A nonlow$_2$ r.e. degree with the extension of embeddings properties of a low$_2$ degree

Richard A. Shore*
Department of Mathematics
Cornell Univeristy
Ithaca NY 14853

Yue Yang
Department of Mathematics
National University of Singapore
Singapore 119260

August 23, 2000

Abstract

We construct a nonlow$_2$ r.e. degree $\mathbf{d}$ such that every extension of embedding property that holds below every low$_2$ degree holds below $\mathbf{d}$. Indeed, we can also guarantee the converse so that there is a low r.e. degree $\mathbf{c}$ such that that the extension of embedding properties true below $\mathbf{c}$ are exactly the ones true below $\mathbf{d}$. Moreover, we can also guarantee that no $\mathbf{b} \leq \mathbf{d}$ is the base of a nonsplitting pair.

1 Introduction

Our goal in this paper is to show how constructions establishing properties shared by all low$_2$ r.e. degrees (i.e. ones $\mathbf{c}$ such that $\mathbf{c}'' = \mathbf{0}''$) can be modified to construct a nonlow$_2$ degree with the same properties. This implies that none of the properties considered (nor indeed all of them together) can separate the low r.e. degrees from the nonlow$_2$ ones. In particular, none of them can define the low$_2$ r.e. degrees inside the structure $\mathcal{R}$ of all the r.e. degrees.

As examples, we consider the properties established for all low$_2$ r.e. degrees in Shore and Slaman [1990]. In particular, in §1 we consider extension of embedding properties. (If $\mathcal{X} \subseteq \mathcal{Y}$ are partial orderings with the same least and greatest elements 0 and 1, we say that a partial ordering $\mathcal{P}$ (with 0 and 1) satisfies the extension of embedding property for $\mathcal{X}$ and $\mathcal{Y}$ if every embedding of $\mathcal{X}$ into $\mathcal{P}$ (preserving 0 and 1) can be extended to one of $\mathcal{Y}$.) Thus we construct a nonlow$_2$ r.e. D such that every extension of embedding

*Partially supported by NSF Grant DMS-9802843 and a Visiting Distinguished Professorship at the National University of Singapore.

property that holds below every low$_2$ r.e. degree (i.e. in $\mathcal{R}(\leq c)$, the r.e. degrees below $\mathbf{c}$, for every low$_2$ r.e. $\mathbf{c}$) holds below $\mathbf{d}$. We then explain in §2 how, in addition, we can make any such property which fails below some low$_2$ r.e. degree also fail below $\mathbf{d}$. As there is also a low r.e. $\mathbf{c}$ with this same property, we see that even the intersection of all of these properties fails to distinguish between the low or low$_2$ r.e. degrees and the nonlow$_2$ ones. In §3 we show how to simultaneously obtain a result previously obtained directly by Cooper, Li and Yi [ta] by making $\mathbf{d}$ not bound any base of a nonsplitting pair, i.e. for every $\mathbf{u} < \mathbf{v}$ with $\mathbf{u} \leq \mathbf{d}$ there are $\mathbf{v}_0$ and $\mathbf{v}_1$ such that $\mathbf{u} < \mathbf{v}_0, \mathbf{v}_1 < \mathbf{v}$ and $\mathbf{v}_0 \oplus \mathbf{v}_1 \equiv_T \mathbf{v}$.

In the case of low$_2$ degrees, all these constructions proceed by converting what would, without the assumption of low$_2$-ness, be infinite injury requirements into finite ones. The crucial point typically is that some positive or negative action would become infinite when some functional involved in the requirement appears total by having the lim sup of its domain of convergence go to infinity but is not actually total. This problem is typically handled by using the low$_2$-ness of the oracle to guess at the totality the functional. The true outcome is guessed infinitely often and the priority tree can be arranged so that it is also the leftmost outcome guessed infinitely often. Thus, if we guess that the function is not total, no action is necessary. On the other hand, if we guess that it is total then we are assured of the guess being correct and we can act accordingly. In our constructions here, we replace the guessing at totality by a requirement that forces the specific functional being considered to actually be total if the lim sup of its domain goes to infinity. Not surprisingly, the natural mechanism for such a procedure is the restraint typically imposed to make a degree low$_2$. What we do here is intersperse these requirements with ones that instead make the set constructed nonlow$_2$. The conflict between these requirement is then the new feature of the construction. It is handled in the usual fashion of $0'''$ type priority arguments.

2 Extension of Embeddings

We first consider satisfying extension of embedding properties. We briefly recall the necessary definitions and notations and refer to Shore and Slaman [1990] for background and other information.

We begin with the nonextendibility conditions. Let

$$\{x_0, \ldots, x_n\} = \mathcal{X} \subseteq \mathcal{Y} = \{x_0, \ldots, x_n, y_0, \ldots, y_m\}$$

be partial orderings (with $x_0 = 0$, the least element of $\mathcal{Y}$ and $x_1 = 1$, the greatest element of $\mathcal{Y}$). The nonextendibility conditions are as follows.

i) For some i and j, $x_i \not\leq y_i$ but

$$\forall x \in \mathcal{X}[\forall x' \in \mathcal{X}(x' < y_i \to x' \leq x) \to x_j \leq x].$$

ii) For some i and j, $y_i \not\leq x_j$ but

$$\forall x \in \mathcal{X}[\forall x' \in \mathcal{X}(y_i \leq x'_i \rightarrow x \leq x') \rightarrow x \leq x_j].$$

iii) For some i and j, $y_i \not\leq y_j$ but

$$\forall x, x' \in \mathcal{X}[\forall x'' \in \mathcal{X}(x'' < y_j \rightarrow x'' \leq x) \ \& \ \forall x'' \in \mathcal{X}(y_i \leq x'' \rightarrow x' \leq x'') \rightarrow x' \leq x].$$

Shore and Slaman [1990] proved the following Non-Extendibility Lemma and Extendibility Lemma for low$_2$ r.e. degrees.

Lemma 2.1 (Shore and Slaman [1990) *] If $\mathcal{X} \subseteq \mathcal{Y}$ are partial orderings satisfying one of the nonextendibility conditions then there is a low$_2$ $\mathbf{c}$ and a monomorphism f : $\mathcal{X} \rightarrow \mathcal{R}(\leq \mathbf{c})$ which cannot be extended to one of $\mathcal{Y}$.*

Lemma 2.2 (Shore and Slaman [1990) *] Let $\mathcal{X} \subseteq \mathcal{Y}$ be finite partial orderings not satisfying any of the nonextendibility conditions. If $\mathbf{c}$ is an r.e. low$_2$ degree and $f : \mathcal{X} \rightarrow \mathcal{R}(\leq \mathbf{c})$ is a monomorphism given by $x_i \mapsto deg\ X_i = \mathbf{x_i}$ for $x_i \in \mathcal{X}$, then there is an extension of f given by $y_i \mapsto deg\ Y_i = \mathbf{y_i}$ for $y_i \in \mathcal{Y}$.*

Next, we recall some notations. For each $y_j \in \mathcal{Y} - \mathcal{X}$ we let

$$X_{g,j} = \oplus\{X_i | x_i \leq y_j\}$$

and

$$X_{l,j} = \oplus\{X_i | \forall x \in \mathcal{X}(x \geq y_j \rightarrow x_i \leq x)\}.$$

Given enumerations W_s and Z_s of r.e. sets W and Z, the associated length of agreement function and maximum length of agreement function are defined as follows:

$$l(e, s, W, Z) = \mu x(\neg[\Phi_{e,s}^{W_s}(x) \downarrow = Z_s(x)])$$

and

$$m(e, s, W, Z) = \max\{l(e, t, W, Z) | t \leq s\}.$$

The strategies to satisfy various comparability and incomparability requirements are as follows.

a) The requirements that $x_i \leq y_j \Rightarrow X_i \leq_T Y_j$ are met by coding, i.e., making $Y_j^{[0]} = \{0\} \times X_{g,j}$.

b) The requirements that $y_j \leq x_i \Rightarrow Y_j \leq X_i$ are met by permitting. Any number entering $Y_j^{[>0]}$ must be permitted by $X_{l,j}$.

c) The requirements that $y_i < y_j \Rightarrow Y_i \leq Y_j$ are also met by coding, i.e., making $Y_j^{[i+1]} = \{i+1\} \times Y_i^{[>0]}$.

d) The requirements that $x_j \not\leq y_i \Rightarrow X_j \not\leq_T Y_i$ are met by Sacks preservation. We keep numbers out of Y_i to preserve the computations of $\Phi_e^{Y_i}$ through the maximum length agreement $m(e, s, Y_i, X_j)$.

e) The requirements that $y_i \not\leq x_j \Rightarrow Y_i \not\leq_T X_j$ are met by Sacks coding. We code $X_{l,i}$ into Y_i up to the maximum length of agreement $m(e, s, X_j, Y_i)$.

f) The requirements that $y_i \not\leq y_j \Rightarrow Y_i \not\leq_T Y_j$ are met by both preservation and coding. We keep numbers out of Y_j to preserve the computations of $\Phi_e^{Y_j}$ up to $m(e, s, Y_j, Y_i)$ and simultaneously code $X_{l,i}$ up to this length of agreement into Y_i.

In Shore and Slaman [1990], the condition that the top set C is low$_2$ is used to guess the totality of various functionals on the true path. We now see that there is a nonlow$_2$ set which has the same extension of embedding properties.

Theorem 2.3 *There is a nonlow$_2$ r.e. degree* $\mathbf{d}$ *such that every extension of embedding property that holds below every low$_2$ r.e. degree holds below* $\mathbf{d}$, *in other words, if $\mathcal{X} \subseteq \mathcal{Y}$ are finite partial orderings not satisfying any of the nonextendibility conditions and $f : \mathcal{X} \to \mathcal{R}(\leq \mathbf{d})$ is a monomorphism given by $x_i \mapsto deg\ X_i = \mathbf{x_i}$ for $x_i \in \mathcal{X}$, then there is an extension of f given by $y_i \mapsto deg\ Y_i = \mathbf{y_i}$ for $y_i \in \mathcal{Y}$.*

Description of Strategies We construct an r.e. set D. The primary requirements are of two types. M_e for the extension of embeddings and R_e for nonlow$_2$-ness.

M_e is the e^{th} instance of an extension of embedding problem as in Lemma 2.2. It specifies finite partial orders $\mathcal{X}_e \subseteq \mathcal{Y}_e$ with elements $x_i, i < n_e$ of $\mathcal{X}_e$ and $y_j, j < m_e$ of $\mathcal{Y}_e - \mathcal{X}_e$ and an ordering $\preceq_e$ and representatives $X_{e,i}, i < n_e$ of the elements of $\mathcal{X}_e$ as r.e. sets along with functionals computing them from D, $\Phi_{e,i}^D(x) = X_{e,i}(x)$. In later constructions, we only use $X_{e,i}$ up to $\Phi_{e,i}^D$, that is, even if a number x has entered $X_{e,i}$ by stage s, we still wait for confirmation that $\Phi_{e,i}^D(x) \downarrow = 1$ before we act on the basis of x's being in $X_{e,i}$. We assure that the positive ordering facts about $\preceq_e$ are reflected in $\leq_T$ on the $X_{e,i}$ by requiring that if $i \prec_e j$ then $X_{e,j}^{[i]} = \{i\} \times X_{e,i}$ for $i, j < n_e$. M_e acts, if it can, to guarantee the totality of the $\Phi_{e,i}$ by a restraint imposing requirement similar to those used to make sets low$_2$.

- M_e: If the maximum length of agreement functions $m(\langle e, i\rangle, s, D, X_{e,i})$ between $\Phi_{e,i}^D$ and $X_{e,i}$ go to infinity for all $i < n_e$ then these functions are all total (and so $\Phi_{e,i}^D = X_{e,i}$ for all $i < n_e$).

This goal is met by requiring that, each time a new maximum length of agreement has been reached for every $i < n_e$, lower priority requirements can injure computations below this new maximum t only a finite number of times altogether. Roughly speaking, each such requirement of lower priority with a candidate to put into D can use its current candidate but all new ones appointed later must wait for the length of agreement at M_e to increase and then be chosen larger than all uses for numbers $\leq t$. Alternatively, a requirement higher up on the priority tree but associated with some global requirement R_i of higher priority than M_e may be allowed to injure all computations of some $\Phi_{e,i}^D(m)$ for all sufficiently large m. However, its injuring M_e infinitely often corresponds to satisfying R_i. In such cases, we restart M_e above this outcome in the typical $0'''$ fashion.

The outcomes for M_e are as usual: ∞ for the maximum length of agreement going to infinity and 0 for its having a finite $\limsup$. In the latter case, we satisfy M_e by default and so do not put any subrequirements for M_e on the tree above the outcome 0.

Above the outcome ∞, we expect to enumerate sets $Y_{e,j}, j < m_e$ so that we produce an order monomorphism from $\mathcal{Y}$ into $\mathcal{R}$ as long as the map given by $x_{e,i} \mapsto \deg(X_{e,i})$ is itself an order monomorphism. The positive aspects of the order preserving map conditions are met by overriding coding and permitting requirements $(a), (b), (c)$ on the enumerations as in Shore and Slaman [1990]. The negative aspects of the order preserving map conditions are achieved by the subrequirements $N_{e,i}, P_{e,i}$ and $Q_{e,i}$ associated with M_e. They correspond to the requirements of type $(d), (e)$ and (f), respectively, in Shore and Slaman [1990]. These subrequirements act the same way as the corresponding requirements in Shore and Slaman [1990] as far as imposing restraint on, and enumerating numbers into, the $Y_{e,j}$. Note that as far as these actions are concerned they do nothing to D and are not influenced by any requirements other than $N_{e,k}, P_{e,k}$ and $Q_{e,k}$ for $k \in \omega$ as in Shore and Slaman [1990]. In addition, they replace the oracle guessing at totality using low$_2$-ness from Shore and Slaman [1990] by imposing low$_2$-like restraint to guarantee the totality of the relevant functionals if the associated maximum length of agreement goes to infinity.

We let $i = \langle a, b, c \rangle$ and Ψ_c list the partial recursive functionals for $c \in \omega$. We list the types of subrequirements. Their actions for coding into and preservation on the $Y_{e,j}$ are as described in Shore and Slaman [1990]. In addition, they each have some functional computed from D. We specify this functional below in terms of oracles involving $X_{e,j}$ and $Y_{e,k}$ but as all of these sets are uniformly computable from D (recall that the set D is $X_{e,1}$, which is the top set of the partial ordering), we may uniformly view the functional as one with oracle D. The requirements attempt to make this functional total if its maximal length of agreement goes to infinity by appropriate restraint. Of course, computations from the given oracles are only accepted when the required information about the $X_{e,j}$ is computed from D. It is worth pointing out that all computations are automatically "believable" ones. In a typical tree argument we are often required to use believable computations, i.e., ones not in conflict with the nodes assumptions about the action of positive requirements below of higher priority, which may be expected to put infinite recursive sets into the oracle. As we shall see later in the construction, if, at s, a node α expects a higher priority node β to put numbers less than s into the oracle, then whenever α is accessible, β has already put in those numbers.

We now list the subrequirements of M_e explicitly to fix our notation.

- $N_{e,i}$: $x_{e,a} \not\preceq y_{e,b} \Rightarrow \Psi_c^{Y_{e,b}} \neq X_{e,a}$; $\Psi_c^{Y_{e,b}}$ total if $\lim m(c,s,Y_{e,b},X_{e,a}) = \infty$.
- $P_{e,i}$: $y_{e,a} \not\preceq x_{e,b} \Rightarrow \Psi_c^{X_{e,b}} \neq Y_{e,a}$; $\Psi_c^{X_{e,b}}$ total if $\lim m(c,s,X_{e,b},Y_{e,a}) = \infty$.
- $Q_{e,i}$: $y_{e,a} \not\preceq y_{e,b} \Rightarrow \Psi_c^{Y_{e,b}} \neq Y_{e,a}$; $\Psi_c^{Y_{e,b}}$ total if $\lim m(c,s,Y_{e,b},Y_{e,a}) = \infty$.

The outcomes for N, P and Q type requirements are again ∞ for the maximum length of agreement going to infinity and 0 for its having a finite $\limsup$. If any of these requirements, say Q, has an infinitary outcome ∞, i.e. the associated maximum length of agreement goes to infinity, then the associated functional is total and we win M_e by demonstrating that one of the negative facts assumed about $\preceq_e$ is false. (In Shore and Slaman [1990], this corresponds to the argument that, as we are there assuming that all these facts about $\preceq_e$ are true, all the outcomes of requirements of type (d), (e) and (f) are finite.) In this case, we restart all requirements of lower priority than M_e above the infinitary outcome of Q but not, of course, M_e (or any of its subrequirements). In particular, note that if there is some R_i originating at a node between M_e and this subrequirement (and so of lower priority than M_e) then R_i is restarted above the infinitary outcome of Q as described below and so no subrequirements of the version of R_i originating between M_e and Q is put on the tree above the infinitary outcome of Q. If, along the true path, all subrequirements of M_e have finitary outcomes, we argue that the sets $Y_{e,j}$ constructed satisfy the requirements to meet all the goals of M_e.

We now consider the requirements for making D not low$_2$. We build a single functional $\Lambda(D;x,m)$ and guarantee (via requirements R_e) that for the eth Σ_3 set

$$\Sigma_e = \{x | \exists m \forall y \exists z (\theta_e(x,m,y,z) \downarrow)\}$$

there is an x such that $\Lambda_x^D = \{m | \Lambda(D;x,m) \downarrow\}$ is infinite if and only if $x \notin \Sigma_e$. The point here is that $\{x | \Lambda_x^D \text{ is infinite}\}$ is a Π_2^D set and these requirements will guarantee that it is not a Σ_3 set. Thus D cannot be low$_2$.

- R_e: There is an x such that $\Lambda_x^D = \{m | \Lambda(D;x,m) \downarrow\}$ is infinite if and only if $x \notin \Sigma_e$.

R_e begins by choosing a witness x for diagonalization. As time goes by and R_e is accessible, it enumerates axioms for $\Lambda(D;x,m)$ for each m. It has subrequirements $S_{e,x,m}$ for each m. They test for m being a witness to the Σ_3 fact that $x \in \Sigma_e$ by looking for verifications of the associated Π_2 fact $\forall y \exists z (\theta_e(x,m,y,z)) \downarrow$ for successive y's. The outcomes of such a subrequirement are either ∞ or 0 depending on whether the Π_2 fact is true or not. Each time it finds a witness for a new y, it tries to make $\Lambda(D;x,n) \uparrow$ for each $n \geq m$ to the extent that its priority allows by enumerating a number less than the use $\lambda(x,n)$ into D. In the ∞ outcome for $S_{e,x,m}$ we get a win on R_e by guaranteeing that $\Lambda(D;x,n)\uparrow$ for all sufficiently large n. Above this outcome, we restart all requirements of lower priority than R_e but of course not R_e itself nor any of its subrequirements. In particular, an M requirement between R_e and $S_{e,x,m}$ will be restarted, because M will be injured when $S_{e,x,m}$ puts the use $\lambda(x,n)$ $(n \geq m)$ into D (these uses are selected by R_e which is of higher priority than M). The cost here is an infinite recursive set is put into D and its members are chosen at the node at which R_e appointed x. (Of course, in the finite outcome only finitely many elements are put into D.) If an M_i of higher priority than R_e is satisfied by some subrequirement, say $Q_{i,j}$, having an infinitary outcome, we restart R_e at an appropriate later node by choosing a new witness x' and putting the corresponding subrequirements $S_{e,x',m}$ on the tree above the version of R_e with witness x'.

Fix a recursive priority list of the requirements and subrequirements such that the subrequirements $N_{e,i}, P_{e,i}$, and $Q_{e,i}$ ($S_{e,x,m}$, respectively) appear after M_e (R_e, respectively).

We define the priority tree T recursively in the usual manner. We label each node on T with a requirement or a subrequirement. We assume that the tree T grows upwards. The root node on T is labelled M_0. Suppose that τ is a node on T. If τ is labelled R_e then τ has a unique outgoing edge labelled ∞; otherwise τ has two outgoing edges labelled ∞ and 0, with $\infty <_L 0$.

Let $\vec{t}$ be a finite path in T starting from the root node and ending with the node τ and α a node on $\vec{t}$ labelled M_e. We say that M_e is Σ_3-*injured* at α on $\vec{t}$ if there is a node β labelled R_i (with witness x), and a node γ labelled $S_{i,x,m}$ such that

$$\beta\hat{\ }\infty \subseteq \alpha \subset \gamma\hat{\ }\infty \subseteq \tau.$$

(*Namely, M_e is between the pair R_i and $S_{i,x,m}$, where $S_{i,x,m}$ demonstrates that R_i has a Σ_3-outcome.*) We define R_e is Σ_3-*injured* at α on $\vec{t}$ similarly.

We say that a requirement M_e is *satisfied* on $\vec{t}$ if there is a node $\alpha \in \vec{t}$ labelled M_e such that either $\alpha\hat{\ }0$ is in $\vec{t}$ or α is not Σ_3-injured on $\vec{t}$ and there is γ labelled with a subrequirement of M_e such that

$$\alpha \subset \gamma\hat{\ }\infty \subseteq \tau.$$

If M_e is satisfied on $\vec{t}$ then all its subrequirements are satisfied on $\vec{t}$. Similarly we can define R_e and its subrequirements being *satisfied* on $\vec{t}$.

Continuing the recursive definition of T, if $\vec{t}$ is a finite path in T which ends with the node τ, then τ is labelled with the highest priority U such that U is either a requirement which never appeared before on $\vec{t}$ or all its copies are Σ_3-injured on $\vec{t}$ or U is a new subrequirement of an unsatisfied requirement on $\vec{t}$.

Before we give the formal construction and verifications, let us analyze the conflicts and how they are solved informally.

We first describe the interactions between a typical requirement $Q_{e,i} = Q$ at node β imposing restraint on D to make its functional total and the various requirements $S_{j,x,m}$ that attempt to put numbers into D. Suppose Q at node β is a subrequirement (or the requirement itself, for that matter) of the last version of M_e on the current path which is at $\alpha \subseteq \beta$. Note that we can ignore the believability issue here too. As we shall see, the subrequirements of M_e will be arranged in such a way that no nodes between α and β labelled with a subrequirement of M_e can have infinite outcome. Therefore, we consider a node δ above β devoted to some $S_{j,x,m}$ with R_j originating at γ. If $\delta \supseteq \beta\hat{\ }0$ then Q's actions are finitary and impose only finite restraint on $S_{j,x,m}$. If $\gamma \supseteq \beta\hat{\ }\infty$ then $\lim m(c, s, Y_b, Y_a) = \infty$ (really, of course the β version of this functional) and Q tries to make $\Psi_c^{Y_b}$ total and so attempts to impose infinitary restraint on δ.

The case when $R_j < M_e < Q_{e,i} < S_{j,x,m}$. If R_j has higher priority than M_e, then by construction $\gamma \subseteq \alpha$. In this case we allow δ to injure all computations from D needed by Q on inputs larger than (the code of) δ. If the outcome of δ is ∞ then, of course, we may ruin Q by making it diverge at all numbers bigger than δ but we get a global win on R_j and so restart M_e above this outcome. On the other hand, if the outcome of δ is 0 then it destroys the computation needed by Q at each number greater than δ only finitely often. This allows Q to eventually impose its restraint successfully against lower priority requirements.

The case when $M_e < Q_{e,i} < R_j < S_{j,x,m}$. If R_j has lower priority than M_e, $\gamma \supseteq \alpha$. As we are assuming that $\delta \supseteq \beta$, the construction guarantees that $\gamma \supseteq \beta$ as well. (There are no subrequirements of nodes between α and β put on the tree above $\beta\hat{\ }\infty$.) In this case, the axioms for $\Lambda(D; x, m)$ that δ tries to kill are chosen at $\gamma \supseteq \beta$. Suppose some action for δ has injured a computation at z required by Q. The next time a choice of axiom for $\Lambda(D; x, m)$ is made must be at a stage at which the functional reaches a new maximal length of agreement at β. We then require (essentially automatically) that the use of this axiom be larger than that of the functional at z as computed at node β. Thus δ can injure the computation at z at most once. In addition, we require that it can never injure computations at z for $z < \delta$. Thus if $\lim m(c, s, Y_b, Y_a) = \infty$ the restraint imposed by Q at β will succeed in making its functional total. (Only finitely many lower priority requirements can ever injure the computation at a fixed z and each of them can do so only finitely often.) On the other hand, this restraint does not interfere with the success of an $S_{j,x,m}$ whose outcome is ∞. Each time it acts to destroy a computation of $\Lambda(D; x, m)$ and so perhaps *Q's* computation at some z, the next axiom is chosen above the use of Q's functional at z, but Q in no way restrains δ from putting in the use of this new axiom at any later stage.

We now describe the stage by stage construction. At stage s we specify a string δ_s of length s, called the *accessible string*. We initialize all nodes to the right of δ_s, that is, cancel all actions desired by these nodes and cancel all restraint imposed by these nodes and any previous choice of witnesses or killing points. We begin at stage 0 by initializing all nodes.

We say that s is an α-*expansionary* stage, if the associated maximum length agreement function as measured at α reaches a new height at stage s.

Construction (stage s):

The root of the tree is always accessible.

Suppose that α is accessible if $|\alpha| = s$ we let $\alpha = \delta_s$. If $|\alpha| < s$, we define the next accessible node and its actions depending on the label of α.

(1) α is labelled M_e.

We first execute coding strategies (a) and (c), i.e., for each $j < m_e$ if x has entered $X_{g,j}$ by stage $s-1$, then enumerate $\langle 0, x\rangle$ into $Y_{e,j}$; for each $i, j < m_e$ such that $y_{e,i} \prec_e y_{e,j}$, if z has entered $Y_{e,i}^{[>0]}$ by stage $s-1$ then put $\langle i+1, z\rangle$ into $Y_{e,j}$. Furthermore, execute the coding strategies which are promised by any subrequirement of M_e. For example, if $Q_{e,i}$ promised to put the coding marker $\langle c, e, b, a, 1, v\rangle$ into $Y_{e,a}$, it has not been initialized since making that promise and v enters $X_{l,\langle e,a\rangle}$ at stage s, then enumerate $\langle c, e, b, a, 1, v\rangle$ into $Y_{e,a}$.

α measures the maximum length of agreement $m(\langle e, i\rangle, s, D, X_{e,i})$ for all $i < n_e$. If s is an α-expansionary stage, then $\alpha\hat{\ }\infty$ is accessible. If s is not α-expansionary, then $\alpha\hat{\ }0$ is accessible.

(2) α is labelled a subrequirement, say $Q_{e,i}$, of M_e. The actions for $N_{e,i}$ and $P_{e,i}$ are similar and even simpler, because Q has to deal with both preserving and coding strategies, whereas N (P respectively) only deals with preserving (coding, respectively).

α measures the length of agreement $m(c, s, Y_{e,b}, Y_{e,a})$. If s is an α-expansionary stage, then $\alpha\hat{\ }\infty$ is accessible. We act as follows.

(i) (preservation) set a restraint on $Y_{e,b}$ with value

$$\max\{\psi_c^{Y_{e,b}}(v, s) | v < m(c, s, Y_{e,b}, Y_{e,a})\}.$$

to preserve the computations $\Psi_c^{Y_{e,b}}(v)$ for all $v < m(c, s, Y_{e,b}, Y_{e,a})$.

(ii) (coding) Promise that at any stage $t > s$, if α has not been initialized between s and t, we will put $\langle c, e, b, a, 1, v\rangle$ into $Y_{e,a}$ for every $v < m(c, s, Y_{e,b}, Y_{e,a})$ such that $v \in X_{l,\langle e,a\rangle,t}$ and $v \notin X_{l,\langle e,a\rangle,s}$. (The number 1 in the coding tuple is used as a tag for subrequirements of type Q. The number 0 is used for subrequirements of type P.)

If s is not an α-expansionary stage, then $\alpha\hat{\ }0$ is accessible and no action is taken.

(3) α is labelled R_e. Let x be α's witness for R_e. (If x is not defined, then pick one which is *big*, that is, larger than any number which we have seen so far in this

construction.) Let n be the least number at which $\Lambda^D(x,n)\uparrow$. Pick a use $\lambda^D(x,n)$ which is big. Define $\Lambda^D(x,p)=1$ with use $\lambda^D(x,n)$ for $s>p\geq n$. Let $\alpha\hat{}\infty$ (which is the only leaving edge anyway) be accessible.

(4) α is labelled $S_{e,x,m}$. Let t_0 be the last stage at which α was initialized. Let y be the least number such that $(\forall z<s_0)\theta_e(x,m,y,z)\uparrow$, where s_0 is the stage when α was last accessible or initialized. Choose a killing point m^* as follows. Let γ be the last node below α which has label R_e. For any β between γ and α labelled with a requirement of type M or any of its subrequirements, such that M is not Σ_3-injured at α, let u_β be the biggest use at β for numbers less than or equal to $|\alpha|$. For any node δ below α labelled with $S_{e',x',m'}$ such that $\delta\hat{}\infty\subset\alpha$, let m_δ be the killing point chosen by δ. For any node δ below α labelled with $S_{e,x,m'}$ such that $\delta\hat{}0\subset\alpha$, let λ_δ be the number $\lambda^D(x,m_\delta)$ where m_δ is the killing point chosen by δ. Let m^* be

$$\max(\{t_0\}\cup\{u_\beta:\gamma\subset\beta\hat{}\infty\subset\alpha\}\cup\{m_\delta:\delta\hat{}\infty\subset\alpha\}\cup$$

$$\{\lambda_\delta:\delta\hat{}0\subset\alpha\ \text{ and } \delta \text{ is labelled } S_{e,x,m'}\}).$$

If $\exists z<s\theta_e(x,m,y,z)\downarrow$ then enumerate $\lambda^D(x,m^*)$ into D. Let $\alpha\hat{}\infty$ be accessible. Otherwise, let $\alpha\hat{}0$ be accessible and do nothing.

This finishes the construction.

We now verify that the construction works.

Lemma 2.4 *Suppose P is an infinite path in T. For every requirement or subrequirement U, there is a unique node α on P labelled U such that either for all $n>|\alpha|$, U is never Σ_3-injured on $P\upharpoonright n$ or for all $n>|\alpha|$, it is satisfied on $P\upharpoonright n$.*

Proof. The Lemma follows by an easy induction on e where U is M_e, R_e or one of their subrequirements. □

Let P be the *true path* in T, that is, P is the leftmost path which is accessible infinitely often. We argue by induction along P that every requirement is satisfied or, more precisely, that the following Lemma holds.

Lemma 2.5 *For each $e\in\omega$ all requirements R_e and M_e are satisfied. Indeed, if U is a requirement or subrequirement and α is the unique node on the true path P associated with U as defined in Lemma 2.4, then*

(a) If U is R_e, then there is a stage s_0 such that R_e does not change its witness x at any $s>s_0$.

(b) *If U is $S_{e,m,x}$ and is working for R_e at node $\gamma \subseteq \alpha$ with witness x, as defined in Lemma 2.4, then there is a stage s such that for all $t > s$, α does not change its selection of killing point m^* at stage t. Moreover, if $\alpha\hat{}0 \subset P$, then for all $n \leq m^*$, $\Lambda^D(x,n)$ is defined; if $\alpha\hat{}\infty \subset P$, then for all $n \geq m^*$, $\Lambda^D(x,n)$ is undefined. In fact this is true for any node on the true path labelled with $S_{e,m,x}$.*

(c) *If U is M_e and $\alpha\hat{}0 \subset P$, then $\mathcal{X}_e$ is not a partial ordering as specified by M_e; if $\alpha\hat{}\infty \subset P$, then for each z the computation $\Phi^D_{e,i}(z)$ is eventually preserved and so the functionals $\Phi^D_{e,i}$ are total.*

(d) *If U is $Q_{e,i}$ (or $N_{e,i}$, $P_{e,i}$ respectively), $i = \langle a,b,c\rangle$ and $\alpha\hat{}0 \subset P$, then $\Psi_c^{Y_{e,b}} \neq Y_{e,a}$ (or $\Psi_c^{Y_{e,b}} \neq X_{e,a}$, $\Psi_c^{X_{e,b}} \neq Y_{e,a}$, respectively); if $\alpha\hat{}\infty \subset P$, then the functional $\Psi_c^{Y_{e,b}}$ (or $\Psi_c^{Y_{e,b}}$, $\Psi_c^{X_{e,b}}$, respectively) is total and the third (first, second, respectively) nonextendibility condition is satisfied or $\mathcal{X}_e$ is not a partial ordering as specified by M_e.*

Proof. We prove (a)-(d) by simultaneous induction and then argue that the requirements R_e and M_e are satisfied.

(a) Let α be the unique node on the true path labelled R_e as defined in Lemma 2.4. Let s_0 be the first stage after which no node to the left of α is accessible. Let x be the witness for R_e at α chosen at some stage $s_1 \geq s_0$. Then for all $s > s_1$, α never changes the witness x it uses for R_e because R_e only changes x when it is either initialized or some previous version of R_e is Σ_3-injured at α.

(b) We first argue that the killing point m^* used by α is eventually fixed. As α is on the true path, all nodes below it are also on the true path and there is a last stage t_0 at which it is initialized. By induction, the statements (a)-(d) apply to the nodes below α. For any δ below α labelled with $S_{e',x',m'}$, the killing point m_δ chosen by δ is eventually fixed by the inductive hypothesis for (b). Furthermore, if $\delta\hat{}0 \subset \alpha$, then $\lambda^D(x, m_\delta)$ is eventually fixed. For any β' labelled with a requirement M or one of its subrequirement such that $\gamma \subset \beta'\hat{}\infty \subset \alpha$, the computations at β' on numbers less than or equal to $|\alpha|$ are eventually preserved by (c) and (d). In particular, the uses $u_{\beta'}$ for those computations are eventually fixed. By the definition of the killing point m^*, it is fixed once the numbers t_0, $u_{\beta'}$, m_δ and λ_δ are fixed.

Now suppose $\alpha\hat{}0 \subset P$. Let s_0 be the first stage after which α's killing point m^* is fixed and $\alpha\hat{}0$ is accessible. When γ is first accessible after stage s_0, say at s_1, it will define $\Lambda^D(x,m^*)$ with big use $\lambda^D(x,m^*)$ (if it is not already defined). We claim that this computation of $\Lambda^D(x,m^*)$ will never be injured, which also implies that for all $n \leq m^*$, $\Lambda^D(x,n)$ is defined. The reason is as follows. The nodes below γ will not make $\Lambda^D(x,m^*)\uparrow$, because those nodes act before $\Lambda^D(x,m^*)$ is defined and at later stages choose only larger numbers as candidates to enter D; the nodes to the right of $\alpha\hat{}0$ are handled by the initialization. (Ones to the right of γ are initialized at s_1; ones above γ and to the right of $\alpha\hat{}0$ were initialized at s_0 and have not been accessible between s_0

and s_1.) For the nodes β' between γ and α, we argue based on the outcome of β'. If $\beta'\hat{}\infty \subset \alpha$, then by the assignment of requirements on the priority tree we know that β' is labelled $S_{e',j,n}$ for some $e' \neq e$, moreover the last copy of $R_{e'}$ is assigned to a node $\gamma' \supset \gamma$. Thus when $R_{e'}$ chooses its axiom λ' at or after stage s_1, λ' will be larger than $\lambda^D(x, m^*)$. (Note that β acted at s_0 and so $R_{e'}$ needs to choose a new axiom for β's killing point when it is first accessible at $s \geq s_1$.) Therefore the action of β', which puts λ' into D will not injure $\Lambda^D(x, m^*)$. If $\beta'\hat{}0 \subset \alpha$, then by the choice of s_0 it will not act, otherwise, the node $\beta'\hat{}\infty$ which is to the left of α would be accessible. It remains to look at the nodes above $\alpha\hat{}0$, they will not injure $\Lambda^D(x, m^*)$ by their choices of killing points. Therefore $\Lambda^D(x, m^*)$ is defined.

Finally, suppose that $\alpha\hat{}\infty \subset P$. Let s be the stage after which no node to the left of $\alpha\hat{}\infty$ is accessible and the killing point m^* is fixed. The action of α, when it is accessible, makes $\Lambda^D(x, n)$ undefined for all $n \geq m^*$.

(c) Let α be the node on the true path which is labelled M_e and never Σ_3 injured on $P \upharpoonright n$ for all n greater than the length of α. For notational simplicity, we drop the index e of $\mathcal{X}$ and $\mathcal{Y}$ and their elements in the discussions below.

If $\alpha\hat{}0 \subset P$, then clearly the X_i's cannot be r.e. sets as specified by M_e.

If $\alpha\hat{}\infty \subset P$, we show that the $\Phi^D_{e,i}$'s are total. Let s_0 be a stage after which no node to the left of $\alpha\hat{}\infty$ is accessible. For each i and z, we argue that the computation $\Phi^D_{e,i}(z)$ can only be injured finitely often after stage s_0. Let us look at the nodes on different regions on the tree. We only need to consider those nodes δ which are labelled with $S_{j,x,m}$ type subrequirements, because only those nodes put numbers into D. If δ is to the left of α, it is never accessible after s_0 and so never injures any computation after s_0. If δ is strictly to the right of $\alpha\hat{}\infty$, it is initialized whenever $\alpha\hat{}\infty$ is accessible and so never considers putting in any number less than the use of the computations seen by α. If δ is below α, then when α is accessible, so is δ. Thus when we see the computation $\Phi^D_{e,i}(z)\downarrow$ at α for the first time, δ has acted. At later stages, when R_j, which is the main requirement of $S_{j,x,m}$, chooses its use λ, λ will be larger than the use $\varphi^D_{e,i}(z)$. Hence the action of δ will not injure the computation $\Phi^D_{e,i}(z)$.

The only remaining worry is those δ above $\alpha\hat{}\infty$ labelled $S_{e',x,m}$, which are subrequirements of some $R_{e'}$. First, let us consider the case that $R_{e'}$ is assigned to a node above $\alpha\hat{}\infty$. When $\Phi^D_{e,i}(z)$ is first defined at some stage after s_0, there are only finitely many such R's which have chosen their λ's and each of them can injure $\Phi^D_{e,i}(z)$ at most once. The reason is as follows. At the stage when $S_{e',x,m}$ puts λ into D destroying the computation $\Phi^D_{e,i}(z)$, it also makes all axioms having uses larger than λ undefined. At the next stage when R requirements (not necessarily the same $R_{e'}$) which are assigned above $\alpha\hat{}\infty$ select their new use λ' after $\Phi^D_{e,i}(z)$ has reconverged, hence λ' will be larger the new use for $\Phi^D_{e,i}(z)$. Finally, we look at the case that $R_{e'}$ is assigned to a node γ below α. If $|\delta| \leq z$, then $S_{e',x,m}$ can only act finitely often, otherwise, the version of $S_{e',x,m}$ on the true path would have outcome ∞, (because they all test the same Π_2 condition $(\forall y)(\exists z)\theta_{e'}(x, m, y, z)\downarrow$), which implies M_e is Σ_3-injured by the pair γ and δ. Let s_1 be the stage after which all these finitary δ's will not act. If $|\delta| > z$ and α is not Σ_3-injured at δ, then $S_{e',x,m}$ will not injure $\Phi^D_{e,i}(z)$ by the choice of its killing point. If α is Σ_3-injured at δ, let (γ_0, δ_0) be the innermost pair of R and S requirements, which Σ_3-injures α. After stage s_1, the δ's having length less than z will not act, thus we may assume that $|\delta_0| > z$. Then δ_0's killing point m^*_0 is too large to injure $\Phi^D_{e,i}(z)$, so is the killing point m^* for δ, since $m^* > m^*_0$.

(d) Let us assume that α is labelled $Q_{e,i}$ and $i = \langle a, b, c\rangle$. The arguments for P and N type subrequirements are similar and we omit them.

If $\alpha\hat{\ }0 \subset P$, then clearly $\Psi_c^{Y_{e,b}} \neq Y_{e,a}$.

If $\alpha\hat{\ }\infty \subset P$, then $\Psi_c^{Y_b}$ is total because it is recursive in D and so our explicit actions to preserve it succeed as for the functional preserved by M_e requirements.

We now show that the third nonextendibility condition is satisfied or $\mathcal{X}_e$ is not the partial ordering specified by M_e.

We are given that $y_a \not\leq y_b$. As $\Psi_c^{Y_b}$ is total and $\lim m(c, s, Y_b, Y_a) = \infty$, $\Psi^{Y_b} = Y_a$. Now, the only requirements affecting the sets Y_a and Y_b are subrequirements of M_e. By the definition of the priority tree, all of them on P of higher priority than α have outcome 0 and so act only finitely often. The usual argument for Sacks coding strategies then shows that $X_{l,a} \leq_T Y_a$: Given a number z larger than all higher priority restraint on Y_a, wait for a stage $t > s_0$ such that $m(e, t, Y_b, Y_a) > z$, then $z \in X_{l,a}$ if and only if either $z \in X_{l,a,t}$ or the coding marker $\langle c, e, b, a, 1, z\rangle$ is in Y_a.

On the other hand, the usual argument for Sacks preservation strategies shows that we can compute Y_a from $X_{g,b}$: Given a number z, wait for a stage t after all positive action for higher priority strategies affecting Y_b have ceased acting such that $m(c, t, Y_b, Y_a) > z$ and $Y_b^{[0]} \upharpoonright u = X_{g,b} \upharpoonright u$, where u is the maximal use in the computation $\Psi_c^{Y_b}(v)$ for $v < m(c, t, Y_b, Y_a)$. Then this computation is never injured. Thus, $Y_a \leq X_{g,b}$. Consequently, we have $X_{l,a} \leq X_{g,b}$.

We now assume that $\mathcal{X}_e$ is a partial ordering (with respect to $\leq_T$) as specified by M_e and show that $\mathcal{X}$ and $\mathcal{Y}$ satisfy the nonextendibility condition (iii) with witnesses y_a and y_b. For any $x_i \in \mathcal{X}$, if $\forall x'' \in \mathcal{X}[x'' < y_b \to x'' < x_i]$, then $X_i \geq_T X_{g,b}$. For any $x_j \in \mathcal{X}$, if $\forall x'' \in \mathcal{X}[y_a < x'' \to x' < x'']$, then $X_j \leq_T X_{l,a}$. Since $X_{l,a} \leq_T X_{g,b}$, $X_j \leq_T X_i$. Thus $x_j \leq x_i$ as required in (iii).

Now we argue that the requirements R_e are satisfied. We consider two cases based on whether R_e has a global Σ_3 outcome.

Case 1 There is a node β on P labelled $S_{e,x,m}$ such that $\beta\hat{\ }\infty \subset P$.

At each of the infinitely many stages when $\beta\hat{\ }\infty$ is accessible, β finds a new y such that $(\exists z)\theta_e(x, m, y, z)\downarrow$. Therefore, $(\forall y)(\exists z)\theta_e(x, m, y, z)\downarrow$. Hence m is the Σ_3-witness that x is in Σ_e. On the other hand, by (a), $\Lambda^D(x, n)$ undefined for all $n \geq m^*$. It follows that Λ^D_x is finite.

Case 2 For all nodes β_m on P labelled $S_{e,x,m}$, $\beta_m\hat{\ }0 \subset P$.

In this case, we argue that x is not in Σ_e and $\Lambda_x^D = \omega$ by showing that for all m in ω, $(\exists y)(\forall z)\theta_e(x,m,y,z)\uparrow$ and $\Lambda^D(x,m)\downarrow$.

Fix m. Let s_m be the stage after which no node to the left of β_m is accessible. Let y_m be the least number such that $(\forall z < s_m)\theta_{e,s_m}(x,m,y_m,z)\uparrow$. Then for all $z \geq s_m$ we still have $\theta_e(x,m,y_m,z)\uparrow$, since otherwise $\beta_m\hat{\ }\infty$ would be accessible. Since this is true for all m, x is not is Σ_e. On the other hand, by (a), every $\Lambda^D(x,m)$ is eventually defined.

In both cases, the requirement R_e is satisfied.

Finally we argue that for all e, M_e is satisfied.

If α is labelled M_e is as in Lemma 2.4 and has outcome 0 on P, then M_e is trivially satisfied by (c). If it has outcome ∞ on P, we consider the following two cases.

Case 1. For all β on P labelled with some subrequirement of M_e,

$$\alpha\hat{\ }\infty \subset \beta\hat{\ }0 \subset P.$$

In this case we argue that we successfully extended the monomorphism $f : \mathcal{X} \to \mathcal{R}$ to an extension of f which has domain $\mathcal{Y}$.

The argument is easy. We consider the requirements (a)-(f) as listed before Theorem 2.3. For the comparability requirements, the direct coding requirements (a) and (c) are satisfied by the action in case (1) when $M_e\hat{\ }\infty$ is accessible; (b) is done by permitting as stated in action (2)(ii). Let us consider the incomparability requirements (d). Suppose that $x_a \not\leq y_b$. Then, by statement (c) of our Lemma, the finitary outcome of $N_{e,i}$ on true path shows that $\Psi_c^{Y_b} \neq X_a$ for all c. The arguments for requirements (e) and (f) are similar.

Case 2. There is β which is labelled some subrequirement of M_e, say $Q_{e,i}$, such that

$$\alpha\hat{\ }\infty \subset \beta\hat{\ }\infty \subset P.$$

In this case, (c) of our Lemma says that $\mathcal{X}$ and $\mathcal{Y}$ satisfy the third nonextendibility condition or $\mathcal{X}_e$ is not a partial ordering as specified by M_e.

Thus the requirements M_e are satisfied. $\Box$

3 Nonextension of Embeddings

Our goal in this section is to show that there are individual r.e. degrees **c** and **d** which are low and nonlow$_2$, respectively, such that each extension of embedding property (as specified by a pair $\mathcal{X} \subseteq \mathcal{Y}$) is satisfied below each of them if and only if it is satisfied below every low$_2$ r.e. degree, i.e. $\mathcal{X}$ and $\mathcal{Y}$ fail to satisfy each of the nonextendibility conditions (i)-(iii) listed at the beginning of §2. The existence of such a low **c** follows directly from the proof of Lemma 2.1 in Shore and Slaman [1990]. It is shown there that given $\mathcal{X}$ and $\mathcal{Y}$ which satisfy one of the nonembeddability conditions, it suffices to get an embedding of $\mathcal{X}$ into a finite distributive lattice $\mathcal{L}$ with the supremum of the image of one subset A of $\mathcal{X}$ greater than equal to the infimum of the image of another subset B of $\mathcal{X}$. One then embeds $\mathcal{L}$ in $\mathcal{R}$ as a lattice with top degree low. The image of $\mathcal{X}$ under the composition of these embeddings can then not be extended to one of $\mathcal{Y}$. Now each of these lattices $\mathcal{L}$ can be embedded as a lattice with 0 and 1 into the atomless Boolean algebra. Thus we can take $\mathbf{c}$ to be the low top of an embedding of the atomless Boolean algebra into $\mathcal{R}$. In this case all of these lattices can be embedded into $\mathcal{R}(\leq \mathbf{c})$ preserving 0 and 1 and so any extension of embedding problem satisfying one of the nonextendibility conditions fails in $\mathcal{R}(\leq \mathbf{c})$.

Now we could argue that we can make the nonlow$_2$ degree $\mathbf{d}$ constructed in §1 simultaneously the top of an embedding of the atomless Boolean algebra. This construction seems rather difficult so instead we revert to an earlier analysis of the nonembedding conditions that is easier to implement. In Fejer and Shore [1985] it is shown that one can divide the argument into two pieces. If the set B mentioned above does not consist precisely of just the 1 of the partial ordering $\mathcal{X}$ then it suffices to embed the lattice $\mathcal{L}$ into the degrees below $\mathbf{d}$ preserving just 0. Thus to take care of these cases, it suffices to make the $\mathbf{d}$ of Theorem 2.3 be above the top of an embedding of the atomless Boolean algebra into $\mathcal{R}$. The cases in which $B = \{1\}$ are handled separately by showing that one can assume that $A = \{x | x < 1\}$ and that there is more than one maximal element of A. It then suffices to embed $\mathcal{X}$ into $\mathcal{R}(\leq \mathbf{d})$ in such a way that the supremum of the image of A is $\mathbf{d}$. Thus it suffices to prove the following two propositions.

Proposition 3.1 *The degree* $\mathbf{d}$ *of Theorem 2.3 can be constructed so that, in addition to the properties described there, there is an r.e.* $\mathbf{a} \leq_T \mathbf{d}$ *such that there is an embedding of the atomless Boolean algebra into* $\mathcal{R}(\leq \mathbf{a})$ *preserving* 0 *and* 1*.*

Proof. We incorporate a standard minimal pair and finite injury construction (as, for example, in Soare [1987]) of an embedding of the atomless Boolean algebra into $\mathcal{R}$ with top $\mathbf{a}$ into the construction of Theorem 2.3 replacing $\mathbf{d}$ by $\mathbf{d} \oplus \mathbf{a}$. The only interactions between our construction and that of the embedding is that ours imposes additional finite restraints on the (minimal pair type) construction of $\mathbf{a}$ and the embedding construction requires additional finite positive actions which, of course, go into $\mathbf{d} \oplus \mathbf{a}$. Clearly the addition of these finitary positive requirements does not cause any new problems for the restraint that our construction requires. The positive actions from our construction of $\mathbf{d}$ do not affect the sets built for the embedding of the atomless Boolean algebra below $\mathbf{a}$ as they do not put numbers into $\mathbf{a}$. Similarly, they are not affected by the minimal pair restraint on $\mathbf{a}$ which applies only to numbers going into $\mathbf{a}$. Thus the minimal pair type restraints needed to construct $\mathbf{a}$ as required are successful as well. □

Proposition 3.2 *For every r.e. degree* $\mathbf{d}$ *and every partial ordering* $\mathcal{X}$ *with* 0 *and* 1 *and more than one maximal element strictly below* 1*, there is an embedding of* $\mathcal{X}$ *into* $\mathcal{R}(\leq \mathbf{d})$ *preserving* 0 *and* 1 *such that the join of the images of all the elements* x *of* $\mathcal{X}$ *with* $x < 1$ *is* $\mathbf{d}$*.*

Proof. Let $\mathcal{X} = \{0, 1, x_2, \ldots, x_n\}$. As in Fejer and Shore [1985], we use the technique of the Sacks splitting theorem to produce degrees $\mathbf{d}_2, \ldots \mathbf{d}_n$ such that $\oplus\{\mathbf{d}_i | 2 \le i \le n\} \equiv_T \mathbf{d}$ and $\mathbf{d}_i \not\le_T \oplus\{\mathbf{d}_j | 2 \le j \ne i\}$ for each $i \ge 2$. The required embedding of $\mathcal{X}$ into $\mathcal{R}(\le \mathbf{d})$ is now given by sending x_i to $\oplus\{\mathbf{d}_j | x_j \le x_i\}$, 0 to $\mathbf{0}$ and 1 to $\mathbf{d}$. $\square$

Thus we have a low r.e. degree $\mathbf{c}$ and a nonlow$_2$ one $\mathbf{d}$ which satisfy exactly the same extension of embedding properties. This shows that no collection of such properties can separate the low r.e. degrees from the nonlow$_2$ ones in $\mathcal{R}$ and so that no set of such properties can define the class of low$_2$ degrees in $\mathcal{R}$.

4 Nonsplitting bases

Finally, we modify the construction in §1 and §2 to make sure that no degree $\mathbf{u} \le \mathbf{d}$ is the base of a nonsplitting pair.

Theorem 4.1 *There is an r.e. nonlow$_2$ degree $\mathbf{d}$ such that it satisfies the statements of Theorem 2.3 and Proposition 3.1 and for any $\mathbf{u} \le \mathbf{d}$ and any $\mathbf{v} > \mathbf{u}$ there is a splitting $\mathbf{v}_0, \mathbf{v}_1$ of $\mathbf{v}$ above $\mathbf{u}$, i.e. $\mathbf{u} <_T \mathbf{v}_0, \mathbf{v}_1 <_T \mathbf{v}$ and $\mathbf{v}_0 \oplus \mathbf{v}_1 \equiv_T \mathbf{v}$.*

For simplicity, let us call the set we are constructing in Proposition 3.1 D rather than $D \oplus A$. In addition to the requirement of that construction, we have the following splitting requirements T_e for $e \in \omega$.

- T_e: If $U_e = \Phi_e^D$ and $V_e \ge_T U_e$ then there exist $V_{e,0}$ and $V_{e,1}$ such that

$$V_{e,0} \sqcup V_{e,1} = V_e \;\wedge\; V_e \not\le_T U_e \oplus V_{e,0}, U_e \oplus V_{e,1}.$$

The requirements T_e list the candidate degrees $\mathbf{u} \le \mathbf{d}$ and companion degrees $\mathbf{v} \ge \mathbf{u}$. They are responsible for an enumeration of a set splitting $V_{e,0}, V_{e,1}$ of V_e and have subrequirements $Z_{e,j,k}$ that try to guarantee that $\Psi_j^{U \oplus V_{e,k}} \ne V_e$.

- $Z_{e,j,k}$: $\Psi_j^{U_e \oplus V_{e,k}} \ne V_e$.

These subrequirements act by preserving $U \oplus V_{e,k}$ on the Ψ_j use up to the maximal length of agreement. The use on U is preserved by restraining D; that on $V_{e,k}$, by putting numbers enumerated in V_e into $V_{e,1-k}$ as in the Sacks splitting theorem. If one of these subrequirements has an infinitary outcome it makes $V_{e,k}$ recursive and so preserving $\Psi_j^{U \oplus V_{e,k}}$ is equivalent to making a functional with oracle D total if the corresponding length of agreement goes to infinity.

Without loss of generality, we can assume that $V_e^{[0]} = U_e$. The strategy for T_e is as follows. T_e tries to make Φ_e^D total if the length of agreement with U_e has infinite lim sup as we did for M_e in Theorem 2.3. In addition, it enumerates sets $V_{e,k}$, $k = 0, 1$. When accessible it enumerates every number that has appeared in V_e since the last such stage into precisely one of the $V_{e,k}$ in accordance with the priority of the desires of its subrequirements as in the Sacks splitting theorem. That is, it determines the subrequirement $Z_{e,i,k}$ of highest priority that would be injured by enumerating the new numbers that have entered V_e into $V_{e,k}$ and puts them all into $V_{e,1-k}$.

The strategy for subrequirement $Z_{e,j,k}$ is as follows. We compute the maximal length of agreement between $\Psi_j^{U_e \oplus V_{e,k}}$ and V_e as usual. $Z_{e,j,k}$ tries to protect these computations (and so in effect asks that no number below the use corresponding to the maximal length of agreement be enumerated into $V_{e,k}$). In addition, it acts to guarantee, by the low$_2$-like strategy, that $\Psi_j^{U_e \oplus V_{e,k}}$ is total if this maximum length of agreement goes to infinity. Note that its restraint for the low$_2$-like preservation strategy is imposed directly only on U_e via its computation from D but not on the $V_{e,k}$. The restraint on the $V_{e,k}$ is imposed by the Sacks splitting type strategy implemented by T_e and produced by its respecting the requests of the $Z_{e,j,k}$ for preserving computations. If the length of agreement goes to infinity, the combined effect of these two procedures is to make $V_{e,k}$ recursive and $\Psi_j^{U_e \oplus V_{e,k}}$ total. In this case, we win the global requirement T_e because we have shown that $V_e \leq_T U_e$.

Thus below the infinitary outcome of $Z_{e,j,k}$, there are no more subrequirements for T_e and all strategies of lower priority than T_e are restarted.

If the length of agreement has a finite limit then the actions of $Z_{e,j,k}$ are finitary. It then behaves like the finitary subrequirements of M_e and so does not interfere with satisfying the R requirements. Moreover, if the $Z_{e,j,k}$ are finite for all j, k, then the usual argument for the Sacks splitting theorem shows that we satisfy T_e.

The interactions between the T_e requirements and their subrequirements $Z_{e,j,k}$ with the nonlow$_2$-ness requirements R and the infimum requirements N of Theorem 2.3 are essentially the same as those of the M_e and their subrequirement. There are no interactions between the M_e and T_e requirements or their subrequirements. Indeed, other than the interactions with the R type requirements, the $Z_{e,j,k}$ interact only with other subrequirements of the same T_e and then only by directing it as to which $V_{e,k}$ numbers enumerated in V_e should be put into by T_e. As in the analysis for Proposition 3.1 the interactions with the embedding construction are also trivial. Hence all the strategies can be combined.

5 Bibliography

S. B. Cooper, A. Li and X. Yi [ta], On the distribution of Lachlan nonsplitting bases, University of Leeds, Department of Pure mathematics, 1998 Preprint series no. 37.

P. A. Fejer and R. A. Shore [1985], Embeddings and extension of embeddings in the r.e. tt and wtt degrees, in *Recursion Theory Week: Proceedings, Oberwolfach 1984*, H. D.

Ebbinghaus, G. H. Müller and G. E. Sacks, eds., Springer-Verlag, Berlin, 1985, 121-140.

R. A. Shore and T. A. Slaman [1990], Working below a low$_2$ recursively enumerable degree, Arch. Math. Logic **29**, 201-211.

R. I. Soare [1987], *Recursively Enumerable Sets and Degrees*, Perspectives in Mathematical Logic, Springer–Verlag, Heidelberg, 1987.