

REDUCING ORDINAL RECURSION¹

C. F. KENT²

Ordinal recursion has been used to generate hierarchies of recursive functions in which computational difficulty is related to the order-type of the underlying well-ordering. Peter [PFR] proves the equivalence of n -fold recursion and ordinal recursion of type ω^n , and the ordinal-recursive functions of finite type used by Gentzen and Ackermann in consistency proofs clearly relate order-type to computational difficulty. However, no simple connection exists in view of results of Myhill [M 53] and Routledge [R 53], which exhibit ordinal recursions over well-orderings of type ω for any recursive function. Myhill's ordering is even primitive recursive. Liu's paper [L 60] contains a proof of Myhill's result. In Liu's proof, the original defining equations for the function are discarded and it is shown that the function (extensionally regarded) is definable by an ordinal-recursive scheme, of type ω , which is essentially unrelated to the original scheme, by which the function may have been presented. Routledge, on the other hand, retains the original scheme but loses much of the constructivity of the original ordering.

This paper presents results extending those of Routledge, in which the original scheme is retained where possible, and in which the original well-ordering, $<$, is replaced by one of type ω , $<_\omega$, primitive recursive in $<$. The replacement of $<$ by $<_\omega$ can be accomplished whether the original recursion is nested or unnested, so long as no parameters are involved. When parameters are involved, it is generally impossible to retain the original scheme, but a method is given for modifying the scheme to one which can be retained.

Notation. The notation and definitions given here are the same as those in Tait [T 61]. The symbol $<$ denotes an irreflexive well-ordering of the set of natural numbers, N , in which 0 is the least element. Let f, g, f_1, \dots, f_m be function symbols and $t(a, b)$ be a numerical term constructed from variables a_1, \dots, a_r, b ; constants 0, ', =;

Presented to the Society, January 24, 1964 under the title *Reduction of ordinal recursion*; received by the editors August 10, 1965 and, in revised form, August 16, 1966.

¹ This research was sponsored by the Office of Naval Research under Grant NONR 00038 (63).

² I am indebted to the referee for pointing out that the original version of Theorem 2, which stated that $<_\omega$ is primitive recursive if $<$ and Ψ are recursive, had not been proved. That clause is false, as shown by the example following the present Theorem 5.

and the function letters. A system of equations is said to be a *nested ordinal-recursion scheme over \prec* , N.O.R. \prec , if it is of the form:

$$(1) \quad \begin{aligned} f(a_1, \dots, a_r, 0) &= g(a_1, \dots, a_r), \\ f(a_1, \dots, a_r, b') &= t(a_1, \dots, a_r, b), \end{aligned}$$

where, for each occurrence $f(s_1, \dots, s_r, s)$ of f in t , s_1, \dots, s_r and s are subterms which, in general, contain \hat{a}, b and the function letters, but s is a function of \hat{a}, b alone, $s = s(\hat{a}, b)$, and it is true that

$$(2) \quad \forall \hat{a} \forall b (s(\hat{a}, b) \prec b').$$

We note that the first four primitive-recursive schemes of Kleene [KIM, p. 219] are special instances of (1), but that scheme V, for primitive recursion, is not necessarily included. Because of the recursively undecidable condition (2) (which is really a condition on the uniformity of the definition scheme (1)), even if \prec is a recursive predicate, the class of ordinal-recursive schemes by this definition is not recursive, or even recursively enumerable. By the use of a "limiting function"

$$(3) \quad \begin{aligned} h_{\prec}(a, b) &= b && \text{if } b \prec a \\ &= 0 && \text{otherwise} \end{aligned}$$

which is clearly primitive recursive in \prec , we obtain a sufficient class of schemes which is recursive by requiring, instead of (2), that the last entry of each f occurrence in t be of the form $h_{\prec}(b', s)$.

A function $\phi(x_1, \dots, x_r, y)$ defined by a sequence of nested ordinal-recursion schemes in which the function letters g, f_1, \dots, f_m are interpreted as the functions $\gamma, \phi_1, \dots, \phi_m$, is said to be defined by *nested ordinal recursion over \prec* , n.o.r. \prec (uniformly, subject to the conditions (2)) in $\gamma, \phi_1, \dots, \phi_m$. If every function definable by primitive recursion is also definable by n.o.r. \prec we say that primitive recursion is *reducible* to n.o.r. \prec (or to o.r. \prec). Included in the class of n.o.r. \prec functions is the class of functions (unnested) *ordinal-recursive over \prec* , o.r. \prec , which is the class definable by a sequence of Kleene's schemes I-IV and the scheme (4) of restricted *ordinal recursion over \prec* , O.R. \prec :

$$(4) \quad f(\hat{a}, 0) = g(\hat{a}), \quad f(\hat{a}, b') = f_i(\hat{a}, b', f(\hat{a}, f_j(\hat{a}, b'))),$$

where $\forall a \forall b (f_j(a, b') \prec b')$. Again, the h -function can be used.

We shall use the symbols $\tau_r, \pi_1^r, \dots, \pi_r^r$, for the primitive recursive functions so that τ_r maps r -tuples of natural numbers one-to-one onto the natural numbers and, for all $a, \tau_r(\pi_1^r(a), \dots, \pi_r^r(a)) = a$. We de-

fine the *lexicographic jump*, \prec^* , of \prec as follows. The relation $x \prec^* y$ means: $x = 0$ and $y \neq 0$, or, $x = 1$ and $y > 1$, or, $x > 1$ and, if p_{n_1}, \dots, p_{n_k} are the prime divisors appearing to different powers in x and in y , ordered so that $n_1 \prec n_2 \prec \dots \prec n_k$, then p_{n_k} appears to a higher power in y than in x . The ω -product, \prec' , is defined as follows. The relation $x \prec' y$ means: $\pi_2^2(x) \prec \pi_2^2(y)$, or, $\pi_2^2(x) = \pi_2^2(y)$ and $\pi_1^2(x) \prec \pi_1^2(y)$. Clearly, if \prec is of order-type α , then \prec^* is of order-type ω^α , and \prec' of type $\omega \cdot \alpha$.

We note that the role of the well-ordering \prec in the schemes (1) and (4) is strictly mediative. It has no real role to play in calculating the function, given (2) or its counterpart, but only serves as a mechanism for proving that the calculation halts. We recognize that any other well-ordering, \subset , which satisfied (2) would serve the same purpose. Accordingly, we call a well-ordering, \subset , *compatible* with (1) if it satisfies condition (2), in place of \prec . Our earlier announced goal of replacing \prec by \prec_ω can be rephrased as finding a well-ordering \prec_ω , primitive recursive in \prec , and compatible with (1) (or with (4), as the case may be).

Removal of parameters and reduction of the order-type.

THEOREM 1. *If the function $\phi(x, y)$ is defined by a N.O.R. \prec scheme (1) (uniformly) in functions $\Psi = \{\gamma, \phi_1, \dots, \phi_m\}$, with parameters a_1, \dots, a_r , then ϕ can also be defined by a N.O.R. scheme without parameters over \prec' (or \prec^*) (uniformly) in Ψ and certain primitive recursive functions (if primitive recursion is reducible to n.o.r. \prec' then (uniformly) by n.o.r. \prec' in Ψ).*

PROOF. The idea is quite simple. We have $\tau_2(p, m) \prec' \tau_2(q, n)$ whenever $m \prec n$. The parameters in (1) are eliminated by “transferring” the recursion to numbers of the form $c = \tau_2(\tau_r(a_1, \dots, a_r), b)$ in the \prec' order (or, to numbers of the form $c = p_b^{\tau_r(a_1, \dots, a_r)}$ in the \prec^* order). We transform the scheme (1) to a scheme (1') as follows:

- (i) Change the first equation to $f'(0) = g(0, \dots, 0)$.
- (ii) Write: $f'(c) = g(\pi_1^r(\pi_1^2(c)), \dots, \pi_r^r(\pi_1^2(c)))$ if $\pi_2^2(c) = 0$.
- (iii) In the second equation of (1):
 - (iii.1) in t , substitute $\pi_1^r(\pi_1^2(c)) = a_1, \dots, \pi_r^r(\pi_1^2(c)) = a_r, \pi_2^2(c) = b$, call the resulting term t' ;
 - (iii.2) beginning with the innermost occurrences of f -terms, $f(s_1, \dots, s_r, s)$, in t' , replace by $f'(\tau_2(\tau_r(s_1, \dots, s_r), s))$.
- (iv) Combine (ii) and (iii) into one equation primitive recursively. Replace the left side by $f'(c)$.

Because of condition (2) on the original scheme (1), and the definition of \prec' , it is clear that (1') is a N.O.R. \prec' scheme, and has no parameters. Moreover, it is easily established by induction on the order \prec that:

$$\forall \hat{a} \forall b (f(\hat{a}, b) = f'(\tau_2(\tau_r(\hat{a}), b)))$$

This proves Theorem 1.

THEOREM 2. *If*

$$(5) \quad f(0) = 0^{(k)}, \quad f(b') = t(b),$$

is a N.O.R. \prec scheme (without parameters) there is a well-ordering, \prec_ω , primitive recursive in \prec, f_1, \dots, f_m ,³ and of type ω , which is compatible with (5).

PROOF. Let $\sigma_1, \dots, \sigma_k$ denote the terms which appear in the last (recursion variable) positions in the occurrences of f in t . Note that $\lambda_i(b, a) = \lambda b a \sigma_i^{(a)}(b)$ (where the superscript (a) denotes the a -fold iteration of σ_i , beginning with b) is recursive in f_1, \dots, f_m for $1 \leq i \leq k$, and let it have Godel Number e_i relative to $S = \{f_1, \dots, f_m\}$. We define a predicate $F(n, y')$ primitive recursive in S , as follows.

$$\begin{aligned} F(n, y') \equiv & y' < n \ \& \ \forall i_{1 \leq i \leq k} \exists z_{z \leq p_{2n}}^n \{ \forall w_{1 \leq w \leq 2n} ((z)_w < n) \\ & \& \ \forall w_{1 \leq w \leq n} T_2^S(e_i, y', w, (z)_w) \\ & \& \ \forall w_{1 \leq w \leq n} (U((z)_w) = (z)_{n+w} \ \& \ (z)_{2n} = 0) \} . \end{aligned}$$

This predicate expresses the proposition that: for each i , the σ_i -splinter beginning with y' (the set $\{y', \sigma_i(y'), \sigma_i(\sigma_i(y')), \dots\}$) is of cardinal less than n , is entirely composed of numbers less than n , and that all computations in the splinter have numbers less than n . Note that if f_1, \dots, f_m denote recursive functions, then $F(n, y')$ is an absolutely primitive recursive predicate. Next, we give a definition of \prec_ω primitive recursive in F and in \prec .

Step 0: Let 0 be the least element of \prec_ω .

Step 1: Define $0 \prec_\omega 1$.

Step 2: If $F(1, 1)$ (never true), define $1 \prec_\omega 2$. If not $F(1, 1)$, define $1 \prec_\omega 2$ if $1 \prec 2$, otherwise, define $2 \prec_\omega 1$.

⋮

³ This mixture of symbols is a mild abuse of the notation.

Note that just prior to the p th step the numbers $0, 1, \dots, p-1$ have been placed in a linear array, 0 leftmost.

Step p : Beginning at the left end of the array, advance p to the right until it stands before one of its \prec -successors, q , for which not $F(p, q)$, or to the right end of the array if no such q is found.

The placement of p relative to $0, \dots, p-1$ does not disturb the order of the previously placed numbers, and makes possible the relative primitive recursion. Clearly, the above instruction results in a linear ordering of all the natural numbers, in which each number $p \neq 0$ has an immediate predecessor, for there is an n such that $F(n, p)$, and no number will be placed as an immediate predecessor of p after the n th step of the instruction. And, it is also clear from the definition of \prec_ω that it is compatible with (5). Lemmas 1 and 2 complete the proof by showing that \prec_ω is a well-ordering of type ω . If $r < p$ and $r \prec p$, but $p \prec_\omega r$, the pair (p, r) will be called an *inversion* in \prec_ω .

LEMMA 1. *There are no inversions in the order \prec_ω .*

PROOF. Suppose there were an inversion. No generality is lost by assuming that the p th step is the first at which an inversion occurs, $p > 1$. Since $r < p$, then r has been placed before the p th step and, hence, when p is placed at the p th step, it is placed as a \prec_ω predecessor of r . Now p cannot be placed immediately before r because $r \prec p$. Thus, there is a number q among $0, 1, \dots, p-1$ so that p is initially placed immediately before q , and $q \prec_\omega r$. It must then be true that $p \prec q$ and hence $r \prec p \prec q$. If $r < q$, then q is placed after r is placed, but $q \prec_\omega r$ while $r < q$, and an inversion would have occurred at a step earlier than the p th, contrary to the choice of p . Hence, $q < r$.

Since r is placed after q is in place, but $q \prec_\omega r$ while $r < q$, it follows that each splinter of q is composed of numbers less than r , i. e. $F(r, q)$. A fortiori, since $r < p$, p is not a member of a splinter of q , and $F(p, q)$. Thus, from the \prec_ω instruction, p would not be initially placed as the immediate predecessor of q . This contradiction proves Lemma 1.

LEMMA 2. *Each number p has finitely many \prec_ω predecessors.*

PROOF. By a comment above, each number p in the \prec_ω order has an immediate predecessor. Let $C = \dots \prec_\omega p_3 \prec_\omega p_2 \prec_\omega p_1 \prec_\omega p$ be a predecessor chain starting at p . Let $C' = \dots \prec_\omega p_{i_2} \prec_\omega p_{i_1} \prec_\omega p$ be the subchain of C produced by choosing p_{i_1} to be the p_n for least n so that $p_n > p$, p_{i_2} the p_m for least m so that $p_m > p_{i_1}$, and so on.

By Lemma 1, since $p_{i_j+1} <_{\omega} p_{i_j}$ and $p_{i_j+1} > p_{i_j}$, we must have $p_{i_j+1} < p_{i_j}$. Hence, C' is a decreasing chain in the $<$ order. Thus C' , and also C , are finite. This proves Lemma 2.

COROLLARY 3. *A function $\phi(x)$ defined from ϕ_1, \dots, ϕ_m by a sequence of N.O.R. $<$ schemes, all without parameters, is defined by the same sequence of schemes, from the same given functions, but by n.o.r. $<_{\omega}$ over a well-ordering, $<_{\omega}$, which is primitive recursive in $<$ and in ϕ_1, \dots, ϕ_m , and is of type ω . If ϕ_1, \dots, ϕ_m are recursive functions, then $<_{\omega}$ is primitive recursive in $<$ alone.*

PROOF. Since none of the schemes used in the definition of ϕ involves parameters, it is plain that an $F(n, y')$ predicate can be defined for all of the σ -terms appearing in each of the schemes, simultaneously, and the proof will go through just as the proof of Theorem 2.

In the case of recursion with parameters it is easy to demonstrate that the order $<_{\omega}$, above, must depend upon the parameters, for if $<$ were the ordering $0, 2, 4, \dots, 1, 3, 5, \dots$, of order type $\omega \cdot 2$, and

$$\begin{aligned} \lambda(a, 0) &= 0, & \lambda(a, b') &= b' - 2 & \text{if } b' \text{ is even} \\ & & &= 2(b' - 1) + 2a & \text{if } b' \text{ is odd} \end{aligned}$$

where λ is one of the functions appearing as a recursion term in some f in t , then no single ordering of type ω is compatible with the scheme for all a . For fixed parameter values the procedure of Theorem 2 still yields an ordering of type ω , of course, but the result does not appear meaningful. If the sequence of recursions used to define ϕ involves parameters, then it can be modified as in Theorem 1 to rid the affected schemes of parameters by raising the underlying ordering from $<$ to $<'$. Then, if we apply Theorem 2 to the resulting schemes we obtain the following analogue to Corollary 3.

COROLLARY 4. *A function $\phi(x)$ defined from ϕ_1, \dots, ϕ_m by a sequence of N.O.R. $<$ schemes is also definable by a modified sequence of schemes, from the same given functions, but by n.o.r. $<_{\omega}$ over a well-ordering, $<_{\omega}$, which is primitive recursive in $<$ and in ϕ_1, \dots, ϕ_m , and is of type ω . If ϕ_1, \dots, ϕ_m are recursive functions then $<_{\omega}$ is primitive recursive in $<$ alone.*

We may note that if the original scheme was unnested, the reductions in Theorems 1 and 2 leave it unnested. Thus we may use our results together with the following theorem of Tait [T 61] to reduce a nested recursion with parameters, of any order type, to an unnested recursion without parameters over a well-ordering of type ω .

THEOREM (TAIT). *If primitive recursion is reducible to $\text{o.r.}\prec$, and $\text{o.r.}\prec$ to $\text{o.r.}\prec^*$ and $\phi(x, y)$ is defined by $\text{n.o.r.}\prec$ (uniformly) in Ψ , then it is definable by $\text{o.r.}\prec^*$ (uniformly) in Ψ .*

While it simplifies the discussion to assume that primitive recursion is reducible to $\text{o.r.}\prec$, it is only necessary to have certain primitive recursive functions $\text{n.o.r.}\prec^*$, or to add them to Ψ . Similarly, to reduce $\text{o.r.}\prec$ to $\text{o.r.}\prec^*$ only the two functions $p(x) =$ the prime of index x , and $p(x) = 0$ if x is not prime, and y if $x = p(y)$, are needed, by Tait's Lemma 1. Tait further requires that the characteristic function of \prec be $\text{o.r.}\prec$. Anderson [A 61] has shown that this is always true.

THEOREM 5. *If primitive recursion is reducible to $\text{o.r.}\prec$, and $\text{o.r.}\prec$ to $\text{o.r.}\prec^*$, and if $\phi(x, y)$ is defined by $\text{n.o.r.}\prec$ (uniformly) in Ψ , then $\phi(x, y)$ is defined by $\text{o.r.}((\prec^*)')_\omega$ (uniformly) in Ψ for a well-ordering $((\prec^*)')_\omega$ which is primitive recursive in \prec and Ψ and of type ω . If the functions of Ψ are recursive, then $((\prec^*)')_\omega$ is primitive recursive in \prec .*

PROOF. First we use Tait's theorem to pass from the original $\text{n.o.r.}\prec$ to an $\text{o.r.}\prec^*$. Then Theorem 1 is used to remove parameters by passing to $(\prec^*)'$ and, finally, Theorem 2 is used to reduce the order type to ω with $((\prec^*)')_\omega$.

Although the dependence of the recursion for \prec_ω from \prec and ϕ_1, \dots, ϕ_m upon the ϕ 's vanishes if the ϕ 's are recursive (see the proof of Theorem 2), it is not possible to remove the dependence upon \prec , even if it is a recursive predicate. It is possible to construct a (general) recursive well-ordering, \prec , of the natural numbers, of order ω , and having a recursive predecessor function, ρ , but so that \prec is not primitive recursive. Any other well-ordering, \subset , compatible with ρ must coincide with \prec . Hence, no primitive-recursive well-ordering is compatible with ρ .

REFERENCES

- [A 61] D. Anderson, *Nested ordinal recursive functions and a subrecursive hierarchy*, Ph. D. Thesis, Duke University, Durham, N. C., 1961.
 [KIM] S. Kleene, *Introduction to metamathematics*, Van Nostrand, Princeton, N. J., 1952.
 [L 60] S. Liu, *A theorem on general recursive functions*, Proc. Amer. Math. Soc. 11 (1960), 184–187.
 [M 53] J. Myhill, *A stumbling block in constructive mathematics*, J. Symbolic Logic 18 (1953), 190–191.
 [PRF] R. Peter, *Rekursive Funktionen*, Akademischer Verlag, Budapest, 1951.
 [R 53] N. Routledge, *Ordinal recursion*, Proc. Cambridge Philos. Soc. 49 (1953), 175–182.
 [T 61] W. Tait, *Nested recursion*, Math. Ann, 143 (1961), 236–250.

CASE WESTERN RESERVE UNIVERSITY AND
LAKEHEAD UNIVERSITY