ASSIGNMENT OF ORDINALS TO TERMS FOR PRIMITIVE RECURSIVE FUNCTIONALS OF FINITE TYPE *

W. A. HOWARD

Introduction. Gentzen [2] showed that the consistency of first order (classical or intuitionistic) arithmetic can be proved by methods which are finitistic except for the use of the descending chain principle for the ordinals less than ε_0. On the other hand, Gödel [4] gave an interpretation of first order intuitionistic arithmetic $\mathscr{H}$ in a quantifier-free theory $\mathscr{T}$ of primitive recursive functionals of finite type, thereby reducing the consistency of $\mathscr{H}$ to that of $\mathscr{T}$.

In the following, the terms of $\mathscr{T}$ will be given in the λ-calculus provided with a finite type structure. We establish a direct connection between $\mathscr{T}$ and ε_0 by giving an assignment of ordinals less than ε_0 to the terms of $\mathscr{T}$ with the property that the reduction of a term (in the sense of λ-conversion) lowers the corresponding ordinal. Unfortunately we have been able to do this only for *restricted* reductions; i.e., reductions arising essentially from the contraction of closed subterms (§ 1). However, in § 4 we extend our result to the case of arbitrary reductions by the use of non-unique assignments of ordinals to terms. This allows us to assign to every reduction sequence of terms

$$A_1 \text{ red } A_2 \text{ red } \ldots \text{ red } A_n \text{ red } \ldots$$

a corresponding descending sequence of ordinals

$$a_1 > a_2 > \ldots > a_n > \ldots.$$

If the ordinals less than ε_0 and the terms of $\mathscr{T}$ are enumerated in a natural way, our discussion can be formalized in Skolem (free variable, first order, primitive recursive) arithmetic. Thus the 'computability' of the

* Mention should be made of the paper of L. E. Sanchis, Functionals defined by recursion, *Notre Dame J. Formal Logic* **8** (1967) 161–174.

terms of $\mathscr{T}$ in the sense of Tait [5], namely the reducibility of each term of $\mathscr{T}$ to normal form, follows from the descending chain principle for the ordinals less than ε_0. Tait uses combinator-terms rather than λ-terms but it is easy to model combinators in the system of λ-terms.

Our treatment is developed in a rather general form because we intend in the future to extend our results to certain extensions of $\mathscr{T}$: in particular to type-zero-bar-recursive functionals. For extensions of $\mathscr{T}$, the ordinals less than ε_0 may no longer be adequate. Indeed, we shall show in the future that for the analysis of the extension of $\mathscr{T}$ just mentioned, the appropriate ordinals are those less than Bachmann's ordinal $\varphi_c(0)$, $c = \varepsilon_{\Omega+1}$ (as developed in H. Gerber's paper [3]).

1. Terms. The terms of $\mathscr{T}$ are obtained from the *prime terms* of $\mathscr{T}$ by applying the following operations finitely often.

(i). Application: from terms A and B get AB, with the restriction on the pair A, B given in clause (a) under ***Type levels*** below. The *interpretation* of AB is the value of the functional A when applied to the argument B.

(ii). λ-abstraction: from any variable X and term A get $\lambda X.A$.

Type levels. The general situation to which the method of the present paper applies is: one has a system of terms, generated as just described, in which it is possible to assign to each term A a non-negative integer level(A) in such a way that whenever AB is well-formed, level(A) > level(B) and level(A) $\geqslant$ level(AB). In the case of $\mathscr{T}$ this assignment is obtained by first assigning to each term A a *type symbol*, called the type of A, as follows. Type symbols are generated from a prime type symbol 0 by means of the operation: from type symbols σ and τ get the type symbol $(\sigma)\tau$. After type symbols are assigned to the prime terms of $\mathscr{T}$ (see ***Prime terms*** below) the assignment of type symbols to the remaining terms of $\mathscr{T}$ is determined by the following two clauses:

(a). If A has type $(\sigma)\tau$ and B has type σ, then AB is well-formed and has type τ.

(b). If the variable X has type σ and A has type τ, then $\lambda X.A$ is well-formed and has type $(\sigma)\tau$.

The *level* of a type symbol is defined inductively by the following two clauses:

(i). The level of the type symbol 0 is zero.

(ii). The level of $(\sigma)\tau$ is the maximum of $1+\text{level}(\sigma)$ and level(τ).

It is easily seen that every type symbol σ other than 0 has the form $(\sigma_1)\ldots(\sigma_n)0$ and that the level of σ is the maximum of

$$1+\text{level}(\sigma_1),\ldots,1+\text{level}(\sigma_n).$$

The level of a term A is defined to be the level of the type symbol assigned to A.

The *length* of a term is defined by the following clauses:

(i). Prime terms have length 1.

(ii). The length of AB is $\text{length}(A)+\text{length}(B)$.

(iii). The length of $\lambda X.A$ is $1+\text{length}(A)$.

Prime terms. The prime terms of $\mathscr{T}$ are as follows:

(i). Variables of every type.

(ii). A numeral $\boldsymbol{n}$ of type 0 corresponding to each non-negative integer n.

(iii). A constant $\mathfrak{z}$ of type (0)0 for the successor function.

(iv). For each type symbol σ, a constant R of type $(0)((0)(\sigma)\sigma)(\sigma)\sigma$ for a 'primitive recursion functional'.

(v). For each type symbol σ and non-negative integer n, a constant R^n of type $((0)(\sigma)\sigma)(\sigma)\sigma$ for a 'restricted primitive recursion functional'.

Free and bound variables. In the generation of terms by the two processes of application and λ-abstraction, the (possibly vacuous) occurrences of a variable X in a term are said to be *free* until the stage is reached where the operation λX is applied; thus the free occurrences of X in a term A become bound occurrences in $\lambda X.A$. A term is said to be *closed* if it has no non-vacuous occurrences of a free variable.

Subform and subterm. In order to define the notion of subform, let $|B|$ denote the set of subforms of a term B. Then

(i). $|B| = \{B\}$ if B is prime.

(ii). $|AB| = |A| \cup |B| \cup \{AB\}$.

(iii). $|\lambda X.A| = |A| \cup \{\lambda X.A\}$.

The notion of *subterm* is defined similarly except that clause (iii) (where $|A|$ now stands for the set of subterms of A) is replaced by the clause

(iii)*. $|\lambda X.A| = (|A| \cup \{\lambda X.A\}) - \mathfrak{T}$,

where $\mathfrak{T}$ denotes the set of all those subterms of A which contain a (non-vacuous) free occurrence of X.

Notation. For terms A, B and C, ABC denotes $(AB)C$. More generally, $A_1A_2A_3 \ldots A_k$ denotes $(\ldots(A_1A_2)A_3 \ldots)A_k$. The result of substituting a term B for all free occurrences of a variable X in A is denoted by $[B/X]A$.

Contractions. In the following schemata, A contr B means: A contracts into B.

(i). λ-contraction: $(\lambda X.A)B$ contr $[B/X]A$ so long as no free occurrence of a variable in B becomes bound in $[B/X]A$.

(ii). $\mathfrak{S}n$ contr $n+1$.

(iii). Rn contr R^n.

(iva). R^0HG contr G.

(ivb). $R^{n+1}HG$ contr $Hn(R^nHG)$.

Restricted reductions. A red B if B arises from A by contracting one occurrence of a sub*term* of A.

General reductions. A red B if B arises from A by contracting one occurrence of a sub*form* of A.

Motivation for our formulation. In case our formulation of the terms of $\mathscr{T}$ appears peculiar, the following motivation should be borne in mind. As mentioned in the introduction, the results of the present paper, together with the descending chain principle for the ordinals less than ε_0, yield the computability of the terms of $\mathscr{T}$; i.e., the reducibility of each term A to an irreducible term: the so-called normal form of A. Thus in the present paper we are concerned with an analysis of the *terms* of $\mathscr{T}$, whereas $\mathscr{T}$ as a *formal system* contains equations between terms and propositional combinations of such equations. Once the computability of the closed terms of $\mathscr{T}$ has been established, *then* intensional equality can be introduced: two terms are said to be intensionally equal if they have normal forms which are congruent (i.e., the same except for changes of bound variables). For this purpose it is necessary to be assured that two normal forms of the same term are congruent. Tait [5] achieves this by imposing a rule which uniquely determines for each term A the subterm of A that is allowed to be contracted in reducing A. For more general reduction procedures, the uniqueness (up to congruence) of the normal form of a term is assured by a well-known theorem of Church and Rosser.

The notion of intensional equality, just described, provides a truth valuation for the closed formulae of $\mathscr{T}$ (i.e. propositional combinations of equations between closed terms). Thus one gets a proof-theoretic analysis

of the system $\mathscr{T}^c$ obtained by restricting $\mathscr{T}$ to closed formulae. Finally one gets a proof-theoretic analysis of $\mathscr{T}$ itself from the observation that a proof in $\mathscr{T}$ can be transformed into a proof in $\mathscr{T}^c$ by replacing all free variables by suitable constants, starting from the end of the proof and working back, the induction rule in $\mathscr{T}$ being eliminated by the calculation of numerical constants.

In the discussion of $\mathscr{T}$ (and its extensions), we permit any methods formalizable in Skolem (free variable, first order, primitive recursive) arithmetic. Hence sets of axioms of the form A contr B are permissible so long as the number of B (in a natural enumeration of terms) is a primitive recursive function of the number of A.

In view of the motivation just described, the contractions given above for the constants R and R^n are sufficient; i.e., it is not necessary to require the stronger contraction $\lambda x.R(\mathfrak{S}x)HG$ contr $\lambda x.Hx(RxHG)$.

2. A theory $\mathscr{E}$ of expressions

Introduction. In the present section we introduce a theory $\mathscr{E}$ whose objects are *expressions*. Expressions are generated from constants and from variables x_j^r by two operations: from expressions f and g obtain $f+g$ and (f, g). By a *vector of level n* is meant an $n+1$-tuple $\boldsymbol{h} = \langle h_0, \ldots, h_n \rangle$ of expressions h_i. We introduce operations $\square$ and δ^r which produce vectors $\boldsymbol{f} \square \boldsymbol{g}$ and $\delta^r \boldsymbol{f}$ from vectors $\boldsymbol{f}$, $\boldsymbol{g}$.

The motivation of the present section is as follows. In § 3 we shall define a mapping from the terms of $\mathscr{T}$ into vectors which assigns to each term of type level n a vector of the same level. In particular, to the variable X^r of type level n is assigned the vector $\boldsymbol{x}^r = \langle x_0^r, \ldots, x_n^r \rangle$, it being assumed that the variables of $\mathscr{T}$ have been enumerated: $X^0, X^1, \ldots, X^r, \ldots$. More generally, the presence of a free variable X^r in a term H is reflected by the presence of the variables $x_0^r, \ldots, x_n^r$ in some of the components h_i of the vector $\boldsymbol{h}$ assigned to H. It is crucial, however, that x_j^r is not contained in h_k for $k > j$ (i.e. $\boldsymbol{h}$ belongs to the class $\boldsymbol{C}$ defined in the present section). If $\boldsymbol{f}$ and $\boldsymbol{g}$ are assigned to F and G, respectively, then to FG is assigned the vector $\boldsymbol{h}$, of the proper level, whose components are equal to the corresponding components of $\boldsymbol{f} \square \boldsymbol{g}$. If $\boldsymbol{h}$ is assigned to H, then $\delta^r \boldsymbol{h}$ is assigned to $\lambda X^r.H$. Thus δ^r is a kind of 'abstraction' operator which maps a vector $\boldsymbol{h}$ which contains the variables x_j^r into another vector $\delta^r \boldsymbol{h}$ which does not contain the variables x_j^r.

As will be mentioned in the present section, the theory $\mathscr{E}$ of expressions has *interpretations* in which the constants are ordinals and the variables range over a set of ordinals (or ordinal notations). In such interpretations an expression is interpreted as a function (more precisely: intensional function) of the variables that it contains.

In § 3 by the expression assigned to H is meant the initial component h_0 of the vector $\boldsymbol{h}$ assigned to H. When expressions are interpreted by means of ordinals the expression h_0 becomes an ordinal if H is closed, and this is taken to be the ordinal assigned to H.

The theory $\mathscr{E}$. We construct expressions from: constants, variables x_j^r, the symbol $+$, and the symbol $(\cdot, \cdot)$ as follows:

(i). A constant is an expression.

(ii). A variable x_j^r is an expression.

(iii). If f and g are expressions, so are $f+g$ and (f, g).

It is assumed that among the constants of $\mathscr{E}$ there are three constants denoted by 0, 1 and ω.

The theory $\mathscr{E}$ is an axiomatic theory of a relation $\prec$ between expressions. Equality between expressions is treated axiomatically: it is assumed to be reflexive and to obey the replacement axiom. $f \succ g$ means $g \prec f$. $f \preccurlyeq g$ means: $f \prec g$ or $f = g$.

Axioms for $\mathscr{E}$

2.1. If $f \prec g$ and $g \prec h$, then $f \prec h$.

2.2. If $f \prec g$, then $f \neq g$.

2.3. $f+g = g+f$; $f+(g+h) = (f+g)+h$.

2.4. If $f \prec g$, then $f+h \prec g+h$.

2.5. $f+g = f$ if and only if $g = 0$.

2.6. $0 \preccurlyeq f$; $0 \prec 1 \prec \omega$.

2.7. If $f \prec \omega$ and $g \prec \omega$, then $f+g \prec \omega$.

2.8. $(f, g+h) = (f, g)+(f, h)$.

2.9. If $g \prec c$ and $h \prec c$, then $(g,f)+(h,f) \preccurlyeq (c,f)$.

2.10. If $f \prec g$, then $(h,f) \prec (h, g)$.

2.11. If $f \prec g$ and $h \neq 0$, then $(f, h) \prec (g, h)$.

2.12. $(0,f) = f$.

2.13. $(f, (g, h)) = (f+g, h)$.

From 2.5 and 2.8 it is easy to prove: $(f, 0) = 0$. From 2.4, 2.5 and 2.9 it is easy to prove

2.14. If $g \succ 0$ and $h \succ 0$, then $(g,f)+(h,f) \preccurlyeq (g+h,f)$.

Interpretation of $\mathscr{E}$. For the analysis of primitive recursive functionals in § 3, the following interpretation of $\mathscr{E}$ is used. The variables range over ordinals (or ordinal notations) less than ε_0; expressions are interpreted as functions, of the variables contained in them, in a manner to be described presently; $a \prec b$ is interpreted as meaning: $a < b$ (the ordinary ordering of the ordinals) for all values of the variables. $a+b$ is interpreted as the natural (i.e. Hessenberg) sum $a \# b$ which is defined as follows (Bachmann [1]):

Represent a and b in Cantor normal form $a = \omega^{a_1} + \ldots + \omega^{a_n}$ and $b = \omega^{b_1} + \ldots + \omega^{b_m}$, where $a_1 \geqslant \ldots \geqslant a_n$ and $b_1 \geqslant \ldots \geqslant b_m$; then

$$a \# b = \omega^{c_1} + \ldots + \omega^{c_{n+m}},$$

where the sequence $c_1 \geqslant \ldots \geqslant c_{n+m}$ is a rearrangement of the sequence $a_1, \ldots, a_n, b_1, \ldots, b_m$.

To define (a, b) in the case $b \neq 0$, represent b in Cantor normal form to the base 2: $b = 2^{b_1} + \ldots + 2^{b_n}$, where $b_1 > \ldots > b_n$. Then take (a, b) to be $2^{c_1} + \ldots 2^{c_n}$, where $c_i = a \# b_i$ $(1 \leqslant i \leqslant n)$. Finally take $(a, 0)$ to be 0.

It is easy to see that this interpretation satisfies the axioms 2.1–2.13.

Expression vectors. If $f_0, \ldots, f_n$ are expressions, the $n+1$-tuple $f = \langle f_0, \ldots, f_n \rangle$ is called a *vector of level n*; and for $0 \leqslant i \leqslant n$ the expression f_i is called the ith *component* of f. We also use $(f)_i$ to denote the ith component of f. If $i >$ level(f) then $(f)_i$ is defined to be 0. We shall often write f_i for $(f)_i$. We define $f+g$ to be the vector h of level max {level(f), level(g)} such that $h_i = f_i + g_i$ $(0 \leqslant i \leqslant$ level$(h))$.

The operation $f \square g$. Let n denote max {level(f), level(g)}. Then $f \square g$ is defined to be the vector $h = \langle h_0, \ldots, h_n \rangle$ such that

$$h_n = f_n + g_n,$$

$$h_i = (h_{i+1}, f_i + g_i) \quad \text{for } 0 \leqslant i \leqslant n.$$

Clearly $f \square g = g \square f$, by axiom 2.3.

The following four lemmas are easy to prove from axioms 2.1–2.13 by downward induction on i.

LEMMA 2.1. *If $f_i \succ 0$ for all $i \leqslant n$, then $(f \square g)_i \succ 0$ for all $i \leqslant n$.*

LEMMA 2.2. *$(f \square g)_i \succcurlyeq f_i$ for all i.*

LEMMA 2.3. *Assume* $\text{level}(f) = \text{level}(g) = n$, *and* $f_i \succcurlyeq g_i$ *for* $0 \leqslant i \leqslant n$. *Then* $(f \square h)_i \succcurlyeq (g \square h)_i$ *for all* i.

LEMMA 2.4. *Under the assumption of lemma* 2.3 *and the additional assumption* $f_i \succ g_i$ *for* $0 \leqslant i \leqslant k$, *some* $k \leqslant n$, *we have*: $(f \square h)_i \succ (g \square h)_i$ *for* $0 \leqslant i \leqslant k$.

We now prove:

LEMMA 2.5. *Assume* $\text{level}(f) = \text{level}(g) = n+1 > \text{level}(h)$ *and* $f_i \succ 0$, $g_i \succ 0$ *for* $0 \leqslant i \leqslant n+1$. *Then*

$$(f \square h)_i + (g \square h)_i \preccurlyeq ((f+g) \square h)_i \quad \textit{for } 0 \leqslant i \leqslant n+1.$$

PROOF. The proof is by downward induction on i. The lemma is clearly true for $i = n+1$. To obtain the induction step, let a, b and c_i denote $f \square h$, $g \square h$ and $f_i + g_i + h_i$, respectively. Observe that by lemma 2.1 $a_{i+1} \succ 0$ and $b_{i+1} \succ 0$ for all $i < n+1$. Hence by 2.14

$$(a_{i+1}, c_i) + (b_{i+1}, c_i) \preccurlyeq (a_{i+1} + b_{i+1}, c_i).$$

It is easy to see that

$$(f \square h)_i + (g \square h)_i \prec (a_{i+1}, c_i) + (b_{i+1}, c_i),$$

whereas by induction hypothesis

$$a_{i+1} + b_{i+1} \preccurlyeq ((f+g) \square h)_{i+1},$$

so

$$(a_{i+1} + b_{i+1}, c_i) \preccurlyeq ((f+g) \square h)_i.$$

Notation. kf denotes $f + \ldots + f$ (k summands).

LEMMA 2.6. *Let the assumptions be as in lemma* 2.5. *Let* d *be a vector such that* $2f_{n+1} + 2g_{n+1} \prec d_{n+1}$ *and* $f_i + g_i \preccurlyeq d_i$ *for all* $i \leqslant n$. *Then*

$$2((f \square h) \square (g \square h))_i \prec (d \square h)_i \quad \textit{for all } i \leqslant n+1.$$

PROOF. Let a, b and c_i denote $f \square h$, $g \square h$ and $f_i + g_i + h_i$, respectively. Let e denote $a \square b$. The induction hypothesis is $2e_{i+1} \prec (d \square h)_{i+1}$ for a given $i < n+1$. We must prove $2e_i \prec (d \square h)_i$. Observe that

$$a_i + b_i = (a_{i+1}, f_i + h_i) + (b_{i+1}, g_i + h_i).$$

Hence

$$a_i + b_i \prec (a_{i+1}, c_i) + (b_{i+1}, c_i).$$

But $e_i = (e_{i+1}, a_i+b_i)$. Hence by use of the axioms 2.10, 2.8 and 2.13 we conclude

$$e_i \prec (e_{i+1}+a_{i+1}, c_i)+(e_{i+1}+b_{i+1}, c_i).$$

By lemma 2.1 $b_{i+1} \succ 0$. Also $e_{i+1} \succcurlyeq a_{i+1}+b_{i+1}$. Hence $e_{i+1} \succ a_{i+1}$. Similarly $e_{i+1} \succ b_{i+1}$. Hence $e_i \prec (2e_{i+1}, c_i)$ by axiom 2.9. Hence $2e_i \prec ((\boldsymbol{d} \sqcap \boldsymbol{h})_{i+1}, c_i)$ by the induction hypothesis and axiom 2.9. The desired result $2e_i \prec (\boldsymbol{d} \sqcap \boldsymbol{h})_i$ now follows from $c_i \preccurlyeq d_i+h_i$.

The classes C_i and C. Recall that the variables occurring in *expressions* are taken from a list of variables x_i^r. We now define the classes C_i of expressions by four clauses.

Starting clauses:

(i). If the expression h contains no variables, then h is in C_i.

(ii). For every r, the variable x_i^r is in C_i.

Inductive clauses:

(iii). If f and g are in C_i, then so is $f+g$.

(iv). If f is in C_{i+1} and g is in C_i, then (f, g) is in C_i.

A crucial property of expressions in C_i is given by the following lemma.

LEMMA 2.7. *If h is in C_i then h contains no variable x_j^r such that $j < i$.*

PROOF. By induction on the number of applications of clauses (iii) and (iv) in the definition of the classes C_i.

The class C is defined to consist of all vectors $\boldsymbol{h}$ such that $(\boldsymbol{h})_i$ is in C_i $(0 \leqslant i \leqslant \text{level}(\boldsymbol{h}))$.

LEMMA 2.8. *If f and g are in C, then so is $f \sqcap g$.*

PROOF. Immediate from clauses (iii) and (iv) in the definition of the classes C_i.

Notation. $[e/x_j^r]h$ denotes the result of substituting e for all occurrences of x_j^r in the expression h.

LEMMA 2.9. *Suppose e is in C_j. Then the operation of substituting e for x_j^r transforms each class C_i into itself* $(i = 0, 1, 2, \ldots)$.

PROOF. We must prove the assertion: for all i and all h, if h is in C_i then $[e/x_j^r]h$ is in C_i. It is easy to verify this assertion if h is as in clauses (i) or (ii), and then prove the assertion by induction on the number of applications of clauses (iii) and (iv) in the formation of h.

The operation δ^r. It will now be assumed that by means of a function $n(r)$ we have obtained a list of vector variables $\boldsymbol{x}^r = {}^{\mathbf{p}}\langle x_0^r, \ldots, x_{n(r)}^r\rangle$ $(r = 0, 1, 2, \ldots)$. To each h in $\bigcup C_i$ we associate a vector $\delta^r h$ in $\boldsymbol{C}$, such that $\delta^r h$ has level $n(r)+1$ and does not contain any component of $\boldsymbol{x}^r$, as follows. In order to avoid ambiguity in clause (a), below, we shall regard h in C_i as being completely given only when the particular C_i, to which h belongs, is specified.

Starting clauses:

(a). If h is in C_i and contains no component of $\boldsymbol{x}^r$, then $\delta^r h$ is the vector of level $n(r)$ such that $(\delta^r h)_i = h+1$ and $(\delta^r h)_j = 1$ when $j \neq i$, $0 \leqslant j \leqslant n(r)+1$.

(b). If h is x_i^r, then $(\delta^r h)_j = 1$ $(0 \leqslant j \leqslant n(r)+1)$.

Inductive clauses:

(c). If h contains a component of $\boldsymbol{x}^r$ and $h = f+g$, where f and g are in C_i, then $\delta^r h = \delta^r f+\delta^r g$.

(d). If h contains a component of $\boldsymbol{x}^r$ and $h = (f, g)$, where f is in C_{i+1} and g is in C_i, then

$$(\delta^r h)_j = (\delta^r f)_j+(\delta^r g)_j \quad \text{if } 0 \leqslant j \leqslant n(r),$$

and

$$(\delta^r h)_j = 2(\delta^r f)_j+2(\delta^r g)_j+1 \quad \text{if } j = n(r)+1.$$

We also define δ^r as acting on *vectors* $\boldsymbol{h} = \langle h_0, \ldots, h_p\rangle$ in $\boldsymbol{C}$ as follows:

$$(\delta^r \boldsymbol{h})_j = (\delta^r h_0)_j+ \ldots +(\delta^r h_p)_j \quad \text{if } 0 \leqslant j \leqslant n(r)+1,$$

and

$$(\delta^r \boldsymbol{h})_j = h_j+1 \quad \text{if } n(r)+1 < j \leqslant p.$$

By lemma 2.7 $\delta^r h$ contains no component of $\boldsymbol{x}^r$.

LEMMA 2.10. *Suppose e is in C_j and contains no component of $\boldsymbol{x}^r$. Suppose $s \neq r$. Then for any h in $\bigcup C_i$*

$$\delta^r[e/x_j^s]h = [e/x_j^s]\delta^r h.$$

PROOF. By induction on the number of applications of clauses (iii) and (iv) in the definition of δ^r.

COROLLARY. *Suppose $\boldsymbol{e}$ is in $\boldsymbol{C}$ and contains no component of $\boldsymbol{x}^r$. Suppose $s \neq r$. Then for any $\boldsymbol{h}$ in $\boldsymbol{C}$*

$$\delta^r[\boldsymbol{e}/\boldsymbol{x}^s]\boldsymbol{h} = [\boldsymbol{e}/\boldsymbol{x}^s]\delta^r \boldsymbol{h}.$$

(We assume level$(\boldsymbol{x}^s)$ = level$(\boldsymbol{e})$.)

LEMMA 2.11. *Let e be a vector of level $n(r)$ and assume h is in C_i. Then* $((\delta^r h) \square e)_i \succ [e/x^r]h$.

PROOF. By induction on the number of applications of clauses (iii) and (iv) in the definition of δ^r. Clause (iii) is handled by lemma 2.5. Clause (iv) is handled by lemma 2.6. Namely, suppose $h = (f, g)$, where f is in C_{i+1} and g is in C_i. Denote $(\delta^r f) \square e$ and $(\delta^r g) \square e$ by u and v, respectively. Then $((\delta^r h) \square e)_i \succcurlyeq (u \square v)_i$ by lemma 2.6. But $(u \square v)_i \succcurlyeq (u_{i+1}, v_i) \succ (f, g)$, since $u_{i+1} \succ f$ and $v_i \succ g$ by induction hypothesis.

COROLLARY. *If h is in C and e has level $n(r)$, then $((\delta^r h) \square e)_i \succ ([e/x^r]h)_i$ for all $i \leqslant level(h)$.*

LEMMA 2.12. *Let a, b and h be expressions. Suppose $a \succ b$. Then* $[a/x_i^r]h \succcurlyeq [b/x_i^r]h$.

PROOF. By induction on the number of times the operations (iii) under ***Expressions*** above are used in building up h.

LEMMA 2.13. *Let a, b and h be expressions. Suppose $a \succ b$. Suppose further that h is in C_0 and that x_0^r is contained in h non-vacuously. Then* $[a/x_0^r]h \succ [b/x_0^r]h$.

PROOF. By induction on the number of times the operations (iii) and (iv) under ***The classes C_i and C*** above are used in building up h. In particular, when h is (f, g), where $f \in C_1$ and $g \in C_0$, we know by lemma 2.7 that x_0^r is contained non-vacuously in g; so axiom 2.10 can be applied.

3. Assignment to terms of $\mathscr{T}$. The main discussion in this and the following section applies not only to $\mathscr{T}$ but to any extension $\mathscr{U}$ of $\mathscr{T}$. Of course we treat only the contractions given in § 1; i.e. λ-contractions and the contractions peculiar to $\mathscr{T}$. It is assumed that the terms of $\mathscr{U}$ have been provided with type levels satisfying the conditions given in § 1; namely, whenever FG is well-formed, then $\text{level}(F) > \text{level}(G)$ and $\text{level}(F) \geqslant \text{level}(FG)$. Indeed, more generally, we need make no restriction on the formation of terms FG so long as we allow the contraction of FG only when the above conditions on the type levels are satisfied.

If a term H of $\mathscr{U}$ is not of the form FG or $\lambda X.F$, then H is said to be prime. In the present section we shall make assignments of vectors to the prime terms of $\mathscr{T}$ and shall give prescriptions for passing from the assignment of vectors to F and G to the assignment of vectors to FG and $\lambda X.F$. Hence for our assignment to be complete it is necessary to have, in addition,

a prescription for assigning vectors to the prime terms of $\mathscr{U}$ which are not prime terms of $\mathscr{T}$. For the purpose of this and the following section we need only assume that such a prescription exists and satisfies the condition that if $\boldsymbol{h}$ is assigned to the prime term H of level n, then $h_i \succ 0$ for all $i \leqslant n$.

It is assumed that the variables of $\mathscr{U}$ have been enumerated: $X^0, X^1, \ldots, X^r, \ldots$, and that the vector variables $\boldsymbol{x}^r$ of § 2 have been listed in such a way that $\boldsymbol{x}^r$ has the same level as X^r. We shall assign to each term F in $\mathscr{U}$ a vector $\boldsymbol{f}$ in $\boldsymbol{C}$ such that level($\boldsymbol{f}$) = level(F). By the *expression* assigned to F is meant the initial component f_0 of the vector $\boldsymbol{f}$ assigned to F.

Assignment to prime terms of $\mathscr{T}$

(i). To the variable X^r is assigned $\boldsymbol{x}^r$.

(ii). To the numeral $\boldsymbol{n}$ is assigned $\langle 1 \rangle$.

(iii). To the constant $\mathfrak{s}$ for the successor function is assigned $\langle 1, 1 \rangle$.

(iv). To the primitive recursion constant R of type level n is assigned the vector $\langle 1, \ldots, 1, \omega \rangle$ of level n.

(v). To the 'restricted primitive recursion' constant R^k of type level n is assigned the vector $\langle 1, \ldots, 1, 2(k+1) \rangle$ of level n.

Terms formed by application. Suppose $\boldsymbol{f}$ and $\boldsymbol{g}$ have been assigned to terms F and G, respectively; and that FG is well-formed and has type level n. Then to FG is assigned the vector $\boldsymbol{h}$ such that $(\boldsymbol{h})_i = (\boldsymbol{f} \square \boldsymbol{g})_i$ $(0 \leqslant i \leqslant n)$.

Terms formed by λ-abstraction. Suppose $\boldsymbol{h}$ has been assigned to the term H. Then to $\lambda X^r.H$ is assigned the vector $\delta^r \boldsymbol{h}$, where δ^r is the operator defined in § 2.

LEMMA 3.1. *Let X^s be a free variable of H. Let F be a term of the same type as X^s and assume no free variable of F becomes bound in $[F/X^s]H$. Suppose $\boldsymbol{f}$ and $\boldsymbol{h}$ are vectors assigned to F and H, respectively. Then $[\boldsymbol{f}/\boldsymbol{x}^s]\boldsymbol{h}$ is a vector assigned to $[F/X^s]H$.*

PROOF. By induction on the number of times the operations of application and λ-abstraction are used in building up H from prime terms. To handle λ-abstraction, suppose H is $\lambda X^r.G$, where $r \neq s$. Then $\delta^r \boldsymbol{g}$ is assigned to H, where $\boldsymbol{g}$ is the vector assigned to G. By induction hypothesis $[\boldsymbol{f}/\boldsymbol{x}^s]\boldsymbol{g}$ is assigned to $[F/X^s]G$. Hence from the fact that $\lambda X^r.[F/X^s]G$ is $[F/X^s]H$ we conclude that $\delta^r[\boldsymbol{f}/\boldsymbol{x}^s]\boldsymbol{g}$ is assigned to the latter term. But by lemma 2.10 and corollary $\delta^r[\boldsymbol{f}/\boldsymbol{x}^s]\boldsymbol{g} = [\boldsymbol{f}/\boldsymbol{x}^s]\delta^r \boldsymbol{g}$.

Addition of constants R^v to $\mathscr{T}$. Besides the constants R^n, let us introduce a constant R^v, of the same type, for each expression v. To R^v assign the vector $\langle 1, \ldots, 1, 2(v+1)\rangle$ of the appropriate level.

LEMMA 3.2. *Let e and f be the vectors assigned to $H0(R^vHG)$ and $R^{v+1}HG$, respectively. Then $e_i \prec f_i$ for $0 \leqslant i \leqslant level(e)$.*

PROOF. Let n denote level(e); so G, H and R^v have levels n, $n+1$ and $n+2$, respectively. Let a be the vector of level $n+1$ defined as follows: $a_{n+1} = 1+h_{n+1}$ and $a_i = (a_{i+1}, 1+h_{i+1})$ for $0 \leqslant i \leqslant n$. Let b, c and t be the vectors assigned to R^vH, G and R^vHG, respectively. Then for $i \leqslant n$

$$e_i \preccurlyeq (a \square t)_i \preccurlyeq (a \square (b \square c))_i.$$

Clearly $a_i \preccurlyeq (a \square c)_i$ for all $i \leqslant$ level(a). Hence $e_i \preccurlyeq ((a \square c) \square (b \square c))_i$ for all $i \leqslant n$. Hence it is sufficient to prove $((a \square c) \square (b \square c))_i \prec (d \square c)_i$ for all $i \leqslant n+1$, where d is the vector assigned to $R^{v+1}H$. By lemma 2.1 $b_i \succ 0$ for $i \leqslant n+1$. Clearly $a_i \succ 0$ for $i \leqslant n+1$. Hence by lemma 2.6 it is sufficient to prove

$$2a_{n+1}+2b_{n+1} \prec d_{n+1}, \tag{3.1}$$

$$a_i+b_i \preccurlyeq d_i \quad \text{for } i \leqslant n+1. \tag{3.2}$$

To prove (3.1) observe $2a_{n+1} = 2(0, 1+h_{n+1}) \preccurlyeq (1, 1+h_{n+1})$ by axiom 2.9; and $2b_{n+1} \preccurlyeq (2v+3, 1+h_{n+1})$ by axiom 2.9. Hence $2a_{n+1}+2b_{n+1} \prec \prec 2(2v+3, 1+h_{n+1}) \preccurlyeq d_{n+1}$ by axiom 2.9. To prove (3.2) by downward induction on i, observe that for $i \leqslant n$, a_i+b_i is equal to

$$(a_{i+1}, 1+h_i)+(b_{i+1}, 1+h_i).$$

Hence $a_i+b_i \preccurlyeq (a_{i+1}+b_{i+1}, 1+h_i)$ by 2.14. Hence, by induction hypothesis, $a_i+b_i \preccurlyeq (d_{i+1}, 1+h_i) = d_i$.

Remark. In lemma 3.2 it would be desirable to replace R^{v+1} by R^w such that $w \succ v$. The difficulty would then arise, in the present proof, of establishing (3.1). This difficulty vanishes if we assume that $\mathscr{E}$ has the property $4(x, y) \preccurlyeq (z, y)$ whenever $x \prec z$. This property holds for the interpretation of $\mathscr{E}$ given in § 2 if, in the definition of (a, b), one uses the Cantor normal form to a base greater than or equal to 4 instead of 2.

THEOREM 3.1. *Suppose A contr B, and let a and b be the vectors assigned to A and B, respectively. Then $a_i \succ b_i$ for all $i \leqslant level(a)$.*

PROOF. It is easy to verify theorem 3.1 for the contractions $\mathfrak{S}n$ contr $n+1$

and Rn contr R^n. The verification of theorem 3.1 for the contractions R^0HG contr G and $R^{n+1}HG$ contr $Hn(R^nHG)$ is provided by lemmas 2.1 and 3.2, respectively. It remains to examine the contraction $(\lambda X^s H)F$ contr $[F/X^s]H$. Let $\boldsymbol{f}$ and $\boldsymbol{h}$ be the vectors assigned to F and H, respectively. Then $[\boldsymbol{f}/\boldsymbol{x}^s]\boldsymbol{h}$ is the vector assigned to $[F/X^s]H$ by lemma 3.1. But for all $i \leqslant \text{level}(\boldsymbol{h})$ the ith component of the vector assigned to $(\lambda X^s \cdot H)F$ is $((\delta^s \boldsymbol{h}) \sqcap \boldsymbol{f})_i$.

The desired result now follows from the corollary to lemma 2.11.

THEOREM 3.2. *Suppose F reduces to G by contraction of one occurrence of a subterm A of F (restricted reduction of F). Let $\boldsymbol{f}$ and $\boldsymbol{g}$ be the vectors assigned to F and G, respectively. Then $f_i \succcurlyeq g_i$ for $1 \leqslant i \leqslant level(\boldsymbol{f})$, and $f_0 \succ g_0$.*

PROOF. It is easy to see that since A is a subterm of F, there is a term H such that F is $[A/X^r]H$, where X^r has one free occurrence in H. Suppose A contracts to B. Then G is $[B/X^r]H$. Let $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{h}$ be the vectors assigned to A, B and H, respectively. By lemma 3.1, $\boldsymbol{f}$ and $\boldsymbol{g}$ are $[\boldsymbol{a}/\boldsymbol{x}^r]\boldsymbol{h}$ and $[\boldsymbol{b}/\boldsymbol{x}^r]\boldsymbol{h}$, respectively. It is easy to see that the non-vacuous free occurrence of X^r in H implies the non-vacuous occurrence of x_0^r in h_0. Theorem 3.2 now follows from theorem 3.1 together with lemmas 2.12 and 2.13.

Assignment of ordinals to terms. To obtain an assignment of ordinals less than ε_0 to the terms of $\mathscr{T}$ proceed as follows. First assign expressions f_0 to terms F in the manner described in the present section. Next apply the interpretation of expressions given in § 2 under ***Interpretation of $\mathscr{E}$***. This results in the assignment of a function to F; so apply this function to a suitable constant (say 0), in all argument places, to get an ordinal: this is the ordinal assigned to F. By theorem 3.2 reduction of F lowers the ordinal assigned to F.

4. General reductions. At first sight it might appear that the assignment given in § 3 is valid for *general* reductions; i.e. that theorem 3.2 is true even when A is a subform. Namely, one might try to prove this generalization of theorem 3.2 by induction on the length of F. This attempt fails when F is of the form $\lambda X^r.H$ for the following reason. Suppose $\lambda X^r.H$ is reduced to $\lambda X^r.G$ by contracting a subform of H, and let $\boldsymbol{h}$ and $\boldsymbol{g}$ be the vectors assigned to H and G, respectively. Then H red G, so by induction hypothesis $h_i \succcurlyeq g_i$ for all $i \leqslant \text{level}(\boldsymbol{h})$, and $h_0 \succ g_0$. Unfortunately this does not imply $(\delta^r \boldsymbol{h})_i \succcurlyeq (\delta^r \boldsymbol{g})_i$ for all $i \leqslant \text{level}(\delta^r \boldsymbol{h})$, and $(\delta^r \boldsymbol{h})_0 \succ (\delta^r \boldsymbol{g})_0$ (at least I can not prove that it does).

The above difficulty can be surmounted, if non-unique assignments are allowed, as follows. To $\lambda X^r.H$ we allow the assignment of $\delta^r \boldsymbol{d} + [\boldsymbol{e}/\boldsymbol{x}^r]\boldsymbol{h}$, where $\boldsymbol{e}$ is the vector $\langle 1, 1, \ldots, 1\rangle$ of the same level as $\boldsymbol{x}^r$, $\boldsymbol{h}$ is any vector assigned to H, and $\boldsymbol{d}$ is any vector assigned to a term D such that D reduces to H in a finite number of steps: $D = D_0$ red D_1 red $\ldots$ red $D_k = H$.

The following integer $t(F)$, associated with F and the manner in which a vector is assigned to F, is useful for the proof of the remaining results of this section: $t(F) = 1$ if F is prime;

$$t(AB) = t(A)+t(B);$$

$$t(\lambda X^r.H) = 1+t(D_0)+\ldots+t(D_k)$$

where $D_0, \ldots, D_k$ are described above.

Lemma 3.1 holds for the new method of assignment: with the help of the proof in § 3 it is easy to get a proof for the new assignments by induction on $t(H)$.

Theorem 4.1 below will be proved under the assumption of the following axiom for expressions b, f and g:

4.1. If $f \succcurlyeq g$, then $[b/x_i^r]f \succcurlyeq [b/x_i^r]g$.

Axiom 4.1 clearly holds under the interpretations of the system $\mathscr{E}$ used in the present paper.

THEOREM 4.1. *Suppose F reduces to G by contraction of a subform (general reduction). Let $\boldsymbol{f}$ be a vector assigned to F. Then there exists a vector $\boldsymbol{g}$ assigned to G such that $f_i \succcurlyeq g_i$ for $1 \leqslant i \leqslant$ level$(\boldsymbol{f})$, and $f_0 \succ g_0$.*

PROOF. By induction on $t(F)$. We handle λ-contractions as follows. Suppose $F = (\lambda X^r.H)B$ red $[B/X^r]H$, where $\delta^r \boldsymbol{d} + [\boldsymbol{e}/\boldsymbol{x}^r]\boldsymbol{h}$ and $\boldsymbol{b}$ are assigned to $\lambda X^r \cdot H$ and B, respectively. By induction assumption there exists an assignment $\boldsymbol{d}^k$ to the term D_k described above such that $d_i \succcurlyeq d_i^k$ for $1 \leqslant i \leqslant$ level$(\boldsymbol{d})$, and $d_0 \succ d_0^k$. Hence by axiom 4.1 $([\boldsymbol{b}/\boldsymbol{x}^r]\boldsymbol{d})_i \succcurlyeq \succcurlyeq ([\boldsymbol{b}/\boldsymbol{x}^r]\boldsymbol{d}^k)_i$ for $1 \leqslant i \leqslant$ level$(\boldsymbol{d})$, and $([\boldsymbol{b}/\boldsymbol{x}^r]\boldsymbol{d})_0 \succ ([\boldsymbol{b}/\boldsymbol{x}^r]\boldsymbol{d}^k)_0$. But D_k is H. Hence by lemma 3.1 $[\boldsymbol{b}/\boldsymbol{x}^r]\boldsymbol{d}^k$ is an assignment to $[B/X^r]H$. It now follows from the corollary to lemma 2.11 that this assignment has the required properties.

The case in which F is AB, and G arises from F by contraction of a subform of A or B, is handled by lemma 2.4. The case in which F is $\lambda X^r.H$ is handled by the induction hypothesis and (4.1).

5. More efficient assignment. Let $\omega_1 = \omega$ and $\omega_{k+1} = \omega^{\omega_k}$ $(k = 1, 2, \ldots)$. Let $\mathscr{T}_k$ denote the set of terms of $\mathscr{T}$ which contain primitive recursion constants R and R^n of level at most k. It is easy to see that the assignment of § 3, in assigning ordinals to terms of $\mathscr{T}_k$, makes use of the ordinals less than ω_{k+1}. However the work of Tait [6], and independent work of C. Parsons (private communication to the author), suggests that it is the ordinals less than ω_k rather than ω_{k+1} that should be assigned to the terms of $\mathscr{T}_k$. The purpose of this section is to provide such an assignment which works so long as the following restrictions are made: if G has type level zero, the contractions $(\lambda X.F)G$ contr $[G/X]F$, R^0HG contr G, and $R^{n+1}HG$ contr $Hn(R^nHG)$ are allowed only when G is a numeral.

We shall modify the assignment, described in § 3, of vectors to terms so that to a term F will be assigned a vector $\boldsymbol{f}$ of level $\max\{1, \text{level}(F)\}$. This assignment will have the property that if F red G (restricted reduction) and if $\boldsymbol{g}$ is assigned to G, then $f_1 \succ g_1$. The expression f_1 will be assigned to F. When expressions are interpreted by ordinals as in § 3, this will yield an assignment to the terms of $\mathscr{T}_k$ by ordinals less than ω_k.

The modifications are as follows. When r is such that X^r has type level zero, the corresponding x^r is taken to be $\langle x^r_0, x^r_1\rangle$ rather than $\langle x^r_0\rangle$ (cf. § 2, ***The operation*** δ^r).

To numerals assign the vector $\langle 1, 0\rangle$. If X^r has type level zero, and $\boldsymbol{h}$ is assigned to H, then to $\lambda X^r.H$ assign the vector $\delta^r[0/x^r_1]\boldsymbol{h}$. If $\boldsymbol{f}$ and $\boldsymbol{g}$ have been assigned to F and G, respectively, then to FG assign the vector $\boldsymbol{h}$ such that $\text{level}(\boldsymbol{h}) = \max\{1, \text{level}(FG)\}$ and $h_i = (\boldsymbol{f} \square \boldsymbol{g})_i$ for $0 \leqslant i \leqslant \text{level}(\boldsymbol{h})$.

Using the method of § 3 and the fact that, for all $\boldsymbol{b}$, $(\boldsymbol{b} \square \langle 1, 0\rangle)_i = (\boldsymbol{b} \square \langle 1\rangle)_i$ for all i, it is easy to verify that the modified assignment has the required properties.

University of Illinois at Chicago circle.

REFERENCES

[1] H. Bachmann, *Transfinite Zahlen*, Springer, Berlin (1955).

[2] G. Gentzen, Neue Fassung des Widerspruchsfreiheitsbeweises für die reine Zahlentheorie, *Forsch. Logik u. Grundl. exakten Wissenschaften* N.S. **4**, Hirzel, Leipzig (1938) 19–44.

[3] H. Gerber, An extension of Schütte's Klammersymbols, *Math. Ann.* **174** (1967) 203–216.

[4] K. Gödel, Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes, *Dialectica* **12** (1958) 280–287.

[5] W. Tait, Intensional interpretations of functionals of finite type, *J. Symb. Logic* **32** (1967) 198–212.

[6] W. Tait, Constructive reasoning, to appear.