PACIFIC JOURNAL OF MATHEMATICS
Vol. 76, No. 2, 1978

# CHARACTERIZING THE ORDERS CHANGED BY PROGRAM TRANSLATORS

MARGARET SHAY AND PAUL YOUNG

**The ways in which translators from one programming system for the recursively enumerable sets to another such programming system can change the orders of the sets being translated are characterized using the computable functions which permute infinitely many initial segments.**

In [3], it is shown (Corollary, p. 194) that every translator from one programming system for the recursively enumerable (r.e.) sets to another such programming system must preserve every order of enumeration of every r.e. set on infinitely many of the programs which enumerate the set in the given order. It was also conjectured there that for every translator, many sets of cardinality greater than one *never* have their order of enumeration changed by the translation of *any* of their programs. In this paper, we show that this conjecture is false, although "nearly" true, and we characterize the orders which can be changed by program translators. Specifically, we show that given any r.e. sequence of effective permutations which permute infinitely many initial segments, we can build a translator which changes every (infinite) order of enumeration by every permutation in this set. On the other hand, if a program enumerates a set sufficiently slowly, then no translation of the program can change the order of enumeration by a permutation which is not of this form. Thus for any translation, many sets (those having only slow enumerations) have *all* of their enumeration orders preserved *modulo* such permutations of their initial segments.

In [3], the vague conjecture that "the only general method of translation is simulation (of the source programs)" is discussed. The results presented here are compatible with that conjecture.

We use without further discussion the notation and the definitions of [3], and we assume some familiarity with the results of that paper.

DEFINITION Let $p$ by a function on the natural numbers, i.e., $p\colon N \xrightarrow[\text{onto}]{1-1} N$. Then $p$ permutes initial segments if there are infinitely many $n$ such that $\{p(i) \mid i < n\} = \{i \mid i < n\}$.

We first show that, in a very strong sense, translations can change orders of enumerations by functions which permute initial segments. Intuitively, if $p$ is such a permutation, and we want to build a translator $\tau$ such that the order $\prec_{\tau(i)}$ is the $p$-permutation of $\prec_i$, we get in trouble if $W_i$ is finite since $W_i$ may not contain enough elements to *complete* the permutation called for by $p$. To overcome this difficulty, we define a "pretranslator" $\tau'$ such that $W_{\tau'(i)}$ is obtained by using as much of $W_i$ as we are able to successfully permute. Since $W_{\tau'(i)} \subseteqq W_i$, we can then define the desired translation roughly as the *inverse* of $\tau'$, using only enough elements of $W_{\tau^{-1}(i)}$ to make $W_{\tau^{-1}(i)} = W_i$. We give the details as:

THEOREM 1. *Let $\lambda i W_i$ be any standard indexing of the* r.e. *sets and let $p$ be a computable function on $N$ which permutes initial segments. Then there is a translator $\tau$ from $\lambda i W_i$ to itself which changes every order of every infinite set by $p$.*

*Proof.* In view of the order isomorphism theorem (5) of [3], it suffices to prove this result for any of the familiar enumeration techniques, such as Turing machines, in which standard intuitive operations on orders can be performed. We assume such a technique in the following proof. (For the same reason, to prove the result for a translation from one enumeration technique to another, it suffices to have the result for a translation from any one enumeration technique to itself.)

First define the "pretranslator" $\tau'$ having recursive range as follows: Given $i$ and $n$ obtain the $n$th element of $W_{\tau'(i)}$ by:

(1) Compute $p(x)$ for $x = 0, 1, \cdots$ until finding $k = \mu y \geqq n$ such that $\{z \mid z \leqq y\} = \{p(z) \mid z \leqq y\}$.

(2) Enumerate $W_i$ until $k$ elements have been enumerated.

(3) If and when step 2 terminates, output the $p^{-1}(n)$th element of $W_i$.

Clearly these instructions are effective and the order padding lemma ([3]) for $\lambda i W_i$ can be used to make the range of $\tau'$ recursive by making $\tau'$ strictly monotonically increasing.

Note that if $W_i$ is infinite, then $W_{\tau'(i)} = W_i$ and $\prec_i$ is a $p$-permutation of $\prec_{\tau'(i)}$. The key observation is that if $W_i$ is finite then $W_{\tau'(i)} \subseteqq W_i$ *and some initial segment of $\prec_i$ is a $p$-permutation of $\prec_{\tau'(i)}$*. Thus $\tau'$ is just the inverse of the translation we want, except that $\tau'$ does not translate finite sets whose cardinalities are not the lengths of initial segments on which the permutation $p$ is fixed.

We now define the translator $\tau$ of the theorem as follows, for all $i$:

(i) If $i \notin$ range $\tau'$, let $\tau(i) = i$.

(ii) Otherwise let $m = \tau'^{(-1)}(i)$; to get the $n$th element of $W_{\tau(i)}$, run $i$ and $m$ until both have enumerated $n$ elements; if and when this happens, put out the $n$th element enumerated by $m$.

Then for all $i$ in the range of $\tau'$, $\prec_{\tau(i)}$ is a $p$-permutation of $\prec_i$. In view of the order isomorphism theorem of [3], all orders of every infinite r.e. set appear infinitely often in every enumeration technique. Clearly for Turing machines, if $\prec_i$ is such an order and $W_i$ happens to be infinite, we can find $i'$ such that $W_{i'} = W_i$ and $\prec_{i'}$ is a $p$-permutation of $\prec_i$. Since $<_{i'}$ is also a $p$-permutation of $\prec_{\tau'(i)}$, we see that $\prec_i = \prec_{\tau'(i)}$. Thus since all orders for every infinite set are in the range of $\tau'$, $\tau$ changes all orders of every infinite set by $p$.

Theorem 1 shows that permutations which permute initial segments can be realized by translations. It is natural to ask what other permutations can be realized. There is a sense in which every permutation can obviously be realized if we know that $W_i$ is infinite or if $W_i$ is infinite and happens to be enumerated "quickly," then as observed at the end of the preceding proof we can change the order $\prec_i$ in any way we please. On the other hand, if we do not *a priori* know whether $W_i$ is infinite and if $p$ does not permute infinitely many initial segments, then intuitively it would seem impossible for any *uniform* method, and hence for any translator, to change the order of $W_i$ by the permutation $p$ since it would appear necessary for the translator to periodically make judgments as to whether $W_i$ is finite or infinite in order to effect the permutation. This is essentially the content of our next theorem:

THEOREM 2. *Let $\lambda i W_i$ be any standard indexing of the* r.e. *sets and let $\tau$ be any translator from $\lambda i W_i$ to itself. Then there is a recursive function $b$ such that for any $i$, if $A_i(n) > b(n)$ infinitely often, then $\tau$ cannot change the order of $i$ by any permutation which does not permute initial segments.*

*Proof.* (Recall that $A_i(n)$, defined in [3] and in [6], is, intuitively, the time required for program $i$ to enumerate $n$ elements.) Note that if $p$ is a permutation which does not permute initial segments and if $\prec_{\tau(k)}$ is a $p$-permutation of $\prec_k$, (with $W_k$ infinite), then for all but finitely many $n$,

$$\{e \mid e \text{ is one of the first } n \text{ elements of } W_{\tau(k)}\}$$
$$\neq \{e \mid e \text{ is one of the first } n \text{ elements of } W_k\}.$$

Using this fact we can define the function $b$ of the theorem by diagonalizing over the run times of all sets $j$ for which $\tau$ changes the order of $j$ by some permutation which does not permute infinitely many initial segments:

Let $b(0) = 1$
and $b(n) = b(n-1) + 1 +$
$\max_{j \leq n} \{A_j(n) \mid (\exists m)[A_j(m) \leqq b(n-1)$, and for all $r$ such that
$m \leqq r < n$
$\{$the first $r$ elements of $W_j\} \neq \{$the first $r$ elements of $W_{\tau(j)}\}$ .

For any translator $\tau$ this $b$ is a total recursive function. (Note that if $\{$the first $r$ elements of $W_j\} \neq \{$the first $r$ elements of $W_{\tau(j)}\}$ then $W_j$ and $W_{\tau(j)}$ have at least $r+1$ elements.) For all $i$, if $\prec_{\tau(i)}$ is a $p$-permutation of $\prec_i$ for some $p$ which does not permute initial segments, then $A_i(n) < b(n)$ almost everywhere. (Just as with Theorem 1, because of the order isomorphism theorem of [3], the extension of Theorem 2 to translations between any *two* standard indexings of the r.e. sets is immediate.)

As a corollary to Theorem 2, we observe that if $p$ is a permutation which does not permute initial segments, then for any translator $\tau$, there are many orders of enumeration which $\tau$ fails to change by $p$:

COROLLARY. *Let $p$ be any computable function which fails to permute infinitely many initial segments, and let $\tau$ be any translator. Then there are infinite sets $W_i$ such that $\tau$ does not permute any order of enumeration of $W_i$ by $p$. Also, for every infinite set $W_i$, $W_i$ has some orders of enumeration which $\tau$ does not permute by $p$.*

*Proof.* It is well known that some infinite r.e. sets are difficult to enumerate (for every order of enumeration). (See, e.g., [6].) Furthermore, it is proven in [3] that every infinite r.e. set has *some orders* in which it is difficult to enumerate the set. Thus the corollary follows from Theorem 2.

We close by extending the proofs of Theorems 1 and 2 to provide a complete characterization of the orders which can be changed by program translators:

THEOREM 3. (a) *Let $p_0, p_1, p_2, \cdots$ be any enumeration of computable permutations each of which either is a finite permutation mapping $\{0, 1, 2, \cdots, m\}$ onto $\{0, 1, 2, \cdots, m\}$ for some $m$ or an infinite permutation which permutes initial segments. Then from the list $p_0, p_1, p_2, \cdots$ we can effectively find a translator $\tau$ such that if $W_i$ is infinite, either $\prec_i = \prec_{\tau(i)}$ or $\prec_i$ and $\prec_{\tau(i)}$ differ by some infinite $p_j$. Furthermore if $p_j$ and $W_i$ are infinite, then the order of enumeration $\prec_i$ is changed by $p_j$.*

(b) *Conversely, let $\tau$ be any translator. Then from $\tau$ we can effectively find a list $p_0, p_1, p_2, \cdots$ such that each $p_j$ is either a finite permutation mapping $\{0, 1, 2, \cdots, m\}$ onto $\{0, 1, 2, \cdots, m\}$ for some $m$, or else $p_j$ is an infinite recursive function which permutes initial segments, and for some $i$ for which $W_i$ is infinite $\prec_i$ and $\prec_{\tau(i)}$ differ by $p_j$. Furthermore, if $W_i$ is infinite and $A_i$ is sufficiently slow, then $\prec_i$ and $\prec_{\tau(i)}$ do differ by some $p_j$.*

*Proof.* The proof of (a) is an obvious and easy extension of the proof of Theorem 1. One begins by using order-padding [3], to obtain from $\lambda i \prec_i$ an infinite listing $\lambda i \lambda j \prec_{\langle i,j \rangle}$ such that if $W_i$ and $p_j$ are infinite then $\prec_i = \prec_{\langle i,j \rangle}$. One then calculates $\tau'$ exactly as in the proof of Theorem 1, except that one replaces $i$ by $\langle i, j\rangle$ and $p$ by $p_j$. That is, one attempts to permute the order $\prec_{\langle i,j \rangle}$ by the permutation $p_j$. Since this construction is uniform, there is no difficulty in so computing $\tau'$ and $\tau$. The proof is now exactly as the proof of Theorem 1, except that we must consider the possibility that $p_j$ is finite. But in this case, we still have that $W_{\tau'\langle i,j \rangle} \subseteqq W_{\langle i,j \rangle}$ and that some initial segment of $\prec_{\langle i,j \rangle}$ is a $p_j$-permutation of $\prec_{\tau'\langle i,j \rangle}$. Thus the proof still reads exactly as the proof of Theorem 1, with $\langle i, j\rangle$ replacing $i$, $\langle i', j\rangle$ replacing $i'$, and $p_j$ replacing $p$.

To prove (b), we observe that, given $\tau$, we can, for each $i$, begin listing the permutation $p_i$ which permutes in the obvious way the longest initial segments of $\prec_i$ and $\prec_{\tau(i)}$ on which $\prec_i$ and $\prec_{\tau(i)}$ do permute the initial segments. It is clear that if $W_i$ is finite, $p_i$ is a finite permutation which correctly permutes $\prec_i$ and $\prec_{\tau(i)}$. If $W_i$ is infinite and $A_i$ is sufficiently slow, then by Theorem 2 $\prec_i$ and $\prec_{\tau(i)}$ differ by an infinite permutation which permutes initial segments and $p_i$ must be this permutation. To complete the proof we observe that if $W_i$ is infinite but $\prec_i$ and $\prec_{\tau(i)}$ do not differ by a permutation which permutes initial segments (which can only happen if $A_i$ is fast), then $p_i$ will obviously be finite, proving (b).

In closing, we remark that the translators $\tau$ of Theorem 1 and of 3(a) can (using order-padding [3]) via the usual sort of isomorphism proofs, be constructed to be isomorphisms. On the other hand, in Theorem 3(b), we cannot obtain a more elegant characterization by requiring *each* of the $P_j$'s to be an infinite permutation which permutes initial segments, essentially because we can code into such a sequence $p_0, p_1, p_2 \cdots$ any enumerable sequence of computable functions, each of whose domain is some finite or infinite initial segment of the integers; since we can obtain *every* total recursive function in such a sequence, if we could then eliminate the finite permutations we would have an enumeration of all the total recursive functions.

## References

1. A.M. Dawes, *Splitting theorems for speedup related to order of enumeration,* Dept. of Math., Univ. West Ontario, (1977), 1-17.

2. J. Gill, J. Helm, A. Meyer and P. Young, *Notes on difficulties of enumerations,* SIAM J. Comp., to appear.

3. J. Helm, A. Meyer and P. Young, *On orders of translations and enumerations,* Pacific J. Math., **46** (1973), 185-195.

4. M. Machtey, K. Winklmann and P. Young, *Simple Gödel numberings, isomorphisms, and programming properties,* SIAM J. Comp., **7** (1978), 39-60.

5. M. Machtey and P. Young, *An Introduction to the General Theory of Algorithms,* Elsevier-North Holland, New York, 1978.

6. P. Young, *Toward a theory of enumerations,* J. Assoc. Comp. Math., **16** (1969), 328-348.

7. ———, *Speed-ups by changing the order in which sets are enumerated,* Math. Systems Theory, **5** (1971), 148-156. (Minor correction, Ibid, **7** (1974), 352.)

Received October 16, 1975 and in revised form November 23, 1977. Supported by NSF Grant MC 75-09212. The authors are indebted to R.W. Ritchie for helpful suggestions for the presentation of this material.

PURDUE UNIVERSITY
LAFAYETTE, IN 47906
AND
UNIVERSITY OF NEW MEXICO
ALBUQUERQUE, NM 87131