Annals of Pure and Applied Logic 52 (1991) 155–179
North-Holland

The density of infima in the recursively enumerable degrees

Theodore A. Slaman*

Department of Mathematics, University of Chicago, 5734 University Avenue, Chicago, IL 60637, USA

Communicated by A. Nerode
Received December 1989
Revised July 1990

Abstract

Slaman, T.A., The density of infima in the recursively enumerable degrees, Annals of Pure and Applied Logic 52 (1991) 155–179.

We show that every nontrivial interval in the recursively enumerable degrees contains an incomparable pair which have an infimum in the recursively enumerable degrees.

Theorem. *If A and E are recursively enumerable sets such that $A >_{\mathrm{T}} E$, then there are recursively enumerable sets B, C and D such that*:

(1) $A >_{\mathrm{T}} B$, $A >_{\mathrm{T}} C$.
(2) $B >_{\mathrm{T}} D$, $C >_{\mathrm{T}} D$.
(3) $D \geqslant_{\mathrm{T}} E$.
(4) *For sets X, if X is recursive in B and in C, then X is recursive in D.*

Let A and E be given so as to satisfy the hypotheses of the theorem. It is safe to assume that E is not recursive since the theorem is known otherwise, by either [1, Fejer] or [3, Lachlan]. Further, notice that the Sacks density theorem [4] implies that we may obtain strict inequality in (3). Given A strictly above E, we use the density theorem to obtain E^* strictly between A and E and apply the above theorem to A and E^*.

We present the construction of B, C and D in two installments. First, we show how to build B, C and D satisfying (2)–(4). Then we describe a modified construction with the following special property. For all n and each of B, C and D either n is enumerated into the set or A is able to enumerate a state of the construction which precludes n's subsequent enumeration. Thus, B, C and D are recursive in A.

* The author's research was supported by NSF grant DMS-8601856 and Presidential Young Investigator Award DMS-8451748.

1. The upward density of pairs with infima

In this section, there are three sorts of requirements. Omitting indices, they are:

$\mathscr{C}$: The coding requirements. *$D \geqslant_{\mathrm{T}} E$, $B \geqslant_{\mathrm{T}} D$ and $C \geqslant_{\mathrm{T}} D$.*

These requirements have the highest priority of all. They are directly satisfied by coding E into D, D into B and D into C. We fix a recursive coding function c (such as $c : n \mapsto 2n$) and ensure that n is an element of E if and only if $c(n)$ is an element of D. Similarly, n is an element of D if and only if $c(n)$ is an element of B if and only if $c(n)$ is an element of C.

$\mathscr{R}$: The infimum requirements. *If $\Psi(B) = \Phi(C) = X$, then $D \geqslant_{\mathrm{T}} X$.*

We satisfy $\mathscr{R}$ by a positive strategy as in Fejer [1]. We describe the enumeration of a Turing functional by the recursive placement of a family of movable markers $\{\Gamma(n) \mid n \in \mathbb{N}\}$. Placing $\Gamma(n)$ on position p during stage s defines the value of $\Gamma(n, D)[s]$ to be the common value of $\Psi(n, B)[s]$ and $\Phi(n, C)[s]$. The computation of $\Gamma(n, D)[s]$ uses the oracle $D[s]$ on all numbers less than or equal to p. We ensure that the only $\Gamma(n, -)$ computation that applies to D is the last one enumerated, by putting a number less than or equal to the position of $\Gamma(n)$ into D each time $\Gamma(n)$ is moved. For each n, we ensure that $\Gamma(n, D) = X(n)$ by moving $\Gamma(n)$ whenever both $\Psi(n, B)$ and $\Phi(n, C)$ change to a new common value.

Precisely, the strategy R for $\mathscr{R}$ requires that the markers will be governed by the following rules. For each stage s, if $\Gamma(n)$ has a position during stage s, let $\gamma(n, s)$ denote that position.

(1) If n is less than s, $\Psi(B)[s] \upharpoonright n + 1 = \Phi(C)[s] \upharpoonright n + 1$ and $\Gamma(n)$ does not have a position then $\Gamma(n)$ must be assigned a position $\gamma(n, s)$. $\gamma(n, s)$ must be larger than any number previously mentioned in the construction. (*We ensure that the domain of $\Gamma(D)$ appears to be at least as large as the domain on which $\Psi(B)$ appears equal to $\Phi(C)$.*)

(2) If $\Gamma(n)$ has position $\gamma(n, s)$ assigned during an earlier stage t and the common value of $\Psi(n, B)[s]$ and $\Phi(n, C)[s]$ is different than it was during stage t, then $\Gamma(n)$ must be moved from $\gamma(n, s)$. Note, divergence is considered a new common value. (*We correct $\Gamma(D)$ when it appears to be wrong.*)

(3) If $\Gamma(n)$ is removed from its position $\gamma(n, s)$ during stage s then, for each $m > n$, $\Gamma(m)$ is also moved from its position. Further, some number less than or equal to $\gamma(n, s)$ must enter D. (*We ensure that the markers stay in order and keep a record of the movement of the markers in D.*)

(4) Except for finitely many exceptions, $\Gamma(n)$ cannot be moved from position $\gamma(n, s)$ unless at least one of $\Psi(B)$ or $\Phi(C)$ changes its computation at some m less than or equal to n after the stage when $\Gamma(n)$ was assigned position $\gamma(n, s)$. Note, if one of $\Psi(n, B)[s]$ or $\Phi(n, C)[s]$ diverges, then we treat the computation as having changed. (*We ensure that if* $\Psi(B)$ *and* $\Phi(C)$ *are equal and total then* $\Gamma(D)$ *is total.*)

By the parenthetical remarks, we have the following proposition.

1.1. Proposition. *If* B, C, D *and* Γ *are constructed so as to follow the rules* (1)–(4), *then* $\mathcal{R}$ *is satisfied.*

The above rules do not completely determine the motion of the Γ markers. They make it necessary to move marker $\Gamma(n)$ when $\Psi(n, B)$ and $\Phi(n, C)$ change their common value. They allow a marker to move if one of $\Psi(n, B)$ or $\Phi(n, C)$ changes its computation. In Rule (4), we are allowed to let each marker be moved indiscriminately by a finite collection of finite outcome strategies.

R's-recovery process

If a strategy is to respect R's rules, when it enumerates a number x into D and its code into B and C, it should also examine the status of $\Psi(B)$ and $\Phi(C)$ to check whether x's entering D has changed the computations associated with any of R's markers. If so, then we must enumerate the least such marker's position into D and codes into B and C to allow the computation of $\Gamma(D)$ to recover. Then, the process must be iterated, since the enumeration of any number may require the enumeration of a smaller number in order to recover. We define R's *recovery process* as follows.

> If there is an x such that $\Gamma(x)$ has position $\gamma(x, s)$ during stage s which was assigned earlier but neither $\Psi(x, B)$ nor $\Phi(x, C)$ has a current computation giving the same value as their common value when $\Gamma(x)$ was assigned its position, then enumerate the least such $\gamma(x, s)$ into D, enumerate $c(\gamma(x, s))$ into both B and C and cancel the positions of all $\Gamma(y)$ such that y is greater than or equal to x. Repeat the recovery process.
>
> Otherwise, end the recovery process.

One aspect of respecting R's rules is that R's recovery process must follow every enumeration of a number into B, C or D. The other aspect is that markers must eventually be assigned positions. This will be done by R explicitly.

Of course, we have only described the atomic strategy to satisfy $\mathcal{R}$. Given two requirements $\mathcal{R}_1$ and $\mathcal{R}_2$ where $\mathcal{R}_1$ has higher priority than $\mathcal{R}_2$, we will have several strategies for $\mathcal{R}_2$ depending upon the outcome of R_1. In the strategy R_2 based on the Π_3-outcome of R_1 (in which every $\Gamma_1(n)$ has a limit), we dynamically assign a finite set of R_1 markers to each R_2 marker and move the R_2 marker each time an R_1 marker from the assigned finite set is moved. For each R_2-marker, if R_1 has the Π_3-outcome then each element of its associated R_1-markers has a limit. In this case, the R_2 motion inherited from R_1 is finite. Backup strategies will be used to satisfy $\mathcal{R}_2$, in the event of a Σ_3-outcome in which there is an $\mathcal{R}_1$ marker that does not have a limit position.

$\mathcal{S}$: The inequalities. $\Theta(D) \neq B$; $\Theta(D) \neq C$.

These requirements are symmetric in B and C so it is sufficient to discuss just the first. The strategies S to satisfy $\mathcal{S}$ depend upon how many sets of markers are imposing their rules on S.

In the trivial case, when S does not have to contend with any markers at all, the requirements can be satisfied by the Sacks coding/preservation strategy used in his proof of the Density Theorem. The more complicated strategies are built upon this one recursively.

We retain the Sacks idea to 'code' enough of A into B and 'preserve' enough of D over E to ensure

$$(1.2) \qquad \Theta(D) = B \Rightarrow E \geqslant_{\mathrm{T}} B, \qquad \Theta(D) = B \Rightarrow B \geqslant_{\mathrm{T}} A.$$

The assumption that A is strictly above E makes the conjunction of the two consequences impossible. Thus, ensuring the two implications ensures that the requirement is satisfied. The interesting ingredient is E's simulation of the evaluation of Θ on D to compute B. To satisfy the first implication in (1.2), E is required to enumerate the $\Theta(D)$ computations that establish $\Theta(D) = B$. E's problem is that a number may enter D and change the evaluation of $\Theta(D)$ for one of three reasons: it may be the $\mathcal{C}$-code for an atomic fact about E, it may be one of a recursive set of numbers enumerated by a higher priority strategy with recursively described outcome or it may record the movement of some Γ-marker. The first two are recursive in E. S ensures that E can enumerate when the third case will not occur.

Consider the last point in more detail. S cannot keep higher priority strategies from moving markers positioned within $\Theta(D)$ computations. Instead, for each $\Theta(D)$ computation, S can try to establish a configuration within the construction which proves to E that no strategy of higher priority than S will move the markers involved in the computation. The failure of S to find such a configuration will be linked to a Σ_3-outcome of an $\mathcal{R}$-strategy of higher priority, and hence to progress on that Π_3-strategy.

We begin by discussing the simplest case. Suppose that the only constraint on S is that it must obey the rules associated with R's family of markers $\{\Gamma(k) \mid k \in \mathbb{N}\}$.

Making $\Theta(D)$ recursive in E

To reduce the question whether $\Theta(n, D)$ is correctly computed during stage s to E, S acts as follows. For each marker $\Gamma(k)$ such that $\gamma(k, s)$ is less than

$\vartheta(n, D)[s]$ (the use of the computation of $\Theta(n, D)$ during stage s), E will either see $\Gamma(k)$ move, S will ensure that $\Gamma(k)$ will not be moved by any strategy of higher priority than S or S will produce an n and ensure that R has the Σ_3-outcome for some $m \leqslant n$. In the last case, S will produce an n such that for some $m \leqslant n$ either $\Psi(m, B) \neq \Phi(m, C)$ or $\Phi(m, C)$ is not defined.

R is the only strategy of higher priority than S that could move a marker. Focusing on a single marker, R will move $\Gamma(k)$ if $\Psi(k, B)$ and $\Phi(k, C)$ change their common value. For the sake of making B compute A, A will be coded into B. This puts the $\Psi(k, B)$ computation more or less beyond E's understanding. But S can still try to control the evaluation of $\Phi(k, C)$. S will reduce E's problem of trying to recognize a correct D computation, that could be changed by E or R, to trying to recognize a correct C computation, that could only be changed by E. Of course, the same considerations appear; since D is coded into C, the movement of a Γ-marker causes a change in C. However, S can attempt to stabilize this phenomenon. This is based on the observation that a Γ-marker is only required to move to reflect an earlier change in C. Thus, R never acts to make the first change in C.

1.3. Definition. A stage s *C-configuration to preserve $D \restriction l$ respecting R* is an initial segment of $C[s]$ of length l' such that:

(1) $l' \geqslant l$.

(2) For all m, if $c(\gamma(m, s)) < l'$ then $\varphi(m, C)[s] < l'$ and $\Phi(m, C)[s] = \Gamma(m, D)[s]$. (Here $c(\gamma(m, s))$ is the code of the stage s position of $\Gamma(m)$; $\Gamma(m, D)[s]$ is the mutual value of $\Psi(m, B)[t]$ and $\Phi(m, C)[t]$ during the stage t when $\Gamma(m)$ was assigned position $\gamma(m, s)$.)

If $C[s] \restriction l'$ is a C-configuration to preserve $D \restriction l$, $C[s] \restriction l'$ is preserved with priority S and E does not change after stage s to cause a code to be enumerated into D and from there into C below l', then $C \restriction l'$ will equal $C[s] \restriction l'$. The reason is that a marker $\Gamma(m)$ with code below l' will only move if both $\Psi(B)$ and $\Phi(C)$ first change their values at some number less than or equal to m. This requires a change in C below l' to initiate Γ-marker movement by R's recovery process since the uses of all of the relevant $\Phi(m, C)$ computations are less than l'. Hence, the first number below l' to enter C after stage s does so to reflect some number's entering E below l'.

Thus, being a permanent C-configuration is recursive in E. S implements the Sacks preservation strategy by attempting to find a permanent C-configuration for each compoutation $\Theta(n, D)$ where $B \restriction n$ is equal to $\Theta(D) \restriction n$.

Let n be fixed. S acts as follows to ensure that if $\Theta(D) \restriction n$ is equal to $B \restriction n$, $\Theta(n, D)$ converges and $\Psi(B) = \Phi(C)$ then there is a permanent C-configuration preserving $D \restriction \vartheta(n, D)$.

(1) S waits for a stage s such that $\Theta(D)[s] \upharpoonright n$ is equal to $B[s] \upharpoonright n$ and $\Theta(n, D)[s]$ is defined.

(2) Given a stage as in (1), S *takes control* of a marker $\Gamma(m_n)$, that has not yet been assigned a position and hence will have position greater than $\sup(\{\vartheta(n', D)[s] \mid n' \leqslant n\})$, and keeps the code of its position clear of the C-computation $\varphi(m_n, C)$ and hence clear of a C-configuration preserving $D \upharpoonright \sup(\{\vartheta(n', D)[s] \mid n' \leqslant n\})$. S moves $\Gamma(m_n)$ every time the approximation to the computation of $\Phi(m_n, C)$ changes, enumerates its former position into D, enumerates the code of that position into B and C, and follows all by R's recovery process.

(3) If the computation of $\Theta(n, D)$ changes then S *releases control* of all $\Gamma(m)$ where m is greater than or equal to m_n. A new and larger value for m_n will be chosen the next time that S reaches (2).

Notice that this action of S does not violate the rules of R. Each time that S moves $\Gamma(m_n)$ is in response to a change in the computation of $\Phi(n, C)$. This change in $\Phi(C)$ grants R-permission for a movement of $\Gamma(m_n)$ according to Rule (4). Further, if $\Theta(D) \upharpoonright n = B \upharpoonright n$ and $\Theta(n, D)$ converges, then S will settle on a final value for m_n and maintain the inequality $\gamma(m_n, s) > \varphi(m_n, C)[s]$. Thus, S ensures that during each stage s, either S moves $\Gamma(m_n)$ or $C[s] \upharpoonright \gamma(m_n, s)$ is a C-configuration to preserve $D \upharpoonright \sup(\{\vartheta(n', D)[s] \mid n' \leqslant n\})$.

For the sake of strategies of lower priority than S, we will show that S imposes a finite amount of permanent control, or $\Psi(b)$ and $\Phi(C)$ are not equal, or one of $\Psi(B)$ or $\Phi(C)$ is not total. In each case, the permanent restraint due to S is finite.

The coding of A into B

We use a minor variation on the Sacks coding method to ensure the second implication in (1.2): $\Theta(D) = B \Rightarrow B \geqslant_{\mathrm{T}} A$.

In the Sacks coding strategy, the use function of a Turing functional which is provably partial determines the coding locations in a set under construction (such as B), for the atomic facts about a given set (such as A). We apply the same strategy using the length of C-configurations to determine the coding locations. The Sacks contradiction will prove that the function mapping n to the length of the nth C-configuration is also partial.

We code the information whether $n \in A$ at a location in B that is greater than the stage s when $\Theta(D)[s] \upharpoonright n = B[s] \upharpoonright n$ and there is a permanent C-configuration preserving $D \upharpoonright \sup(\{\vartheta(n', D)[s] \mid n' \leqslant n\})$. So, if there is an n such that $\Theta(n, D)$ is not equal to $B(n)$ or there is no permanent C-configuration preserving $D \upharpoonright \sup(\{\vartheta(n', D)[s] \mid n' \leqslant n\})$, then S codes a recursive set into B. The same feature appears in the Sacks density construction. The only difference between the two situations is that Sacks could directly preserve an initial segment of his set where we can only preserve an initial segment of D by preserving a larger initial segment of C.

The formal three requirement construction $\mathscr{C}$, $\mathscr{R}$, $\mathscr{S}$

Before describing the full construction, we describe the rudimentary version implementing the direct coding strategies together with one strategy for each of the requirements $\mathscr{R}$ and $\mathscr{S}$. The three strategy construction operates by recursion, as dictated by the stage by stage actions of the following strategies.

We adopt the following conventions. For each functional under consideration, if the computation of that functional changes between consecutive stages in execution of a strategy, then the strategy regards the functional as divergent during the second of these stages. Also, we regard the use functions of all Turing functionals as strictly increasing. That is to say that the evaluation of a functional at m involves first computing its values on all arguments less than m.

Coding: If n has been enumerated into E and $c(n)$ has not been enumerated into $D(n)$, then *Coding* enumerates $c(n)$ into D. Similarly, *Coding* enumerates $c(n)$ into both C and D if n has been enumerated into D. Further, *Coding* restrains $c(n)$ from entering any set except in response to n's entering a set being coded. Of course, we choose the recursive function c so that the complement of the range of c is an infinite recursive set.

R: During stage s, R first implements its recovery process to correct $\Gamma(D)$ from the effects of *Coding* according to Rules (2) and (3). Then R assigns positions according to Rule (1), following the convention that markers are assigned positions in increasing order of argument.

S: As described above, the stage s activity of S is divided into ensuring the two implications in (1.2). For the first of these, S implements its version of the Sacks preservation strategy. Formally, we let $n=0$, $m_{-1}=-1$, and $r(-1,s)=0$ and start the cycle for $n=0$.

The nth cycle works as follows. Fix an infinite recursive set P for S's exclusive use.

(1) If $\Theta(D)[s] \restriction n$ is equal to $B[s] \restriction n$ and $\Theta(n, D)[s]$ is defined, then go to (2). Otherwise, preserve $C \restriction r(n-1, s)$; for each i greater than or equal to n, cancel the value of m_i; end stage s activity of S with tentative outcome $\langle d, n, r(n-1, s)\rangle$. (*$d$ is an abbreviation for 'disagree'. If S stops at step* (1) *in the nth cycle infinitely often, then either $\Theta(n-1, D)$ converges with value unequal to $B(n-1)$ or the approximation to the computation of $\Theta(n, D)$ does not have a limit. Either of these implies that $\Theta(D)$ is not equal to B. The limit on s of $r(n-1, s)$ is the restraint on C needed to preserve the first n C-configurations.*)

(2) If m_n is defined then go to (3). Otherwise, let m_n be the least number greater than any number previously mentioned in the construction and not in the range of the direct coding function c. (*In particular, $\Gamma(m_n)$ has never been assigned a position.*)

(3) If $\Gamma(m_n)$ does not have a position or if the computation of $\Phi(C) \restriction m_n+1$ has changed between s and the stage when $\Gamma(m_n)$ was assigned its current

position, then proceed as follows. Enumerate $\gamma(m_n, s)$ into D and $c(\gamma(m_n, s))$ into B and C, if $\Gamma(n)$ has a position; for each y greater than or equal to m_n, cancel the position of $\Gamma(y)$; for each n' greater than n, cancel the values of $m_{n'}$ and $\delta(n')$ (the coding locations for the coding half of S's activity); preserve $C \restriction r(n-1, s)$; end stage s activity with the tentative outcome $\langle c, m_n, r(n-1, s)\rangle$. *(If this outcome is repeated infinitely often, S is never able to preserve a C-configuration to preserve $D \restriction \vartheta(n, D)$ because $\Phi(C)$ is not total.)*

Otherwise, define $r(n, s)$ to be $\gamma(m_n)[s]$. If $\delta(n)$ does not have a value, define $\delta(n, s)$ to be the least element of P that is larger than any number previously mentioned in the construction. Begin the $n+1$st cycle in step (1). *(Note,. $C[s] \restriction r(n, s)$ is a C-configuration to preserve D's computation of $\Theta(D) \restriction n+1$. By our convention, $\varphi(m_n, s)$ is greater than m_n, hence greater than $\sup\{\vartheta(m, D)[s] \mid m \leq n\}$. Further, since we also take φ to be monotone, for any marker positioned less than $\varphi(m_n, C)[s]$ during stage s, that marker has argument less than m_n. Hence, since we have assumed that φ is monotone, there is a C-computation of length less than $\varphi(m_n, C)[s]$ that keeps R from moving it. This is the reason that S takes steps to ensure $\gamma(m_n, s) > \varphi(m_n, C)[s]$.)*

During stage s, S's coding mechanism acts as follows.

If $\delta(k, s)$ is defined, let t be the stage when $\delta(k)$ was assigned its current value $\delta(k, s)$. If k is an element of $A[s] - A[t]$ but $B[s] \restriction \delta(k, s)+1 = B[t] \restriction \delta(k, s)+1$, then enumerate $\delta(k, s)$ into B, cancel the values of δ on all numbers greater than or equal to n, and invoke R's recovery process.

Suppose that B, C and D are constructed while respecting the above strategies. Suppose that $\Theta(D)$ is equal to B and for every n, S finds a permanent C-configuration preserving $D \restriction \vartheta(n, D)$. Then, we can conclude that A is recursive in E. First note that E can enumerate the collection of permanent C-configurations found by S to preserve $\Theta(D) = B$. This implies that B is recursive in E. Now, to compute whether n is an element of A, we use E to compute S's first n permanent C-configurations and the stage s when these are first preserved by S. The value of $\delta(n, s)$ can only be reassigned in response to a change in a C-configuration associated with the computation of $\Theta(m, D)$ for some m less than or equal to n. By the choice of s, there are no such changes. By the action taken in the coding strategy in (2), n is an element of A if and only if n has been enumerated into A by the stage s_1 when $B[S_1] \restriction \delta(n, s)+1$ is equal to $B \restriction \delta(n, s)+1$. Since B is recursive in E, we can uniformly compute s_1 and read off the value of A at n.

For our theorem, we may assume that A is not recursive in E. Thus, either $\Theta(D)$ is not equal to B or there is an n such that $\Theta(D) \restriction n = B \restriction n$ and $\Theta(n, D)$ converges but S is not able to preserve a permanent C-configuration preserving $D \restriction \sup(\{\vartheta(m, D) \mid m \leq n\})$. In either case, there are only finitely many C-configurations permanently preserved by S. At worst, there is an n such that

$\delta(n, s)$ is reassigned infinitely often and $\lim_{s\to\infty} \delta(n, s) = \infty$. This implies that the set of coding locations enumerated into B by S is recursive.

Outcomes in the three strategy construction

In the first type of outcome $\langle d, n, r\rangle$, r is the limit of the lengths of the C-configurations which preserve the computation of $\Theta(D)$ on arguments less than n and for infinitely many s, $\Theta(D)[s] \upharpoonright n+1 \neq B[s] \upharpoonright n+1$. In this case, $\mathcal{S}$ is satisfied since $\Theta(D)$ is not equal to B. S imposes a permanent restraint on C of size r. S has permanent control only over the finite set of markers $\{\Gamma(m_i) \mid i<n\}$ and each of these has a limiting position. Thus, the behavior of the R markers is permanently modified only on a finite set.

The second type of infinitary outcome is $\langle c, m_n, r\rangle$. Here, there are permanent computations establishing equality between $\Theta(D)$ and B below n, a permanent computation for $\Theta(n, D)$ and permanent C-configurations to preserve the computation of $\Theta(D)$ on arguments less than n but S is unable to preserve a C-configuration to preserve the computation of $\Theta(n, D)$. Then, it must be the case that either S encounters infinitely many stages during which $\Gamma(m_n)$ does not have a position or infinitely many stages during which the computation of $\Phi(C) \upharpoonright m_n+1$ has changed, in which case S moves $\Gamma(m_n)$ infinitely often. Given that the construction respects R's rules, either of these conditions implies that $\Phi(C)$ is not total. Thus, $\mathcal{R}$ is satisfied. Let m be the least number less than or equal to m_n such that its Γ marker has no limit position. Either there is a stage in the construction after which $\Gamma(m)$ is never assigned a position or $\Gamma(m)$ is moved infinitely often, temporarily occupying each of an increasing recursive sequence of positions. Thus, the effect of R is actually recursive, described by the finite motion of the markers that have a limiting position and the recursive motion of those above m. This opens the way for later strategies to ignore the complexity of $\mathcal{R}$. S does not ensure the satisfaction of $\mathcal{S}$, but provides the witness to the fact that R has a Σ_3-outcome.

If we were to continue with the three requirement construction, the most natural next strategy to follow this outcome would be the pure Sacks strategy, reducing D to E (plus the recursive input of R) and coding A into B. This strategy would be guaranteed to satisfy $\mathcal{S}$.

Depth-n strategies

The injury of a strategy by the action of one of higher priority was seen in a primitive form in the way that a Σ_3-strategy S could fail to satisfy $\mathcal{S}$, by showing that $\mathcal{R}$ was satisfied in a trivial way and reducing R to a recursively describable Σ_3-outcome. Injury between Π_3-strategies appears when we introduce more than one R strategy. Of particular interest is the strategy S designed to work on $\mathcal{S}$ while respecting $R_1, \ldots, R_k$.

Let R_i be the ith strategy that S must respect; R_i uses markers $\Gamma_i(n)$ to ensure

$$\Psi_i(B) = \Phi_i(C) = X_i \Rightarrow D \geqslant_T X_i.$$

Inductively, R_i is based on the assumption that for all j less than i and for each n, $\Gamma_j(n)$ has a limiting position.

We define the $R_1, \ldots, R_k$ *recovery process* in the natural way. We iterate all the individual recovery processes until each of them terminates without enumerating any numbers into D.

1.4. Definition. A *C-configuration for the preservation of* $D \restriction l$ *respecting* $R_1, \ldots, R_k$ is an initial segment of $C[s]$ of length l' together with placement of markers for Γ_i, $i \leqslant k$, such that:

(1) $l' \geqslant l$.

(2) For all $i \leqslant k$ and for all m, if $c(\gamma_i(m, s))$ is less than l', then $\varphi_i(m, C)[s] < l'$. (*Here,* $\gamma_i(m, s)$ *is the position of* $\Gamma_i(m)$ *during stage* s.)

As in its atomic version, S threatens to make the computation of $\Theta(D)$ recursive in E by finding a permanent C-configuration of length $\gamma(m_n, C)$ for each permanent computation establishing $\Theta(D) \restriction n+1 = B \restriction n+1$. Earlier, S took control of a marker $\Gamma(m_n)$ and ensured that if $\Phi(C)$ were total, $\Theta(D) \restriction n$ were equal to $B \restriction n$ and $\Theta(n, D)$ were convergent, then there would have been a permanent C-configuration to preserve this convergence. S kept $\Gamma(m_n)$ clear of any interaction with $D \restriction \vartheta(n, D)$ by moving it to stay clear of the computation of $\Phi(m_n, C)$. When there are at least two sets of markers, Γ_1 and Γ_2 written in order of priority, our problem becomes a little more subtle due to their interaction. $\Gamma_1(m)$ cannot be moved when the computation for $\Phi_2(m, C)$ changes. This is because R_1 does not depend on the outcome of R_2. In short, we cannot move all of the markers whenever a computation for one of the Φ_i's changes.

Instead, S keeps a sequence of markers $\Gamma_i(m_{n,i})$ clear of the configuration. But the elements of the sequence change depending on the apparent convergence or divergence of the relevant $\Phi_j(m_{n,j}, C)$. Whenever $\Gamma_i(m_{n,i})$ is without position or the evaluation of $\Phi_i(m_{n,i}, C)$ changes, S acts as follows. For all j less than i, S releases control of $\Gamma_j(m_{n,j})$ and cancels the values of $m_{n,j}$. Larger values are assigned for these parameters when a new computation for $\Phi(m_{n,i}, C)$ is found. For all j greater than or equal to i, S moves $\Gamma_j(m_{n,j})$ from its current position, if any, and enumerates the least such position into D, enumerates its code in B and C and invokes the $R_1, \ldots, R_k$ recovery process. Thus, all of these markers will go to infinity if $\Phi_i(m_{n,i}, C)$ diverges. This ordering of events is consistent with the heuristic principle that the strategies of higher priority than R_i are not effected by the outcome of R_i but all of the lower priority R_j are canceled when R_i is seen to have a Σ_3-outcome.

The preservation of C by S

We implement the preservation strategy as follows. Let P be an infinite recursive set reserved for the sole use of S. First, let $n = 0$ and let $r(-1, s) = 0$.

Proceed through the construction by recursion on cycles, starting with the cycle for 0. The nth cycle operating during stage s of the construction consists of the following steps.

(1) If $\Theta(D)[s] \upharpoonright n = B[s] \upharpoonright n$ and $\Theta(n, D)[s]$ is defined, go to (2). Otherwise, for all i less than or equal to k and for all n' greater than or equal to n, cancel the values of $m_{n',i}$ and $\delta(n')$; preserve $C \upharpoonright r(n-1, s)$. End the stage s activity of S with tentative outcome $\langle d, n, r\rangle$.

(2) If for each j less than or equal to k, $m_{n,j}$ is defined, then go to (3). Otherwise, by recursion on j assign a value to each $m_{n,j}$ that is not defined. In the recursion step, let i be the minimal j such that $m_{n,j}$ is not defined. Define $m_{n,i}$ to be the least element of P that is greater than any number previously mentioned in the construction. In particular, when $m_{n,i}$ is defined $\Gamma(m_{n,i})$ has never been assigned a position.

(3) Proceed by recursion on i, starting from k and working down. If $\Gamma_i(m_{n,i})$ does not have a position, if the computation of $\Phi_i(C) \upharpoonright m_{n,i}+1$ has changed between stage s and the stage when $\Gamma_i(m_n)$ was assigned its current position or if there is a j greater than i such that $\gamma_j(m_{n,j}, s) < \varphi(m_{n,i}, C)[s]$, then proceed as follows. Enumerate the least element of $\{\gamma(m_{n,j}, s) \mid j \geq i\}$ into D and its code into B and into C, if any such $\Gamma_j(m_{n,j})$ has a position, followed by $R_1, \ldots, R_k$'s recovery process; for each j greater than or equal to i and each y greater than or equal to $m_{n,i}$, cancel the position of $\Gamma_j(y)$; for each j less than i, cancel the value of $m_{n,j}$; for each n' greater than n and j less than or equal to k, cancel the values of $m_{n',j}$; for each n' greater than or equal to n, cancel the value of $\delta(n')$; preserve $C \upharpoonright r(n-1, s)$; end stage s activity with the tentative outcome $\langle c, m_{n,i}, i, r(n-1, s)\rangle$. *(If this outcome is repeated infinitely often, S can trace its inability to preserve a C-configuration to preserve $D \upharpoonright \vartheta(n, D)$ to $\Phi_i(C)$'s not being total.)*

None of the above cases hold, then define $r(n, s)$ by

$$r(n, s) = \inf(\{\gamma_i(m_{n,i}, s) \mid i \leq k\}).$$

If $\delta(n)$ does not have a value, let $\delta(n, s)$ be the least element of P that is larger than any number previously mentioned in the construction. Begin the $n+1$st cycle in step (1). *(Note, as before, $C[s] \upharpoonright r(n, s)$ is a C-configuration to preserve D's computation of $\Theta(D) \upharpoonright n+1$ respecting $R_1, \ldots, R_k$. The only complication is that we have to ensure that all of the markers $\Gamma_i(m_{n,i})$ are clear of the supremum of the computations $\varphi_j(m_{n,j}, C)[s]$.)*

The coding of A into B by S

The coding of A into B by S is the same as in the atomic strategy. The only difference is that the coding location for the nth atomic fact about A is greater than the nth k-fold C-configuration to preserve D up to $\sup(\{\vartheta(n', D)[s] \mid n' \leq n\})$.

1.5. Proposition. *In the context of a construction in which S acts infinitely often, suppose that*

(1) S can restrain all numbers from entering B, C or D except the complete enumeration into B, C and D of fixed sets recursive in E and the coding of the movement of markers associated with $R_1, \ldots, R_k$;

(2) S can restrain any strategy from moving a marker except for the $R_1, \ldots, R_k$ recovery process.

Then, during any stage S's activity is finite, S works with numbers greater than the stage when S was first implemented and one of (I) *or* (II) *holds.*

(I) There is an n such that S finds C configurations preserving $D \restriction \sup(\{\vartheta(n', D)[s] \mid n' < n\})$ and either $\Theta(n-1, D)$ converges and is not equal to $B(n-1)$ or the computation of $\Theta(n, D)$ changes infinitely often. Let s_0 be the stage after which the first n many C-configurations have stabilized and let r be the supremum of their lengths. There will be infinitely many stages s after s_0 when S is active, during which either $\Theta(n-1, D)$ appears unequal to $B(n-1)$ or $\Theta(n, D)$ appears to diverge (using convention that a change in computation between S active stages is viewed as divergence). During any such stage, S has tentative outcome $\langle d, n, r\rangle$ and the only constraint due to S is to restrain numbers less than or equal to r from being enumerated. This restraint r is also imposed by S during every stage after s_0. The only markers over which S maintains permanent control are the $\Gamma_j(m_{n',j})$ with $n' < n$. During the stages when S has tentative outcome $\langle d, n, r\rangle$, the only markers with positions that will ever later be controlled by S are those that S permanently controls.

(II) There is an n such that $\Theta(D) \restriction n = B \restriction n$, $\Theta(n, D)$ has a permanent computation and S finds permanent C-configurations preserving $D \restriction \sup(\{\vartheta(n', D)[s] \mid n' < n\})$ but no C-configuration preserving $D \restriction \sup(\{\vartheta(n', D)[s] \mid n' \leqslant n\})$. Then, there is an i less than or equal to k with the following properties. Let s_0 be the stage after which the first n many C-configurations have stabilized and the computation of $\Theta(n, D)$ has reached its limit. For all j greater than or equal to i, $m_{n,j}$ has a limit value. Let s_1 be a stage after s_0 when these are achieved. For all $j > i$, there are permanent computations of $\Phi_j(C) \restriction m_{n,j}+1$ but there is no permanent computation of $\Phi_i(C) \restriction m_{n,i}+1$. During the stages s after s_1, when $\Phi_i(C) \restriction m_{n,i}$ does not converge by means of a computation with $\varphi_i(m_{n,i}, C)[s] < \inf(\{\gamma_j(m_{n,j}, s) \mid j \geqslant i\})$, S has tentative outcome $\langle c, m_{n,i}, i, r\rangle$, where r is equal to $r = \inf(\{\gamma_j(m_{n-1,j}) \mid j \leqslant k\})$; for each j less than i, S releases all control over markers $\Gamma_j(m)$ with m greater than $m_{n-1,j}$; for each j greater than or equal to i, if $\Gamma_j(m_{n,j})$ has a position during stage s, then S moves $\Gamma_j(m_{n,j})$; the only negative constraint due to S on strategies of lower priority than S is to restrain numbers less than or equal to r from being enumerated. The prohibition against enumerating numbers less than r is also in effect during every stage after s_1.

In either case, S's coding apparatus enumerates a recursive set into B.

Proof. Assume (1) and (2). Let W be the E-recursive set which is enumerated into C, not subject to S restraint and not due the strategies $R_1, \ldots, R_k$. Since every marker appearing in a C-configuration is kept from being moved by its associated R-strategy by a computation of length less than or equal to the configuration, the only way that a C-configuration respecting $R_1, \ldots, R_k$ which is preserved by S can change is if a number enters W. Since W is recursive in E, the set of permanent C-configurations is recursive in E. If for each n, the computation of $\Theta(n, D)$ were preserved by a permanent C-configuration then the evaluation of $\Theta(D)$ could be recursively simulated by E. If also $\Theta(D)$ were equal to B then B would be recursive in E. But if these would both occur then the coding of A into B would converge on every argument and so A would be recursive in B. This would contradict the assumption that E does not compute A.

Thus, S is ensured of having a Σ_3-outcome. There is an n such that for all n' less than n the equality between $\Theta(n', D)$ and $B(n')$ is preserved by a permanent C-configuration and either $\Theta(D) \upharpoonright n+1 \neq B \upharpoonright n+1$ is seen during infinitely many stages of the construction or the computation of $\Theta(D) \upharpoonright n+1$ is not permanently preserved by any C-configuration found by S. These two possibilities correspond to case (I) with outcome $\langle d, n, r\rangle$ and case (II) with one of the outcomes $\langle c, m_{n,i}, i, r\rangle$, respectively.

$\langle d, n, r\rangle$. Let n be the least number such that either $\Theta(D) \upharpoonright n-1 = B \upharpoonright n-1$ and $\Theta(n-1, D)$ converges but is not equal to $B(n-1)$ or $\Theta(D) \upharpoonright n = B \upharpoonright n$ and $\Theta(n, D)$ diverges. Assume that permanent C-configurations are eventually found to preserve $\Theta(D) \upharpoonright n$. Let r be the infimum of $\{\gamma_j(m_{n-1,j}) \mid j \leqslant k\}$, as calculated after the stage t when the associated configurations have stabilized. For all $s \geqslant t$, $n' \leqslant n$ and $j \leqslant k$, $m_{n',j}$ is not canceled nor is $\Gamma(m_{n',j})$ moved during stage s.

Say that s is *S-correct* if during stage s S's tentative outcome is $\langle d, n, r\rangle$. By clause (1) in the specification of S, any stage after t during which $\Theta(n-1, D)$ appears to disagree with $B(n-1)$, in the first case, or the computation of $\Theta(n, D)$ appears to diverge, in the second case, will be S-correct. By convention, any change in the evaluation of $\Theta(n, D)$ causes a momentary divergence. Thus, there will be infinitely many S-correct stages.

During any S-correct stage s, all of the values of $m_{n',j}$ for $n' \geqslant n$ are canceled. Thus S will not move any marker with argument less than s after stage s. Secondly, S only imposes restraint on C to preserve the C-configurations to preserve the computations of $\Theta(D)$ up to an including the first disagreement with B. Thus, during an S-correct stage, S will only preserve C on the numbers less than or equal to r. Lastly, any (temporary) C-configuration that appears after stage s will have length greater than s due to the position of markers $\Gamma(m_{n,j})$ where $m_{n,j}$ is greater than s. Thus, any coding by S after stage s will either involve the finite coding of $A \upharpoonright n$ or will contribute only numbers larger than s to D. These remarks verify that the action of S satisfies (I) and contributes a recursive set to B, if S has outcome $\langle d, n, r\rangle$.

$\langle c, m_{n,i}, i, r\rangle$. Let n be least such that $\Theta(D) \upharpoonright n = B \upharpoonright n$ and $\Theta(n, D)$ is defined but there is no permanent C-configuration to preserve $D \upharpoonright \vartheta(n, D)$. Let i be minimal such that $m_{n,i}$ has a limiting value but $\Phi_i(m_{n,i}, C)$ has no final computation. Such an i will exist since $m_{n,k}$ could not be canceled without a change in $D \upharpoonright \vartheta(n, D)$ or a change in the C-configuration associated with a smaller argument. This contribution to cancellation is bounded, showing that $m_{n,k}$ has a limiting value. Now show by induction that if $m_{n,i}$ has a limiting value m and $\Phi_i(m, C)$ converges then $m_{n,i-1}$ has a limiting value. Finally, if all of the $\Phi_i(m_{n,i}, C)$ converge, then S finds and preserves a permanent C-configuration contrary to the assumption of this case.

Say that a stage s is *S-correct* in this outcome if s is large enough so that $m_{n,i}$ has achieved its final value and $\Phi(m_{n,i}, C)[s]$ is not defined or there is a j such that j is greater than or equal to i and $\gamma_j(m_{n,j}, s)$ is less than or equal to $\varphi_i(m_{n,i}, C)[s]$. Again there will be infinitely many S-correct stages during which S has tentative outcome $\langle c, m_{n,i}, i, r\rangle$. During each such stage, step (3) will be executed for $m_{n,i}$. Then, the only restraint imposed by S is r, defined as above. r has a limiting value by the choice of n and i.

Every time the evaluation of $\Phi_i(m_{n,i}, C)$ changes, the locations for S to code A into B are increased on all numbers greater than or equal to n. As above, S's coding of A into B contributes a recursive set to B.

Finally, during each S-correct stage s, S cancels the values of $m_{n,j}$ for all j less than i and cancels the values of $m_{n',j}$ for all $n' > n$ and all $j \leqslant k$. Any subsequent assignment of values to these parameters will involve only numbers greater than s. Thus, for each R_j with $j < i$, S will not move any of the first s markers associated with R_j after stage s. For $j \geqslant i$, S moves all markers $\Gamma_j(m)$ where m is greater than or equal to $m_{n,j}$.

Finally consider the stage by stage construction. Assuming that the construction is finite up to the point when S is invoked, there are only finitely many markers $\Gamma_j(m)$ that have been assigned positions. If the activity of S is not curtailed for any other reason, then S will be eventually define a value for $m_{n,j}$ so large that $\Gamma_j(m_{n,j})$ does not have a position. Then S will end activity in step (3ii). Thus, S's action during any stage is finite. □

Examining cases (I) and (II) gives us:

1.6. Corollary. *Working in the same context as in Proposition* 1.5, *S ensures that either $\mathcal{S}$ is satisfied or that some higher priority $\mathcal{R}_i$ is satisfied.*

The above analysis verifies the inductive properties needed to combine strategies to construct sets satisfying all of the requirements. It ensures that given an appropriate environment each strategy is guaranteed to make some progress toward satisfying one of the requirements of its or higher priority. Further, the effect of each strategy is to produce an appropriate environment.

2. Producing sets recursive in A

In the construction outlined in the previous section, there are three possible ways for a number to enter one of B, C or D.

(1) A number is enumerated into E and this fact is reflected by the $\mathscr{C}$-codings of E into D and of D into both B and C.

(2) A number enters A and this is coded into one of B or C for the sake of some diagonal strategy S.

(3) A marker $\Gamma(m)$ is moved and its former position p is enumerated into D; then the code for p enters both B and C.

The first two of these are the possibilities that appear in the density theorem. A can compute with little difficulty whether a strategy will want to enumerate a number for one of these two reasons. (3) is a new case.

Let us consider R in isolation. Suppose that R is the only strategy and $\Gamma(n)$ has position p. R will move $\Gamma(n)$ if and only if both $\Psi(n, B)$ and $\Phi(n, C)$ are seen to have simultaneously changed their values at n. Note, the evaluation of $\Psi(n, B)$ could change while ending with the same answer. If during the period of $\Psi(B)$ change at n, $\Phi(n, C)$ did not change, then R would not be triggered to move $\Gamma(n)$. There is even the possibility that both $\Psi(n, B)$ and $\Phi(n, C)$ could be undefined in the limit without $\Gamma(n)$'s ever moving. There is no way that A could recursively predict this infinitary behavior in the construction. Luckily, the ability of the construction to exhibit this behavior is in no way related to the satisfaction of its requirements.

The density requirement for a marker $\Gamma(n)$ and position p is that either $\Gamma(n)$ moves past p without occupying it, $\Gamma(n)$ occupies position p and is later removed from that position or A can enumerate a permanent configuration for a computation of either $\Psi(n, B)$ or $\Phi(n, C)$ which keeps R from moving $\Gamma(n)$. The first two are recursively enumerable possibilities; the third will make essential reference to A and *use of a density strategy* M_p invoked during the construction.

We also make a modest modification in R so that if both $\Psi(n, B)$ and $\varphi(n, C)$ diverge in the limit, then R will eventually be triggered to move $\Gamma(n)$ from position p. M_p ensures that either R moves $\Gamma(n)$ or there is a permanent B/C-configuration (to be defined) which keeps $\Gamma(n)$ from moving. As in S, we keep a collection of markers clear and impose restraint so that if a permanent computation for either of $\Psi(n, B)$ or $\Phi(n, C)$ exists then A will recognize it. The only difference between what we do here and what we did with S is that the density strategy can use B or C computations.

In the final analysis, the strategy to enable A to enumerate the fact that $\Gamma(n)$ will never move again must result in a finite A-computation. Therefore, it itself must have a finite outcome. Given that all of its activity will be finitely bounded, an M-strategy can be allowed to move any marker at will, by Rule (4).

The modification of R

The modification of R uses the fact that E is not recursive to ensure that if $\Psi(n, B)$ diverges and $\Phi(n, C)$ diverges, then R will eventually want to move $\Gamma(n)$. We define max-min(n, s) to be the maximum of lengths of the shortest computations that have kept R from moving $\Gamma(n)$ during the stages between t, when $\Gamma(n)$ was assigned its position and stage s. As a function of s, max-min(n, s) is non-decreasing and goes to infinity if and only if both $\Psi(n, B)$ and $\Phi(n, C)$ diverge. We modify r by adding another rule.

(5) If $\Gamma(n)$ has a position during stage s, max-min(n, s) is defined and E has changed below max-min(n, s) between stage s and the stage when max-min(n, s) first achieved its current value, then $\Gamma(n)$ must be removed from its stage s position.

If $\Psi(B)$ and $\Phi(C)$ are defined and equal on all $n \in \mathbb{N}$, then for all n, max-min(n, s) has a finite limit. Thus, (5) can only introduce a finite number of movements for any marker. This implies that the modified version of R still ensures the satisfaction of $\mathcal{R}$. Secondly, as in a simple permitting argument, if $\lim_{s\to\infty}$ max-min$(n, s) = \infty$, then there must be a number x that enters E during a stage s with max-min$(n, s) > x$. If not, we could compute whether x is an element of E by checking whether x is an element of $E[s]$ for the least s such that max-min$(n, s) > x$. Thus, if $\Phi(n, B)$ and $\Psi(n, C)$ diverge, then (5) will eventually ensure that $\Gamma(n)$ is moved by R.

The modification of S

Of course, introducing (5) to R makes it more difficult for S to preserve D. We must ensure that the notion of configuration includes assurance against an invocation of (5). We modify the definition of B or C-configuration as follows.

2.1. Definition. (1) Define max-min(i, n, s) to be the value of max-min(n, s) associated with R_i's marker $\Gamma_i(n)$.

(2) A *stage s C-configuration to preserve $D \restriction l$ respecting $R_1, \ldots, R_k$* is an initial segment of $C[s]$ of length l' such that during stage s:

(i) $l' \geq l$.

(ii) For all $i \leq k$, for all n, if $\gamma_i(n, s) < l'$ then $\varphi_i(n, C)[s] < l'$ and max-min$(i, n, s) < l'$.

(3) A stage s B-configuration is defined similarly.

As before, E can be recursively enumerate the permanent configurations. We leave it to the reader to observe that this modification does not change the validity of Propositon 1.5, which describes the possible outcomes of an S-strategy.

The density strategies

We consider the reasonably general case when there is only one strategy R of higher priority. The job of M_l is to allow A to recognize the stage when all of B, C and D have settled down on the numbers less than or equal to l. In particular, if $\Gamma(n)$ reaches a permanent position below l, then A must be able to recursively enumerate this condition. M_l operates similarly to an S-strategy in that it attempts to preserve a configuration showing that R will not initiate any change below l. Then the only remaining source of change below l would have to come from the coding strategies, whose effects are uniformly recursive in A. M_l has an advantage over S in that M_l does not need to enumerate any numbers into B or C for the sake of making these sets complicated. Thus M_l can preserve both B and C computations. Further, the fact that $\Gamma(n)$ has not been moved by R implies that there is a computation relative to one of these sets to be preserved.

2.2. Definition. A *stage s B/C-configuration for l respecting R* is an initial segment of $B[s]$ and $C[s]$ of length l' such that l' is greater than or equal to l and for all n, if $c(\gamma(n, s)) < l'$, then max-min(n, s) is less than l'.

If A and E do not introduce a change after stage s in one of B, C or D below l' through a coding strategy, then $B \upharpoonright l' = B[s] \upharpoonright l'$, $C \upharpoonright l' = C[s] \upharpoonright l'$, and $D \upharpoonright l' = D[s] \upharpoonright l'$.

M_l acts to ensure that only a fixed finite set of markers become involved in the minimal B/C-configuration for l. Its action is both positive and negative. M_l fixes a number $move_l$ in advance. If the shortest B/C-configuration for l changes or involves the position of $\Gamma(move_l)$, M_l moves all of the markers $\Gamma(y)$ for y greater than or equal to $move_l$ and records this movement by enumerating a number into D. Otherwise, M_l preserves the shortest available B/C-configuration. Note, there will be a B/C-configuration since there are only finitely many markers with positions during any stage and any marker without an associated B or C computation can be removed from its position, see R's Rule (2).

By modifying R, we have ensured that each Γ-marker is either without position for all but finitely many stages, moved infinitely often or assigned a permanent position with a permanent B or C computation keeping it from being moved. Consider the first $move_l$ markers. Suppose that some initial subset of these reaches its limit position with associated final computations during stage s. Let *max-conv* be the maximum value less than $move_l$ for which Γ has a limit. The markers for arguments greater than *max-conv* will have moved beyond the computations for the smaller markers by stage s. Further, $\Gamma(move_l)$ is clear of these computations by the action of M_l. Thus, there is a permanent configuration for l established during stage s. After stage s, the only effect of M_l is to preserve this configuration. Thus, M_l is a Σ_2-strategy, i.e. finite action.

This simple strategy implements the general plan to ensure that B, C and D are fixed below l as soon as no strategy of higher priority than M_l initiates any change below l'. During the course of the construction, M_l can recognize a configuration in which any change would have to be initiated by a change in A. M_l preserves the shortest such configuration.

The general strategy M_l^σ

In general, there are finitely many requirements of higher priority than $\mathcal{M}_l$. For each sequence σ of strategies and their outcomes associated with these higher priority requirements, there is a strategy M_l^σ associated with $\mathcal{M}_l$.

A sequence σ will be a finite string of nodes of the form $\langle R_j^\beta\rangle$, $\langle S_j^\beta, c, m_{n,i}^\beta, i, r^\beta\rangle$, $\langle S_j^\beta, d, n^\beta, r^\beta\rangle$ or $\langle M_j^\beta, r^\beta\rangle$. The value of r^β is the restraint imposed by the strategy S_j^β or M_j^β. The other parameters describe the possible outcomes of their associated strategies. β represents the initial segment of σ up to and not including the strategy indexed by β.

2.3. Definition. A *stage s B/C-configuration for l respecting* σ is an l' greater than or equal to l such that l' passes the following test.

The test is defined by induction on m along σ. Assume that l' has not failed the test before m nor has it passed the test.

(1) $\sigma(m) = \langle R_i^\beta\rangle$. The markers associated with R_i^β are denoted by Γ_i^β.
   (a) If for all x such that $\Gamma_i^\beta(x)$ is located at position $\gamma_i^\beta(x, s)$ such that $c(\gamma_i^\beta(x, s))$ is less than l', max-min(i, x, s) is defined and is also less than l', then l' passes the test at step m.
   (b) Otherwise, l' fails the test.

(2) $\sigma(m)$ is one of the other possibilities.
   (a) If l' is less than r^β, then l' fails the test.
   (b) If l' is equal to r^β, then l' passes the entire test.
   (c) Otherwise, l' passes the test at step m.

Say that l' passes the test if either there is a step during which l' passes the entire test or if l' passes the test during all steps less than or equal to the length of σ.

The strategy M_l^σ

During a stage when all of the strategies on σ act with tentative outcomes as specified by σ, M_l^σ acts as follows. By recursion on l', M_l^σ applies the test. If M_l^σ finds l' that passes the test but there is a β_0 contained in σ such that $\gamma_i^{\beta_0}(move_l, s)$ is less than l', then M_l^σ enumerates the least position of such a marker into D, enumerates its code into B and C, moves all markers $\Gamma_i^\beta(n)$ associated with strategies appearing on σ with n greater than or equal to $move_l$ and applies the recovery process for the sequence of R-strategies that appear on σ. After recovery, M_l^σ starts over with the above recursion on l'.

Note, no markers are assigned positions during the action of M_l^σ so there are only finitely many ways by which M_l^σ might be required to initialize its recursion.

Ultimately, M_l^σ computes the least l' that passes the test for l and this l' is less than the positions of the markers $\Gamma^\beta(move_l)$ for all β contained in σ. Then, M_l^σ restrains all strategies of lower priority from enumerating any numbers less than l' into B, C or D.

2.4. Lemma. *Suppose that l' passes the test for M_l^σ during stage s, all of the strategies on σ act with tentative outcomes as specified by σ and*

(1) $A \upharpoonright l'+1 = A[s] \upharpoonright l'+1$ *and* $E \upharpoonright l'+1 = E[s] \upharpoonright l'+1$;

(2) *if n is less than the length of σ, then the only strategies that can be the first to enumerate a number less than $r^{\sigma \upharpoonright n}$ into B, C or D are the basic coding strategies and those mentioned before n in σ.*

Then $B \upharpoonright l' = B[s] \upharpoonright l'$, $C \upharpoonright l' = C[s] \upharpoonright l'$ *and* $D \upharpoonright l' = D[s] \upharpoonright l'$.

Proof. By the explicit application of coding and the recovery process following the enumeration of any number into B, C or D by any of the strategies on σ, neither *Coding* nor any of the R-strategies indexed by nodes along σ will require that numbers below l' be enumerated to record events that occurred between their implementation during stage s and the implementation of M_l^σ.

Assumption (1) implies that neither the direct coding of E into D nor any coding of A into B or C by an S-strategy can cause a number less than l' to enter B, C or D after stage s. It is enough to prove the claim: No strategy in σ is the first to cause a marker located below l' to move.

Proceed by induction along σ to prove the claim. Suppose the claim is true for all strategies in $\beta \subset \sigma$. In the case that β is empty, let r equal 0. If β is not empty, let r be the supremum of the restraints due to elements of β. Let $\sigma(m)$ be the next element of σ.

The first cast is when $\sigma(m)$ is $\langle R_i^\beta \rangle$. R_i^β will only move a marker $\Gamma_i^\beta(n)$ located below l' when either E changes below max-min(i, n, s) or both of R_i^β's computations $\Psi_i(B)$ and $\Phi_i(C)$ change value at an argument below $n+1$. Since l' passes the test for $\langle R_i^\beta \rangle$, the value of max-min$(i, n, s)$ is less than l'. Thus, assumption (1) implies that R_i^β will not move $\Gamma_i^\beta(n)$ for the sake of a change in E. Similarly, since max-min(i, n, s) is less than l', there is a computation of length less than l' for one of $\Psi_i(B) \upharpoonright n+1$ or $\Phi_i(C) \upharpoonright n+1$ which has the same value as the one in effect during the stage when $\Gamma_i^\beta(n)$ was assigned its current position. Thus, R_i^β will not require the movement of $\Gamma_i^\beta(n)$ for the sake of recording a new common value without a number first being enumerated below l' to destroy the computation available during stage s.

For the remaining types of strategies, there are two possibilities. First, l' could pass the entire test at step m by clause (b). In this case, l' is equal to the restraint imposed by the mth strategy in σ. If this strategy is of the form $M_i^{\sigma \upharpoonright m}$, then we can conclude the result by induction, since $r^{\sigma \upharpoonright m}$ is the least number that passed the test for $M_i^{\sigma \upharpoonright m}$. Otherwise, the mth strategy is of the form $S_i^{\sigma \upharpoonright m}$. This strategy enumerates numbers to code A into one of B or C and moves markers in an attempt to produce configurations. $S_i^{\sigma\restriction m}$ does not enumerate numbers or move markers below its imposed restraint unless prompted to do so by a change in one of A or E below this restraint (see Proposition 1.5). Assumption (1) ensures that neither A nor E will prompt any such action by $S_i^{\sigma\restriction m}$. Thus, by induction, no strategy above $S_i^{\sigma\restriction m}$ will enumerate a number less than l', the restraint imposed by $S_i^{\sigma\restriction m}$, $S_i^{\sigma\restriction m}$ itself will not enumerate any numbers less than l' and no strategy below $S_i^{\sigma\restriction m}$ can enumerate any such numbers because they are all restrained from doing so by $S_i^{\sigma\restriction m}$. In this case, the lemma is proven.

The final case is when for every m less than the length of σ, l' passes the test at step m by clause (c). If the mth strategy is $R^{\sigma\restriction m}$, we have already seen that it will not initiate the enumeration of any number less than l'. If the mth strategy is of the form $M_i^{\sigma\restriction m}$, then by induction this strategy is restraining a permanent B/C-configuration for i. Thus, its activity is completed. Otherwise, the mth strategy is of the form $S_i^{\sigma\restriction m}$ with tentative outcome either $\langle d, n^\beta, r^\beta\rangle$ or $\langle c, m_{n,i}^\beta, i, r^\beta\rangle$. In either case, the markers that $S_i^{\sigma\restriction m}$ would move to infinity do not have a position when l' is tested. When next assigned positions, those positions will be larger than l'. Thus, subsequent movement of those markers or any other markers later chosen by $S_i^{\sigma\restriction m}$ will only result in the enumeration of numbers greater than l'. Then the only way that $S_i^{\sigma\restriction m}$ could cause the enumeration of a number less than l' would be to move a marker associated with a smaller value of n. This will not happen, because assumptions (1) and (2) imply that the B or C configurations preserved by $S_i^{\sigma\restriction m}$ during stage s are permanent. The claim follows. □

2.5. Corollary. *In the same context as in the previous lemma, any restraint imposed by the strategies appearing in σ is permanent.*

2.6. Lemma. *For all σ and for all l and s there is a stage s configuration for l respecting σ.*

Proof. As we argued above, since no markers are assigned positions during the execution of M_l^σ, M_l^σ can only remove finitely many markers to restart its recursion. Once M_l^σ stops moving markers, any l' that is greater than any computation associated with a marker that is indexed by an initial segment of σ and associated with a marker that has a position will pass the test. □

2.7. Lemma. *Suppose that*

(1) *The strategies in σ play infinitely often in some construction with tentative outcomes as indicated by σ.*

(2) *The restraints appearing in the outcomes listed in σ are respected after they are imposed by the strategies in σ.*

(3) *M_l^σ plays whenever the strategies in σ have the tentative outcomes listed in σ.*

Then, there is a stage s and a l' such that for every $t \geq s$, l' is a stage t configuration for l respecting σ.

Proof. Let s_0 be large enough that the restraints appearing in the outcomes listed in σ have all been imposed by the strategies in σ.

If some restraint imposed by a strategy mentioned in σ passes the entire test for M_l^σ, then it will do so during every stage when σ's outcomes are the tentative outcomes of the strategies mentioned in σ. In this case, the claim is proven.

Otherwise, the proof is the same as in the special case when σ is equal to $\langle\langle R_i\rangle\rangle$. Because M_l^σ moves all markers with argument greater than or equal to $move_l^\sigma$ whenever the shortest configuration for l changes, there is a fixed set of markers F_l which can be involved in any B/C-configuration. By Rule (5), these either have a permanent position and a permanent B or C computation keeping them in position or have no limiting position. Choose s so that s_0 is less than s and for each member $\Gamma_i^{\sigma'}(n)$ of F_l, if $\Gamma_i^{\sigma'}(n)$ has a limiting position, then max-min(i, n, s) is equal to the limit of max-min$(i, n, -)$ and the computation keeping $\Gamma_i^{\sigma'}(n)$ from moving is permanent.

Let l' be the minimum number greater than or equal to l and the supremum of all the restraints imposed by the strategies appearing in σ in the outcomes mentioned in σ such that l' is greater than or equal to sup($\{$max-min$(i, n, s) \mid \Gamma_i^{\sigma'}(n) \in F_l\}$). Note, any marker that does not have a limit position will have position greater than l' after stage s.

Since l' is explicitly chosen to be sufficiently large, l' cannot fail the M_l^σ test by being strictly less than a restraint appearing in σ. Similarly by the choice of l', for each strategy $R_i^{\sigma\restriction m}$ appearing in σ, l' cannot fail the test by being between a marker's position and its associated max-min. Then l' passes the M_l^σ test during every stage greater than or equal to s in which σ seems to correctly predict the eventual behavior of the construction. □

2.8. Proposition. *Suppose, in the context of some construction, that the strategies in σ play infinitely often with tentative outcomes specified in σ. Also, suppose that the only strategies which do not respect the restraint imposed by the strategies in σ are their predecessors in σ. If M_l^σ is played after every play of σ, then the following conditions hold:*

(1) *M_l^σ initiates only finitely many movements of markers.*

(2) *M_l^σ's eventual activity is to preserve some permanent B/C-configuration for l respecting σ.*

(3) *The least A and E correct B/C-configuration for l respecting σ which is preserved by M_l^σ is permanent.*

Proof. The proposition follows from the earlier lemmas. Note that Lemma 2.7 is needed to show that during stage s, M_l^σ eventually decides on some action during stage s. □

3. The tree of strategies and the construction

The strategies used to fulfill the requirements are organized with their outcomes so as to form a tree T. For each strategy, there is an implicit well ordering of its outcomes generated from the following rules.

(1) The βth strategy is R_i^β. $\langle R_i^\beta \rangle$ represents R_i^β and its unambiguous outcome.

(2) The βth strategy is S_i^β.

(a) $\langle S_i^\beta, d, n, r_1 \rangle < \langle S_i^\beta, d, n, r_2 \rangle$, if $r_1 < r_2$;

(b) $\langle S_i^\beta, d, n, r_1 \rangle < \langle S_i^\beta, c, m_{n,i}, i, r_2 \rangle$, for all values of $m_{n,i}$, r_1, r_2 and i;

(c) $\langle S_i^\beta, c, m_{n,i_1}, i_1, r_1 \rangle < \langle S_i^\beta, c, m_{n,i_2}, i_2, r_2 \rangle$, if $i_1 > i_2$, $i_1 = i_2 \;\&\; m_{n,i_1} < m_{n,i_2}$ or $i_1 = i_2 \;\&\; m_{n,i_1} = m_{n,i_2} \;\&\; r_1 < r_2$;

(d) $\langle S_i^\beta, c, m_{n,i}, i, r_1 \rangle < \langle S_i^\beta, d, n+1, r_2 \rangle$, for all value of $m_{n,i}$, i, r and r_2.

By comparing least difference, there is an implicit notion of α's being to the left of β.

The activity involved in stage s of the construction is to determine a finite path $\sigma[s]$ in T and to allow the strategies mentioned in $\sigma[s]$ to take action.

If a node β lies to the right of $\sigma[s]$, then all of the work taken by X^β, the strategy indexed by β, during previous stages will be canceled during stage s. For example, values for r^β, $m_{n,i}^\beta$ or $move_i^\beta$ will be erased. The strategy X^β will start in its initial state the next time that it plays.

If β lies to the left of $\sigma[s]$, then all of the work done by X^β will be left intact during stage s. In particular, if X^β imposed a restraint at an earlier stage which has not been canceled, then that restraint will be replaced during stage s.

Definition of T

The set of nodes σ in T is defined by induction on the length of σ. Also, some auxiliary notions are needed to keep the induction going.

(1) If $\sigma \in T$, say that $\mathcal{R}_i$ *is satisfied in* σ if there are m, r and j such that $\sigma(m) = \langle S_j^{\sigma\restriction m}, c, m_{n,i}, i, r^{\sigma\restriction m} \rangle$.

(2) If $\sigma \in T$ and $\sigma(m) = \langle R_i^{\sigma\restriction m} \rangle$, say that $R_i^{\sigma\restriction m}$ *is injured in* σ if there is an m' greater than m and a k less than i, such that there are j, n, $m_{n,k}$ and r with $\sigma(m') = \langle S_j^{\sigma\restriction m'}, c, m_{n,k}, k, R \rangle$.

(3) Say that $\mathcal{S}_i$ *is satisfied in* σ if there are m and r such that $\sigma(m) = \langle S_i^{\sigma\restriction m}, d, r \rangle$.

Suppose that "$\sigma \in T$" has been defined for all sequences σ of length k. Let σ be an element of T of length k. We follow σ with an implementation of a strategy for the first requirement in the priority list

$$\mathcal{C}, \mathcal{R}_1, \mathcal{S}_1, \mathcal{M}_1, \mathcal{R}_2, \mathcal{S}_2, \mathcal{M}_2, \ldots$$

that does not seem to be satisfied by an element of σ. Let i be minimal so that either $\mathcal{R}_i$ has not been satisfied in σ and every strategy $R_i^{\sigma\restriction m}$ appearing in σ is injured in σ, or $\mathcal{S}_i$ is not satisfied in σ or there is no strategy for $\mathcal{M}_i$ in σ.

Case 1. $\mathcal{R}_i$ is not satisfied in σ and every strategy $R_i^{\sigma\restriction m}$ appearing in σ is injured in σ. The only immediate successor of σ in T is $\sigma^\frown\langle R_i^\sigma\rangle$.

Case 2. Case 1 does not hold and $\mathcal{S}_i$ is not satisfied in σ. Let

$$\text{Active-index}(\sigma) = \{j \mid j \leq i \text{ and } \mathcal{R}_j \text{ is not satisfied in } \sigma\}.$$

Let

$$\text{Outcome} = \{\langle c, m_{n,j}, j, r\rangle, \langle d, n, r\rangle \mid j \in \text{Active-index}(\sigma) \,\&\, \{n, m_{n,i}, r\} \subset \mathbb{N}\}.$$

The immediate successors of σ in T are the sequences of the form $\sigma^\frown\langle S_i^\sigma, o\rangle$ where $o \in$ Outcome.

Case 3. σ does not fall into either of the first two cases. The immediate successors of σ in T are of the form $\sigma^\frown\langle M_i^\sigma, r\rangle$ where $r \in \mathbb{N}$.

The construction

We uniformly assign infinite disjoint recursive sets P_σ to the nodes in T. During the construction, the strategy associated with the immediate successors of σ will only use numbers taken from P_σ.

The construction takes place in stages. During each stage s there are s many substages. We begin substage m with a value for $\sigma[s] \restriction m$, lying in T. In the definition of T, every immediate successor of $\sigma[s] \restriction m$ in T is associated with the same strategy X^σ.

3.1. Definition. Suppose that $\sigma \in T$. Let Active-strategy(σ) be defined by

$$\text{Active-strategy}(\sigma) = \{R_j^{\sigma\restriction m_j} \mid j \in \text{Active-index}(\sigma) \,\&\, R_j^{\sigma\restriction m_j} \text{ is not injured in } \sigma\}.$$

We modify the description of the strategies given in the earlier sections so that X^σ only uses numbers from P_σ greater than any restraint imposed by a strategy on or to the left of $\sigma[s] \restriction m$. Further, X^σ only works with the markers associated with strategies in Active-strategy(σ) and whenever X^σ enumerates a number it applies the Active-strategy(σ) recovery process.

We play X^σ until it completes its activity with tentative outcome o. Then $\sigma[s](m)$ is defined to be $\langle X^\sigma, o\rangle$.

If during stage s, $\sigma[s]$ is to the left of β in T, then the history of the activity of the strategy associated with β is canceled.

Following Harrington [2], the leftmost path visited infinitely often by $\sigma[s]$ in T is called the true path TP of the construction. The requirements are seen to be satisfied by a finite injury analysis along TP.

3.2. Proposition. (1) *If γ is right of TP, then the strategy X^γ associated with γ moves any particular marker only finitely often.*

(2) *Suppose s_1 is less than s_2 and for each stage t in $[s_1, s_2)$, $\sigma[t]$ lies on or to the right of $\sigma[s_1]$. If a strategy X enumerates a number into B, C or D during stage s_2 violating a restraint imposed by an element of $\sigma[s_1]$, then X is associated with a node on or to the left of $\sigma[s_1]$.*

(3) *Let $\sigma \in TP$. For all m, there is a stage s such that for all $t \geqslant s$ the following conditions hold. If R_i^β is a member of* Active-strategy(σ), *then the only strategy mentioned in σ which ever moves $\Gamma_i^\beta(m)$ during stage t* is R_i^β, *in the context of an application of R_i^β's recovery process. Further, if X is the strategy named in the immediate successors of σ and X acts to move $\Gamma_i^\beta(m)$ during stage t, then $\Gamma_i^\beta(m)$ is moved by X during every stage when X's tentative outcome is equal to its true outcome and $\Gamma_i^\beta(m)$ has a position.*

(4) *The numbers enumerated in B or C by strategies on or strictly to the right of σ which are not members of* Active-strategy(σ) *form a set that is recursive in E.*

(5) *TP is infinite.*

Proof. (1) holds because X^γ starts over infinitely often. Each time that it does so, X^γ chooses new markers to move in its attempt to find a permanent configuration. The choices are always larger than any number previously mentioned in the construction, in particular larger than any argument of a marker previously moved by X^γ.

(2) holds because strategies on σ respect the restraint of their predecessors. Strategies to the right of σ start over after stage s_1 and hence use only numbers that are larger than any number mentioned during stage s_1.

Conditions (3) and (4) are proven by induction on TP. Assume that σ is in TP and (3) and (4) are true along σ. Let X be the strategy associated with the immediate successors of σ in T. We must show that the action of X satisfies (3) and (4). To conclude (5), we also show that there is a leftmost immediate successor of σ that acts infinitely often.

Suppose that $X = R_i^\sigma$. In this case the lemma is clear. R_i^σ only moves its own markers. σ's unique immediate successor $\sigma^\frown\langle R_i^\sigma\rangle$ lies on $\sigma[s]$ during every stage s greater than its length during which σ lies on $\sigma[s]$.

In the other two cases, $X = S_i^\sigma$ and $X = M_k^\sigma$, the inductive hypotheses (1)–(4) imply that Propositions 1.5 and 2.8 apply respectively. These imply that the inductive hypotheses are preserved by the actions of their respective strategies. □

It is implicit in Proposition 3.2 that the outcomes paired with strategies on TP are actually their true outcomes.

3.3. Proposition. *For all i, $\mathcal{S}_i$ is satisfied.*

Proof. Each M_k appears exactly once on TP. Each R_j can appear no more than j times, reappearing each time it is injured by the Σ_3-outcome of a higher priority $R_{j'}$. Similarly, an S_j can appear no more than $j+1$ times, reappearing each time it is injured by an $R_{j'}$ such that $j' \leqslant j$. Eventually, some $S_i^{TP\restriction l}$ appears on TP that is not injured. Then Proposition 1.5, Case I must apply to $S_i^{TP\restriction l}$. So, $\mathcal{S}_i$ is satisfied by the action of $S_i^{TP\restriction l}$. □

2.4. Proposition. *For all i, $\mathcal{R}_i$ is satisfied.*

Proof. The strategies $R_i^{TP\upharpoonright m}$ on TP can be injured only finitely often since each injury involves eliminating some $R_{j'}$ where j' is less than i. Applying Proposition 1.5, if there are k, m and r such that $TP(m)$ is equal to $\langle S_k^{TP\upharpoonright m}, c, m_{n,i}, i, r\rangle$, then $\mathcal{R}_i$ is satisfied. Otherwise, there is an m and an r such that $\langle R_i^{TP\upharpoonright m}, r\rangle$ is not injured on TP. We use Propositions 1.1, 1.5, 2.8 and 3.2 to show that the actions of $R_i^{TP\upharpoonright m}$ satisfy $\mathcal{R}_i$. Proposition 1.1 implies that $R_i^{TP\upharpoonright m}$ will satisfy $\mathcal{R}_i$ provided that its rules are respected. Since strategies $R_k^{TP\upharpoonright l}$ do not move markers associated with $R_i^{TP\upharpoonright m}$, these strategies respect $R_i^{TP\upharpoonright m}$'s rules. Proposition 1.5 states that all of the rules of $R_i^{TP\upharpoonright m}$ are respected by any strategy $S_k^{TP\upharpoonright l}$. By Proposition 2.8, any strategy $M_k^{TP\upharpoonright l}$ makes only finitely many moves of any particular marker; Proposition 3.2 implies the same for strategies off of TP. Thus, Proposition 1.1 can be applied and $\mathcal{R}_i$ is satisfied. $\square$

3.5. Proposition. *B, C and D are recursive in A.*

Proof. A computes $B \upharpoonright k$, $C \upharpoonright k$ and $D \upharpoonright k$ by finding the least A and E correct B/C-configuration for k preserved by some strategy in the construction. Since there is an m such that $M_k^{TP\upharpoonright m}$ appears on TP, we can apply Lemma 2.7 to see that there is such a configuration. Moreover, by Proposition 2.8 and Proposition 3.2, if there is a σ such that $\sigma \subseteq \sigma[s]$ and M_k^{σ} preserves an A and E correct B/C-configuration, then σ is an element of TP or σ lies to the left of TP. Thus the restraint imposed by σ is permanent and the computation of B, C and D is correct. $\square$

References

[1] P. Fejer, The structure of definable subclasses of the recursively enumerable degrees, Ph.D. Dissertation, University of Chicago, 1980.
[2] L.A. Harrington, Understanding Lachlan's monster paper, Handwritten notes.
[3] A.H. Lachlan, Decomposition of recursively enumerable degrees, Proc. Amer. Math. Soc. 79 (1980) 629–634.
[4] G.E. Sacks, The recursively enumerable degrees are dense, Ann. of Math. 80 (1964) 300–312.
[5] R.I. Soare, Recursively Enumerable Sets and Degrees, Perspectives in Mathematical Logic (Springer, Berlin, 1987).