On Turing Reducibility

MARCIA J. GROSZEK
THEODORE A. SLAMAN

ABSTRACT. We show that the transitivity of pointwise Turing reducibility on the recursively enumerable sets of integers cannot be proven in $P^- + I\Sigma_1$, first order arithmetic with induction limited to Σ_1 predicates. We produce a example of intransitivity in a nonstandard model of $P^- + I\Sigma_1$ by a finite injury priority construction.

§1. INTRODUCTION

In Turing reducibility, a set of integers X is computable from, or relative to, another set Y ($X \leq_w Y$), if there is an algorithmic procedure answering any atomic question about X by referring to finitely many atomic conditions on Y. Relative computability is a transitive relation: Suppose that X is computable from Y by procedure P and Y is computable from Z by procedure Q. We can use Z to compute X as follows. Follow the steps in P; whenever required to ask a question about Y, use Q to compute the answer from Z. This composition of algorithms is itself an algorithm. The computation of a particular value of X from Y involves only finitely many questions to Y; each of the answers is computed in finitely many steps from Z. The computation from Z is finite by this simple fact: A finite union of finite sets is finite.

Consider this situation from an axiomatic point of view. The statement that a finite union of finite objects is still finite is formally phrased as an instance of replacement or bounding: for any number b and any function f with domain the numbers less than b, there is a bound on the range of f. In the previous paragraph, we applied an instance of bounding relative to a function that was recursive in Z.

We will work with models of $P^- + I\Sigma_0$, a weak fragment of arithmetic. First, we will show that the natural proof of the transitivity of $\leq_w$ can be formulated and applied to the recursively enumerable sets in a straight forward way using only $P^- + B\Sigma_2$, bounding for Σ_2 formulas.

The authors' research was supported by NSF grants DMS-8601777 and DMS-8601856, respectively. In addition, Slaman was supported by Presidential Young Investigator Award DMS-8451748 and by the Japan Society for the Promotion of Science. The authors wish to thank Professors Hájek and Kučera whose paper [**4**] ignited their interest.

The majority of this paper is devoted to showing that the transitivity of $\leq_w$ on the recursively enumerable sets is independent of $P^- + I\Sigma_1$, induction for Σ_1 sets. Here, the proof is more interesting. First, we take a model $\mathcal{N}$ of full Peano Arithmetic with nonstandard element p. In $\mathcal{N}$, we form $\mathcal{M}$, a Skölem hull of $\{p\}$, by closing under the Skölem functions for Σ_1 and bounded Π_1 formulas. $\mathcal{M}$ is a model of $P^- + I\Sigma_1$. Moreover, there is a function μ from ω onto $\mathcal{M}$ that is the limit of a recursive sequence in $\mathcal{M}$.

In $\mathcal{M}$, or any other nonstandard model with these properties, there are recursively enumerable sets $A \leq_w B \leq_w C$ so that $A \not\leq_w C$. We use a finite injury priority construction in $\mathcal{M}$ to enumerate A, B and C. The failure of $B\Sigma_2$ given by μ is used to give a priority ordering of order-type ω to the requirements. Namely, if i is less than j then the requirements indexed by $\mu(i)$ have higher priority than those indexed by $\mu(j)$. We prove that all of the requirements are satisfied by induction on ω.

§2. Basic Definitions

Fragments of Peano Arithmetic. Following standard usage, say that a formula is Σ_0 or Π_0 if all of its quantifiers are bounded. A formula is Σ_{k+1} if it is a string of existential quantifiers followed by a formula in Π_k. A formula is Π_{k+1} if it is the negation of a Σ_{k+1} formula. If we include an additional symbol for a predicate Z, then we form the formulas Σ_n in Z (Σ_n^Z), similarly.

2.1. Definition. (1) By P^- we mean the subtheory of Peano Arithmetic with addition, multiplication and exponentiation containing no instances of the induction scheme. It consists of the universal closures of the following axioms.

$$x' \neq 0$$
$$(x' = y') \implies (x = y)$$
$$x \neq 0 \implies 0' \leq x$$
$$x < y \iff (\exists t)[x + t' = y]$$
$$x < y \lor x = y \lor x > y$$
$$x + y = y + x; \quad x \cdot y = y \cdot x$$
$$x + (y + z) = (x + y) + z; \quad x \cdot (y \cdot z) = (x \cdot y) \cdot z$$
$$x + 0 = x; \quad x \cdot 0 = 0; \quad x^0 = 1$$
$$x + y' = (x + y)'; \quad x \cdot y' = (x \cdot y) + x; \quad x^{y'} = x^y \cdot x$$

$$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$$
$$x + y = x + z \implies y = z$$

Here, we use the abbreviations: $x \leq y$ for $(\exists t)[x + t = y]$ and $x < y$ for $x \leq y$ & $x \neq y$.

(2) If Γ is a set of formulas, $B\Gamma$ is the *bounding scheme* for elements of Γ. It consists of the universal closures of the formulas

$$(2.2.\ ) \qquad (\forall x)\left[(\forall y < x)(\exists w)\varphi(y, w) \implies (\exists b)(\forall y < x)(\exists w < b)\varphi(y, w)\right]$$

for each φ in Γ. Formula (2.2) states that the search for the witnesses $w(y)$ for a bounded set $\{\varphi(y, -) \mid y < x\}$ of instances of φ, is itself bounded.

(3) A related and usually stronger scheme is $I\Gamma$, the *induction scheme* for elements of Γ. It consists of the universal closures of the formulas

$$\Big(\varphi(0) \,\&\, (\forall x)[\varphi(x) \implies \varphi(x+1)]\Big) \implies (\forall x)\varphi(x)$$

for φ in Γ.

(4) *Peano Arithmetic* (PA) is defined by

$$PA = P^- \cup \bigcup_{k \in \omega} I\Sigma_k.$$

The basic result in the area of fragments of arithmetic is due to Kirby and Paris, 1977.

2.3. THEOREM [**5**]. *For all k, the following implications hold in the presence of $P^- + I\Sigma_0$.*

$$I\Sigma_{k+1} \implies B\Sigma_{k+1} \implies I\Sigma_k$$
$$I\Sigma_k \iff I\Pi_k$$

Further, the only true implications are the ones indicated.

Recursion Theoretic Definitions. We take as our base theory $P^- + I\Sigma_0$. In this theory, one can prove the basic facts of number theory. Following standard development, see Enderton [**2**], we can use the exponential function to represent sets of elements by single elements. The number representing a set is called its code.

2.4. Definition. Suppose that $\mathcal{M}$ is a model of $P^- + I\Sigma_0$. A set F is $\mathcal{M}$-*finite* if and only if it has a code in $\mathcal{M}$.

We will treat the $\mathcal{M}$-finite sets exactly like elements of $\mathcal{M}$. Similarly, we will treat $\mathcal{M}$ as if it were closed under $\mathcal{M}$-finite sequences. We will denote the sequence with elements $m_1, \ldots, m_k$ by $\langle m_1, \ldots, m_k \rangle$. This can be done without affecting the complexity of formulas which quantify over $\mathcal{M}$-finite sets, even in P^-; see Enderton [**2**].

2.5. Notation. If $\mathcal{M}$ is a model of $P^- + I\Sigma_0$ and m is an element of $\mathcal{M}$, let $<m$ denote the set of x in $\mathcal{M}$ such that x is less than m.

2.6. Definition. If $\mathcal{M}$ is a model of $P^- + I\Sigma_0$ and Z is contained in $\mathcal{M}$, then Z is an *amenable* subset of $\mathcal{M}$ if for every m in $\mathcal{M}$, the intersection of Z with $<m$ is $\mathcal{M}$-finite.

For the basics of recursion theory, we follow the standard presentation. For example, see Soare [**7**].

2.7. Definition. A subset W of $\mathcal{M}$ is *recursively enumerable* in $\mathcal{M}$ if W is Σ_1 with parameters in $\mathcal{M}$.

2.8. Definition. Suppose that $\mathcal{M}$ is a model of $P^- + I\Sigma_0$.

(1) A *Turing reduction* or *Turing functional* Φ is a recursively enumerable set of quadruples $\langle x, y, P, N \rangle$ where x and y are elements of $\mathcal{M}$ and P and N are $\mathcal{M}$-finite sets. We refer to the elements of Φ as *computations*.
(2) Suppose that X is a subset of $\mathcal{M}$. $\Phi(x, X) = y$ if and only if there is a computation $\langle x, y, P, N \rangle$ in Φ so that $P \subseteq X$ and $N \cap X = \emptyset$. In this case, we say that the computation $\langle x, y, P, N \rangle$ *applies* to X. Similarly, we say that $\langle x, y, P, N \rangle$ is a computation from X, or relative to X, or an X-computation. Note, we allow the possibility that $\Phi(-, X)$ does not define a function; it may not be single valued. P is called the *positive condition,* and N the *negative condition,* of the computation. The pair $\langle P, N \rangle$ is called a *neighborhood condition* satisfied by X.
(3) If X and Y are subsets of $\mathcal{M}$ then X is *weakly recursive* in Y, $X \leq_w Y$, if there is a Turing reduction Φ such that, for all x in $\mathcal{M}$, $X(x) = \Phi(x, Y)$. Here, we are identifying the set X with its characteristic function.

(4) X is *strongly recursive* in Y, $X \leq_s Y$, if the sets

$$\left\{P \,\middle|\, P \text{ is an } \mathcal{M}\text{-finite subset of } X\right\}$$
$$\left\{N \,\middle|\, N \text{ is an } \mathcal{M}\text{-finite set with } N \cap X = \emptyset\right\}$$

are weakly recursive in Y.

The important difference between the weak and strong reducibilities is that strong reducibility is transitive by its very definition. The transitivity of the weaker reducibility depends on whether $\mathcal{M}$-finitely many Y-computations can be amalgamated to produce a single $\mathcal{M}$-finite computation.

The distinction between weak and strong reducibilities was studied in depth in α-recursion theory. Driscoll [**1**] proved in 1968 that the metarecursive analogue of $\leq_w$ is not transitive on the metarecursively enumerable sets. The distinction was rediscovered by Hájek and Kučera [**4**] for fragments of arithmetic.

We begin with the positive results.

2.9. Proposition. *($P^- + I\Sigma_0$)*

(1) *If $X \leq_s Y \leq_s Z$ then $X \leq_s Z$.*
(2) *Suppose that Z is an amenable subset of $\mathcal{M}$ and $\mathcal{M}$ is a model of $B\Sigma_1^Z$. If $Y \leq_w Z$, then $Y \leq_s Z$, Y is amenable and $\mathcal{M}$ is a model of $B\Sigma_1^Y$. Hence, if $X \leq_w Y \leq_w Z$ then $X \leq_s Y \leq_s Z$ and so $X \leq_s Z$.*

Proof: We work in $\mathcal{M}$ and prove the propositions in order.

First suppose that $X \leq_s Y$ and that $Y \leq_s Z$. That is, there are Turing reductions Φ, $\bar{\Phi}$, Ψ, $\bar{\Psi}$ such that:

$$A = \{P \mid P \text{ is an } \mathcal{M}\text{-finite subset of } X\} = \Phi(-, Y)$$
$$B = \{N \mid N \text{ is an } \mathcal{M}\text{-finite subset of } \mathcal{M} - X\} = \bar{\Phi}(-, Y)$$
$$C = \{P \mid P \text{ is an } \mathcal{M}\text{-finite subset of } Y\} = \Psi(-, Z)$$
$$D = \{N \mid N \text{ is an } \mathcal{M}\text{-finite subset of } \mathcal{M} - Y\} = \bar{\Psi}(-, Z)$$

To show $X \leq_s Z$, we must show $A \leq_w Z$ and $B \leq_w Z$; the two cases are the same, so we consider A. Define the Turing reduction Γ to be the set of computations $\langle x, y, P, N\rangle$ such that:

$$(\exists H)(\exists \bar{H})(\exists F')(\exists F'')(\exists G')(\exists G'') \\ \left[\begin{array}{c} F' \subseteq P \ \ \& \ \ F'' \subseteq P \ \ \& \ \ G' \subseteq N \ \ \& \ \ G'' \subset N \ \ \& \\ \left\langle H, 1, F', G'\right\rangle \in \Psi \ \ \& \ \ \left\langle \bar{H}, 1, F'', G''\right\rangle \in \bar{\Psi} \ \ \& \ \ \left\langle x, y, H, \bar{H}\right\rangle \in \Phi \end{array}\right]$$

(Intuitively, Γ is the composition of the procedure Φ with Ψ and $\bar{\Psi}$.) Because Ψ, $\bar{\Psi}$ and Φ are recursively enumerable, so is Γ. In case $\langle x, y, F, G\rangle \in \Gamma$, $F \subseteq Z$ and $G \cap Z = \emptyset$, we have (by Ψ) $H \subseteq Y$, and (by $\bar{\Psi}$) $\bar{H} \cap Y = \emptyset$; thus, (by Φ) $A(x) = y$. Conversely, by taking the union of two neighborhood conditions, if $A(x) = y$, we can find $F \subseteq Z$ and G such that $G \cap Z = \emptyset$, such that $\langle x, y, F, G\rangle \in \Gamma$. Thus, $A(x) = \Gamma(x, Z)$, as desired.

For the second claim, we first show that if $\mathcal{M}$ is a model of $B\Sigma_1^Z$ and $Y \leq_w Z$ then $Y \leq_s Z$. Fix Φ such that $Y = \Phi(-, Z)$. We show how to compute $A = \{F \mid F \text{ is an } \mathcal{M}\text{-finite subset of } Y\}$ from Z.

First, we define a modification of Φ, Φ', the set of computations $\left\langle x, y, H, \bar{H}\right\rangle$ such that:

$$(\exists J)(\exists \bar{J}) \left[J \subseteq H \ \ \& \ \ \bar{J} \subseteq \bar{H} \ \ \& \ \ \left\langle x, y, J, \bar{J}\right\rangle \in \Phi\right].$$

Φ' has the property that for all X, $\Phi'(-, X) = \Phi(-, X)$, and in addition, Φ' is monotone: If $\left\langle x, y, J, \bar{J}\right\rangle \in \Phi'$, $J \subseteq H$, and $\bar{J} \subseteq \bar{H}$ then $\left\langle x, y, H, \bar{H}\right\rangle \in \Phi'$.

Next, we show that if F is $\mathcal{M}$-finite, there is a single neighborhood condition $\left\langle G, \bar{G}\right\rangle$ satisfied by Z such that

$$(\forall m \in F)(\exists x) \left[\left\langle m, x, G, \bar{G}\right\rangle \text{ is a } Z\text{-computation from } \Phi'\right].$$

This is an application of $B\Sigma_1^Z$: Suppose $F \subseteq {<}n$. Since $Y = \Phi(-, Z)$,

$$(\forall m < n)(\exists w) \left[w \text{ is a } Z\text{-computation of } \Phi'(m, Z)\right].$$

By applying $B\Sigma_1^Z$, we get

$$(\exists b)(\forall m < n)(\exists w < b) \left[w \text{ is a } Z\text{-computation of } \Phi'(m, Z)\right].$$

Fix b. Since "w is a Z-computation" means w codes some $\left\langle m, y, H, \bar{H}\right\rangle$ where $H \subseteq Z$ and $\bar{H} \cap Z = \emptyset$, by the nature of the coding, $H \subseteq$

$<w \subseteq <b$. By the amenability of Z, $G = \{z \mid z < b \ \& \ z \in Z\}$ and $\bar{G} = \{z \mid z < b \ \& \ z \notin Z\}$ are $\mathcal{M}$-finite. By the monotonicity of Φ' and the choice of b, $\left\langle G, \bar{G} \right\rangle$ has the desired property.

Now, this implies, for our given $\mathcal{M}$-finite F and this neighborhood condition,

$$(\forall m \in F)(\exists i \in \{0,1\})(\left\langle m, i, G, \bar{G} \right\rangle \in \Phi') \ \&$$
$$\begin{pmatrix} F \in A \iff (\forall m \in F)\left[\left\langle m, 1, G, \bar{G} \right\rangle \text{ is a } Z\text{-computation from } \Phi'\right] \\ F \notin A \iff (\exists m \in F)\left[\left\langle m, 0, G, \bar{G} \right\rangle \text{ is a } Z\text{-computation from } \Phi'\right] \end{pmatrix}$$

Finally, we can define a Turing reduction Ψ such that $A = \Psi(-, Z)$: Let Ψ be the set of $\left\langle F, y, G, \bar{G} \right\rangle$ such that

$$\left(y = 0 \ \& \ (\exists m \in F)\left[\left\langle m, 0, G, \bar{G} \right\rangle \in \Phi'\right]\right) \text{ or}$$
$$\left(y = 1 \ \& \ (\forall m \in F)\left[\left\langle m, 1, G, \bar{G} \right\rangle \in \Phi'\right]\right).$$

The definition of Ψ is Σ_1 by a standard application of $B\Sigma_1$, to interchange the bounded quantifier $(\forall m \in F)$ with the unbounded quantifier in Φ'.

To see that Y is amenable, choose n, and show that $Y \cap < n$ is $\mathcal{M}$-finite. As above, there is a neighborhood condition on Z, $\left\langle G, \bar{G} \right\rangle$, such that

$$(\forall m < n)(\exists y)\left[\left\langle m, y, G, \bar{G} \right\rangle \text{ is a } Z\text{-computation of } \Phi'(m, Z) = Y(m)\right].$$

Fix G and $\bar{G}$. Then, by an additional use of $B\Sigma_1$, there is a common bound b to the witnesses to $\left\langle m, y, G, \bar{G} \right\rangle \in \Phi'$, for all $m < n$ and $y = 0$ or $y = 1$. This shows

$$Y \cap < n = \left\{m \mid m < n \ \& \ (\exists w < b)\left[w \text{ witnesses } \left\langle m, 1, G, \bar{G} \right\rangle \in \Phi'\right]\right\}.$$

This is a bounded Σ_0 set, hence, by $I\Sigma_0$, $\mathcal{M}$-finite.

Given that $Y \leq_s Z$, $y \in Y$ and $y \notin Y$ are Σ_1^Z predicates on y. Thus, any Σ_1^Y formula is equivalent to a Σ_1^Z formula (using $B\Sigma_1^Z$ to interchange bounded quantifiers.) Therefore, $\mathcal{M}$ is also a model of $B\Sigma_1^Y$, so if $X \leq_w Y$, then $X \leq_s Y$ and $X \leq_s Z$. $\diamondsuit$

The commonly used properties of the recursively enumerable sets have proofs at different levels of the hierarchy of induction schemes. The following is a list of the results we will use. The reader is asked to consult [**6**] for the proof.

2.10. Definition. Suppose that $\mathcal{M}$ is a model of $P^- + I\Sigma_0$ and W is recursively enumerable in $\mathcal{M}$. Then, we can order the elements of W in the order of their being witnessed to belong to W. We call this order the *natural enumeration* of W and let $W[s]$ be the subset of $W \cap {<} s$ enumerated by witnesses less than s.

Note that $x \in W[s]$ is a Σ_0 predicate on x and s.

2.11. Theorem. (1) *(H. Friedman) Suppose that $\mathcal{M}$ is a model of $P^- + I\Sigma_k$ and A is a Σ_k predicate on $\mathcal{M}$. Then A is amenable.*

(2) *Suppose that $\mathcal{M}$ is a model of $P^- + B\Sigma_{k+1}$, i is less than or equal to k and A is a Σ_i predicate on $\mathcal{M}$. If W is recursively enumerable in A, then W is Σ_{i+1} in $\mathcal{M}$.*

2.12. Corollary. *If $\mathcal{M}$ is a model of $P^- + B\Sigma_2$ then $\leq_w$ is transitive on the recursively enumerable predicates in $\mathcal{M}$.*

Proof: Suppose that $\mathcal{M}$ is given as above and that W is recursively enumerable in $\mathcal{M}$. Theorem 2.11 shows that W is amenable and that $\mathcal{M}$ satisfies $B\Sigma_1$ relative to W. By proposition 2.9, $\leq_w$ is equal to $\leq_s$ below W and so is transitive there. $\diamondsuit$

2.13. Corollary. *Suppose that $\mathcal{M}$ is a model of $P^- + I\Sigma_1$ and W is recursively enumerable in $\mathcal{M}$. For any Turing reduction Φ, $\Phi(x, W) = y$ if and only if there is a computation in Φ that applies to $W[t]$ for all sufficiently large t.*

Proof: By Theorem 2.11, W is an amenable subset of $\mathcal{M}$. Suppose that $\langle x, y, F, G\rangle$ is a computation that applies to W. Let m be an upper bound on the elements of F. By $B\Sigma_1$ in $\mathcal{M}$, fix s so that $W \cap {<} m$ is equal to $W[s] \cap {<} m$. For any t greater than or equal to s, $\langle x, y, F, G\rangle$ applies to $W[t]$. The converse is clear. $\diamondsuit$

Corollary 2.13 gives us the ability to apply the basic intuition of the priority method. During a construction of A, we can wait for a stage when $\Phi(x, A[s])$ is defined. If no such stage appears, then $\Phi(x, A)$ will be undefined.

§3. The Model

We will focus our attention on the difference between $I\Sigma_1$ and $B\Sigma_2$. In particular, we will work with a model of $I\Sigma_1$ that is not a model of $B\Sigma_2$.

We obtain this model by forming a limited Skölem hull in a nonstandard model of Peano Arithmetic.

During our discussion of nonstandard models and their elements, we will reserve the word *integer* for a standard element of ω.

Let $\mathcal{N}$ be a model of Peano Arithmetic with nonstandard element p. We build a Σ_1-substructure $\mathcal{M}$ of $\mathcal{N}$ with an additional property: If X is Π_1 definable in $\mathcal{N}$ with parameters from $\mathcal{M}$ and has an element less than some element of $\mathcal{M}$, then the $\mathcal{N}$-least element of X is in $\mathcal{M}$.

Let $\vec{\varphi}$ and $\vec{\psi}$ be standard recursive enumerations of the Σ_1 and Π_1 formulas of arithmetic, respectively. Note, in $\mathcal{N}$, we can specify $\mathcal{N}$-recursive enumerations having $\vec{\varphi}$ and $\vec{\psi}$ as initial segments.

We build $\mathcal{M}$ in ω many steps. Begin by putting p into $\mathcal{M}$.

At the beginning of the $k+1$st step, we have determined that $\mathcal{M}[k] = m_1, \dots, m_i$ is contained in $\mathcal{M}$. First, for each of the first k statements in $\vec{\varphi}$ about elements in $\mathcal{M}[k]$ true in $\mathcal{M}$, if there is no witness to this (existential) statement among the elements of $\mathcal{M}[k]$, we add the $\mathcal{N}$-least witness to $\mathcal{M}$. Secondly, for each ψ among the first k formulas in $\vec{\psi}$ and each sequence of parameters $\vec{m}$ from $\mathcal{M}[k]$, if there is an x in $\mathcal{N}$ such that x is a solution to ψ relative to $\vec{m}$ and x is less than some element of $\mathcal{M}[k]$, then we add the $\mathcal{N}$-least solution of $\psi(-,\vec{m})$ to $\mathcal{M}$. $\mathcal{M}[k+1]$, the set of elements we have now determined lie in $\mathcal{M}$, consists of the union of $\mathcal{M}[k]$ with the set of elements added at step $k+1$.

3.1. Proposition. *If $\mathcal{N}$ is a nonstandard model of P and $\mathcal{M}$ is constructed from $\mathcal{N}$ as above then the following conditions hold.*

(1) *$\mathcal{M}$ is a Σ_1-substructure of $\mathcal{N}$, written as $\mathcal{M} \prec_{\Sigma_1} \mathcal{N}$. That is, for any Σ_1-formula φ and elements $\vec{m}$ of $\mathcal{M}$,*

$$\mathcal{M} \models \varphi(\vec{m}) \iff \mathcal{N} \models \varphi(\vec{m}).$$

Note that (1) implies that there is no difference in the interpretation of Σ_1 or Π_1 formulas between $\mathcal{N}$ and $\mathcal{M}$. We will no longer specify where such a formula is to be interpreted.

(2) *Suppose that θ is a Π_1-formula and $\vec{m}$ is contained in $\mathcal{M}$. If $\theta(\vec{m},-)$ defines a nonempty subset X of $\mathcal{M}$, then the $\mathcal{N}$-least element of X is an element of $\mathcal{M}$. Consequently, $\mathcal{M}$ is a model of $I\Sigma_1$.*

(3) *Let μ denote the function that maps i to the ith element added to $\mathcal{M}$. In $\mathcal{M}$, μ is recursive in $\emptyset'$, the complete Σ_1 set.*

(4) *In $\mathcal{M}$, there is a recursive function of two variables $\mu(-,s)$ such that,*

for all i in ω,

$$\mathcal{M} \models \lim_{s \to \infty} \mu(i,s) = \mu(i).$$

Proof: For the first claim, note that each element of $\mathcal{M}$ is introduced at some finite point in the recursion. Once $\vec{m}$ is known to be contained $\mathcal{M}$, the remaining steps of the construction include the action of adding to $\mathcal{M}$ a witness for each Σ_1 statement true about $\vec{m}$ in $\mathcal{N}$. This ensures that $\mathcal{M} \prec_{\Sigma_1} \mathcal{N}$.

The second claim follows by an analogous argument. We took active steps to ensure that the $\mathcal{N}$-least element of a Π_1 subset of $\mathcal{M}$ is the same as the $\mathcal{M}$-least one. Since $\mathcal{N}$ is a model of Peano Arithmetic, every Π_1 subset of $\mathcal{N}$ that is not empty has a least element. The same is then true in $\mathcal{M}$. This condition is equivalent to induction for all Σ_1 formulas in $\mathcal{M}$ (see [**5**, Kirby-Paris]).

For the third claim, note that the action taken during the kth step of the construction is determined by the satisfaction predicate in $\mathcal{N}$ on a finite set of Σ_1 and Π_1 statements about elements from $\mathcal{M}$. As $\mathcal{M}$ is a Σ_1-substructure of $\mathcal{N}$, the satisfaction predicate for these statements is identical between $\mathcal{M}$ and $\mathcal{N}$. Thus, the action taken during this step is accurately described by $\emptyset'$, the complete Σ_1 subset of $\mathcal{M}$.

For the final claim, consider the function

$$\mu(i,s) = \begin{cases} \mu(i, \emptyset'[s]), & \text{if this function converges by a computation} \\ & \text{less than } s; \\ 0, & \text{otherwise.} \end{cases}$$

By corollary 2.13, the values of $\mu(i,s)$ will be correct for all sufficiently large s. $\diamondsuit$

§4. An instance of intransitivity in $\leq_w$ among the recursively enumerable sets

Let $\mathcal{M}$ be the model described in the previous section. In particular, $\mathcal{M}$ is a model of $P^- + I\Sigma_1$ and not a model of $B\Sigma_2$. In $\mathcal{M}$ there is a function μ from ω onto $\mathcal{M}$, which is the limit of a recursive function of two variables $\mu(-,s)$: For all i in ω, $\mathcal{M} \models \lim_{s \to \infty} \mu(i,s) = \mu(i)$.

4.1. Proposition. *In $\mathcal{M}$, there are recursively enumerable sets A, B and C such that $A \leq_w B \leq_w C$ but $A \not\leq_w C$.*

The proposition has the next theorem as an immediate corollary.

4.2. Theorem. *There is no proof from $P^- + I\Sigma_1$ that $\leq_w$ is transitive on the recursively enumerable sets.*

The remainder of this section is devoted to presenting a proof of Proposition 4.1.

Priority constructions. We follow the conventional form for a priority construction using *stages* and *strategies*. A stage is just an element s of $\mathcal{M}$ viewed in the context of a definition by recursion.

In the usual dynamic picture of a priority argument, the construction of a set A is viewed as an enumeration of A (or of the sequence $\langle A[s] \mid s \in \mathcal{M} \rangle$), guided at stage s by a finite collection $\Sigma(s)$ of strategies. $\Sigma(s)$ also specifies for each strategy certain parameters whose role will become clear later, such as the *state* of the strategy and the *restraint* imposed by the strategy, and generally specifies a *priority ordering* on the strategies. We specify the construction by giving a recursive procedure that, given the construction $\langle A[s], \Sigma(s) \mid s < t \rangle$ up to stage t, specifies the construction $\langle A[t], \Sigma(t) \rangle$ at stage t. Informally, we interpret $\Sigma(t-1)$ as a collection of constraints on the continuation of the construction, and give a procedure to produce a continuation that respects as many of those constraints as possible. Formally, we define a recursive set X (of appropriate partial constructions) such that, given any sequence $\langle A[s], \Sigma(s) \mid s < t \rangle$ in X, there is at least one $\langle A[t], \Sigma(t) \rangle$ such that $\langle A[s], \Sigma(s) \mid s \leq t \rangle$ is in X. From this, we conclude that there is a recursive construction $\langle A[s], \Sigma(s) \mid s \in \mathcal{M} \rangle$, such that each initial segment $\langle A[s], \Sigma(s) \mid s < t \rangle$ is in X. The theory $I\Sigma_1$ is just strong enough to justify such a conclusion. Therefore, the dynamical presentation of a construction (given a construction up to stage t, take certain recursively specified actions to determine the construction at stage t) is valid when working in a model of $I\Sigma_1$. (Generally, the proof that a construction presented in this manner satisfies the desired properties may require a stronger form of induction.)

Informally, a strategy σ is a recursive procedure to be applied during a enumeration of sets and functionals (Turing reductions). During each stage of the enumeration, the strategy exerts enough control over what happens during that stage to force the desired properties in the limit.

The direct effects of a strategy are called *actions*. One possible action of a strategy is to impose constraints on the possible continuation of the enumeration. For example, σ may *restrain* w from entering A by not allowing the construction to enumerate w. Similarly, σ may require that a particular number enter A or that the intersection of A with some nonempty $\mathcal{M}$-finite set is not empty. In the last case, σ does not determine which element enters A, but leaves that decision open to the effects of other strategies.

A second possibility is that a strategy may constrain the actions of later strategies in a way that is not so directly phrased in terms of the sets being enumerated. For example, we will use strategies that prohibit later strategies from imposing too large a restraint.

In addition to actions, a strategy may change its actions in response to the construction's taking a favorable turn. We call this *changing state*. The actions taken by a strategy in a fixed state produce an *environment* for the remaining strategies.

All of our strategies are recursively specified in terms of some parameters from $\mathcal{M}$. The parameters for a specific strategy will be chosen during the construction so as to be consistent with the environment produced by the actions of earlier strategies. When we choose these parameters, we say that we have *initialized* the strategy.

When we refer to a *construction*, we will always mean an enumeration of sets and functionals together with the evolution in the states and actions of the strategies involved. The construction will be presented by a recursion wherein the continuation of the construction from stage $s-1$ to stage s is recursively determined from the earlier stages.

We say that a construction is an *execution* of a strategy if for every stage the actions of the construction during that stage are compatible with the constraints of the strategy.

To prove the theorem, we will represent its statement as an infinite conjunction of requirements. Each requirement will be assigned a set of strategies designed to influence the construction so as to produce sets that satisfy the requirement. At the beginning of stage s, we will choose a specific sequence of strategies. Some initial segment of these will have been in use during the previous stage and will have reached a particular state by the end of that stage. Our first move during stage s is to initialize the new strategies so as to work in the environment imposed by earlier strategies. Once initialized, each strategy determines a constraint on the allowed actions in the construction during that stage. The actions we take are chosen to respect the longest possible initial segment of the sequence of strategies.

During the construction, we will use $[s]$ as a suffix to indicate that the preceding expression is to be approximated by the information available at the beginning of stage s. Consistently with the notation already established, we use $A[s]$ to indicate the numbers that have already been enumerated into A by earlier action. Similarly, $\Phi(x, A) = y[s]$ indicates that there is a computation of Φ at x relative to $A[s]$ which is less than s and gives value y.

The reader interested in a more elaborate exposition and foundational treatment of the priority method may consult [**3**, Groszek-Slaman], to appear.

We make the following definitions for the sake of this proof.

4.3. Definition. An *enumeration* of a set A is a recursive sequence $\langle A[s] \mid s \in I\rangle$ (I some initial segment of $\mathcal{M}$) of $\mathcal{M}$-finite sets such that $A[0]$ is the empty set and, for all s, $A[s] \subseteq A[s+1]$.

If $I = \mathcal{M}$, this is a *full enumeration*; if $I = {<}t+1$, a *finite enumeration* of length t. The set enumerated is $A = \cup\{A[s] \mid s \in I\}$. We say that x is *enumerated into* A *at stage* s if and only if $x \in A[s+1]$ and for all $r \leq s$, $x \notin A[r]$. We leave the following lemma to the reader.

4.4. Lemma. (In $P^- + I\Sigma_1$) *Suppose that $\langle A[s] \mid s \in I\rangle$ is a recursive enumeration. For all $s < t$, $A[s] \subseteq A[t]$. Furthermore, if $x \in A$, then there is a unique s such that x is enumerated into A at stage s.*

Requirements. We will enumerate sets A, B and C in $\mathcal{M}$ by means of a Π_1-priority construction (finite injury). The function μ from ω onto $\mathcal{M}$ will be used to arrange the requirements in order type ω. We use μ so that we can impose an $\mathcal{M}$-finite bound on the number of strategies which are active at any stage. This allows us to prove that the requirements are satisfied by induction on ω. This is useful, because $\mathcal{M}$ satisfies only a limited induction scheme, whereas the proof of the theorem uses a stronger instance of induction than is satisfied by $\mathcal{M}$. Since $\mu(-)$ is not recursive, during the construction we will use its recursive approximation $\mu(-, s)$ and change the priority order when the approximation changes.

4.5. Definition. The value $\mu(i)$ *changes at stage* s if $\mu(i, s) \neq \mu(i, s-1)$. If $\mu(j)$ changes at stage s for some j less than i, then $\mu \restriction i$ *changes at stage* s.

By the properties of μ, for any i, there is a stage t such that $\mu \restriction i$ does not change at any stage greater than t, if and only if i is an integer.

The following lemma is a standard application of induction.

4.6. Lemma. (In $P^- + I\Sigma_1$) *If there is any i such that $\mu(i)$ changes at stage s, then there is a least such i.*

There are three conditions to be satisfied:

(1) $B \leq_w C$;
(2) $A \leq_w B$;
(3) $A \not\leq_w C$.

Each of the first two will be ensured by a family of coding strategies. The third condition is divided into a conjunction of diagonal requirements, $\Theta(-,C) \neq A$, for each Turing functional Θ in $\mathcal{M}$. The coding strategies must be gentle enough so that the diagonal strategies $D(\Theta)$ are dense, that is, are always allowed the option of taking action.

The coding strategy $C(B \to C)$. Globally, the coding strategy is fairly passive. For a particular Turing functional Φ, $C(B \to C)$ will act at each stage s to enforce $\Phi(-,C) = B[s]$. This will ensure that, if $\Phi(m,C)$ is defined in the limit, it is equal to $B(m)$. For every m in $\mathcal{M}$, we use an additional strategy $C(B \to C, m)$ to ensure that $\Phi(m,C)$ is defined in the limit. Note, even when given the correct value for $B(m)$ it will not be possible to recursively identify the limiting computation of $\Phi(m,C)$.

We recursively view $\mathcal{M}$ as isomorphic to $\mathcal{M} \times \mathcal{M}$. Let the nth column of C be defined by

$$C^{(n)} = \left\{ x \,\middle|\, \langle n, x\rangle \in C \right\}.$$

We enumerate m into $C^{(n)}$ by enumerating $\langle n, m\rangle$ into C.

4.7. Definition. x is *enumerated into $C^{(m)}$ at stage s* if and only if $\langle x, m\rangle$ is enumerated into C at stage s. $C^{(m)}[s] = (C[s])^{(m)}$.

$C(B \to C)$ codes B into C as follows: For all m, there is a $c(m)$ such that $C^{(m)}$ is equal to $<c(m)$, and m is an element of B if and only if $c(m)$ is odd.

The stage-by-stage effect of $C(B \to C)$ is to require, for every stage s and every m in $\mathcal{M}$ less than s, if $c(m)$ is the least element not in $C^{(m)}[s]$ then $C^{(m)} = <c(m)$ and

$$m \in B[s] \iff c(m) \text{ is odd.}$$

In particular, to ensure the first condition, $C(B \to C)$ requires if $\langle m, c\rangle$ is enumerated into C and x is less than c, then $\langle m, x\rangle$ must also enter C (if not already in C). Further, $C(B \to C)$ imposes a condition on the use of later strategies: no strategy can restrain the mth column of C without also restraining m from entering B.

The additional effect of the strategy $C(B \to C, m)$ is to ensure that if m is not enumerated into B during stage s, then the least element of the complement of $C^{(m)}[s]$ is not enumerated into $C^{(m)}$ during stage s. Consider the functional Φ that is computed as follows. For argument m relative to predicate X, first compute the least element $c(m)$ of the complement of $X^{(m)}$; if $c(m)$ is odd give the value 1 and if $c(m)$ is even give the value 0.

The elementary fact that these strategies together ensure that $\Phi(-, C) = B$, is recorded below.

4.8. Definition. An enumeration of sets B and C *respects the coding strategy* $C(B \to C)$ *at stage* s if and only if, for all m,

(1) For all x, if x is enumerated into $C^{(m)}$ at stage s, so is every number less than x which is not already in $C^{(m)}[s]$.
(2) If m is not enumerated into B at stage s, then the least number not in $C^{(m)}[s+1]$ has the same parity as the least number not in $C^{(m)}[s]$.
(3) If m is enumerated into B at stage s, then the least number not in $C^{(m)}[s+1]$ is odd.

4.9. Definition. An enumeration of sets B and C *respects the coding strategy* $C(B \to C, m)$ *at stage* s if and only if, if m is not enumerated into B at stage s then nothing is enumerated into $C^{(m)}$ at stage s.

4.10. Lemma. *Suppose an enumeration of B and C respects $C(B \to C)$ at every stage s, and for every m there is a t, such that the enumeration respects $C(B \to C, m)$ at every stage s greater than or equal to t. Then, $B \leq_w C$.*

Proof: We show, for each m, there is a number $c(m)$ such that $C^{(m)}$ is equal to $< c(m)$ (i.e. $x \in C^{(m)}$ if and only if $x < c(m)$,) and furthermore, $c(m)$ is odd if and only if $m \in B$. From this it is easy to show that $B \leq_w C$.

Let m be given. By $I\Sigma_1$ and the fact that the enumeration respects $C(B \to C)$ at every stage, for each s there is a $c(m, s)$ such that $C^{(m)}[s] = < c(m, s)$, and $c(m, s)$ is odd if and only if $m \in B[s]$.

Now let t be a stage such that the enumeration respects $C(B \to C, m)$ at every stage s greater than or equal to t, and if $m \in B$, then $m \in B[t]$. By $I\Sigma_1$ and $C(B \to C, m)$, for all $s \geq t$, $C^{(m)}[s] = C^{(m)}[t] = < c(m, t)$; $c(m, t)$ is odd if and only if $m \in B[t]$ if and only if $m \in B$. Therefore, letting $c(m)$ equal $c(m, t)$, we are done. $\diamondsuit$

Note, it is not hard to produce enumerations of this sort. Given any enumeration of a set B, and any function $f : \mathcal{M} \to \mathcal{M}$ which is a limit of recursive approximations (in the sense that μ is), there is an enumeration of a set C such that the pair of enumerations satisfies the hypotheses of the lemma, and for all m, $C^{(m)} \neq C^{(m)}[f(m)]$. (This uses $I\Sigma_1$.)

The coding strategy $C(A \to B)$. The strategies for coding A into B are very similar to, but not identical with, those that code B into C. The difference is that we code A into B using a nonstandard negative neighborhood condition on B.

Recall that p is a nonstandard element of $\mathcal{M}$. We define the strategies $C(A \to B)$ and $C(A \to B, m)$ analogously to $C(B \to C)$ and $C(B \to C, m)$ so that the pth element of the complement of $B^{(m)}$ records whether m is an element of A. (This is in analogy to saying the first element of the complement of $C^{(m)}$ records whether m is an element of B.) That is, during each stage s, $C(A \to B)$ enforces the condition

$$m \in A \iff \text{the } p\text{th element of the complement of } B^{(m)} \text{ is odd.}$$

$C(A \to B, m)$ enforces the condition that the first p elements of the complement of $B^{(m)}$ can only be changed under the condition that m enters A.

4.11. Convention. We will take p to be odd so that before any numbers are enumerated in any set Ψ is correctly computing A from B. Note, 0 is the first element of $\mathcal{M}$.

Let $\Psi(x, X)$ be computed by finding the pth element $b(x)$ of the complement of $X^{(x)}$ and giving answer 0 if $b(x)$ is even and answer 1, otherwise.

4.12. Definition. An enumeration of sets A and B *respects the coding strategy* $C(A \to B)$ at stage s if and only if, for all m,

(1) For all x, if x is enumerated into $B^{(m)}$ at stage s ($x \in B^{(m)}[s+1]$), then at most $p-1$ numbers less than x are not in $B^{(m)}[s+1]$.
(2) If m is not enumerated into A at stage s, the pth element in the complement of $B^{(m)}[s+1]$ has the same parity as the pth element in the complement of $B^{(m)}[s]$.
(3) If m is enumerated into A at stage s, the pth element of the complement of $B^{(m)}[s+1]$ is odd.

4.13. Definition. An enumeration of sets A and B *respects the coding strategy* $C(A \to B, m)$ *at stage* s if and only if, if m is not enumerated into A at stage s, then nothing is enumerated into $B^{(m)}$ at stage s.

4.14. Lemma. *Suppose that an enumeration respects $C(A \to B)$ at every stage, and for each m there is a t, such that the enumeration respects $C(A \to B, m)$ at every stage s greater than or equal to t. Then $A \leq_w B$.*

Proof: Exactly analogous to the coding of B into C in Lemma 4.10. $\diamondsuit$

Again, it is easy to find such enumerations and even to find enumerations of sets A, B and C such that the hypotheses of Lemmas 4.10 and 4.14 are satisfied.

The diagonal strategies.

4.15. Definition. Given a Turing reduction (functional) Φ, we say $\Phi(x, C) = y[s]$ if and only if there is a $C[s]$-computation $\langle x, y, P, N\rangle$ in $\Phi[s]$.

Under $I\Sigma_1$, $\Phi(x, C) = y$ if and only if there is a computation $\langle x, y, P, N\rangle$ in Φ which is a $C[s]$-computation for all sufficiently large s.

Let Θ be a given Turing functional, and suppose that q is a nonstandard element of $\mathcal{M}$. We will describe a strategy $D(\Theta)$ to work in an environment where the columns of C with indices in F are restrained from being changed, and some $\mathcal{M}$-finite set may be restrained from A.

(0) First, chose w so that w is not prohibited from entering A but has not yet done so. The number w will be a witness to inequality. We restrain the construction from enumerating w in A and go to (1). *This is the initialization of the strategy; it takes place whenever a strategy of higher priority changes state. The move from (0) to (1) does not constitute a change of state for $D(\Theta)$.*

(1) If $\Theta(w, C) = 0[s]$ by a computation whose negative condition involves less than q many columns of C whose indices are not mentioned in F, then go to (2).

Otherwise: If $\Theta(w, C) = 0[s]$ *(with negative condition necessarily spanning q or more C-columns not mentioned in F),* we require that some element from the negative condition associated with the least such computation must be enumerated into C during stage s. *This "destroys" the potential computation $\Theta(w, C) = 0$ seen at stage s. Thus, we force the construction to produce a predicate C such that, if $\Theta(w, C)$ is defined then its computation uses a negative condition on C that is contained in the union of less than q many columns of C beyond those indexed in F.*

(2) *We are given that $\Theta(w, C) = 0[s]$ by means of a computation with permitted negative condition.* We restrain all of the elements of the C-columns beyond those in F mentioned in the negative part of the computation of $\Theta(w, C)[s]$ from entering C. Further, if $C^{(m)}$ is restrained then we also restrain m from entering B. In addition, we require that w be enumerated into A. *This is the usual diagonal step: preserve a computation from C and change A so as to diagonalize.*

4.16. Definition. An enumeration of sets A and C, sequence $st = \langle st(r) \mid r > t\rangle$ (of $D(\Theta)$-states), and sequence $res = \langle res(r) \mid r > t\rangle$ (of $D(\Theta)$-restraints) *respects the strategy $D(\Theta)$ at stage $s \geq t$* (with parameter q, witness w, initialization stage t, and initial restraint F on columns of C) if and only if:

(1) If $m \in F$ then nothing is enumerated into $C^{(m)}$ at stage s. (The columns of C indexed by F are restrained.)
(2) If $s = t$: $st(t+1) = 1$ ($D(\Theta)$ is in state 1 at the end of stage t), $res(t+1) = \emptyset$ ($D(\Theta)$ is restraining no columns of C at the end of stage t), and $w \notin A[t+1]$.
(3) If $res(s) = X$ and $m \in X$, then nothing is enumerated into $C^{(m)}$ at stage s. (The columns of C indexed by $res(s)$ are restrained.)
(4) If $s > t$ and $st(s) = 1$ then:
   (a) If $\Theta(w, C) \neq 0[s]$, then $st(s+1) = st(s)$, $res(s+1) = res(s)$, and w is not enumerated into A at stage s.
   (b) If $\Theta(w, C) = 0[s]$ and the least such computation in $\Theta[s]$ has a negative condition G spanning q or more columns of C whose indices are not in F, then an element of G is enumerated into C at stage s ("destroying" that computation), $st(s+1) = st(s)$, $res(s+1) = res(s)$, and w is not enumerated into A at stage s.
   (c) If $\Theta(x, C) = 0[s]$ and the least computation of this has a negative condition G spanning columns of C whose indices are in $F \cup X$, where X is disjoint from F and has size less than q, then w is enumerated into A at stage s (so $A[s+1](w) = A(w) = 1$), for any $m \in X$ nothing is enumerated in $C^{(m)}$ at stage s (so the computation $\Theta(w, C) = 0[s]$ is preserved, giving at this stage an inequality between $\Theta(w, C)$ and $A(w)$), $st(s+1) = 2$, and $res(s+1) = X$. (The columns of C indexed by X are restrained.)
(5) If $s > t$ and $st(s) = 2$, $res(s) = X$, then $st(s+1) = 2$ and $res(s+1) = X$. (Continue to restrain the columns of C indexed by X, preserving an inequality between $\Theta(w, C)$ and $A(w)$.)

4.17. Notation. We use $D(\Theta, q, w, t, F)$ to refer to the strategy $D(\Theta)$ with parameter q, witness w, initialization stage t, and initial restraint F on the columns of C.

4.18. Lemma. *If a full enumeration of sets A and C, sequence st, and sequence res, respect $D(\Theta, q, w, t, F)$ at every stage s greater than or equal to t, then $\Theta(w, C) \neq A(w)$.*

Proof: First, by $I\Sigma_1$, the sequences st and res are uniquely defined by the enumerations of A and C. The state $st(r)$ is 1 if $w \notin A[r]$ and 2 if $w \in A[r]$. The restraint $res(r)$ is empty if $w \notin A[r]$ and X if $w \in A[r]$, where X is defined in (4c) above.

If $w \in A$, then w is enumerated into A at some stage $s > t$; $st(s) = 1$, and case (4c) holds at stage s. I.e. $\Theta(w, C) = 0[s]$ via a computation $\langle w, 0, P, N\rangle$ whose negative condition N spans columns of $C[s]$ indexed by $F \cup X$. We set $st(s + 1) = 2$, $res(s + 1) = X$. By $I\Sigma_1$, for $m \in F \cup X$ and $r \geq s$, $C^{(m)}[r] = C^{(m)}[s]$, so $C^{(m)} = C^{(m)}[s]$. This means, since $N \cap C[s] = \emptyset$, $N \cap C = \emptyset$; i.e., $\langle w, 0, P, N\rangle$ is a C-computation. Thus, $\Theta(w, C) = 0$; but $A(w) = 1$, as desired.

If $w \notin A$, we must show that it is not the case that $\Theta(w, C) = 0$; suppose then that $\Theta(w, C) = 0$. Using the amenability of C, we can show that there is a least C-computation $\langle w, 0, P, N\rangle$ in Θ, and furthermore, there is a stage s greater than t such that it is the least $C[s]$ computation in $\Theta[s]$. At stage s, then, we are in case (4b) (it cannot be case (4c), since we assume $w \notin A$.) But we then enumerate an element of N into C at stage s, so $N \cap C \neq \emptyset$, contradicting the assumption that $\langle w, 0, P, N\rangle$ is a C-computation. $\diamondsuit$

Enumerating sets A and C to satisfy these hypotheses is not a problem; when we try to combine $D(\Theta)$ with $C(B \to C)$ and $C(A \to B)$, we first see a potential conflict: In case (4c), $D(\Theta)$ requires that w enter A and certain numbers are kept out of C. But as w enters A, $C(A \to B)$ requires that something enter B, and so $C(B \to C)$ requires that something enter C; conceivably, one of the same numbers that $D(\Theta)$ requires be kept out of C.

This is why the distinction is made between cases (4b) and (4c): in case w is to enter A at stage s, we are in case (4c), and fewer than $|F| + q$ many columns of C are being restrained; as w enters A, we have a choice of p many numbers (the first p elements of the complement of $B^{(w)}[s]$) which can enter B in order to satisfy $C(A \to B)$; if $p \geq |F| + q$, one of these must index a column of C which is not being restrained.

To see why case (4c) is necessary at all (in other words, why not let $q = 0$,) we need to consider the nature of finite injury priority arguments. Usually, in a finite injury construction, we consider a sequence of strategies $S_1, S_2, S_3, \ldots$ ordered according to decreasing priority. At a given stage s, some initial segment $S_1, \ldots, S_n$ of strategies has been initialized (with certain parameters) and each is currently imposing some constraints on the construction. We look to see which is the least S_i that requires action involving a change of state. At stage s we take that action, while continuing to respect the constraints imposed by the strategies of higher priority $S_1, S_2, \ldots, S_{i-1}$. (In general, S_i has been initialized so that this is possible; e.g., if $S_i = D(\Theta)$, the initial restraint F on columns of C is the restraint imposed by $S_1, \ldots, S_{i-1}$, and the witness w is a number not restrained from entering A by any of the $S_1, \ldots, S_{i-1}$.) We do not, however, worry about respecting $S_{i+1}, \ldots, S_n$; these lower priority strategies are *injured*, and must be re-initialized to be consistent with the new state of S_i. This combinatorial pattern works provided any strategy, if not injured, will only change state finitely often: S_1 will never be injured, so eventually will stop acting. After that, S_2 will not be injured, so eventually will stop acting, and so forth. Thus, we can (using $I\Sigma_2$ on the priority ordering) put together a construction which eventually respects every S_i.

Each $D(\Theta)$, if never injured (i.e. always respected,) changes state at most once; however, it may act positively as in case (4b) infinitely often. This action is not a problem, if it is taken in a way that does not injure any strategy, even those of lower priority. The way to guarantee this is to choose q so large that the total restraint imposed by strategies of lower priority at any stage will involve fewer than q columns of C; thus, in case (4c) there will always be an element of G whose column is not indexed in F and is not restrained by any strategy. This element of G can be enumerated into C without injuring anything.

In the final construction, we will show how to choose parameters q for each $D(\Theta)$ so the sizes of their restraints fit together appropriately. The following lemma illustrates the use of the function μ to order strategies, in such a way that a single $D(\Theta)$ can be respected in a way which is compatible with all of the coding strategies.

4.19. Lemma. *Suppose Θ is a Turing functional, F is an $\mathcal{M}$-finite set, and q and r are non-standard integers, such that $|F| + q \le p$ and $r < q$. Let w and t be any numbers in $\mathcal{M}$. Then, there are full enumerations of sets A, B, C, and sequences st and res, satisfying the hypotheses of Lemmas 4.10, 4.14, and 4.18.*

Proof: For $s \le t$, let $A[s] = B[s] = C[s] = \emptyset$. We will inductively define these enumerations, for $s \ge t$, so that at stage s they respect $C(A \to B)$, $C(B \to C)$, $C(A \to B, m)$ and $C(B \to C, m)$ for any $m = \mu(x, s)$ for $x < r$. (Simultaneously, of course, we will define suitable sequences st and res.) Since for all m, there is a standard x such that $m = \mu(x, s)$ for sufficiently large s, this will prove the lemma.

Define $st(t+1) = 1$, $res(t+1) = \emptyset$, $A[t+1] = B[t+1] = C[t+1] = \emptyset$.

Given $s > t$, suppose the enumerations are defined up to stage s. At stage s, we have four possible cases for $D(\Theta)$.

In case (4a) or (5) no action is called for: let $A[s+1] = A[s]$, $B[s+1] = B[s]$, $C[s+1] = C[s]$, $st(s+1) = st(s)$, and $res(s+1) = res(s)$.

In case (4b), we need to enumerate an element of G into C: G spans at least q columns of C whose indices are not in F; as $r < q$, at least one of those columns has an index m not in $\{\mu(x,s) \mid x < r\}$. Enumerate that element into C, along with any other elements of the mth column needed to guarantee $C(B \to C)$ is respected (i.e. all smaller numbers, and the next largest if that is necessary to preserve the correct parity of the least element of the complement of $C^{(m)}[s]$.) Enumerate nothing into any other column of C, and let $A[s+1] = A[s]$, $B[s+1] = B[s]$, $st(s+1) = st(s)$, $res(s+1) = res(s)$.

In case (4c), $D(\Theta)$ changes state: we must enumerate w into A (let $A[s+1] = A[s] \cup \{w\}$) and let $st(s+1) = 2$, $res(s+1) = X$. We must also enumerate an element m into B, which is one of the first p elements of the complement of $B^{(w)}[s]$, in order to respect $C(A \to B)$; let $B[s+1] = B[s] \cup \{m\}$. Finally, we enumerate the least element of the complement of $C^{(m)}[s]$ in order to respect $C(B \to C)$ (and enumerate nothing else into C.) In order to respect $D(\Theta)$, we must choose m properly, so that $m \notin F$ and $m \notin X$. But by our choice of F and q, $|F| + |X| < p$; thus we can chose m so that $m \notin F \cup X$. To specify the construction unambiguously, let m be the least such number.

It is easy to check that, in any case, we have respected all the necessary strategies at stage s. ♢

Compatibility between strategies. It only remains to chose strategies for each requirement so that all of the strategies are compatible and, for each requirement, the strategies associated with it are dense. This will follow from a judicious choice of parameters q for the strategies $D(\Theta)$.

4.20. Definition. Let r be any non-standard number, and let $p = 2^{2r} + 1$. For $0 < i < r$, let $q_i = \frac{p-1}{2^i} = 2^{2r-i}$.

4.21. Lemma. (1) $\sum_{j=1}^{r-1} q_j + r = \sum_{k=r+1}^{2r-1} 2^k + r = 2^{2r} - 2^{r+1} + r < p$.
(2) $\sum_{j=i+1}^{r-1} q_j + r = 2^{2r-i} - 2^{r+1} + r = q_i - 2^{r+1} + r < q_i$.

Proof: Elementary calculation. ♢

4.22. Definition. An enumeration of sets A, B, and C is an *execution* of the sequence of strategies $\langle D(\Theta_i, q_i, w_i, t_i, F_i), C(A \to B, x_i), C(B \to C, x_i) \rangle$ provided

(1) Each t_i is less than the length of the enumeration.
(2) $C(A \to B)$ and $C(B \to C)$ are respected at every stage.
(3) $C(A \to B, x_i)$ and $C(B \to C, x_i)$ are respected at every stage $s \geq t_i$.
(4) $D(\Theta_i, q_i, w_i, t_i, F_i)$ is respected at every stage $s \geq t_i$. (This means there are appropriate sequences st and res defined from the enumer-

ation and parameters Θ_i, q_i, w_i, t_i, F_i; let X_i denote the final, or in the case of a full enumeration the limiting, value of $res(s)$.)

(5) $F_i = \bigcup\{X_j \mid j < i\} \cup \{x_j \mid j < i\}$. ($F_i$ is the set of columns of C restrained by the strategies of higher priority, or equivalently, of lower index, than $D(\Theta_i)$.)

(6) $w_i \notin \{w_j \mid j < i\}$. (No strategy of higher priority than $D(\Theta_i)$ restrains w_i from entering A.)

4.23. Corollary. *If this is a finite sequence with length at most r, then for any i, $|F_i| + q_i < p$, and $|\,\{x_j \mid j \geq i\} \cup \bigcup\{X_j \mid j > i\}\,| < q_i$.*

Proof: As each X_j has size less than q_j, this is a corollary to Lemma 4.21. ♢

4.24. Lemma. *If a full enumeration of sets A, B and C is an execution of a sequence of strategies $\langle D(\Theta_i, q_i, w_i, t_i, F_i), C(A \to B, x_i), C(B \to C, x_i)\rangle$ such that each Turing reduction Θ appears as one of the Θ_i, and each number m appears as one of the x_i, then $A \leq_w B$, $B \leq_w C$, but $A \not\leq_w C$.*

Proof: $A \leq_w B$ and $B \leq_w C$ follow from Lemmas 4.10 and 4.14. $A \not\leq_w C$ follows from Lemma 4.18, since if $A \leq_w C$, we must have $A = \Theta_i(C)$ for some Θ_i, which is impossible. ♢

To complete the proof of the theorem, we will construct in $\mathcal{M}$ an enumeration of sets A, B and C, which is an execution of a sequence of strategies $\langle D(\Theta_i, q_i, w_i, t_i, F_i), C(A \to B, x_i), C(B \to C, x_i) \mid 1 \leq i < \omega\rangle$ such that each Turing reduction Θ appears as one of the Θ_i and each number m appears as one of the x_i. This sequence of strategies will of course not be recursive, but it will be the limit of a recursive sequence which we will construct simultaneously with the enumerations.

First we prove a lemma showing how such a construction can be extended at a given stage.

4.25. Lemma. *Suppose a finite enumeration of sets A, B and C of length t is an execution of a sequence of strategies*

$$\langle D(\Theta_i, q_i, w_i, t_i, F_i), C(A \to B, x_i), C(B \to C, x_i) \mid 1 \leq i < r\rangle .$$

(The q_i are as defined in Definition 4.20.) Let k be such that no strategy $D(\Theta_i)$ for $i < k$ needs to change state at stage t.

Then there is a continuation of the enumeration to length $t+1$ which is an execution of

$$\langle D(\Theta_i, q_i, w_i, t_i, F_i), C(A \to B, x_i), C(B \to C, x_i) \mid 1 \leq i \leq k\rangle . \tag{4.26}$$

Furthermore, given any sequences $\left\langle \bar{\Theta}_i \mid k < i < r \right\rangle$ *and* $\langle \bar{x}_i \mid k < i < r \rangle$*, the continuation of the enumeration can be chosen to be an execution of the sequence of strategies in (4.26) extended by*

$$\left\langle D(\bar{\Theta}_i, q_i, \bar{w}_i, t, \bar{F}_i), C(A \to B, \bar{x}_i), C(B \to C, \bar{x}_i) \mid k < i < r \right\rangle$$

for an appropriate choice of $\bar{w}_i$ *and* $\bar{F}_i$*.*

PROOF: There are two cases to consider.

Case 1. $D(\Theta_k)$ needs to change state at stage t, due to a computation with negative condition spanning columns in C indexed by $F_k \cup X$, where $|X| < q_k$. Then we enumerate w_k into A at stage t ($A[t+1] = A[t] \cup \{w_k\}$.) Since $w_k \notin \{w_i \mid i < k\}$ this is compatible with respecting $D(\Theta_i)$ for all $i \leq k$.

To continue to respect the coding strategies, we choose $m = \langle y, w_k \rangle$ such that y is one of the first p elements of the complement of $B^{(w_k)}[t]$, and $m \notin F_k \cup X$. We enumerate m into B at stage t, and we enumerate the least element of the complement of $C^{(m)}[t-1]$ into $C^{(m)}$ at stage t. (We will also enumerate other numbers into C.) This is compatible with respecting $D(\Theta_i)$ for $i \leq k$; we can choose $m \notin F_k \cup X$ since $|F_k \cup X| < p$ by Corollary 4.23.

We enumerate other numbers into C at stage t as follows. In order to respect $D(\Theta_i)$ for $i \leq k$, we must avoid columns of C indexed by $F_k \cup X$. In order to respect $C(B \to C, x_i)$ for $i \leq k$ and $C(B \to C, \bar{x}_i)$ for $i > k$, we must avoid columns of C indexed by x_i, $i \leq k$, or $\bar{x}_i$, $i > k$; i.e. the elements of a set of size r, which we call Y. For $i < k$, X_i is already given; set $X_k = X$. For each $i < k$ such that $D(\Theta_i)$ requires enumerating something into C because of a computation $\langle w_i, 0, P, N \rangle$, N spans at least q_i columns of C not indexed by F_i. The number of columns which must be avoided in addition to those indexed by F_i (i.e., those indexed by $Y \cup \bigcup \{X_j \mid i < j \leq k\}$) is by Corollary 4.23 less than q_i. Hence we can find a column of C, $C^{(m)}$, which meets N, such that $m \notin F_k \cup X_k \cup Y$; enumerate into that column an element of N, and other elements as necessary to satisfy $C(B \to C)$. This is compatible with respecting $D(\Theta_j)$, $j \leq k$, and all coding requirements with parameters x_j, $j \leq k$, and $\bar{x}_j$, $j > k$.

Finally, we must initialize $D(\bar{\Theta}_i)$, $i > k$, so the enumeration respects these at stage t. To do so, choose $\bar{w}_i \notin A[t+1]$, such that the $\bar{w}_i$ are all distinct and disjoint from the w_i. Define $\bar{F}_i$ inductively as in Definition 4.22 (the definition of execution.) We say the original strategies indexed by $i > k$ are *canceled*. Note t_i for $i > k$ has been replaced by t.

Case 2. $D(\Theta_k)$ does not need to change state at stage t. Then enumerate nothing into A or B. Enumerate numbers into C for the sake of $D(\Theta_i)$, $i < k$, as in Case 1; also, do so for $D(\Theta_k)$ if necessary. Initialize the $D(\bar{\Theta}_i)$, $i > k$, as in Case 1. $\diamondsuit$

Note that in Case 1 it was necessary to re-initialize the $D(\Theta_i)$ for $i > k$ even if $\Theta_i = \bar{\Theta}_i$, since F_i will no longer satisfy the definition which $\bar{F}_i$ must satisfy.

It is important to note that $A[t]$, $B[t]$, $C[t]$ (and $\bar{w}_i$, $\bar{F}_i$, for $i > k$) can be defined uniformly and recursively from the enumeration up to stage t and the sequences

$$\langle D(\Theta_i, q_i, w_i, t_i, F_i), C(B \to C, x_i), C(A \to B, x_i) \mid 1 \leq i \leq k \rangle$$

and $\left\langle \bar{\Theta}_i, \bar{x}_i \mid k < i < r \right\rangle$.

Assigning priority. It only remains to recursively specify which sequence of strategies to use during each stage. For this we will use the recursive approximation $\mu(x, s)$ to the function $\mu(x)$ that maps ω onto $\mathcal{M}$.

4.27. Definition. For each s, let $l(s)$ be the greatest x less than or equal to s and less than r such that

$$(\forall y \leq x)\,[\mu(y, s-1) = \mu(y, s)]\,.$$

The domain of $\mu(-)$ is ω. This provides us with two dynamic features: first, there will be cofinally many s in $\mathcal{M}$ such that $l(s)$ is an integer; second, for any integer n, there is a t so that for each s greater than t, $l(s)$ is greater than n.

4.28. Definition. Let $\langle \Theta_m \mid m \in \mathcal{M} \rangle$ be an effective list in $\mathcal{M}$ of all of the Turing functionals. Define $P(s)$, the stage s priority sequence, to be the following sequence.

$$\begin{aligned} & C(B \to C), C(A \to B), \\ & \qquad D(\Theta_{\mu(1,s)}), C(B \to C, \mu(1, s)), C(A \to B, \mu(1, s)), \ldots \\ & \qquad\qquad D(\Theta_{\mu(l(s),s)}), C(B \to C, \mu(l(s), s)), C(A \to B, \mu(l(s), s)) \end{aligned}$$

The construction. The construction $\mathcal{C}$ will be a recursive sequence $\langle \mathcal{C}[s] \mid s \in \mathcal{M} \rangle$. It will incorporate enumerations of sets A, B and C; $\mathcal{C}[s]$ will have the form $\langle A[s], B[s], C[s], \Sigma[s] \rangle$. The last component $\Sigma[s]$ will be a sequence of strategies (given by the priority sequence $P(s)$ of Definition 4.28), such that the enumeration of length s given by $\langle \mathcal{C}[t] \mid t \leq s\rangle$ is an execution (Definition 2.22) of $\Sigma[s]$. The parameters, witnesses, initialization stages and initial restraints associated with the $D(\Theta)$ will also be given by the $\Sigma[s]$.

By $I\Sigma_1$, we can define $\mathcal{C}$ by induction on s. We must guarantee that every strategy is respected by a final segment of the enumeration of A, B and C.

4.29. Definition.

$$\mathcal{C} = \langle \mathcal{C}[s] \mid s \in \mathcal{M}\rangle$$
$$\mathcal{C}[s] = \langle A[s], B[s], C[s], \Sigma[s]\rangle$$
$$\Sigma[s] = \left\langle \begin{array}{l} D\left(\Theta_{\mu(i,s)}, q_i, w_{i,s}, t_{i,s}, F_{i,s}\right), \\ C(A \to B, \mu(i,s)), C(B \to C, \mu(i,s)) \end{array} \middle| \, 1 \leq i \leq l(s) \right\rangle$$

The q_i are given by Definition 4.20, $\mu(i,s)$ by Proposition 3.1 and $l(s)$ by Definition 4.27. The rest we define inductively.

$$\begin{aligned} &A[1] = B[1] = C[1] = \emptyset \\ &w_{i,1} = i \\ &t_{i,1} = 0 \\ &F_{i,1} = \{w_{j,1} \mid j < i\} \end{aligned}$$

This guarantees the enumeration of length 1 is an execution of the strategies $\Sigma[1]$. Assuming $\mathcal{C}[s]$ gives an enumeration of length s which is an execution of $\Sigma[s]$, we choose $\mathcal{C}[s+1]$ using Lemma 4.25:

Choose $k(s)$ to be least such that $D\left(\Theta_{\mu(k(s),s)}\right)$ is required to change state at stage s, if there is such a strategy, and $k(s)$ equal to $l(s)$ otherwise. Use the recursive method of Lemma 4.25 to define $\mathcal{C}[s+1]$ such that for $i \leq k$, $w_{i,s+1} = w_{i,s}$, $t_{i,s+1} = t_{i,s}$, $F_{i,s+1} = F_{i,s}$, (and, as $k \leq l(s)$, $D(\Theta_{\mu(i,s+1)}) = D(\Theta_{\mu(i,s)})$) and such that the enumeration of A, B and C through stage s is an execution of $\Sigma[s+1]$. (For i greater than k, this method determines $w_{i,s+1}$ and $F_{i,s+1}$, and sets $t_{i,s+1}$ equal to s.)

4.30. Lemma. *The sets A, B and C enumerated by the construction $\mathcal{C}$ satisfy the following conditions.*

$$A \leq_w B \leq_w C$$
$$A \not\leq_w C$$

PROOF: First, by $I\Sigma_1$ in $\mathcal{M}$, for every s in $\mathcal{M}$, $\mathcal{C}[s]$ is defined, and the enumeration of A, B and C through stage s is an execution of $\Sigma[s]$.

By the conditions on the convergence of $\mu(s,i)$ to $\mu(i)$, for any given x in $\mathcal{M}$, there are an i in ω and a t in $\mathcal{M}$ such that

$$(\forall s > t)\,[l(s) > i \;\;\&\;\; \mu(i,s) = x]\,.$$

(In particular, $\mu(i) = x$.)

Given x in $\mathcal{M}$, let i and t be as above. Then for all $j \leq i$, there is a stage $t(j) \geq t$, such that for $s \geq t(j)$, $k(s) > j$.

This can be proven by induction on j: Suppose $t(j-1)$ is given. Then for $s \geq t(j-1)$, $k(s) \geq j$, so by Definition 4.28,

$$D(\Theta_{\mu(j,s)}, q_j, w_{j,s}, t_{j,s}, F_{j,s}) = D(\Theta_{\mu(j,s+1)}, q_j, w_{j,s+1}, t_{j,s+1}, F_{j,s+1}).$$

By $I\Sigma_1$ (induction on s) this strategy and its associated parameters are unchanged for all $s \geq t(j-1)$. We call this limiting value $D(\Theta_{\mu(j)}, q_j, w_j, t_j, F_j)$. Again by Definition 4.29, for for s greater than or equal to $t(j-1)$, we can have $k(s) = j$ only if this strategy is required to change state at stage s, which can happen only once. If this happens at some stage s greater than or equal to $t(j-1)$, set $t(j)$ equal to $s+1$; if not, $t(j)$ is equal to $t(j-1)$.

This induction on j less than i is valid because i is a standard integer.

Now we have shown that for any x, there are i in ω, and $t(i)$ in $\mathcal{M}$, such that $\mu(i) = x$, and for s greater than or equal to $t(i)$ and j less than or equal to i,

$$D(\Theta_{\mu(j,s)}, q_j, w_{j,s}, t_{j,s}, F_{j,s}) = D(\Theta_{\mu(j)}, q_j, w_j, t_j, F_j).$$

But then the full enumeration of A, B and C is an execution of the sequence of strategies

$$\left\langle D\left(\Theta_{\mu(i)}, q_i, w_i, t_i, F_i\right), C(A \rightarrow B, \mu(i)), C(B \rightarrow C, \mu(i)) \mid 1 \leq i \leq \omega \right\rangle.$$

Since each x in $\mathcal{M}$ occurs as one of the $\mu(i)$, by Lemma 4.24, our conditions are satisfied. $\diamondsuit$

Lemma 4.30 directly implies Proposition 4.1, so we have completed its proof. $\diamondsuit$

References

1. Driscoll, G. C., Jr., *Metarecursively enumerable sets and their metadegrees*, Jour. Sym. Log. **33** (1968), 389-411.
2. Enderton, H. B., "A Mathematical Introduction to Logic," Academic Press, New York, 1972.
3. Groszek, M. J. and Slaman, T. A., *Foundations of the priority method I: finite and infinite injury*, (to appear).
4. Hájek, P. and Kučera, A., *On recursion theory in* $I\Sigma_1$, (to appear).
5. Kirby, L. A. S. and Paris, J. B., Σ_n*-collection schemas in arithmetic*, in "Logic Colloquium '77," North Holland, Amsterdam, 1978, pp. 199-209.
6. Mytilinaios, M. E. and Slaman, T. A., Σ_2*-collection and the infinite injury priority method*, Jour. Sym. Log. (to appear).
7. Soare, R. I., "Recursively Enumerable Sets and Degrees," Perspectives in mathematical logic, Springer-Verlag, Berlin-Heidelberg, 1987.

Keywords. Turing reducibility, priority method, fragments of arithmetic.
1980 *Mathematics subject classifications*: Primary: 03D30.

Department of Mathematics and Computer Science, Dartmouth College, Hanover, NH 03755

Department of Mathematics, The University of Chicago, Chicago, IL 60637