcal{E}$ where $Q_a x = (x, a)$ for each $a \in V_0$.

(viii) The propositional functions *Neg, Cnj, Dsj* $\in \mathcal{E}$, where, for $x, y \in \{0, 1\}$ we have

$$Neg\ 0 = 1, \quad Neg\ 1 = 0 \ ,$$
$$Cnj\ (x,y) = 0 \Leftrightarrow x = 0 \wedge y = 0 \ ,$$
$$Dsj\ (x,y) = 0 \Leftrightarrow x = 0 \vee y = 0 \ .$$

All these (i)–(viii) are easily established.

Remark. Speaking logically, every propositional combination of equations between terms built up from variables and 0 by the $\mathcal{E}$-closure conditions reduces to an equation of the form $t = 0$, by (v) and (viii).

4. Explicitly determined classes and relations. A class A is said to be *explicitly determined,* and we write $A \in \mathcal{E}$, if $c_A \in \mathcal{E}$; we do the same for a class B considered as a relation. The following are easily checked:

(i) $\{0\} \in \mathcal{E}$

(ii) If $f \in \mathcal{E}$ and $A \in \mathcal{E}$ then $f^{-1}A(= \{x|fx \in A\})$ is in $\mathcal{E}$.

(iii) $\mathcal{E}$ is closed under $\cap, \cup$ and $-$ (complementation).

Exercises.

1. For each $f \in \mathcal{E}$, $f^{-1}\{0\}(= \{x|fx = 0\})$ is in $\mathcal{E}$. Hence $V_0 = K_0^{-1}\{0\}$, $\emptyset = K_1^{-1}\{0\}$ and $\{a\} = E_a^{-1}\{0\}$ are in $\mathcal{E}$.
2. $A, B \in \mathcal{E} \Rightarrow A \times B \in \mathcal{E}$,

$$\begin{aligned} \text{by } A \times B &= \{x|x = (P_1x, P_2x) \wedge P_1x \in A \wedge P_2x \in B\} \\ &= \{x|E(x, (P_1x, P_2x)) = 0\} \cap P_1^{-1}A \cap P_2^{-1}B \end{aligned}$$

Abbreviations. $A^1 := A\,, \quad A^{n+1} := A^n \times A.$

A collection $\mathcal{K}$ of classes is said to be $\mathcal{F}$-*closed* if it contains $\{0\}$ and is closed under f^{-1} for each $f \in \mathcal{F}$, $\cup, \cap$ and $-$. $\mathcal{K}$ is said to be an $\mathcal{F}^+$-*closed collection* if it satisfies the same closure conditions except possibly for the complement operation. Note that $\mathcal{E}$ is the least $\mathcal{E}^+$-closed collection since if $A \in \mathcal{E}$ then $A = c_A^{-1}\{0\}$. Most closure conditions stated in the following apply to any $\mathcal{F}^+$-closed collection.

We shall represent $m+1$-tuples of classes $\langle A_0, \ldots, A_m \rangle$ by disjoint sums,

$$\langle A_0, \ldots, A_m \rangle := \bigcup_{i \leq m} A_i \times \{i\}\ .$$

Thus $A_i = Q_i^{-1}\langle A_0, \ldots, A_m \rangle$. We also write $\langle A_i \rangle_{i \leq m}$ (or simply $\langle A_i \rangle$) for $\langle A_0, \ldots, A_m \rangle$.

5. Primitive recursive functions and classes. The functional $\mathcal{R}$ for definition by primitive recursion is given by

$$\mathcal{R}(f, g) = h \text{ where } \begin{cases} h0 = 0 \\ h(x, 0) = fx \\ h(x, (y, z)) = g(x, y, z, h(x, y), h(x, z))\ . \end{cases}$$

We consider this as primitive recursion with one parameter x. By the representation of n-tuples in §1, the very same functional yields primitive recursion with n-parameters for any $n \geq 1$:

$$h(x_1, \ldots, x_n, 0) = f(x_1, \ldots, x_n)$$
$$h(x_1, \ldots, x_n, (y, z)) = g(x_1, \ldots, x_n, y, z, h(x_1, \ldots, x_n, y), h(x_1, \ldots, x_n, z)) \ .$$

To obtain recursion with no parameters, i.e.

$$h0 = a$$
$$h(y, z) = g(y, z, hy, hz) \ ,$$

one applies $\mathcal{R}$ to suitable (f_0, g_0).

A collection $\mathcal{F}$ of functions is said to be $\mathcal{PR}$-*(primitive recursively) closed* if it is $\mathcal{E}$-closed and closed under $\mathcal{R}$. The least such collection is denoted $\mathcal{PR}$. A class A is said to be *primitive recursive,* and we write $A \in \mathcal{PR}$, if $c_A \in \mathcal{PR}$. The $\mathcal{PR}$ classes are also $\mathcal{E}$-closed. For any $\mathcal{F}$, $\mathcal{PR}(\mathcal{F})$ denotes the least $\mathcal{PR}$-closed collection of functions (and thence of classes) which contains $\mathcal{F}$; members of $\mathcal{PR}(\mathcal{F})$ are said to be *primitive recursive in* $\mathcal{F}$.

6. Presentation of primitive recursive functions. Informally, by a *presentation* of any specific $F \in \mathcal{PR}$ we mean a description of how F is defined in some particular way from the basic functions by successive application of the compounding functionals. Later, this will be specified by a *function term* in the formal system FS_0 . However, there are other more *ad hoc* means of presentation, such as provided by the following coding system C.

C is defined as the least class $X \subseteq V_0$ such that:

$$(1) \quad \begin{cases} (0, i) \in X \text{ for } i = 0, \ldots, 4, \text{ and} \\ c_1, c_2 \in X \Rightarrow (j, (c_1, c_2)) \in X \text{ for } j = 1, 2, 3 \ . \end{cases}$$

With each $c \in C$ is associated a function $[c] \in \mathcal{PR}$ by:

$$(2) \quad \begin{cases} [(0,0)] := I, \quad [(0,1)] := K_0, \ [(0,2)] := D, \ [(0,3)] := P_1, \ [(0,4)] := P_2, \\ [(1,(c_1,c_2))] := \mathcal{P}([c_1],[c_2]), \ [(2,(c_1,c_2))] := \mathcal{C}([c_1],[c_2]), \\ [(3,(c_1,c_2))] := \mathcal{R}([c_1],[c_2]) \ . \end{cases}$$

Every $\mathcal{PR}$ function F is $[c]$ for some $c \in C$ (in fact for infinitely many $c \in C$). Presentations of explicit functions are obtained from the subclass C_0 of C generated from the $(0, i)(i = 0, \ldots, 4)$ by closing under $(1, (c_1, c_2))$ and $(2, (c_1, c_2))$ only. Thus $F \in \mathcal{E}$ just in case $F = [c]$ for some $c \in C_0$.

It is interesting to note the following results concerning C, though they are not needed below.

(3) *Substitution ("s-1-1") theorem.* There is an operation $S \in \mathcal{PR}$ such that $c \in C$ implies $S(c,a) \in C$, and $[S(c,a)]x = [c](a,x)$ for all a, x.

(4) *Recursion theorem.* For each $\mathcal{PR}$ function $[f]$ we can find $e \in C$ with $[e]x = [f](e,x)$ for all x.

Proofs. The function S for (3) is simply given by $S(c,a) = (2,(c,(1,(K'a,(0,0)))))$ where $K'0 = (0,1)$, $K'(a,b) = (1,(K'a,K'b))$, so that $[K'a] = K_a$ for all a. Then (4) is proved by the usual diagonalization argument, taking $e = [S(c,c)]$ where $[c](z,x) = [f](S(z,z),x)$.

Note. Both (3), (4) hold for the $\mathcal{E}$-closure of the single function K', in place of $\mathcal{PR}$.

7. Finitary inductive systems. We now come to a central concept of our work. By a *system of finitary inductive closure conditions* for classes $X_0, \ldots, X_m$ we mean a finite set of conditions of the form:

$$(1)\qquad \begin{array}{ll} \text{(i)} & A_i \subseteq X_i \quad (0 \leq i \leq m) \\ \text{(ii)} & \dfrac{y_1 \in X_{k_1}, \ldots, y_{n_j} \in X_{k_{n_j}}}{x \in X_i}\ B_j(x, y_1, \ldots, y_{n_j}) \quad (0 \leq j \leq p)\ . \end{array}$$

Such a system is specified by two sequences of classes $\langle A_i \rangle_{i \leq m}$ (the *basis conditions*) and $\langle B_j \rangle_{j \leq p}$ (the *rules of inference*) and a *signature* σ, which is an assignment to each $j \leq p$ of an (n_j+1)-tuple $(i, k_1, \ldots, k_{n_j})$ with $i \leq m$ and each $k_r \leq m$ $(1 \leq r \leq n_j)$. By the *finitary inductive system* Σ *of signature* σ for (1) we mean the pair $(\langle A_i \rangle_{i \leq m},\ \langle B_j \rangle_{j \leq p})$ specifying these closure conditions.

In more logical form, the closure conditions (1) can be rewritten as:

$$(2)\qquad \begin{aligned} \mathrm{Clos}^{\sigma}_{\Sigma}(X_0, \ldots, X_m) :=& \bigwedge\!\!\!\bigwedge_{i \leq m} \forall x (x \in A_i \Rightarrow x \in X_i) \wedge \\ & \wedge \bigwedge\!\!\!\bigwedge_{j \leq p} \forall x, y_1, \ldots, y_{n_j} \left[\bigwedge\!\!\!\bigwedge_{1 \leq r \leq n_j} (y_r \in X_{k_r}) \wedge B_j(x, y_1, \ldots, y_{n_j}) \Rightarrow x \in X_i \right] . \end{aligned}$$

Clearly there is a least $\langle X_0, \ldots, X_m \rangle$ satisfying these closure conditions; it is denoted

$$(3)\qquad \mathcal{SI}^{\sigma}(\Sigma) := (\mathrm{least} \langle X_0, \ldots, X_m \rangle)\ \mathrm{Clos}^{\sigma}_{\Sigma}(X_0, \ldots, X_m).$$

'$\mathcal{SI}$' is used for *simultaneous induction*. Note that for $\langle X_0, \ldots X_m\rangle = \mathcal{SI}^\sigma(\Sigma)$ we have each $X_i = Q_i^{-1}\mathcal{SI}^\sigma(\Sigma) = \{x|(x,i) \in \mathcal{SI}^\sigma(\Sigma)\}$.

Four measures of complexity of a simultaneous inductive definition $\mathcal{SI}^\sigma(\langle A_i\rangle_{i\leq m}, \langle B_j\rangle_{j\leq p})$ concern us in the following. The first is the number, $m+1$, of classes being determined simultaneously by the closure conditions (1). The second is the number, $p+1$, of rules of inference B_j being applied in (1). The third is the number $n = \max\limits_{0\leq j\leq p} n_j$, given by σ; we call n_j the *number of hypotheses* (or assumptions) of B_j, and n the *maximum number of hypotheses* of this inductive definition. The final measure is qualitative, namely as to how "complicated" are the classes A_i, B_j; this will be described by different collections $\mathcal{K}$ from which these classes may be drawn.

When the signature σ is indicated by context, we shall omit it to simplify notation. We write $\mathcal{SI}_{\leq n}$ for $\mathcal{SI}^\sigma$ when the signature σ gives $n = \max\limits_{0\leq j\leq p} n_j$. The simplest case for m is that of $m = 0$, when we are dealing with a single inductively defined class, and in this case we write $\mathcal{I}_{\leq n}(A, \langle B_j\rangle_{j\leq p})$ for the result of the inductive definition. That is,

(4) $\mathcal{I}_{\leq n}(A, \langle B_j\rangle)$ is the least class X satisfying:

(i) $A \subseteq X$

(ii) $$\frac{y_1, \ldots, y_{n_j} \in X}{x \in X}\ B_j(x, y_1, \ldots y_{n_j})\ (0 \leq j \leq p)$$

for $n = \max\limits_{0\leq j\leq p} n_j$.

Finally, we write $\mathcal{I}_n(A, B)$ for the result of defining a single class using a single rule of inference with n hypotheses, that is,

(5) $\mathcal{I}_n(A, B)$ is the least class X satisfying:

(i) $A \subseteq X$

(ii) $$\frac{y_1, \ldots, y_n \in X}{x \in X}\ B(x, y_1, \ldots, y_n)\ .$$

8. Reduction in the complexity of inductive definitions. In this section we shall show how to reduce the complexities of simultaneous inductive definitions according to the first three measures just described; then we shall accomplish a

reduction in the complexity of basis conditions and rules of inference in the next section. The basic ideas, all quite simple, come from [IPS], pp. 101–102.

8.1 Reduction of $\mathcal{SI}_{\leq n}$ to $\mathcal{I}_{\leq n}$.

To define $\mathcal{SI}_{\leq n}(\langle A_i\rangle_{i\leq m}, \langle B_j\rangle_{j\leq p})$ as $\mathcal{I}_{\leq n}(A', \langle B'_j\rangle_{j\leq p})$ we simply put conditions on $X = \langle X_0, \ldots, X_m\rangle$ by treating $x \in X_i$ as $(x, i) \in X$. The desired closure conditions on X are then given in the form

$$(1)\qquad \begin{array}{l} \text{(i) } \langle A_0, \ldots, A_m\rangle \subseteq X, \text{ and} \\ \text{(ii) } \dfrac{(y_1, k_1) \in X, \ldots (y_{n_j}, k_{n_j}) \in X}{(x, i) \in X}\ B_j(x, y_1, \ldots, y_{n_j})\ . \end{array}$$

This is recast as

$$(2)\qquad \begin{array}{l} \text{(i) } A' \subseteq X, \quad \text{for } A' = \langle A_0, \ldots, A_m\rangle, \text{ and} \\ \text{(ii) } \dfrac{u_1, \ldots, u_{n_j} \in X}{v \in X}\ B'_j(v, u_1, \ldots, u_{n_j}) \end{array}$$

$$\text{for } B'_j(v, u_1, \ldots, u_{n_j}) \Leftrightarrow B_j(P_1 v, P_1 u_1, \ldots, P_1 u_{n_j}) \wedge$$
$$v = (P_1 v, i) \wedge u_1 = (P_1 u_1, k_1) \wedge \ldots \wedge u_{n_j} = (P_1 u_{n_j}, k_{n_j})\ .$$

Note that any $\mathcal{E}^+$-closed $\mathcal{K}$ which contains $\langle A_i\rangle, \langle B_j\rangle$, also contains $A', \langle B'_j\rangle$.

8.2 *Reduction of $\mathcal{I}_{\leq n}$ to $\mathcal{I}_n$.*

Given $A, \langle B_j\rangle_{j\leq p}$ where the B_j are treated as n_j-ary relations, let $n = \max\limits_{0\leq j\leq p} n_j$. Then replace B_j by B'_j where $B'_j(x, y_1, \ldots, y_n) \Leftrightarrow B_j(x, y_1, \ldots, y_{n_j}) \wedge y_{n_{j+1}} = \ldots = y_n = y_1$. This makes $\mathcal{I}_{\leq n}(A, \langle B_j\rangle_{j\leq p}) = \mathcal{I}_{\leq n}(A, \langle B'_j\rangle_{j\leq p})$ with each B'_j n-ary. But then we simply have $\mathcal{I}_{\leq n}(A, \langle B'_j\rangle_{j\leq p}) = \mathcal{I}_n(A, B'')$ for $B'' = \bigcup\limits_{j\leq p} B'_j$. Again, any $\mathcal{E}^+$-closed $\mathcal{K}$ which contains $A, \langle B_j\rangle$ also contains A, B''.

8.3 *Reduction of $\mathcal{I}_n$ to $\mathcal{I}_2$.*

The idea here is to replace the n-hypothesis rule of inference in $\mathcal{I}_n(A, B)$:

(1) $$A \subseteq X; \quad \frac{y_1, \ldots, y_n \in X}{x \in X}\ B(x, y_1, \ldots, y_n)\ ,$$

by the rule with single hypothesis:

(2) $$A \subseteq X; \quad \frac{u \in X^n}{x \in X}\ B(x, P_{n1}u, \ldots, P_{nn}u)\ .$$

However, X^n must be defined simultaneously with X, and for that we need more generally to define X^k simultaneously with X for each $k = 2, \ldots, n$. Thus consider the following conditions on classes $X_1, \ldots, X_n$:

(3) $$\begin{cases} A \subseteq X_1\ ; \quad \dfrac{y \in X_i, z \in X_1}{x \in X_{i+1}}\ x = (y, z), \quad \text{for } 1 \leq i < n\ ; \\ \dfrac{y \in X_n}{x \in X_1}\ B(x, P_{n1}y, \ldots, P_{nn}y)\ . \end{cases}$$

The least $\langle X_1, \ldots, X_n \rangle$ satisfying this is of the form $\mathcal{SI}_{\leq 2}(\langle A'_i \rangle_{1 \leq i \leq n}, \langle B'_j \rangle_{1 \leq j \leq n})$ where $A'_1 = A$, $\quad A'_{i+1} = \emptyset$, $\quad B'_j(x, y_1, y_2) \Leftrightarrow x = (y_1, y_2)$ for $j < n$ and $B'_n(x, y_1, y_2) \Leftrightarrow B(x, P_{n1}y_1, \ldots, P_{nn}y_1) \wedge y_2 = y_1$. Moreover, $\mathcal{SI}_{\leq 2}(\langle A'_i \rangle, \langle B'_j \rangle) = \langle X, X^2, \ldots, X^n \rangle$ where $X = \mathcal{I}_n(A, B)$ is the least solution of (1). Now by 8.1 and 8.2, we re-represent $\mathcal{SI}_{\leq 2}(\langle A'_i \rangle, \langle B'_j \rangle)$ as $\mathcal{I}_2(A'', B'')$ for suitable A'', B'', so $X = Q_0^{-1}\mathcal{I}_2(A'', B'')$. Again, if $\mathcal{K}$ is $\mathcal{E}^+$-closed and contains A, B, we may obtain A'', B'' in $\mathcal{K}$.

9. Inductive closure and the Normal Form Theorem. We define S-IND($\mathcal{K}$) to be the least $\mathcal{PR}^+$-closed collection which contains $\mathcal{SI}^\sigma(\langle A_i \rangle, \langle B_j \rangle)$ for all $A_i, B_j \in \mathcal{K}$. Similarly, $\mathrm{IND}_2(\mathcal{K})$ is defined to be the least $\mathcal{PR}^+$-closed collection which contains $\mathcal{I}_2(A, B)$ for all $A, B \in \mathcal{K}$. Finally, define $\mathcal{IND}$ to be the least $\mathcal{K}$ such that

$$\mathrm{IND}_2(\mathcal{K}) \subseteq \mathcal{K}\ ,$$

i.e. the least $\mathcal{PR}^+$-closed $\mathcal{K}$ such that $\mathcal{K}$ is closed under $\mathcal{I}_2$. These notions can be relativized to any class $\mathcal{F}$ of functions in place of $\mathcal{PR}$, as S-IND$^{(\mathcal{F})}(\mathcal{K})$, $\mathrm{IND}_2^{(\mathcal{F})}(\mathcal{K})$, and $\mathrm{IND}^{(\mathcal{F})}$, and the following results hold for $\mathcal{E}$-closed $\mathcal{F}$ and $\mathcal{F}^+$-closed ($\mathcal{K}$).

First we can summarize the results of 8.1–8.3 by:

THEOREM. *If $\mathcal{K}$ is $\mathcal{E}^+$-closed then* S-IND($\mathcal{K}$) = $\mathrm{IND}_2(\mathcal{K})$.

COROLLARY. S-IND($\mathcal{IND}$) = $\mathcal{IND}$.

Our main result here is the following (which again holds for $\mathcal{IND}^{(\mathcal{F})}$, for any $\mathcal{E}$-closed $\mathcal{F}$ in place of $\mathcal{PR}$).

THE NORMAL FORM THEOREM FOR $\mathcal{IND}$.[6] *For each $X \in \mathcal{IND}$ we can find $A, B \in \mathcal{PR}$ with $X = Q_0^{-1}\mathcal{I}_2(A, B)$.*

Proof. It is simpler to first represent X in terms of $\mathcal{SI}_{\leq 2}$ and then apply the reductions of §8.

LEMMA. *Each $X \in \mathcal{IND}$ is representable as $X = X_0$ for the least solution $\langle X_0, \ldots, X_m\rangle$ of closure conditions of the form*

$$A_i \subseteq X_i \ ; \quad \frac{y_1 \in X_j,\ y_2 \in X_k}{x \in X_i}\ B_{ijk}(x, y_1, y_2)$$

with $A_i,\ B_{ijk} \in \mathcal{PR}$. In other words $X = Q_0^{-1}\mathcal{SI}_{\leq 2}(\langle A_i\rangle, \langle B_{ijk}\rangle)$.

Proof. This proceeds by induction on the generation of $\mathcal{IND}$ as the least $\mathcal{K}$ which is closed under $\mathcal{I}_2(A, B)$ for $A, B \in \mathcal{K}$.

(i) $X = \{0\}$; then $X = Q_0^{-1}$ (Least$\langle X_0\rangle$)($\{0\} \subseteq X_0$).

In the following, assume by induction hypothesis that $Y = Y_0$ for the least $\langle Y_0, \ldots, Y_m\rangle$ satisfying

$$(1) \qquad C_i \subseteq Y_i \ ; \quad \frac{y_1 \in Y_j,\ y_2 \in Y_k}{x \in Y_i}\ C'_{ijk}(x, y_1, y_2)\ ,$$

and that $Z = Z_0$ for the least $\langle Z_0, \ldots Z_q\rangle$ satisfying

$$(2) \qquad D_i \subseteq Z_i \ ; \quad \frac{u_1 \in Z_j,\ u_2 \in Z_k}{z \in Z_i}\ D'_{ijk}(z, u_1, u_2)$$

where all $C_i, D_i, C'_{ijk}, D'_{ijk} \in \mathcal{PR}$. We shall show that $X = f^{-1}Y$ (for $f \in \mathcal{PR}$), $X = Y \cup Z$, $X = Y \cap Z$ and $X = \mathcal{I}_2(Y, Z)$ satisfy the conclusion of the lemma in the following (ii)–(v).

[6] This result is analogous to (but formally simpler than) the Reduction Theorem of Moschovakis [1969], p.331 (also called a Normal Form Theorem), for his inductive approach to the general notion of algorithm.

(ii) To show that $X = f^{-1}Y$ for $f \in \mathcal{PR}$ satisfies the lemma, we take $X = X_0$ for the least $\langle X_0, Y_0, \ldots, Y_m\rangle$ satisfying the closure conditions (1) and

$$\frac{y \in Y_0}{x \in X_0}\ y = fx\ .$$

(As in 8.2, this one-hypothesis rule is trivially transformed into a rule with two hypotheses; the same applies in the following.)

(iii) To show $X = Y \cup Z$ satisfies the lemma, we take $X = X_0$ for the least $\langle X_0, Y_0, \ldots, Y_m, Z_0, \ldots, Z_q\rangle$ satisfying the closure conditions (1), (2) and, in addition,

$$\frac{y \in Y_0}{x \in X_0}\ y = x\ ; \quad \frac{y \in Z_0}{x \in X_0}\ y = x$$

(iv) To show $X = Y \cap Z$ satisfies the lemma, we represent $X = X_0$ for the least $\langle X_0, Y_0, \ldots, Y_m, Z_0, \ldots, Z_q\rangle$ satisfying the closure conditions (1), (2) and, in addition

$$\frac{y_1 \in Y_0,\ y_2 \in Z_0}{x \in X_0} \quad x = y_1 = y_2$$

(v) To show $X = \mathcal{I}_2(Y, Z)$ satisfies this form, we consider first its given representation as the least X satisfying

$$Y \subseteq X\ ; \quad \frac{y_1,\ y_2 \in X}{x \in X}\ (x, y_1, y_2) \in Z\ .$$

Now in place of this, we take $X = X_0$ for the least $\langle X_0, Y_0, \ldots, Y_m,\ Z_0, \ldots, Z_q,\ W\rangle$ satisfying (1), (2) and the following conditions:

$$\frac{y \in Y_0}{x \in X_0}\ x = y\ ; \quad \frac{y_1, y_2 \in X_0}{u \in W}\ u = (y_1, y_2)\ ; \quad \frac{u \in W, z \in Z_0}{x \in X_0}\ z = (x, P_1u, P_2u)\ .$$

The class W is introduced to make $W = X_0^2$ in the least solution and thus to keep to rules with two hypotheses.

This concludes the proof of the Lemma. Now the Normal Form Theorem itself follows by the reduction methods of §8:

$$\begin{aligned} X &= Q_0^{-1}\ \mathrm{Least}\langle X_0, \ldots, X_m\rangle\ \mathrm{Clos}_{\langle A_i\rangle\langle B_{ijk}\rangle}(X_0, \ldots, X_m) \\ &= Q_0^{-1}(\mathrm{Least}\, Y)\mathrm{Clos}_{A', B'}(Y) \end{aligned}$$

where $A' = \langle A_0, \ldots, A_m\rangle = \bigcup_i A_i \times \{i\}$ and

$$\begin{aligned} B'_{ijk}(v, u_1, u_2) \Leftrightarrow v =& (P_1v, i) \wedge u_1 = (P_1u_1, j) \wedge u_2 = (P_1u_2, k) \\ & \wedge B_{ijk}(P_1v, P_1u_1, P_1u_2)\ . \end{aligned}$$

and $B' = \bigcup_{ijk} B'_{ijk}$.

In the following sections we turn to some interesting specific examples of classes defined by elementary closure conditions.

10. The natural numbers. We are using the abbreviation $x' = (x, 0)$ from §1; it follows that $x = P_1 x'$. Moreover, $x' \neq 0$, and $x' = y' \Rightarrow x = y$. Now N is defined as the least class X satisfying

$$0 \in X \ ; \quad \frac{y \in X}{x \in X}\, x = y' \ ,$$

that is, $N = \mathcal{I}_1(\{0\}, \{(x, y) | x = y'\})$. Hence we have:

Closure. $0 \in N \wedge \forall x(x \in N \Rightarrow x' \in N)$.

Induction. $0 \in A \wedge \forall x(x \in A \Rightarrow x' \in A) \Rightarrow N \subseteq A$.

Next note that N *is primitive recursive.* Its characteristic function c_N is defined by the primitive recursion (with no parameters):

$$c_N 0 = 0 \ , \qquad c_N(y, z) = \begin{cases} 0 & \text{if } z = 0 \wedge c_N y = 0 \\ 1 & \text{otherwise.} \end{cases}$$

This is because $(y, z) \in N \Leftrightarrow y \in N \wedge z = 0$.

Primitive recursion on N. We can define an operator $\mathcal{R}_N$ from $\mathcal{R}$ so that

$$h = \mathcal{R}_N(f, g) \Rightarrow h(x, 0) = fx \text{ and } h(x, y') = f(x, y,\ h(x, y)) \ .$$

From this one obtains recursion on N with n parameters for any $n \geq 0$.

Note. h is total on V_0; the equations only show how $h(x, y)$ acts for $x \in V_0,\ y \in N$.

It is immediate that all number-theoretic primitive recursive functions and relations are in $\mathcal{PR}$. But also many closure conditions on $\mathcal{PR}_N$ extend to arbitrary functions. For example, we have:

Bounded quantification. With each f is associated g such that for each $y \in N$ and $x \in V_0$:

$$g(x, y) = 0 \Leftrightarrow \forall z(z < y \Rightarrow f(x, z) = 0) \ .$$

g is defined by recursion on N, with

$$g(x, 0) = 0 \ , \quad g(x, y') = \begin{cases} 0 & \text{if } g(x, y) = 0 \wedge f(x, y) = 0 \\ 1 & \text{otherwise} \end{cases} \ .$$

Similarly with each f is associated g such that for each $y \in N$ and $x \in V_0$,

$$g(x,y) = 0 \Leftrightarrow \exists z(z < y \wedge f(x,z) = 0) \ .$$

Finally, we can introduce the *bounded minimum*

$$g(x,y) = (\mu z < y) f(x,z) = 0$$

for $y \in N$, $x \in V_0$.

11. Sequences. Here we follow [IPS] §3.3: each $x \in V_0$ represents a sequence, with 0 representing the empty sequence and if y represents $\langle y_0, \ldots, y_{n-1} \rangle$ then (y,z) represents $\langle y_0, \ldots, y_{n-1}, z \rangle$.

Sequences from a class. For each class Z, the class $Seq(Z)$, or $Z^{<\omega}$, of finite sequences, all of whose terms belong to Z, is defined as the least class X satisfying:

$$0 \in X \ , \qquad \frac{y \in X}{x \in X} \ x = (y,z) \wedge z \in Z \ .$$

That is, $Seq(Z) = \mathcal{I}_1(\{0\}, \{(x,y) | x = (y, P_2 x) \wedge P_2 x \in Z\})$. Again we have the closure and induction principles for $Z^{<\omega}$. Moreover, $Z^{<\omega}$ is $\mathcal{PR}$ in Z.

Length. This is defined for arbitrary sequences (in other words, arbitrary members of V_0) by means of the primitive recursive definition:

$$Lh0 = 0 \ , \quad Lh(y,z) = Lh(y) + 1 \ .$$

Then it is proved by induction on V_0 that $Lh : V_0 \to N$. The i^{th} term of a sequence x, denoted $Val(x,i)$, is defined recursively by

$$Val(0,i) = 0, \quad Val((y,z),i) = \begin{cases} Val(y,i) & \text{if } i < Lh(y) \\ z & \text{otherwise} \end{cases}$$

Thus if y represents $\langle y_0, \ldots, y_{n-1} \rangle$ we have $Lh(y) = n$ and $y_i = \mathrm{Val}(y,i)$. From now on, we write $y = \langle y_0, \ldots, y_{Lh(y)-1} \rangle$ or $y = \langle y_i \rangle_{i < Lh(y)}$ for arbitrary y, considered as a finite sequence. $y = \langle z \rangle$ is used for a sequence of length 1 with $y_0 = z$.

Concatenation. Again this applies to all sequences, with $x * y$ defined recursively by

$$x * 0 = x, \quad x * (y,z) = (x * y, z) \ .$$

Thus $x * y = \langle x_0, \ldots, x_{Lh(x)-1}, y_0, \ldots, y_{Lh(y)-1} \rangle$ and $x * \langle z \rangle = \langle x_0, \ldots, x_{Lh(x)-1}, z \rangle = (x,z)$.

Restriction of functions to sequences ("apply-to-all"). Define $f \restriction x$ recursively by:

$$f \restriction 0 = 0, \qquad f \restriction (y, z) = (f \restriction y) * \langle fz \rangle \ .$$

Thus $f \restriction \langle x_0, \ldots, x_{Lh(x)-1} \rangle = \langle fx_0, \ldots, f_{Lh(x)-1} \rangle$. (In [IPS] 5.4 we wrote $f \circ x$ for $f \restriction x$.)

Sets from sequences. Given $x = \langle x_0, \ldots, x_{n-1} \rangle$, the set $\{x_0, \ldots, x_{n-1}\}$ can be identified with the class $\{x | x = x_0 \vee \ldots \vee x = x_{n-1}\}$. However, this is not an object in V_0. We can define equivalence of sequences if they determine the same set, by $x \equiv y := [x \subseteq y \wedge y \subseteq x]$, where $x \subseteq y := \forall i < Lh(x)\ \exists j < Lh(y)[x_i = y_j]$.

12. Inductive definitions with variably many hypotheses. In practice one also meets finitary inductive definitions with no pre-assigned bound to the number of hypotheses in the rules of inference, e.g. when defining the class of terms in a language with function symbols of every arity. Using the representation of sequences in the preceding section, we show how such an inductive definition $\mathcal{I}_{<\omega}$ can be reduced to $\mathcal{I}_2$.

A rule of inference R between a conclusion x and a sequence $y = \langle y_0, \ldots, y_{Lh(y)-1} \rangle$ of hypotheses is simply of the form $R(x, y)$. Then we define $\mathcal{I}_{<\omega}(R)$ to be the least X such that

$$\frac{y \in X^{<\omega}}{x \in X}\ R(x, y) \tag{1}$$

Note that no separate base case is necessary, as $0 \in X^{<\omega}$ is always true; thus $A = \{x | R(x, 0)\}$ takes over the base case $A \subseteq X$.

The method of reduction is given in [IPS] 3.4(iv) (p.102). We can take $\mathcal{I}_{<\omega}(R) = X$ for the least $\langle X, Y \rangle$ satisfying the simultaneous inductive definition

$$0 \in Y; \quad \frac{y \in Y, z \in X}{u \in Y}\ u = (y, z)\ ; \quad \frac{y \in Y}{x \in X}\ R(x, y)\ , \tag{2}$$

since the least Y here is then $X^{<\omega}$. This gives $\mathcal{I}_{<\omega}(R)$ in terms of an $\mathcal{SI}_{\leq 2}$, which is reduced to $\mathcal{I}_2$ by §8.

In the same way we can treat more generally a simultaneous inductive definition of $\langle X_0, \ldots, X_m \rangle$ with variably many hypotheses from each X_k.

13. Transitive closure. Informally, the transitive closure of x, $TC(x)$, is the set of y's below x in the build-up of x from 0. $TC(x) = \mathcal{I}_2(\{x\}, B)$ for suitable B. But $TC(x)$ is finite and we can define a function $tc \in \mathcal{PR}$ which gives for each x a sequence $tc(x)$ enumerating $TC(x)$ as follows:

$$tc(0) = 0\ , \quad tc(y,z) = tc(y) * tc(z) * \langle y, z\rangle\ . \tag{1}$$

Thus we can also define $TC(x) := \{y | \exists i < Lh(tc(x))(y = (tc(x))_i)\}$. Another useful function is

$$p\ell(y,x) := \mu i < Lh(tc(x))[y = (tc(x))_i] \tag{2}$$

which gives the place of y in the sequence $tc(x)$ when $y \in TC(x)$. Finally, write $y \prec x$ for $y \in TC(x)$ and $y \preceq x$ for $y \prec x \vee y = x$; this relation is primitive recursive.

Induction with respect to $\prec$. This is the principle

$$\forall x[\forall y(y \prec x \Rightarrow y \in X) \Rightarrow x \in X] \Rightarrow \forall x(x \in X)\ , \tag{3}$$

which follows from the above definition. In addition we have:

Recursion with respect to $\prec$. Given g we can obtain h primitive recursive (uniformly) in g, such that

$$hx = g(x, h \restriction tc(x)) \quad \text{for all } x\ . \tag{4}$$

The idea for this is to first define $\bar{h}x = h \restriction tc(x)$ for each x, so that $hx = g(x, \bar{h}x)$ for each x. $\bar{h}$ is given by the primitive recursion:

$$\bar{h}0 = 0, \quad \bar{h}(y,z) = (\bar{h}y) * (\bar{h}z) * \langle g(y, \bar{h}y), g(z, \bar{h}z)\rangle\ .$$

Rank. Ordinary primitive recursion serves to define the function Rnk with

$$Rnk(0) = 0 \quad Rnk(y,z) = \max(Rnk(y), Rnk(z)) + 1\ .$$

Thus $Rnk(x) =$ the height of x considered as a binary branching tree. We have $Rnk : V_0 \to N$ and $y \prec x \Rightarrow Rnk(y) < Rnk(x)$. There are only finitely many y with $Rnk(y) \leq Rnk(x)$. (Note that this is not true for the corresponding rank function on $V = U^*$ when we have an arbitrary class of urelements.) We can define a primitive recursive function on N which gives for each n a pair of sequences (r_n, r'_n) where r_n enumerates $\{x | Rnk(x) \leq n\}$ and r'_n enumerates $\{x | Rnk(x) = n\}$, since $\{x | Rnk(x) \leq n+1\} = \{(y,z) | Rnk(y) \leq n \wedge Rnk(z) \leq n\}$ and $\{x | Rnk(x) = n+1\} = \{(y,z) | (Rnk(y) \leq n \wedge Rnk(z) = n) \vee (Rnk(y) = n \wedge Rnk(z) \leq n)\}$.

14. Trees and derivations. In order to explain derivations for inductive definitions, we introduce *finitely branching labelled trees.* As all the inductive definitions dealt with here reduce to $\mathcal{I}_2$, it is sufficient to deal with *binary branching labelled trees.* The more general notion is treated in [IPS] pp.110–111, but the special case is simpler to deal with in certain respects.

The informal idea is that $(0, z)$ represent a tree with single node labelled z, and if y_1, y_2 represent (binary) trees then $x = ((y_1, y_2), z)$ represents the tree with label z at its tip and immediate subtrees y_1, y_2. Thus $Tree_2$ is the least X satisfying the closure conditions:

$$(1) \qquad (0, z) \in X \text{ for all } z\ ; \quad \frac{y_1, y_2 \in X}{x \in X}\ x = ((y_1, y_2), P_2 x)$$

The label of any $x \in Tree_2$ is just $P_2 x$, and its immediate subtrees are $y_i = P_i P_1 x$ when $P_1 x \neq 0$. Since each $y_i \prec x$ in the latter case, *recursion on trees* is just a special case of recursion with respect to $\prec$, in the following form.

(2) For each f, g we can find h primitive recursive (uniformly) in f, g with

$$h(0, z) = fz, \text{ and } h((y_1, y_2), z) = g(y_1, y_2, z, hy_1, hy_2)\ .$$

In particular, we can define the *height* $|x|$ of tree by

$$(3) \qquad |(0, z)| = 1, \quad |((y_1, y_2), z)| = \max(|y_1|, |y_2|) + 1\ .$$

Given A, B we define the class $D(A, B) \subseteq Tree_2$ of $\mathcal{I}_2(A, B)$-derivations as the least X which satisfies the following closure conditions:

$$(4) \qquad \begin{array}{ll} \text{(i)} & (0, x) \in X \quad \text{for each } x \in A \\ \text{(ii)} & \dfrac{d_1, d_2 \in X}{d \in X}\ B(P_2 d, P_2 d_1, P_2 d_2) \wedge d = ((d_1, d_2), P_2 d)\ . \end{array}$$

When $d \in D(A, B)$ and $P_2 d = x$ we write $Der_{(A,B)}(d, x)$, or $d \vdash_{(A,B)} x$ or simply $d \vdash x$. (4)(i) expresses that when $x \in A$, the tree d with single node 0 and label x has $d \vdash x$, and (4)(ii) expresses that if $d_1 \vdash y_1$ and $d_2 \vdash y_2$ and $B(x, y_1, y_2)$ then the tree $d = ((d_1, d_2), x)$ has $d \vdash x$. It is thus seen that

$$(5) \qquad \begin{array}{rl} x \in \mathcal{I}_2(A, B) & \Leftrightarrow \exists d(d \in D(A, B) \wedge d \vdash x) \\ & \Leftrightarrow \exists d(d \in D(A, B) \wedge P_2 d = x)\ . \end{array}$$

It will be shown in the next section that $D(A, B)$ is primitive recursive (uniformly) in A, B.

15. Deterministic and decidable inductive definitions. For implementation of logics on computers, we need to know that various syntactic classes dealt with are algorithmically decidable. We show how to do this for certain inductively defined classes of $\mathcal{I}_2$ form (see [IPS] pp.111–112 for the more general case $\mathcal{I}_{<\omega}$).

An inductive definition $\mathcal{I}_2(A,B)$ is said to be *deterministic* if for each $x \in \mathcal{I}_2(A,B)$ there is a unique $d \in D(A,B)$ with $d \vdash x$. A n.a.s.c. for this is that

$$(1) \qquad \forall x \in \mathcal{I}_2(A,B)\{x \in A \vee \exists! y_1, y_2[y_1, y_2 \in \mathcal{I}_2(A,B) \wedge B(x, y_1, y_2)]\}$$

Now for (1) it is *sufficient* that we have *predecessor functions* f_1, f_2 for B, in the sense that:

$$(2) \qquad B(x, y_1, y_2) \Rightarrow y_1 = f_1 x \wedge y_2 = f_2 x \ .$$

In this case, $\mathcal{I}_2(A,B)$ is called *functionally deterministic;* f_1, f_2 are called *transitive predecessor functions* if (2) holds and

$$(3) \qquad B(x, y_1, y_2) \Rightarrow y_1 \prec x \wedge y_2 \prec x \ .$$

THEOREM. *If B has transitive predecessor functions f_1, f_2 then $\mathcal{I}_2(A,B)$ is primitive recursive (uniformly) in A, B, f_1, f_2.*

Proof. Let $J = \mathcal{I}_2(A,B)$. Its characteristic function c_J is defined by the $\prec$ recursion

$$c_J x = \begin{cases} 0 & \text{if } x \in A \text{ or } [B(x, f_1 x, f_2 x) \wedge c_J(f_1 x) = c_J(f_2 x) = 0] \\ 1 & \text{otherwise} \ . \end{cases}$$

COROLLARY. *For each A, B, the class $D(A,B)$ of $\mathcal{I}_2(A,B)$-derivations is primitive recursive in A, B.*

Proof. By the preceding section,

$$D(A,B) = \mathcal{I}_2(A', B')$$

with $A' = \{(0,x) | x \in A\}$ and $B' = \{(d, d_1, d_2) | B(P_2 d, P_2 d_1, P_2 d_2) \wedge d = ((d_1, d_2), P_2 d)\}$. This has the $\mathcal{PR}$ transitive predecessor functions $f_1 d = P_1 P_1 d$, $f_2 d = P_2 P_1 d$.

Note. The facts that $N \in \mathcal{PR}$ (§10), and that $A^{<\omega}$ is $\mathcal{PR}$ in A (§11), both follow from the theorem above.

We can carry out definition by recursion on classes $\mathcal{I}_2(A,B)$ satisfying the hypothesis of the theorem, so that for each g_1, g_2 we obtain h primitive recursive in g_1, g_2 with

$$h(x) = \begin{cases} g_1 x & \text{for } x \in A \\ g_2(x, h(f_1 x), h(f_2 x)) & \text{for each } x \in \mathcal{I}_2(A,B) \text{ with } x \notin A \end{cases}$$

This is again a special case of $\prec$-recursion.

16. The axiomatic theory FS_0 for functions and classes of expressions. We now set up a formal system FS_0 in which all the preceding work may be directly formalized. The language $L(FS_0)$ of this system is three-sorted, for *individuals, functions,* and *classes*. The symbols used for various entities in FS_0 will be similar to those used in the informal development §§1–15, e xcept that we use Roman letters instead of italics; however, we shall still use script letters for the compounding functionals. Thus we use a, b, c, u, v, w, x, y, z for individual variables, f, g, h for function variables and A, B, C, X, Y, Z for class variables (in all cases with or without subscripts).[7] We denote by $InTm$ the class of *individual terms* (for which we use s, t,...), by $FnTm$ the class of *function terms* (F, G,...) and by $ClTm$ the class of *class terms* (R, S, T,...). $InTm$ and $FnTm$ are defined simultaneously as the smallest classes satisfying:

$InTm$ (i) Each individual variable is in $InTm$.
(ii) The constant $\bar{0}$ is in $InTm$.
(iii) If $\mathrm{t}_1, \mathrm{t}_2 \in InTm$ then $(\mathrm{t}_1, \mathrm{t}_2) \in InTm$.
(iv) If $\mathrm{F} \in FnTm$ and $\mathrm{t} \in InTm$ then $\mathrm{Ft} \in InTm$.

$FnTm$ (i) Each function variable is in $FnTm$.
(ii) The constants $\mathrm{I}, \mathrm{D}, \mathrm{P}_1, \mathrm{P}_2$ are in $FnTm$.
(iii) If $\mathrm{t} \in InTm$ then $\mathrm{K(t)} \in FnTm$.
(iv) If $\mathrm{F}, \mathrm{G} \in FnTm$ then $\mathcal{P}(\mathrm{F},\mathrm{G}), \mathcal{C}(\mathrm{F},\mathrm{G}), \mathcal{R}(\mathrm{F},\mathrm{G}) \in FnTm$.

Then $ClTm$ is defined as the smallest class satisfying:

$ClTm$ (i) Each class variable is in $ClTm$.
(ii) The constant $\{\bar{0}\} \in ClTm$.
(iii) If $\mathrm{F} \in FnTm$ and $S \in ClTm$ then $\mathrm{F}^{-1}\mathrm{S} \in ClTm$.
(iv) If $\mathrm{S}, \mathrm{T} \in ClTm$ then $\mathrm{S} \cap \mathrm{T}, \mathrm{S} \cup \mathrm{T}$ and $\mathcal{I}_2(\mathrm{S},\mathrm{T}) \in ClTm$.

The *atomic formulas* of $L(FS_0)$ are just those of the form $(\mathrm{t}_1 = \mathrm{t}_2)$ with $\mathrm{t}_1, \mathrm{t}_2 \in InTm$, and $\mathrm{t} \in \mathrm{S}$ with $\mathrm{t} \in InTm$ and $\mathrm{S} \in ClTm$.

The class of *formulas* $(\phi, \psi, \theta, \ldots)$ is the least class containing the atomic formulas and closed under $\neg, \wedge, \vee, \rightarrow, \forall, \exists$ (applied to variables of any sort). The

[7] These are not the "official" lists of variables.

underlying logic of FS_0 is that of 3-sorted classical predicate calculus with $=$ in the first sort only.

Remark. We regard $\mathrm{F} = \mathrm{G}$ as defined by $\forall \mathrm{x}[\mathrm{Fx} = \mathrm{Gx}]$, and $\mathrm{S} = \mathrm{T}$ as defined by $\forall \mathrm{x}[\mathrm{x} \in \mathrm{S} \leftrightarrow \mathrm{x} \in \mathrm{T}]$. We could add $=$ as a basic symbol in both these sorts and then take these (extensionality) statements as axioms.

The abbreviations introduced in §1 and §2 are also used in FS_0 and we write $\mathrm{S} \subseteq \mathrm{T}$ for $\forall \mathrm{x}(\mathrm{x} \in \mathrm{S} \to \mathrm{x} \in \mathrm{T})$.

Axioms of FS_0 .

I. *Pairing, projections.*

(i) $(\mathrm{x}_1, \mathrm{x}_2) \neq \bar{0}$
(ii) $\mathrm{P}_1(\mathrm{x}_1, \mathrm{x}_2) = \mathrm{x}_1 \ , \quad \mathrm{P}_2(\mathrm{x}_1, \mathrm{x}_2) = \mathrm{x}_2.$

II. *Basic function axioms.*

(i) $\mathrm{Ix} = \mathrm{x}$
(ii) $\mathrm{K(a)x} = \mathrm{a}$
(iii) $[\mathrm{x}_1 = \mathrm{x}_2 \to \mathrm{D}(\mathrm{x}_1, \mathrm{x}_2, \mathrm{y}_1, \mathrm{y}_2) = \mathrm{y}_1] \wedge [\mathrm{x}_1 \neq \mathrm{x}_2 \to \mathrm{D}(\mathrm{x}_1, \mathrm{x}_2, \mathrm{y}_1, \mathrm{y}_2) = \mathrm{y}_2]$
$\wedge(\neg\exists \mathrm{x}_1, \mathrm{x}_2, \mathrm{y}_1, \mathrm{y}_2[\mathrm{u} = (\mathrm{x}_1, \mathrm{x}_2, \mathrm{y}_1, \mathrm{y}_2)] \to \mathrm{Du} = \bar{0}).$

III. *Compound function axioms.*

(i) $\mathrm{h} = \mathcal{P}(\mathrm{f}, \mathrm{g}) \to \forall \mathrm{x}[\mathrm{hx} = (\mathrm{fx}, \mathrm{gx})]$
(ii) $\mathrm{h} = \mathrm{C}(\mathrm{f}, \mathrm{g}) \to \forall \mathrm{x}[\mathrm{hx} = \mathrm{f(gx)}]$

IV. *Recursion axiom.*

$$\mathrm{h} = \mathcal{R}(\mathrm{f}, \mathrm{g}) \to h\bar{0} = \bar{0} \wedge \forall x[h(x, \bar{0}) = fx] \wedge$$
$$\wedge \forall \mathrm{x}, \mathrm{y}, \mathrm{z}[\mathrm{h}(\mathrm{x}, (\mathrm{y}, \mathrm{z})) = \mathrm{g}(\mathrm{x}, \mathrm{y}, \mathrm{z}, \mathrm{h}(\mathrm{x}, \mathrm{y}), \mathrm{h}(\mathrm{x}, \mathrm{z}))]$$

V. *Explicit class constructions axioms.*

(i) $\mathrm{x} \in \{\bar{0}\} \leftrightarrow \mathrm{x} = \bar{0}$
(ii) $\mathrm{x} \in \mathrm{f}^{-1}\mathrm{A} \leftrightarrow \mathrm{fx} \in \mathrm{A}$
(iii) $\mathrm{x} \in \mathrm{A} \cap \mathrm{B} \leftrightarrow \mathrm{x} \in \mathrm{A} \wedge \mathrm{x} \in \mathrm{B}$
(iv) $\mathrm{x} \in \mathrm{A} \cup \mathrm{B} \leftrightarrow \mathrm{x} \in \mathrm{A} \vee \mathrm{x} \in \mathrm{B}$

VI. *Inductive generation axiom.*

$$\text{(i)}\quad \begin{aligned} &\mathrm{C} = \mathcal{I}_2(\mathrm{A},\mathrm{B}) \rightarrow \mathrm{A} \subseteq \mathrm{C} \wedge \\ &\qquad \wedge \forall \mathrm{x},\mathrm{y}_1,\mathrm{y}_2[\mathrm{y}_1 \in \mathrm{C} \wedge \mathrm{y}_2 \in \mathrm{C} \wedge (\mathrm{x},\mathrm{y}_1,\mathrm{y}_2) \in \mathrm{B} \rightarrow \mathrm{x} \in \mathrm{C}] \end{aligned}$$

$$\text{(ii)}\quad \begin{aligned} &\mathrm{C} = \mathcal{I}_2(\mathrm{A},\mathrm{B}) \wedge \mathrm{A} \subseteq \mathrm{X} \wedge \\ &\quad \wedge \forall \mathrm{x},\mathrm{y}_1,\mathrm{y}_2[\mathrm{y}_1 \in \mathrm{X} \wedge \mathrm{y}_2 \in \mathrm{X} \wedge (\mathrm{x},\mathrm{y}_1,\mathrm{y}_2) \in \mathrm{B} \rightarrow \mathrm{x} \in \mathrm{X}] \rightarrow \mathrm{C} \subseteq \mathrm{X}\,. \end{aligned}$$

VII. *Induction on the universe.*

$$\bar{0} \in \mathrm{X} \wedge \forall \mathrm{x},\mathrm{y}[\mathrm{x} \in \mathrm{X} \wedge \mathrm{y} \in \mathrm{X} \rightarrow (\mathrm{x},\mathrm{y}) \in \mathrm{X}] \rightarrow \forall \mathrm{x}(\mathrm{x} \in \mathrm{X}).$$

This completes the description of FS_0 . We shall also consider the subsystem EFS_0 obtained by deleting the $\mathcal{R}$-operator and axiom IV.

17. Models and presentations. The standard model for Axioms I and VII is the structure $\mathcal{V}_0 = \langle V_0, P, P_1, P_2, 0\rangle$. We shall not be concerned here with non-standard models. The function axioms II, III are satisfied by any $\mathcal{E}$-closed collection of functions $f : V_0 \rightarrow V_0$, in particular by $\mathcal{E}$ itself. The axioms II–IV are satisfied by any $\mathcal{PR}$-closed collection of functions.[8] The axiom V is satisfied by any $\mathcal{E}^+$-closed collection of classes in EFS_0, and any $\mathcal{PR}^+$-closed collection of classes in FS_0 . The Axiom VI is satisfied in any collection of classes closed under $\mathcal{I}_2$. In particular, the class axioms are satisfied by $\mathcal{INC}$ (the least $\mathcal{PR}^+$-closed $\mathcal{K}$ which is closed under $\mathcal{I}_2$).

The minimal model of FS_0 is thus given by $\mathcal{V}_0, \mathcal{PR}$ and $\mathcal{INC}$. We shall use $[\![\cdot]\!]$ to associate with each formal term its corresponding informal interpretation in this model. A term of any sort is called *closed* if it contains no variables of any sort.

Each closed individual term t denotes an element $[\![\mathrm{t}]\!]$ of V_0, and each $a \in V_0$ is denoted by a closed individual term $\bar{a}$, given by $\overline{(a,b)} = (\bar{a},\bar{b})$; thus $[\![\bar{a}]\!] = a$.

Each closed function term F of FS_0 denotes a function $[\![\mathrm{F}]\!]$ in $\mathcal{PR}$. We call F a *presentation of* $[\![\mathrm{F}]\!]$. The term F shows exactly how $[\![\mathrm{F}]\!]$ is built up from the basic functions by the compounding functionals.

Each closed class term S of FS_0 denotes a class $[\![\mathrm{S}]\!]$ in $\mathcal{INC}$. We call S a *presentation of* $[\![\mathrm{S}]\!]$. Again, the term S shows exactly how $[\![\mathrm{S}]\!]$ is obtained by the explicit and inductive class construction axioms.

[8] The operation $a \mapsto K(a)$, from individuals to functions, was not needed in $\mathcal{PR}$ since for each $a \in V_0$ we have $K(a) = K_a \in \mathcal{PR}$. However, if we just took K_0 in the axioms, we would not be able to define K(a) with 'a' variable. The latter is necessary if we are to prove that for each term $\mathrm{t}[\mathrm{x}], \exists \mathrm{f} \forall \mathrm{x}(\mathrm{fx} = \mathrm{t}[\mathrm{x}])$, as well as recursion with no parameters.

By a *presentation of a finitary inductive system* Σ (of signature σ), we mean a presentation of a pair $(\langle A_i\rangle_{i\leq m},\ \langle R_j\rangle_{j\leq p})$ of classes specifying (with σ) the closure conditions for Σ.

18. Primitive recursive and inductive completeness of FS_0. *C-InTm*(*C-FnTm*, *C-ClTm*) denotes the class of closed individual (function, class) terms. *C-InTm* and *C-FnTm* are obtained by a simultaneous inductive definition simply by omitting the basis condition for variables in §16; similarly for *C-ClTm*.

LEMMA. *If* $a, b \in V_0$ *and* $a \neq b$ *then* $EFS_0 \vdash \bar{a} \neq \bar{b}$.

THEOREM 1. *(i) If* $\mathrm{t} \in$ *C-InTm and* $[\![\mathrm{t}]\!] = a$ *then* $FS_0 \vdash \mathrm{t} = \bar{a}$.

(ii) If $\mathrm{F} \in$ *C-FnTm and* $[\![\mathrm{F}]\!]a = b$ *then* $FS_0 \vdash \mathrm{F}\bar{a} = \bar{b}$.

Proof. This is carried out by induction on the simultaneous generation of *C-InTm* and *C-FnTm*. In the cases that F has the form $\mathcal{R}(\mathrm{G}, \mathrm{H})$, we carry out a subsidiary induction on its argument $a \in V_0$.

COROLLARY. *(i) For each* $\mathrm{t} \in$ *C-InTm there exists a with* $FS_0 \vdash \mathrm{t} = \bar{a}$

(ii) For each $\mathrm{F} \in$ *C-FnTm and each a there exists b with* $FS_0 \vdash \mathrm{F}\bar{a} = \bar{b}$.

Note. The corollary may be proved directly without appeal to the semantic interpretations $[\![\cdot]\!]$.

THEOREM 2. *If* $\mathrm{S} \in$ *C-ClTm and* $a \in [\![\mathrm{S}]\!]$ *then* $FS_0 \vdash (\bar{a} \in \mathrm{S})$.

Proof. This is carried out by induction on the generation of *C-ClTm*. In the case that $\mathrm{S} = \mathcal{I}_2(\mathrm{T}, \mathrm{R})$ we have a subsidiary induction on the inductive generation of $[\![\mathrm{S}]\!]$ from $[\![\mathrm{T}]\!]$ by the rule $[\![\mathrm{R}]\!]$.

Similar results hold for EFS_0, when the terms are restricted to $L(EFS_0)$.

19. Functional and class abstraction. By the class of $\exists^+$-formulas of FS_0 we mean the smallest class which contains all equations $\mathrm{t}_1 = \mathrm{t}_2$, inequalities $\mathrm{t}_1 \neq \mathrm{t}_2$, as well as atomic formulas of the form $\mathrm{t} \in \mathrm{T}$ ($\mathrm{t}, \mathrm{t}_1, \mathrm{t}_2$ individual terms, and T a class term), and closed under $\wedge, \vee$, and $\exists \mathrm{x}$ applied to any individual variable.

THEOREM *(i) For each individual term* t *with free variables included in* $\{\mathrm{x}_1, \ldots, \mathrm{x}_n\}$ *we can find a closed function term* F_t *such that*

$$FS_0 \ \vdash \mathrm{F}_\mathrm{t}(\mathrm{x}_1, \ldots, \mathrm{x}_n) = \mathrm{t}\ .$$

(ii) For each function term G *with free variables included in* $\{\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{n}\}$ *we can find a closed function term* F_G *such that*

$$FS_0 \ \vdash \mathrm{F}_\mathrm{G}(\mathrm{x}_1, \ldots, \mathrm{x}_n, \mathrm{y}) = \mathrm{Gy} \ .$$

(iii) For each $\exists^+$*-formula* ϕ *with free variables included in* $\{\mathrm{x}_1, \ldots, \mathrm{x}_n\}$ *we can find a closed class term* S_ϕ *such that*

$$FS_0 \ \vdash (\mathrm{x}_1, \ldots, \mathrm{x}_n) \in \mathrm{S}_\phi \leftrightarrow \phi \ .$$

Proof. Parts (i) and (ii) proceed by simultaneous induction. For part (iii), the essential new point is closure under $\exists$. This comes from the following observation. Given any B, we can define $\{x|\exists y B(x,y)\}$ as $Q_1^{-1}X$ where X is the least class satisfying:

$$(x,0) \in X \text{ for all } x \ ; \quad \frac{(y,0) \in X}{(x,1) \in X} \ (x,y) \in B \ .$$

Remark. We may regard $\mathrm{F_t}$ as $\lambda(\mathrm{x}_1, \ldots, \mathrm{x}_n)\mathrm{t}$, $\mathrm{F_G}$ as $\lambda(\mathrm{x}_1, \ldots, \mathrm{x}_n, \mathrm{y})\mathrm{Gy}$, and S_ϕ as $\{(\mathrm{x}_1, \ldots, \mathrm{x}_n)|\phi\}$.

20. Proof-theoretical strength of FS_0. By formalizing the inductive definition of the class N of natural numbers (§10) and the treatment of $\mathcal{PR}_N$ in terms of $\mathcal{PR}$, we can interpret the system PRA of primitive recursive arithmetic in FS_0. Moreover, the fragment Σ^0_1-IA of (first-order) Peano Arithmetic based on PRA and the Σ^0_1-induction axiom is also contained in FS_0 under this interpretation. For, every Σ^0_1-formula is equivalent to an $\exists^+$-formula, and each such defines a class by §19. Since the induction axiom for N in the form

$$0 \in \mathrm{X} \wedge \forall \mathrm{x}(\mathrm{x} \in \mathrm{X} \rightarrow \mathrm{x}' \in \mathrm{X}) \rightarrow \mathrm{N} \subseteq \mathrm{X}$$

is a consequence of the induction axiom VII for the universe, it follows that we have the induction scheme for all $\exists^+$-formulas.

Now it can also be shown that FS_0 is interpretable in Σ^0_1-IA. The idea for this is that we interpret individual, function, and class variables all as ranging over ω, with the pairing and projection functions taken to be primitive recursive with $(\mathrm{x}, \mathrm{y}) \neq 0$. We interpret fx as $[\mathrm{f}](\mathrm{x})$ where the enumeration $[\mathrm{f}]$ of primitive recursive functions is defined in terms of the more general enumeration $\{\mathrm{z}\}$ of partial recursive functions. Finally $\mathrm{x} \in \mathrm{z}$ is interpreted as $\{\mathrm{z}\}(\mathrm{x})\downarrow$, i.e. as $\mathrm{x} \in \mathrm{W_z}(= \{\mathrm{u}|\exists \mathrm{y} \mathrm{T}_1(\mathrm{z}, \mathrm{u}, \mathrm{y})\})$.

These kinds of arguments lead to the following.

THEOREM. *FS_0 is of the same proof-theoretical strength as (Σ^0_1-IA), and hence of PRA.*[9]

Comparison with the system FM_0, and correction to [IPS]. The system FM_0 in [IPS] used only individual terms built up by pairing and projections, and class terms built up by class comprehension for $\exists^+$-formulas together with $\mathcal{I}_2$ inductive generation. FM_0 is easily seen to be a subsystem of FS_0, in fact already of EFS_0.[10] It was claimed in IPS that FM_0 contains the system Σ^0_1-IA. The idea was first to define a general recursion theory over the universe by means of an inductively defined 3-placed relation $\mathrm{xy} \simeq \mathrm{z}$, and then to obtain primitive recursion as a special case. However, the unicity property for $\simeq$,

$$\mathrm{xy} \simeq \mathrm{z}_1 \wedge \mathrm{xy} \simeq \mathrm{z}_2 \rightarrow \mathrm{z}_1 = \mathrm{z}_2 \ ,$$

in 3.5(vi) (p.102) of [IPS], does not seem to follow from FM_0 as asserted there. FM_0 can be expanded by a basic relation symbol $\simeq$ with the statements 3.5(i)–(vi) of [IPS] as axioms to give a system FM_0^* which *is* equivalent to Σ^0_1-IA. For, we can interpret $\mathrm{xy} \simeq \mathrm{z}$ via $\{\mathrm{x}\}(\mathrm{y}) \simeq \mathrm{z}$ in o.r.t. Instead, we have chosen here to incorporate the primitive recursion part of the recursion theory into the formalism of FS_0. This is more intuitive and closer to the needs of logical practice.

It would be of interest to determine the exact proof-theoretical strength of the systems FM_0 and EFS_0 .

21. Finitary inductively presented logics. In the most general sense, a formal logic is just a finitary inductive system Σ, and a *finitary inductively presented (f.i.p.) logic* is just a presented finitary inductive system, i.e. one represented by a class term S of FS_0 . By carrying out the arguments of §§8–9 in FS_0 , we can even take S in the normal form $\mathrm{Q}_0^{-1}\mathcal{I}_2(\mathrm{T},\mathrm{R})$ where T,R are primitive recursive class terms.

Some among the variety of logics that are met in practice have been mentioned in the introduction. These can all be regarded as f.i.p. logics in the above sense and can be reasoned about in FS_0 . Notions and results about wide classes of logics might be considered to be part of the subject of meta-logic; these can also be formulated in FS_0 at various levels of generality. For example, one might study closure

[9] Parsons [1970] proved that ($\Sigma^0_1 - IA$) and *PRA* are of the same proof-theoretical strength.

[10] The universe of FM_0 is U^* for an unspecified class U of urelements, with $0 \in \mathrm{U}$.

conditions on rules of inference (or consequence relations, as in Avron [1987]), the difference between derived rules and admissible rules, notions of schematic axioms and rules, interpretation of one logic into another, etc.

For illustrative purposes and to get quickly to Gödel's incompleteness theorems, we shall limit ourselves in the following to a very special case, namely logics based on many-sorted first-order classical predicate calculus with equality (in some sorts). A number of details for the single-sorted case have been given in [IPS] §6 (pp.114–119) and will not be repeated here.[11] First one defines (in FS_0) the classes Var, Const, Fun and Rel of variables, constants, function symbols and relation symbols of arbitrary arities; these are all given explicitly. (In the many-sorted case these are supplemented by the class Sort of sorts, and arities are sequences from Sort.) By a *language* is meant an arbitrary subclass L of Const∪Fun∪Rel containing the relation symbol $\mathrm{r}_{0,2}$ for equality. Note that L is treated as a variable class in FS_0 . Then one defines inductively Term(L), Atom(L) and Form(L), all of which have $\mathcal{PR}$ transitive predecessor functions and hence (by §12 and §15) are $\mathcal{PR}$ in L. Next one defines the general notions of being a free variable, and of being a term free for a variable in a formula, and the operation Sub of substitution of a term for a variable in a term or formula; Sub is defined by $\prec$-recursion and is primitive recursive. Finally one defines the ($\mathcal{PR}$) class LogAx of logical axioms, and takes LogAx(L)=LogAx∩Form(L), the class of *logical axioms in* L, which is $\mathcal{PR}$ in L. For the formulation of predicate calculus in [IPS], only two rules of inference were used: *modus ponens* (MP) and *universal generalization* (UG); these relations are in $\mathcal{E}$.

By an *axiomatic system* is meant a pair $\mathrm{S} = \langle \mathrm{L}, \mathrm{A} \rangle$ where L is a language and $\mathrm{A} \subseteq \mathrm{Form(L)}$. A is considered to be the class of "non-logical" *axioms of* S, denoted $\mathrm{A} = \mathrm{Ax(S)}$, and L is denoted L(S). Again $\mathrm{L}, \mathrm{A}, \mathrm{S}$ are treated as arbitrary (variable) classes in FS_0 , subject to the given restrictions. The class Proof(S) is defined as the class of derivation trees for the class $\mathrm{Prov(S)} = \mathcal{I}_2(\mathrm{LogAx(L)} \cup \mathrm{Ax(S)},\ \mathrm{MP} \cup \mathrm{UG})$; by §§14,15, Proof(S) is $\mathcal{PR}$ in S and

$$a \in \mathrm{Prov(S)} \leftrightarrow \exists \mathrm{d}[\mathrm{d} \in \mathrm{Proof(S)} \wedge \mathrm{a} = \mathrm{P}_2\mathrm{d}] \ .$$

We also write $\mathrm{Proof}_\mathrm{S}(\mathrm{d}, \mathrm{a})$ for $\mathrm{d} \in \mathrm{Proof(S)} \wedge \mathrm{a} = \mathrm{P}_2\mathrm{d}$, and $\mathrm{Prov}_\mathrm{S}(\mathrm{a})$ or $\mathrm{S} \vdash \mathrm{a}$ for $\mathrm{a} \in \mathrm{Prov(S)}$.

Examples of elementary meta-logical theorems about Prov(S) which can be proved in FS_0 for arbitrary (variable) S are the Deduction Theorem and the Finiteness Theorem.

[11] In IPS it was assumed that the class U of urelements contains eight basic symbols '0', 'v', 'c', 'f', 'r', '¬', '→', '∀'; these would here be replaced by '0',. . . , '7', resp.

An axiomatic system is said to be *inductively presented* if it is given by a specific closed class term $\mathbf{S}$ of FS_0 . In this case, all the notions leading up to Prov($\mathbf{S}$) are also f.i.p. $\mathbf{S}$ is said to be p.r.p. if it is given in the form $\mathbf{S} = \mathbf{F}^{-1}\{\bar{0}\}(= \{x|\mathbf{F}x = \bar{0}\})$ for a closed function term $\mathbf{F}$; then all the notions leading up to Proof($\mathbf{S}$) are also p.r.p., while Prov(S) is f.i.p.

22. Gödel's incompleteness theorems for finitary inductively presented extensions of FS_0 . With the notions of §21 suitably expanded to the many-sorted case, it is seen that FS_0 is itself a p.r.p. logic given by a $\mathcal{PR}$ class term $\mathbf{FS}_0$ of FS_0 .[12] Gödel's incompleteness theorems are here formulated for arbitrary finitary inductively presented extensions $S = \langle L, A\rangle$ of FS_0 , given by a closed class term $\mathbf{S} = \langle \mathbf{L}, \mathbf{A}\rangle$. Each member ϕ of $Form(L)$ is identified with an element of V_0. Then $\bar{\phi}$ is the canonical closed term of $L(FS_0)$ which denotes ϕ. The function Sb which associates with each ϕ and a the result $Sb(\phi, a) = \phi(\bar{a})$ of substituting $\bar{a}$ for the variable x (or v_0) in ϕ, is in $\mathcal{PR}$, presented by a function term Sb. In particular, given ϕ(x) we can form ψ(x) = ϕ(Sb(x, x)), so that for $\chi = \psi(\bar{\psi}) = Sb(\psi, \psi)$, we have

$$(1) \qquad FS_0 \ \vdash \chi \leftrightarrow \phi(\overline{\chi}) \ .$$

This gives the first ingredient of the incompleteness theorems, the construction of "self-referential" statements. The second ingredient is the inductive completeness of FS_0 from §18, which is here specialized to Prov($\mathbf{S}$):

$$(2) \qquad S \vdash \phi \ \text{ implies } \ FS_0 \ \vdash \mathrm{Prov}_{\mathbf{S}}(\overline{\phi}) \ .$$

For Gödel's *First Incompleteness Theorem*, we apply (1) to form $\chi_{\mathbf{S}}$ with

$$(3) \qquad FS_0 \ \vdash \chi_{\mathbf{S}} \leftrightarrow \neg\mathrm{Prov}_{\mathbf{S}}(\overline{\chi}_{\mathbf{S}}) \ .$$

Then by the usual argument we have:

THEOREM 1. *If $S = [\![\mathbf{S}]\!]$ is a consistent extension of FS_0 then $S \not\vdash \chi_{\mathbf{S}}$.*[13]

Let $\mathrm{Con}_{\mathbf{S}} = \neg\mathrm{Prov}_{\mathbf{S}}(\overline{\phi})$, for $\phi = \neg(\bar{0} = \bar{0})$, be the canonical consistency statement associated with the presentation $\mathbf{S}$ of S. Then what must be shown for Gödel's *Second Incompleteness Theorem* is:

THEOREM 2. $FS_0 \vdash \mathrm{Con}_{\mathbf{S}} \rightarrow \neg\mathrm{Prov}_{\mathbf{S}}(\overline{\chi}_{\mathbf{S}})$, *for $S = [\![\mathbf{S}]\!]$ extending FS_0 .*

[12] In fact, $Ax(FS_0)$ is finite.

[13] This is the first half of Gödel's First Incompleteness Theorem; the second half is that if S is correct for statements of the form $\mathrm{Prov}_{\mathbf{S}}(\overline{\phi})$ then $S \not\vdash \neg\chi_{\mathbf{S}}$.

For the proof of Theorem 2 we need to formalize (2) in FS_0 and for this more generally we need to formalize Theorem 2 of §18. The details of that require a more extended presentation than is possible here. Only one point should be noted. In the proof of Theorem 2 of §18 we are carrying out a double induction, first on the closed class terms of FS_0 and then on $[\![\mathcal{I}_2(\mathrm{A},\mathrm{B})]\!]$ for each inductive class term. However, for fixed **S** there are only a finite number of subterms that must be considered, and so we are reduced to a finite number of individual inductions. Alternatively, using the Normal Form Theorem of §9, we can reduce the inductive argument to a single one. Prior to that one must establish a formal version of the primitive recursive completeness theorem of FS_0 (Theorem 1 of §18); again, only a finite number of inductions need to be made for each specific function.

23. Where do we go from here? Returning to the three basic aims that we set for this work in the introduction—conceptual, pedagogical, and practical (computational)—what has been accomplished here lies mainly in providing a conceptual framework, with indications in the preceding paragraph how this would be spelled out for a pedagogically reasonable exposition of Gödel's incompleteness theorems. That should be carried out in detail and extended to include other results in the "arithmetization of metamathematics" and proof theory, for example concerning provable reflection principles (cf. Kreisel, Levy [1968], Smorynski [1977]), and proof-theoretical conservation results (cf. Feferman [1988]). The use of infinitary methods in proof theory for which finitary formalizations can be given requires particular attention (cf. Feferman [1967], pp.93–95, and Schwichtenberg [1977]). Another step into the transfinite, for which finitary treatments can be given (at least, in part) is provided by the iteration of (non-provable) reflection principles in recursive progressions of theories (Feferman [1962], [1964]). I believe that FS_0 provides a natural framework in which to re-develop these topics in a proper way. Finally, there should be an extension of FS_0 to a theory of *infinitely* branching trees (with primitive recursive functions and inductive classes of such) which would serve as a natural framework in which to formalize essentially infinitary logics, using inductive systems with infinitary closure conditions.

In another direction, one can form a non-finitist extension FS_1 of FS_0 (analogous to the extension FM_1 of FM_0 in [IPS]) by adding complementation as an operation on classes; the resulting system is a conservative extension of PA. As sketched in [IPS] §8, much countable model theory can be formalized in FS_1, in fact already in $FS_0 + WKL$ (Weak König's Lemma). The conservation result of Friedman for Σ^0_1-IA+WKL over PRA gives conservation of $FS_0 + WKL$ over PRA, as can be established directly by finitist methods (e.g. those of Sieg [1985]). But already in FS_0 and closely related systems one can develop non-trivial parts of countable model theory, thus generalizing portions of recursive model theory. Systematic work in this direction is being carried out by my doctoral student, Paolo Mancosu.

Finally, on the computational front, the work of §15 establishes implementability in principle, but clearly this is only the beginning. Whether implementation is feasible and what its value might be can only be judged by actually trying to carry it out. Good test cases for general topics are provided by the examples considered for the ELF project (cf. Avron *et al* [1987]). Elaboration of the preceding section could provide another test case, and here one has the work of Shankar [1985] for an interesting comparison.

References

A. Avron [1987] *Simple consequence relations*, **LFCS Report Series** 87–30, Laboratory for Foundations of Computer Science, University of Edinburgh.

A. Avron, F.A. Honsell, and I.A. Mason [1987], *Using typed lambda calculus to implement formal systems on a machine*, **LFCS Report Series** 87–31, Laboratory for Foundations of Computer Science, University of Edinburgh.

J. Barwise and S. Feferman [1985] (eds.) **Model-theoretic logics**, (Springer-Verlag, Berlin).

S. Feferman
[1960] *Arithmetization of metamathematics in a general setting*, **Fundamenta mathematica** 49, 35–92.
[1962] *Transfinite recursive progressions of axiomatic theories*, **J. Symbolic Logic** 27, 259–316.
[1964] *Systems of predicative analysis*, **J. Symbolic Logic** 29, 1–30.
[1967] *Lectures on proof theory*, **Lecture Notes in Mathematics** 70, 1–107.
[1982] *Inductively presented systems and the formalization of meta-mathematics*, in **Logic Colloquium '80** (D. van Dalen *et al* eds., North-Holland, Amsterdam) 95–128.
[1988] *Hilbert's program relativized: proof-theoretical and foundational reductions*, **J. Symbolic Logic** 53, 364–384.

M. Fitting [1987], **Computability theory, semantics and logic programming**, (Oxford University Press, New York).

K. Gödel [1986], **Collected Works, Volume I. Publications 1929–1936** (S. Feferman *et al*, eds., Oxford University Press, New York).

R. Harper, F. Honsell and G. Plotkin [1987], *A framework for defining logics*, Proc. Second Annual Conference on Logic and Computer Science, Cornell 1987 (to appear).

G. Kreisel [1965], *Mathematical logic*, in **Lectures on Modern Mathematics III** (T.L. Saaty, ed., Wiley, New York) 95–195.

G. Kreisel and A. Levy [1968], *Reflection principles and their use for establishing the complexity of axiomatic systems*, **Zeitschrift f. Mathematische Logik u. Grundlagen d. Mathematik** 14, 97–142.

Y. Moschovakis
[1969], *Abstract first-order computability* I, **Trans. Amer. Math. Soc.** 138, 427–464.
[1984], *Abstract recursion as a foundation for the theory of algorithms*, **Lecture Notes in Mathematics** 1104, 289–364.

C. Parsons [1970], *On a number-theoretic choice schema and its relation to induction*, in **Intuitionism and Proof Theory** (eds. J. Myhill *et al*, North-Holland, Amsterdam) 459–474.

E. Post [1943], *Formal reductions of the general combinatorial decision problem,* **Amer. J. Math. 65**, 197–214.

H. Schwichtenberg [1977], *Proof theory: some applications of cut-elimination*, in **Handbook of Mathematical Logic** (J. Barwise ed., North-Holland, Amsterdam) 867–895.

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

N. Shankar [1985], *Towards mechanical metamathematics*, **J. Automated Reasoning** 1, 407–434.

C. Smorynski [1977], *The incompleteness theorems*, in **Handbook of Mathematical Logic** (J. Barwise ed., North-Holland, Amsterdam) 821–865.

R. Smullyan [1961], **Theory of formal systems** (Princeton University Press, Princeton).

Added March 10, 1992

The following references bear on two of the proposals in §23, namely for the finitary formalization of infinitary methods in proof theory, and for the implementation of FS_0, resp.

W. Buchholz [1991], *Notation systems for infinitary derivations,* **Archive for Mathematical Logic 30**, 277–296.

S. Matthews, A. Smaill, and D. Basin [1991], *Experience with FS_0 as a logical framework,* (preprint) for the Second Logical Frameworks Basic Research Action Workshop, Edinburgh, July 1991.

The work of P. Mancosu mentioned in §23 resulted in a Ph.D. dissertation at Stanford University in 1989, and was later published under the same title as:

P. Mancosu [1991], *Generalizing classical and effective model theory in theories of operations and classes,* **Annals of Pure and Applied Logic 52**, 249–308.