Interpretability and definability in the recursively enumerable degrees

André Nies*
University of Chicago, Chicago IL 60637

Richard A. Shore§
Cornell University, Ithaca NY 14853 USA

Theodore A. Slaman‡
University of Chicago, Chicago IL 60637

Abstract

We investigate definability in $\mathcal{R}$, the recursively enumerable Turing degrees, using codings of standard models of arithmetic (SMA's) as a tool. First we show that an SMA can be interpreted in $\mathcal{R}$ without parameters. Building on this, we prove that the recursively enumerable T–degrees satisfy a weak form of the biinterpretability conjecture which implies that all jump classes $\mathrm{Low_n}$ and $\mathrm{High_{n-1}}$ ($n \geq 2$) are definable in $\mathcal{R}$ without parameters and, more generally, that all relations on $\mathcal{R}$ that are definable in arithmetic and invariant under the double jump are actually definable in $\mathcal{R}$. This partially answers Soare's Question 3.7 [35, XVI].

1. Introduction

Coding by first–order formulas in structures arising from computability theory has been a focus of attention for some time. Traditionally, such coding methods are

*Partially supported by NSF Grant DMS-9500983.

§Partially supported by NSF Grants DMS-9204308, DMS-9503503 and ARO through MSI, Cornell University, DAAL-03-C-0027.

‡Partially supported by NSF Grant DMS-9500878 and a CNR Visiting Professorship at the University of Siena.

initially used to study the complexity of the theory of the structure. Typically, one first proves that theory is undecidable by uniformly coding a sufficiently complex class of structures (with parameters). Then, usually using more sophisticated codings, undecidability of low–level fragments of the theory is established. For instance, Harrington and Shelah [10] sketched a coding of the class of Δ^0_2–partial orders in the structure $\mathcal{R}$ of recursively enumerable Turing degrees to prove the undecidability of $Th(\mathcal{R})$. Later, Lempp, Nies and Slaman [15] showed that the Π_3–theory of $\mathcal{R}$ (as a partial order) is undecidable by a coding of the class of finite bipartite graphs with Σ_1–formulas. Another elaboration of the undecidability of a theory is the determination of its computational complexity. For recursion–theoretic structures which can be coded in $\mathcal{N}$ (like $\mathcal{R}$), the expected result is that the complexity is as high as possible, namely that one can give an interpretation of true arithmetic $Th(\mathcal{N})$ in the theory of the structure. To prove this, one finds coding schemes to code models of arithmetic with parameters and tries to give a first–order condition on parameters which implies that the coded model is standard. We supply such an interpretation and conditions below for $\mathcal{R}$. The bare interpretability of $Th(\mathcal{N})$ in $Th(\mathcal{R})$ has previously been established by Harrington and Slaman and, in a different way, by Slaman and Woodin but without explicitly defining a class of standard models.

Due to the use of parameters, the types of codings mentioned so far have a very local character, so they do not seem to reveal much information about global aspects of the structure beyond the complexity of the theory. Another, seemingly unrelated, line of research is the study of definability in recursion theoretic structures $\mathcal{A}$. While coding results give information about what structures can be recovered from $\mathcal{A}$ (possibly extended by some constants) up to isomorphism, definability results in recursion theory tell us which relations that are apparently external to $\mathcal{A}$ are in fact already inherent in $\mathcal{A}$, namely can be defined in a first–order way. As examples, consider the definability of "arithmetic in" and "hyperarithmetic in" within the structure of $\mathcal{D}$ with the jump operator [14]; the jump operator itself in $\mathcal{D}$ [5]; or the class of promptly simple degrees in $\mathcal{R}$ as the class of degrees which are not half of a minimal pair [3]. Considerable previous work on the structure $\mathcal{D}$ of all the Turing degrees, however, has shown that coding methods can be used to prove global results restricting the possible automorphisms of the structure and that such restrictions on automorphisms can often be converted into very general definability results. Typically, this work showed that all automorphisms of $\mathcal{D}$ are the identity on a cone (the degrees above some fixed degree $\mathbf{c}$) and that all relations on this cone that are definable in second order arithmetic are actually definable in $\mathcal{D}$. (See, for example, [32], [19] and [28].) Similarly, Shore [1981] used coding methods to determine the complexity of $\mathcal{D}(\leq \mathbf{0}')$, the (Turing) degrees below $\mathbf{0}'$. He used these methods in [29] to restrict the possible automorphisms of the structure by showing that the triple jump classes are invariant and then derived corresponding definability results.

Our first goal in this paper is to develop such global coding methods for $\mathcal{R}$. For instance, we will give an interpretation of an SMA without parameters, which of course implies that true arithmetic can be interpreted in $Th(\mathcal{R})$. Our second goal is to apply our global coding results to obtain invariance results for automorphisms of $\mathcal{R}$ and then the corresponding definability results. Once again the jump classes play a crucial role. Recall that an r.e. degree $\mathbf{a}$ is $\mathrm{Low_n}$ ($\mathrm{High_n}$) if $\mathbf{a}^{(n)} = \mathbf{0}^{(n)}$ ($\mathbf{a}^{(n)} = \mathbf{0}^{(n+1)}$).

Various methods have been developed to exploit properties of degrees in the classes Low_1, $\mathrm{High}_1, \mathrm{Low}_2$ and its complement, $\overline{\mathrm{Low}_2}$, in recursion theoretic constructions and there are many structural results that rely on degrees being in each of these classes. The question "which of the jump classes are definable" has motivated a fair amount of research. In particular, the methods developed suggested that there should be "natural" order theoretic definitions of Low_2 and High_1 in $\mathcal{R}$. As an approximation, Shore and Slaman ([30], [31]) give an example of a definable class which includes Low_2, but is disjoint from High_1. In another setting, Downey and Shore [7] show that Low_2 is definable in the r.e. tt degrees.

Here, in a uniform approach similar in outline to that previously used for $\mathcal{D}$ and $\mathcal{D}(\leq \mathbf{0}')$, we establish the definability of all the classes $\mathrm{Low_n}$ and $\mathrm{High_n}$ for $n \geq 2$. In fact we show that each relation $P(\vec{\mathbf{x}})$ which only depends on each coordinate of $\vec{\mathbf{x}}$ up to second jump and is arithmetical (i.e. its index set can be defined in $\mathcal{N}$) is definable in $\mathcal{R}$. A further argument then yields the definability of High_1. In accordance with the formulation of our second goal, the definability of these relations is obtained by considering codings of SMA's, particularly of "effectively" coded SMA's and the interaction of such coded models with a degree $\mathbf{x}$. The interactions between a degree $\mathbf{x}$ and the effective SMA's (via the sets coded in them by degrees below $\mathbf{x}$) show that the isomorphism type of $\mathbf{x}$ within $\mathcal{R}$ determines its double jump. This immediately makes the relations invariant under automorphisms. The interactions between the sets so coded and the arithmetic inherent in the SMA's translate these results into the desired ones on definability. Our improvements on Shore's coding in [29] also enable us to improve the definability results established there for the degrees below $\mathbf{0}'$ by one jump (from triple to double).

We now discuss the coding and definability results in more detail. As mentioned above, the first step is to obtain a coding or, in the terminology of [11], an interpretation of an SMA in $\mathcal{R}$ without parameters. (As an example of interpretations of structures, consider the interpretation of Q on Z given by the quotient field construction.) The existence of an interpretation of $\mathcal{N}$ in $\mathcal{R}$ means that $\mathcal{N}$ can be considered as a "projection" of $\mathcal{R}$. Similar results have been proved for many global degree structures by Slaman and Woodin (see [33]) and, in the setting of the r.e. sets, for the r.e. many–one and wtt–degrees (see [21] and [24]). On the other hand, the lattice $\mathcal{E}$ of recursively enumerable sets under inclusion is an example of a structure where an interpretation of true arithmetic in its theory can be given, but no interpretation of the structure $\mathcal{N}$ in the structure $\mathcal{E}$ exists (see [9] and [12]). In all three cases of r.e. degree structures, the most general concept of interpretations is used: a number is represented by an n–tuple of degrees (for some fixed $n \geq 1$), modulo a definable equivalence relation. (This is also the case for the interpretation of Q in Z, where $n = 2$.) The example of the r.e. many-one degrees, $\mathcal{R}_{m-1}$, shows that the existence of such an interpretation does not necessarily imply that the degree structure is similar to $\mathcal{N}$. For example, $\mathcal{R}_{m-1}$ has continuum many automorphisms.

One way to show that some structure is very much like $\mathcal{N}$ is to prove that the two structures are biinterpretable. Of course, there is an obvious interpretation of $\mathcal{R}$ in arithmetic, so what is needed is an interpretation of $\mathcal{N}$ in $\mathcal{R}$ and a definable relation associating each degree $\mathbf{x}$ in $\mathcal{R}$ to codes (relative to this model) of sets of degree $\mathbf{x}$. The existence of such a model and definable relation is called the *biinterpretability conjecture* for $\mathcal{R}$ (Harrington; Slaman and Woodin). In the setting of just the r.e. degrees, the conjecture is equivalent to the existence of any definable *injective* map $f : \mathcal{R} \to \mathbf{N}$. Here $\mathbf{N}$ is the copy of $\mathcal{N}$ coded inside $\mathcal{R}$, and the definability of f means that the corresponding relation on $\mathcal{R}$ between an argument $\mathbf{a}$ and a tuple $\mathbf{b}$ representing $f(\mathbf{a})$ is definable. Such a map can be viewed as an internal coding map, sending a degree $\mathbf{a}$ to an "index" $f(\mathbf{a}) \in \mathbf{N}$ all taking place inside $\mathcal{R}$. An easy modification of any such f gives a standard index, i.e., for each $\mathbf{a}$ an $e = f(\mathbf{a})$ such that $\deg(W_e) = \mathbf{a}$.

The biinterpretability conjecture has many consequences for the analysis of the structure of $\mathcal{R}$. For instance, it implies that a relation on $\mathcal{R}$ is definable iff it is definable in arithmetic and that $\mathcal{R}$ is rigid (i.e. has no nontrivial automorphisms). Hence it contradicts Cooper's recent announcement [6] that there are automorphisms of $\mathcal{R}$ and indeed ones that take a low degree to a nonlow one. A weaker form of the conjecture which only requires that f and $\mathbf{N}$ be definable from parameters has been established by Slaman and Woodin (see [33]) for $\mathcal{D}_T$ $(\leq \mathbf{0}')$ (for which Cooper has also announced the existence of a nontrivial automorphism). Even in this weaker form, the conjecture implies that the structure is a prime model of its theory, its automorphism group is countable, and a relation is definable iff it is arithmetically definable and invariant under automorphisms.

In this paper we provide another type of approximation to the biinterpretability conjecture. We show that there is a definable $f : \mathcal{R} \to \mathbf{N}$ such that, for each $\mathbf{a}$ and $\mathbf{b}$,

$$\mathbf{a}^{(2)} \neq \mathbf{b}^{(2)} \to f(\mathbf{a}) \neq f(\mathbf{b}). \tag{1.1}$$

(In fact, $f(\mathbf{a})$ will be an index for $\mathbf{a}^{(2)}$ in the sense that $W^{(2)}_{f(\mathbf{a})} = \mathbf{a}^{(2)}$.) Thus the coded model $\mathbf{N}$ still interacts with the structure $\mathcal{R}$ as a whole in a very strong way and we have obtained what we called a "global" coding result.

Let $\mathbf{a} \sim_2 \mathbf{b} \Leftrightarrow \mathbf{a}^{(2)} = \mathbf{b}^{(2)}$. As an immediate consequence of (1) we obtain that

$$\begin{array}{l}\text{any relation } C \subseteq \mathcal{R}_\mathcal{T} \text{ which is arithmetical}\\ \text{and invariant under } \sim_2 \text{ is definable in R.}\end{array} \tag{1.2}$$

We call C *arithmetical* if the corresponding relation on indices is. To see that this definability result follows from (1.1), note that $f(C)$ is definable and $C = f^{-1}(f(C))$.

As a corollary, we obtain the definability results for Low_n, High_n $(n \geq 2)$ mentioned earlier, along with the definability of the relation $\sim_2$ itself. As we show that, in $\mathcal{R}_T$, $\mathbf{x} \in \mathrm{High}_\mathbf{1} \Leftrightarrow \forall \mathbf{z} \exists \mathbf{y} \leq \mathbf{x}(\ \mathbf{y} \sim_2 \mathbf{z})$, High_1 is also definable.

The required coding machinery is introduced in §2 along with statements of the technical results needed to carry it out. §3 considers the issue of relativizing our results to structures $\mathcal{R}^\mathbf{z}$, the degrees r.e. in and above $\mathbf{z}$, for arbitrary degrees $\mathbf{z}$. (Most relativize; some do not and so provide elementary differences between various pairs of such structures.) It also shows how to use our coding ideas to improve the known results on invariance and definability in $\mathcal{D}(\leq \mathbf{0}')$ to be comparable to the ones derived here for $\mathcal{R}$ and applies these results to give a proof of Slaman and Woodin's result that all automorphisms of $\mathcal{D}$ are fixed on all degrees above $\mathbf{0}''$. The remaining sections are devoted to the constructions needed to implement the coding machinery. We begin in §4 with a presentation of the Slaman-Woodin coding scheme for interpreting partial orders (and so models of arithmetic) in $\mathcal{R}$ and various technical extensions and improvements. The next section (§5) provides the results needed to compare models coded in this way and so define an SMA in $\mathcal{R}$. §6 combines lattice embeddings with the codings of the previous sections to produce more effective codings of models of arithmetic that are needed to show that $\mathbf{x}''$ is determined by the sets coded in such models by degrees below $\mathbf{x}$. The final section, §7, which is independent of the previous ones, contains the construction of degrees below any $\mathbf{x}$ coding any given Σ_3^X set in the models constructed in §5. (Of course, these sets determine $\mathbf{x}''$.)

Notations and Conventions: We generally follow [35]. In particular, if $\mathbf{x}$ is an r.e. degree, then X denotes some r.e. set in $\mathbf{x}$. Unless otherwise specified, all sets and degrees will be r.e.

2. The Coding Methods

2.1. Schemes

We intend to uniformly code objects of a certain type (e.g., 1–1 maps or models of some finitely axiomatized theory) into $\mathcal{R}$. Formally, a *scheme* for coding objects of a certain type in $\mathcal{R}$ is given by a sequence of formulas $\varphi_0, \ldots, \varphi_k$ (in the language of partial orderings) with a common list $\overline{p}$ of names for parameters and further free variables, as well as a formula $\psi(\overline{p})$ called the *correctness condition*. The first formula φ_0 defines the domain of the interpreted structure (possibly a set of n-tuples) and the remaining formulas define its functions and relations (including equality if the intention is to identify elements (or n-tuples) according to some equivalence relation). The formula ψ typically says that the φ_i that are intended to define functions actually do so and that the relations and functions defined by the φ_i satisfy various axioms. Precise formulations of these notions can be found in [11, 5.3]. We content ourselves with a couple of examples.

Example 2.1. *A scheme S_g for defining a function g on a particular domain is given by a formula $\varphi_0(x; \overline{\mathbf{p}})$ defining the domain of the function; one $\varphi_1(x, y; \overline{\mathbf{p}})$ defining the relation between inputs and outputs; and a correctness condition $\psi(x, y; \overline{\mathbf{p}})$ which says that a function is actually defined on the intended domain: $\forall x(\varphi_0(x; \overline{\mathbf{p}}) \to \exists! y\ \phi_1(x, y; \overline{\mathbf{p}}))$.*

Example 2.2. *A scheme S_M for coding models of some finitely axiomatized fragment PA^- of Peano arithmetic (in the language $L\ (+, \times)$) is given by the formulas*

$$\varphi_0(x, \overline{p}), \varphi_1(x, y, z; \overline{p}), \varphi_2(x, y, z; \overline{p})$$

and a correctness condition $\psi(\overline{p})$ which says that φ_1 and φ_2 define binary operations on the set $\{x : \varphi_0(x;\overline{p})\}$ which satisfy the finitely many axioms of PA^-. In our applications, the axioms ensure that M has a standard part.

In general, a coding scheme S_X introduces a new type of object. The parameters $\overline{p}$ satisfying $\psi(\overline{p})$ code an object, and S_X acts as a decoding key. Using this coding, it becomes possible to quantify over objects of the new type (a form of second order quantification) in the first–order language of $\mathcal{R}$. We can also perform basic mathematical operations on objects of two possibly different types and obtain a uniform way of coding objects of a yet different type (for example, we can define the composition of maps). Furthermore, we can express basic relationships between coded objects by first order conditions on codes; for instance we can express the relationship "g is a partial map from M_0 to M_1", where M_0, M_1 are coded via S_M and g is coded via S_g, by formulas of $\mathcal{R}$. We use the following convention throughout: If a scheme S_X is given, variables X, X_0, etc. denote objects coded by this scheme for a particular parameter list $\overline{\mathbf{p}}$ satisfying the correctness condition. If it is necessary to mention the parameters explicitly, we write $X(\overline{\mathbf{p}})$, $X_0(\overline{\mathbf{p}})$, etc. We say that $\overline{\mathbf{p}}$ codes $X(\bar{\mathbf{p}})$ via S_X.

Definition 2.3. *A class C of structures with a common signature is uniformly definable in $\mathcal{R}$ if, for some scheme S, C is the class of structures coded via S as the parameters range over $\mathcal{R}$. An object X is definable via S if X is coded via S with a particular choice of parameters.*

2.2. A Scheme for Coding SMA's in $\mathcal{R}$

We now describe a particular coding of the type described in Example 2.2. We begin with special class of definable antichains in $\mathcal{R}$.

Definition 2.4. *A set $G \subseteq \mathcal{R}$ is called an SW–set (short for Slaman–Woodin set) above* $\mathbf{b}$ *if, for some parameters* $\mathbf{p}$, $\mathbf{q}$, $\mathbf{r} \in \mathcal{R}$, *$G$ is the set of minimal degrees* $\mathbf{x}$, $\mathbf{b} \leq \mathbf{x} \leq \mathbf{r}$, *such that* $\mathbf{q} \leq \mathbf{x} \vee \mathbf{p}$. *If* $\mathbf{b} = \mathbf{0}$, *we call G simply an SW-set and may omit* $\mathbf{b}$ *from the list of parameters.*

Note that the class of SW–sets (above some $\mathbf{b}$) is uniformly definable in $\mathcal{R}$. Slaman and Woodin (unpublished) constructed sets of this kind (we supply a construction and proof in Theorem 4.1) and used them to give a uniform coding of an arbitrary recursive partial order $\leq_{\mathcal{P}}$ in $\mathcal{R}$ by proving that there exists a u.r.e. sequence $\langle \mathbf{g}_i \rangle$ of degrees and an $\mathbf{l} \in \mathcal{R}$ such that $\{\mathbf{g}_i : i \in \omega\}$ is an SW–set and, for each $i, j \in \omega$, $i \leq_P j \Leftrightarrow \mathbf{g}_i \leq \mathbf{g}_j \vee \mathbf{l}$. They also gave a coding of $(\omega, +, \times)$ in a particular partial order $\mathcal{P}_A$ which, together with their coding of arbitrary recursive partial orders in $\mathcal{R}$, determines a scheme $S_M(\overline{\mathbf{p}})$ as in Example 2.2 such that some coded model M is standard. To construct this particular partial order $\mathcal{P}_A$ coding arithmetic, one starts with a countable antichain of minimal elements p_n which will represent the numbers n. Then, for each $n, m \in \omega$ one adds an element $c_{n,m}$ to $\mathcal{P}_A$ which represents the pair (p_n, p_m). Next, one adds ascending chains of lengths 2 and 3, respectively, from p_n to $c_{n,m}$ and from p_m to $c_{n,m}$. Finally, to code addition, add a chain of length 4 from p_{n+m} to $c_{n,m}$ and for multiplication, add a chain of length 5 from $p_{n\times m}$ to $c_{n,m}$. We provide the recursion theoretic constructions in §4.

The scheme $S_M(\overline{p}) = S_M(p, q, r, \ell)$ (or $S_M(b, p, q, r, \ell)$) will be extended later by a further correctness condition. For now, we just assume the general correctness condition from Example 2.2. Since the elements of a model M are members of some SW–set, they form an antichain in $\mathcal{R}$. We use subscripting or superscripting by M to indicate the interpretations in M of the usual objects of arithmetic. In particular, i^M denotes the $(i+1)^{st}$element of M for $i \in \omega$ and $+_M$ denotes addition in M.

The above considerations show that some coded M is standard. Moreover, combining the Slaman–Woodin construction with permitting, we will be able to produce a coding with the parameter $\mathbf{r}$ below any given nonzero $\mathbf{a}$. Thus all the elements of M are below $\mathbf{a}$. In addition, we can make $\mathbf{r}$ (and even $\mathbf{r} \vee \mathbf{p} \vee \mathbf{q}$) low (Theorem 4.7).

For an arbitrary M (standard or not), we say that M is *coded below* $\mathbf{a}$ if all elements of M are $\leq$ $\mathbf{a}$ and M is *low* if all elements are low. (This does not restrict the parameters used for coding M.)

2.3. Comparison Maps Between Coded Models

After finding a way to code SMA's with particular parameters in a structure $\mathcal{A}$, we would like to find a first order condition on parameters that picks out a subclass of coded models which are standard. This would give an interpretation of true arithmetic in the theory of $\mathcal{A}$. Various ways to find such a condition have been used in other settings. We employ the method of comparison maps. A *comparison map* between M_0 and M_1 is a 1–1 partial map $M_0 \to M_1$ which extends the isomorphism between the standard parts. We intend to find a uniform way to define such maps between any two (good) models M_0, M_1 coded in $\mathcal{R}$: Then, since some model M^* is standard, a (good) model M is standard iff for each (good) M', such a map $g : M \to M'$ exists which is total.

Theorem 2.5. *There is a scheme S_g such that $\forall M_0 \forall M_1 \exists g$, g is a comparison map from M_0 to M_1.*

Keep in mind that the current definition of S_M will be extended by one more correctness condition to make the scheme "good". This condition will be determined during the proof. The precise way the scheme S_g is obtained will be of little relevance for the rest of the paper.

Proof. We combine recursion theoretic methods with coding tricks. As the recursion theoretic component, we use the following lemma to develop a preliminary scheme S_h which allows us to uniformly define the natural embedding $M_0 \to M_1$, if both M_0, M_1 are low and M_0 is standard. Interpolating with several such embeddings and their inverses, we will find a uniform way to define comparison maps.

Lemma 2.6. *Suppose $\mathbf{q}_0, \ldots, \mathbf{q}_{m-1}, \mathbf{r}_1$ are nonzero low degrees. Then there is a low standard M such that for $i, j < m, i^M \leq \mathbf{q}_i$ and $\mathbf{q}_i \ \Theta\ \mathbf{q}_j \Rightarrow i^M \ \Theta\ \mathbf{q}_j$ while for $j \geq m$, $j^M \ \Theta\ \mathbf{r}_1$.*

To prove Theorem 2.5 (assuming Lemma 2.6 which is a special case of Theorem 5.1), we first describe how to uniformly define the isomorphism $\widetilde{h}$ between the initial segments of M_0 and M_1 up to n^{M_i}, for each $n \in \omega$. Let $\mathbf{q}_i = i^{M_0}$ for $i < n$ and $\mathbf{q}_i = (i-n)^{M_1}$ for $n \leq i < 2n$ and let $\mathbf{r}_1$ be the appropriate parameter in the definition of M_1 bounding all the elements of the model. Note that $(\mathbf{q}_i)_{0 \leq i < n}$ and $(\mathbf{q}_i)_{n \leq i < 2n}$ are antichains. Now let M be the SMA obtained applying the Lemma. For $\mathbf{x} \in M_0$ and $\mathbf{y} \in M_1$ we wish to define the map taking $\mathbf{x}$ to $\mathbf{y}$ if and only if $\mathbf{x} = i^{M_0}$ and $\mathbf{y} = i^{M_1}$ for some $i < n$. Suppose then that $\mathbf{x} \in M_0$ and $\mathbf{y} \in M_1$. We claim that by the incomparability requirements of Lemma 2.6,

$$(\exists i < n)(i^{M_0} = \mathbf{x} \ \&\ \mathbf{y} = i^{M_1}) \Leftrightarrow (\exists \mathbf{z} \in M)(\mathbf{x} \text{ is the } M_0\text{-least } \mathbf{w} \text{ such that } \mathbf{z} \leq_T \mathbf{w} \ \&\ \mathbf{y} \text{ is the } M_1 \text{ least } \mathbf{v} \text{ such that } \mathbf{z} + n^M \leq_T \mathbf{y}).$$

The implication from left to right is immediate from the conditions of Lemma 2.6 with $\mathbf{z} = i^M$. For the converse, let $\mathbf{z} \in M$ be as described. Since $\mathbf{z} + n^M \leq_T \mathbf{y} \in M_1$, the conditions on $\mathbf{r}_1$ in the Lemma guarantee that in M, $\mathbf{z}+n < 2n$ and so $\mathbf{z} < n$ in M. The conditions on $\mathbf{q}_j$ for $j < n$ now guarantee that, if $i^M = \mathbf{z}$ (necessarily with $i < n$), then the M_0-least $\mathbf{w}$ Turing above $\mathbf{z}$ is i^{M_0}. Similarly, those on $\mathbf{q}_j$ for $n \leq j < 2n$ guarantee that the M_1-least $\mathbf{v}$ Turing above $\mathbf{z} + n^M$ is i^{M_1} as required.

So $\widetilde{h}$ can be defined via the formula

$$\begin{aligned}\widetilde{\varphi}(\mathbf{x}, \mathbf{y}; \overline{\mathbf{p}}_0, \overline{\mathbf{p}}_1, \overline{\mathbf{p}}, \mathbf{n}) \equiv [&\mathbf{n} \in M(\overline{\mathbf{p}}) \wedge \mathbf{x} \in M_0 \wedge \mathbf{y} \in M_1 \wedge \exists \mathbf{z} \in M(\overline{\mathbf{p}}) \\ &(\mathbf{x} \text{ is the } M_0\text{-least } \mathbf{w} \text{ such that } \mathbf{z} \leq_T \mathbf{w} \;\&\; \mathbf{y} \text{ is the } M_1 \text{ least } \mathbf{v} \\ &\quad \text{such that } \mathbf{z} + n^M \leq_T \mathbf{y})].\end{aligned}$$

(We denote $M(\overline{\mathbf{p}}_i)$ by M_i.)

Clearly $\widetilde{h}$ satisfies the following condition, which can be expressed as a first order correctness condition α on the parameters involved:

$$\begin{aligned}\widetilde{h}(0^{M_0}) = 0^{M_1} \wedge (\forall \mathbf{x} \in M_0)[&\widetilde{h}(\mathbf{x} +_{M_0} 1) \text{ is defined} \Rightarrow \\ &\widetilde{h}(\mathbf{x}) \text{ is defined and } \widetilde{h}(\mathbf{x} +_{M_0} 1) = \widetilde{h}(\mathbf{x}) +_{M_1} 1]\end{aligned} \tag{2.1}$$

Now, to obtain the desired preliminary scheme S_h, we take the union of such maps defined by parameters satisfying the condition α:

$$\varphi(\mathbf{x}, \mathbf{y}; \overline{\mathbf{p}}_0, \overline{\mathbf{p}}_1) \equiv (\exists \overline{\mathbf{p}})(\exists \mathbf{n}) \alpha(\overline{\mathbf{p}}_0, \overline{\mathbf{p}}_1, \overline{\mathbf{p}}, \mathbf{n}) \wedge \widetilde{\varphi}(\mathbf{x}, \mathbf{y}; \overline{\mathbf{p}}_0, \overline{\mathbf{p}}_1, \overline{\mathbf{p}}, \mathbf{n}).$$

If M_0 is standard, there is, by (2.1), a unique map defined in this way, namely the natural embedding $M_0 \to M_1$. Because we are currently only interested in the case that M_0 is standard, we impose as a correctness condition for the scheme S_h that the relation defined by φ be an embedding $M(\overline{\mathbf{p}}_0) \to M(\overline{\mathbf{p}}_1)$ which satisfies (2.1). (Note that we have not yet expressed the standardness of M_0 by a first order condition on parameters.)

We now define the scheme S_g to code comparison maps between any two models M_0, M_1, at the cost of adding an additional correctness condition to the scheme S_M.

Recall that, by definition, M is coded below $\mathbf{c}$ if the elements of M are $\leq \mathbf{c}$. The following notion is a first order approximation to the lowness of $\mathbf{c}$. We say that $\mathbf{c}$ is *good* via M^* if the model M^* which is coded below $\mathbf{c}$ can be embedded via the scheme S_h into any other model coded below $\mathbf{c}$, i.e.,

$$(\forall M \text{ coded below } \mathbf{c})(\exists h : M^* \to M)[h \text{ is total}].$$

(Recall that h is 1–1 by the correctness condition on S_h.) As mentioned above, some M coded below $\mathbf{c}$ is standard, so M^* must be standard. If $\mathbf{c}$ is low, we can (by Lemma 2.6) take as M^* any SMA coded below $\mathbf{c}$, so each low degree is good. Thus we can safely add as a correctness condition in $S_M(b, p, q, r, l)$ that the upper bound $\mathbf{r}$ is good via M, i.e., M is coded below $\mathbf{r}$ and can be embedded via S_h into any other model coded below $\mathbf{r}$.

Since we can make $\mathbf{r}$ low, there are parameter lists satisfying the condition; moreover, any model coded in that way will be standard. Now, existentially quantifying over such parameter lists, we obtain an interpretation of true arithmetic in $Th(\mathcal{R})$.

For clarity, temporarily call the extended scheme S_M^*. The scheme S_g to uniformly define a comparison map (i.e., the isomorphism) between two models $M_i(\overline{\mathbf{p}}_i)$ coded by S_M^* is obtained by interpolating with two low models.

By the new correctness condition, $\mathbf{r}_i \in \overline{\mathbf{p}}_i$ is good via M_i. Choose low standard models M_i^L coded below $\mathbf{r}_i$ via S_M, and let $h_i : M_i \to M_i^L$ be the natural embedding coded via S_h. We already know that the isomorphism $h : M_0^L \to M_1^L$ is coded via S_h. Now the isomorphism $M_0 \to M_1$ equals $h_1^{-1} \circ h \circ h_0$, and it can then be defined via a scheme S_g including parameters to code all the models involved. With *any* choice of parameters coding intermediate models M_i^L, the composition must be the isomorphism $M_0 \to M_1$ by the correctness condition in S_h requiring that h be an embedding of models. This proves Theorem 2.5 , assuming Lemma 2.6. $\square$

We are now ready to obtain a parameterless interpretation of $\mathbf{N}$.

Theorem 2.7. *(i) There is a uniformly definable class $\mathcal{C}_{st}$ of coded standard models of arithmetic.*
(ii) Let $\tilde{\mathbf{N}} = \{(\mathbf{x}, \overline{\mathbf{p}}) : M(\overline{\mathbf{p}}) \in \mathcal{C}_{st} \wedge\ \mathbf{x} \in M(\overline{\mathbf{p}})\}$. The equivalence relation Q on $\tilde{\mathbf{N}}$ given by

$$(\mathbf{x}, \overline{\mathbf{p}})Q(\mathbf{y}, \overline{\mathbf{q}}) \Leftrightarrow (\exists n \in \omega)[\mathbf{x} = n^{M(\overline{\mathbf{p}})} \wedge \mathbf{y} = n^M(\overline{\mathbf{q}})]$$

is definable in $\mathcal{R}$.
(iii) An SMA $\mathbf{N}$ can be defined on the set of equivalence classes $\tilde{\mathbf{N}}/Q$ without parameters.

Proof. (i) By the above, the scheme S_M (which from now on includes the additional correctness condition) only codes SMA's.
(ii) For $(\mathbf{x}, \overline{\mathbf{p}})$, $(\mathbf{y}, \overline{\mathbf{q}}) \in \tilde{\mathbf{N}}$,

$$(\mathbf{x}, \overline{\mathbf{p}})Q(\mathbf{y}, \overline{\mathbf{q}}) \Leftrightarrow (\exists g)[g : M(\overline{\mathbf{p}}) \to M(\overline{\mathbf{q}}) \wedge g(\mathbf{x}) = \mathbf{y}].$$

(iii) Addition and multiplication of equivalence classes representing numbers in $\tilde{\mathbf{N}}$ can be carried out by choosing tuples with a common parameter list $\overline{\mathbf{p}}$ and carrying out the operations in $M(\overline{\mathbf{p}})$. For instance,

$$[(\mathbf{x}_0, \overline{\mathbf{p}}_0)] + [(\mathbf{x}_1, \overline{\mathbf{p}}_1)] = [(x_2, \overline{\mathbf{p}}_2)] \Leftrightarrow \exists \overline{\mathbf{p}}\ \exists \tilde{\mathbf{x}}_0, \tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2 \in M(\overline{\mathbf{p}})$$

$$(\bigwedge_{i=0,1,2} (\mathbf{x}_i, \overline{\mathbf{p}}_i) Q(\tilde{\mathbf{x}}_i, \overline{\mathbf{p}}) \wedge \tilde{\mathbf{x}}_0 +_{M(\overline{\mathbf{p}})} \tilde{\mathbf{x}}_1 = \tilde{\mathbf{x}}_2). \square$$

2.4. Biinterpretability of $\mathcal{R}$ and $\mathcal{N}$ up to second jump

In this paragraph, we develop the additional coding tools needed to obtain a definable map $f : \mathcal{R} \to \mathbf{N}$ such that $\mathbf{a}^2 \neq \mathbf{b}^2 \Rightarrow f(\mathbf{a}) \neq f(\mathbf{b})$. We first concentrate on an easier result:

$$\text{For each } \mathbf{a}, \text{the class } \{\mathbf{x} : \mathbf{x}^{(2)} = \mathbf{a}^{(2)}\} \text{ is invariant under automorphisms of } \mathcal{R}. \tag{2.2}$$

Recall that (2.2) is a consequence of (1.2) of the introduction which in turn is a consequence of the strongest formulation (1.1). Combining the methods developed here with the comparison map machinery will lead us to the full result (1.1).

We now consider SMA's which are also equipped with an "effectively" coded successor function. This way of coding the successor function was first used by Shore in [26].

Definition 2.8. *An effective SMA is an SMA G above $\mathbf{b}$ with u.r.e. elements $\{\mathbf{g}_i : i \in \omega\}$ (listed in the order of the model G) which is coded by parameters $\mathbf{b}, \mathbf{p}, \mathbf{q}, \mathbf{r}, \mathbf{e}_0, \mathbf{f}_0, \mathbf{e}_1, \mathbf{f}_1$ such that*

- *for each i, $(\mathbf{g}_{2i} \vee \mathbf{e}_1) \wedge \mathbf{f}_1 = \mathbf{g}_{2i+1}$, and $(\mathbf{g}_{2i+1} \vee \mathbf{e}_0) \wedge \mathbf{f}_0 = \mathbf{g}_{2i+2}$.*

Note that the sequence $\langle \mathbf{g}_i \rangle$ is an antichain. Moreover, $\mathbf{g}_{2i+1}\ \Theta\ \mathbf{f}_0$ and $\mathbf{g}_{2i+2}\ \Theta\ \mathbf{f}_1$: If, say, $\mathbf{g}_1 \leq \mathbf{f}_0$, then $\mathbf{g}_2 = (\mathbf{g}_1 \vee \mathbf{e}_0) \wedge \mathbf{f}_0 \geq \mathbf{g}_1$. It follows that we can recognize in a first–order way which pair of parameters $\mathbf{e}_i, \mathbf{f}_i$ to choose to obtain the successor operation σ in the model: for $\mathbf{g}, \mathbf{h} \in G$,

$$\sigma(\mathbf{g}) = \mathbf{h} \Leftrightarrow \text{ for some } i \leq 1, \mathbf{g}\ \Theta\ \mathbf{f}_i \text{ and } \mathbf{h} = (\mathbf{g} \vee \mathbf{e}_i) \wedge \mathbf{f}_i.$$

So we obtain a scheme for coding effective SMA's, denoted by S_G.

Our coding is effective in the sense that there is a way to find an index for $\mathbf{g}_n$ in an arithmetically not too complex way. The reason is that for $n > 0$, $\mathbf{g}_n$ is simultaneously a minimal element in the set $\{\mathbf{x} \leq \mathbf{r} : \mathbf{x} \vee \mathbf{p} \geq \mathbf{q}\}$ and, as an infimum, the greatest element in the set $\{\mathbf{x} : \mathbf{x} \leq (\mathbf{g}_{n-1} \vee \mathbf{e}_i) \wedge \mathbf{x} \leq \mathbf{f}_i\}$ ($i = n \bmod 2$)). Both sets have quantifier free definitions from parameters in the language of uppersemilattices. Using these facts, we will show in Lemma 2.13 that, if a low upper bound on all parameters involved exists, then $g_i = \deg_T\{\beta(i)\}^R$ for some map $\beta \leq_T \emptyset''$.

Now, the idea for proving the invariance result (2.2) is to look at the interaction of $\mathbf{a}$ with sets of numbers coded as subsets of effective SMA's. Since it is too much to ask that all parameters needed be below $\mathbf{a}$, we use the following weaker requirement:

Definition 2.9. *Let* $\mathbf{a} > 0$ *and* $\mathbf{u} > 0$ *(*$\mathbf{u}$ *will be noncappable in the applications).*
(i) G *is coded below* $(\mathbf{a}, \mathbf{u})$ *if* $\mathbf{r} \leq \mathbf{a}$ *and all parameters involved are* $\leq \mathbf{u}$.
(ii) A set $X \subseteq \omega$ *is represented below* $(\mathbf{a}, \mathbf{u})$ *if there is a* G *coded below* $(\mathbf{a}, \mathbf{u})$ *and there are further parameters* $\mathbf{c}, \mathbf{d} \leq \mathbf{a}$ *such that*

$$X = \{i : \mathbf{c} \leq \mathbf{g}_i \vee \mathbf{d}\}.$$

We say that $\mathbf{a}$ is *automorphic* to $\mathbf{c}$ if there is an automorphism φ of $\mathcal{R}$ such that $\varphi(\mathbf{a}) = \mathbf{c}$. Consider the following class of sets:

$$S(\mathbf{a}) = \{X : (\forall \mathbf{u})[\mathbf{u} \text{ noncappable } \Rightarrow X \text{ is represented below } (\mathbf{a}, \mathbf{u})]\}. \tag{2.3}$$

Then

$$\mathbf{a} \text{ automorphic to } \mathbf{c} \Rightarrow S(\mathbf{a}) = S(\mathbf{c}). \tag{2.4}$$

The reason is that an automorphism π such that $\pi(\mathbf{a}) = \mathbf{c}$ maps a representation of X below $(\mathbf{a}, \mathbf{u})$ to one below $(\mathbf{c}, \pi(\mathbf{u}))$. To establish the invariance fact (2.2) we prove the following:

Theorem 2.10. *If* $\mathbf{a} = \deg(A)$*, then* $S(\mathbf{a}) = \Sigma^0_3(A)$.

Thus $\{\mathbf{x} : \mathbf{x} \sim_2 \mathbf{a}\} = \{\mathbf{x} : \Sigma^0_3(X) = \Sigma^0_3(A)\}$ is invariant under automorphisms by (2.4). One immediate corollary shows that many relations are invariant.

Definition 2.11. *An* n*-ary relation* $P(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ *on* $\mathcal{R}$ *is* invariant under the double jump *if, whenever* $\mathcal{R} \models P(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ *and* $\mathbf{x}''_1 \equiv_T \mathbf{y}''_1, \ldots, \mathbf{x}''_n \equiv_T \mathbf{y}''_n$*, it is also true that* $\mathcal{R} \models P(\mathbf{y}_1, \ldots, \mathbf{y}_n)$*. (In these circumstances, we also say that* P *is* invariant under $\sim_2$*.) We say that* P *is* invariant in $\mathcal{R}$ *if whenever* $\mathcal{R} \models P(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ *and* φ *is an automorphism of* $\mathcal{R}$*,* $\mathcal{R} \models P(\varphi(\mathbf{x}_1), \ldots, \varphi(\mathbf{x}_n))$.

Corollary 2.12. *Any relation on $\mathcal{R}$ which is invariant under the double jump is invariant in $\mathcal{R}$.* $\Box$

The Theorem is established in a sequence of auxiliary results. First, to prove that $S(\mathbf{a}) \subseteq \Sigma_3^0(A)$, we exploit the effectivity of our coding of SMA's along with the fact that there is a low noncappable degree.

Lemma 2.13. *Suppose $\mathbf{u}$ is low.*

(i) If G is coded below $(\mathbf{a}, \mathbf{u})$ then there is a total function $\beta \leq_T \emptyset''$ such that $\mathbf{g}_i = \deg\{\beta(i)\}^A$.

(ii) If X is represented below $(\mathbf{a}, \mathbf{u})$ then X is in $\Sigma_3^0(A)$.

Thus, since we quantify over all possible noncappable $\mathbf{u}$ in the definition of $S(\mathbf{a})$, $S(\mathbf{a}) \subseteq \Sigma_3^0(A)$.

Proof. (i) Recall that all the $\mathbf{g}_i$ are below $\mathbf{r}$ which is itself low. Thus, using $\emptyset''$ as an oracle, we can determine if, for given e, $\{e\}^R$ is total. Since $R \leq_T A$, it will be enough to find a function β such that the conclusion of (i) holds with R in place of A. Let $\beta(0)$ be some e such that $\mathbf{g}_0 = \deg(\{e\}^R)$. Inductively, assume we have already determined $\beta(i)$, for even i, say. To find $\beta(i+1)$, look for a j such that $\{j\}^R$ is total, $Z = \{j\}^R \leq_T U$ and the T–reductions which determine the degree $\mathbf{g}_{i+1}$ holds for $Z : Q \leq_T Z \oplus P$, $Z \leq \{\beta(i)\}^R \oplus E_1$ and $Z \leq F_1$. Since all the relevant oracle sets are low and lowness indices for them are known, to search for j we only have to answer some Σ_3^0–questions, i.e., carry out a search for j and for three indices of T–reductions recursively in $\emptyset''$. We know this search will terminate because $\mathbf{G}$ is an effective SMA.
(ii) Note that $i \in X \Leftrightarrow C \leq_T \{\beta(i)\}^A \oplus D$. The clause on the right is a $\Sigma_3^0(A)$ property of i since $C, D \leq_T A$. $\Box$

We now address the more delicate inclusion relations $\Sigma_3^0(A) \subseteq S(\mathbf{a})$. For any $\Sigma_3^0(A)$ set X, and any noncappable $\mathbf{u}$, we have to find a G coded below $(\mathbf{a}, \mathbf{u})$ so that X can be represented using G. First, we show that appropriate low effective SMA's exist. We use the fact from [3] that noncappable degrees are the degrees of promptly simple sets to obtain enough permitting by U.

Theorem 2.14. *For any $\mathbf{a} \neq \mathbf{0}$ and for any promptly simple $\mathbf{u}$, there is an effective SMA G coded below $(\mathbf{a}, \mathbf{u})$ such that $\langle \mathbf{g}_i \rangle$ is a u.r.e. sequence and $\oplus_i \mathbf{g}_i$ is low.*

This follows from Theorem 6.1. Then we prove that all $\Sigma_3^0(A)$–sets can be represented below $(\mathbf{a}, \mathbf{u})$ using a model as above. Note that the sequence $\langle \mathbf{g}_i \rangle$ from Theorem 2.14 satisfies the hypotheses of the following Theorem which we prove later as Theorem 7.1.

Theorem 2.15. *If $\langle \mathbf{g}_i \rangle$ is a u.r.e. antichain, $\oplus_i \mathbf{g}_i$ is low and $(\forall i)[\mathbf{a} \; \Theta \; \mathbf{g}_i]$, then for each $\Sigma_3^0(A)$–set S there are $\mathbf{c}, \mathbf{d} \leq \mathbf{a}$ such that $S = \{i : \mathbf{c} \leq \mathbf{g}_i \vee \mathbf{d}\}$.*

The proof of Theorem 2.14 is another extension of the techniques introduced by Slaman and Woodin, incorporating methods of Downey and Shore [8] while the proof of Theorem 2.15 uses methods of Nies [20] and is independent of the rest of the paper. Clearly, these two theorems together show that $\Sigma_3^0(A) \subseteq S(a)$ and so establish Theorem 2.10. □

We now apply the fact that $S(\mathbf{a}) = \Sigma_3^0(A)$ to prove that, up to second jump, the biinterpretability conjecture holds for $\mathcal{R}$.

Theorem 2.16. *There is a definable map $f : \mathcal{R} \to \mathbf{N}$ such that $(\forall \mathbf{a})[W^{(2)}_{f(\mathbf{a})} = \mathbf{a}^{(2)}]$.*

Proof. To give a first–order definition of f, we have to provide an appropriate definable relation R_f which holds between degrees $\mathbf{a}$ and tuples $(i, \overline{\mathbf{p}})$ representing an equivalence class in $\mathbf{N}$. Note that $\mathbf{a}^{(2)}$ is the least degree $\mathbf{v}$ such that each set in $\Sigma_3^0(A)$ is r.e. in $\mathbf{v}$. (If the last statement holds for $\mathbf{v}$, then $A^{(2)}$ and $\overline{A^{(2)}}$ are r.e. in $\mathbf{v}$.) But $\Sigma_3^0(A) = S(\mathbf{a})$, so if we had a first–order way to obtain, from the degree $\mathbf{a}$, representations of $S(\mathbf{a})$ "inside" $\mathbf{N}$ we could define R_f since finding an index for such a least $\mathbf{v}$ is an arithmetical process.

Fix $\mathbf{a}$ and a list $\overline{\mathbf{p}}$ satisfying the correctness condition for S_M (so $M(\overline{p})$ is standard). If $\mathbf{a} = \mathbf{0}$, we assert that $R_f(\mathbf{a}, (i_0, \overline{\mathbf{p}}))$ holds, where i_0 is some fixed index for the empty set. If $\mathbf{a} \neq \mathbf{0}$, consider the following representation of $S(\mathbf{a})$ as a subset of $M(\overline{p})$:

$$\begin{aligned} S(\mathbf{a}, \overline{\mathbf{p}}) = \{\mathbf{j} \;\; \in \;\; & M(\overline{\mathbf{p}}) : (\forall \mathbf{u} \text{ noncappable})[\{n : M(\overline{p}) \models n \in W_{\mathbf{j}}^{\phi^{(3)}}\} \\ & \text{is represented below } (\mathbf{a}, \mathbf{u})\}]. \end{aligned} \tag{2.5}$$

Thus we need to show that "$\mathbf{j} \in S(\mathbf{a}, \overline{\mathbf{p}})$" is definable. This follows from the existence of comparison maps between any two SMA's whether effective or not. We simply note that

$$\begin{aligned} j \;\; \in \;\; & S(\mathbf{a}, \overline{\mathbf{p}}) \Leftrightarrow (\forall \mathbf{u} \text{ noncappable})(\exists G \text{ coded below } (\mathbf{a}, \mathbf{u})) \\ (\exists g \;\; : \;\; & G \leftrightarrow M(\overline{p}))[g^{-1}(W_j^{\emptyset^{(3)}}) \text{ is represented below } (\mathbf{a}, \mathbf{u}) \text{ via } G]. \end{aligned}$$

As g can be uniformly defined via the scheme S_g, the right hand side can be expressed in a first–order way. Finally, let $R_f(\mathbf{a},(i,\overline{\mathbf{p}}))$ hold if i is the least element of $M(\overline{\mathbf{p}})$ such that for each $\mathbf{j} \in S(\mathbf{a},\overline{\mathbf{p}})$, $W_j^{\emptyset^{(3)}}$ is r.e. in $(W_i)^{(2)}$ and, if the same holds for $i' \in M_{\overline{p}}$ then $W_i^{(2)} \leq_T W_{i'}^{(2)}$.

Since "$\mathbf{j} \in (\mathbf{a},\overline{\mathbf{p}})$" is definable in $\mathcal{R}$ and the other expressions involved can be expressed in the SMA $M_{\overline{p}}$, R_f is definable. $\Box$

We now derive some consequences of Theorem 2.16. Consider the following equivalence relations: $\mathbf{x} \sim_k \mathbf{y} \iff \mathbf{x}^{(k)} = \mathbf{y}^{(k)}$. From Theorem 2.16, the following general definability result can be derived.

Corollary 2.17. *If $C \subseteq \mathcal{R}^n$ is a relation which is invariant under $\sim_2$ such that the corresponding relation on indices of r.e. sets is arithmetical, then C is definable in $\mathcal{R}$ without parameters.*

Proof. Since C invariant under $\sim_2$, it is sufficient to show that the image relation $f(S)$ is definable in $\mathbf{N}$, where $f : \mathcal{R} \to \mathbf{N}$ is the map of Theorem 2.16. But $f(\mathbf{C})$ is arithmetical as a relation on $\mathbf{N}$, since f is definable and C has an arithmetical index relation. $\Box$

We give a few more examples of such relations.

Corollary 2.18. *For each $k \geq 2$ the relation $\mathbf{x}\ \mathrm{s}_k \mathbf{y}$ is definable in $\mathcal{R}$.* $\Box$

Corollary 2.19. *For each $\mathbf{c}$ r.e. in and above $\mathbf{0}''$, the set of r.e. degrees $\mathbf{a}$ with double jump $\mathbf{c}$ is definable in $\mathcal{R}$.* $\Box$

Corollary 2.20. *The jump classes $Low_n = \{\mathbf{a} |\, \mathbf{a}^{(n)} = \mathbf{0}^{(n)}\}$ and $High_n = \{\mathbf{a} |\, \mathbf{a}^{(n)} = \mathbf{0}^{(n+1)}\}$, are definable in $\mathcal{R}$ for $n \geq 2$.* $\Box$

The second example gives a partition of R_T into infinitely many definable automorphism bases, since by [1] already each first-jump class is an automorphism base. So each second-jump class D is a union of orbits, and each automorphism is determined by its action on D.

The definability of $\sim_2$, together with the Robinson Jump Interpolation Theorem (see [36]) and a result by Soare and Stob [36] implies the definability of High_1. For the rest of this section the letters $\mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{s}, \mathbf{t}$ denote degrees which are not necessarily r.e. Recall that $\mathbf{v}$ is *REA* in $\mathbf{u}$ if $\mathbf{v} \geq \mathbf{u}$ and $\mathbf{v}$ is r.e. in $\mathbf{u}$.

Theorem 2.21. *In $\mathcal{R}$, $\mathbf{x} \in \mathrm{High}_1 \Leftrightarrow (\forall \mathbf{y})(\exists \mathbf{z} \leq \mathbf{x})[\mathbf{z} \sim_2 \mathbf{y}]$*

Proof. We quote the Robinson Jump Interpolation Theorem ($RJIT$): if $\mathbf{w} \leq \mathbf{v}$ are r.e. degrees, $\mathbf{s} \geq \mathbf{w}'$ and $\mathbf{s}$ is r.e. in $\mathbf{v}$, then $\mathbf{s} = \mathbf{t}'$ for some r.e. $\mathbf{t}$ such that $\mathbf{w} \leq \mathbf{t} \leq \mathbf{v}$. The $RJIT$ will be used in relativized form, i.e. with "r.e." replaced by "REA in $\mathbf{v}$" for some $\mathbf{v}$.

First suppose $\mathbf{x} \in \mathrm{High}_1$. Given $\mathbf{y}$, let $\mathbf{s} = \mathbf{y}^{(2)}$. We apply the $RJIT$ two times according to the following diagram. The lowest line (except for the first column) contains the degree to which we are relativizing the $RJIT$, and each step to the right means going down by one jump.

$$
\begin{array}{lcc}
 & \mathbf{x}' & \mathbf{x} \\
\mathbf{s} & \mathbf{t} & \mathbf{z} \\
\mathbf{0}^{(2)} & \mathbf{0}' & \mathbf{0} \\
\hline
\mathbf{x}' = \mathbf{0}^{(2)} & \mathbf{x} & \mathbf{0}
\end{array}
$$

In words, first we relativize the $RJIT$ to $\mathbf{x}$ in order to obtain $\mathbf{t}' = \mathbf{s}$ with $\mathbf{t}$ REA in $\mathbf{s}$. We use the highness of $\mathbf{x}$ to ensure that $\mathbf{s}$ is r.e. in $\mathbf{x}'$. Now, by the unrelativized $RJIT$, $\mathbf{t} = \mathbf{z}'$ for some r.e. $\mathbf{z} \leq \mathbf{x}$. So $\mathbf{z} \sim_2 \mathbf{y}$.

Now suppose $\mathbf{x} \notin \mathrm{High}_1$, i.e. $\mathbf{x}' < \mathbf{0}'$. By the main result in [36], relativized to $\mathbf{x}'$, there exists a $\mathbf{u}$ REA in $\mathbf{0}'$ such that $\mathbf{u}$ is not r.e. in $\mathbf{x}'$. Now by a twofold application of the $RJIT$, first relativized to $\mathbf{0}'$ and then unrelativized, $\mathbf{u} = \mathbf{y}'$ for some $\mathbf{y}$. But $\mathbf{u} = \mathbf{z}'$ for some $\mathbf{z} \leq \mathbf{x}$ implies that $\mathbf{u}$ is r.e. in $\mathbf{x}'$. Λ

3. Relativizations of $\mathcal{R}$ and Other Structures

We now use our coding methods to investigate definability in relativizations of $\mathcal{R}$ and to distinguish among different relativizations. We will also see how they can be combined with Shore's methods from [29] to improve the invariance and definability results for $\mathcal{D}(\leq \mathbf{0}')$, the degrees below $\mathbf{0}'$, derived there by one jump to get ones analogous to the ones established here for $\mathcal{R}$. The invariance results for the double jump in $\mathcal{D}(\leq \mathbf{0}')$ will then be used to prove Slaman and Woodin's result that every automorphism of $\mathcal{D}$ fixes every degree above $\mathbf{0}''$. In this section degrees will not be r.e. unless explicitly so specified.

We begin with relativizations of $\mathcal{R}$: $\mathcal{R}^{\mathbf{z}}$ is the partial order of degrees REA in $\mathbf{z}$. We show that, as a consequence of relativizations of our coding results, $\mathbf{z}^{(2)} \not\equiv_T \mathbf{w}^{(2)}$

implies that $\mathcal{R}^{\mathbf{z}}$ Æ $\mathcal{R}^{\mathbf{w}}$. Moreover, under suitable additional conditions, the two structures are not even elementarily equivalent. These results are analogous to those of Shore for relativization of $\mathcal{D}(\leq \mathbf{0}')$ [26] and are improvements of similar results for relativizations of $\mathcal{R}$ in [27].

The relativization of a definition or result such as the ones above to a degree $\mathbf{z}$ is obtained by replacing the notions "recursive" by "recursive in $\mathbf{z}$" and r.e. by "REA in $\mathbf{z}$" (recursively enumerable in and above $\mathbf{z}$). In particular, the role of $\mathbf{0}$ is now played by $\mathbf{z} = \deg(Z)$ (which we also denote by $\mathbf{0}$).

An examination of the arguments in Sections 2, 4, 6 and 7 reveals that Theorem 2.10 relativizes:

$$\text{If } \mathbf{a} \in \mathcal{R}^{\mathbf{z}}, \text{ then } S(\mathbf{a}) = \Sigma^0_3(A). \tag{3.1}$$

For instance, Lemma 2.13, which showed that $S(\mathbf{a}) \subseteq \Sigma^0_3(A)$, relativizes because now in the search to find $\beta(i+1)$ we have to evaluate $\Sigma^0_3(Z)$ questions, so we obtain a map $\beta \leq_T Z^{(2)}$. But Z is recursive in all sets involved, so, as before, $X \in \Sigma^0_3(A)$ if X is represented below $(\mathbf{a}, \mathbf{u})$, where $\mathbf{u}' = \mathbf{z}'$.

Theorem 3.1. *If* $\mathbf{z}^{(2)} \not\equiv_T \mathbf{w}^{(2)}$, *then* $\mathcal{R}^{\mathbf{z}}$ Æ $\mathcal{R}^{\mathbf{w}}$.

Proof. It is sufficient to recover $\Sigma^0_3(Z)$ from the structure $\mathcal{R}^{\mathbf{z}}$ in a way that depends only on the isomorphism type of the structure. To do so, note that

$$\Sigma^0_3(Z) = \bigcap_{\mathbf{a} \neq \mathbf{0}} S(\mathbf{a}) \tag{3.2}$$

since one can apply (3.1) to some $\mathbf{a} \in \mathcal{R}^{\mathbf{z}} - \{\mathbf{0}\}$ such that $\mathbf{a}' = \mathbf{z}'$. □

We now consider definability issues for relativized versions of $\mathcal{R}$ and implications for their theories. One can check that the comparison map machinery, along with its extension to effective SMA's, relativizes to any $\mathcal{R}^{\mathbf{z}}$ (via the same schemes). In particular, the correctness condition in S_M on $\overline{p}$ that the parameter $\mathbf{r}$ be good implies that $M_{\overline{\mathbf{p}}}$ is standard, and Theorem 2.7 defines a standard model $\mathbf{N}^{\mathbf{z}}$ in each relativization $\mathcal{R}^{\mathbf{z}}$. However, when we consider relativizing the discussion of sets coded in the model $\mathbf{N}^{\mathbf{z}}$, we can no longer simply use an index in $\mathbf{N}^{\mathbf{z}}$ as a code. Instead we use pairs of degrees outside the model and interpret the sets coded as prescribed by Theorem 2.15. Thus, while the notions of invariant and invariant under the double jump remain the same as in the unrelativized case, we must adjust our definition of "definable in arithmetic" accordingly. We now allow free set variables in our formulas ψ and the usual binary relation symbol $\in$ for membership (i.e., the membership of a degree coding a natural number in these coded sets).

An n-ary relation P on degrees is then said to be *definable in arithmetic* if there is such a formula ψ such that $P = \{\langle \deg(X_1), \ldots, \deg(X_n)\rangle | \mathcal{N} \models \psi(X_1, \ldots, X_n)\}$. (Of course, this agrees with the definition in terms of indices when all the sets X_i are r.e.) With this definition, we immediately get the appropriate relativized version of Theorems 2.7 and 2.16 and most of the associated corollaries:

Theorem 3.2. *For every degree* $\mathbf{z}$*, there is a definable copy* $\mathbf{N}^{\mathbf{z}}$ *of the structure* $(\mathcal{N}, +\times)$ *in* $\mathcal{R}^{\mathbf{z}}$*and a definable relation associating each degree* $\mathbf{a}$ *REA in* $\mathbf{z}$ *with codes for sets of degree* $\mathbf{a}''$. $\square$

Corollary 3.3. *For every degree* $\mathbf{z}$*, any relation on* $\mathcal{R}^{\mathbf{z}}$ *which is invariant under the double jump is invariant in* $\mathcal{R}^{\mathbf{z}}$. $\square$

Corollary 3.4. *For every degree* $\mathbf{z}$*, any relation on* $\mathcal{R}^{\mathbf{z}}$ *which is definable in arithmetic (as redefined above) and invariant under the double jump is definable in* $\mathcal{R}$. $\square$

Corollary 3.5. *For every degree* $\mathbf{z}$*, and for each* $k \geq 2$ *the relation* $\mathbf{x}\ \mathrm{s}_k \mathbf{y}$ *defined by* $\mathbf{x}^{(k)} \equiv_T \mathbf{y}^{(k)}$ *is definable in* $\mathcal{R}^{\mathbf{z}}$. $\square$

Corollary 3.6. *For every degree* $\mathbf{z}$*, the jump classes* $Low_n^{\mathbf{z}} = \{\mathbf{a} \in \mathcal{R}^{\mathbf{z}} | \, \mathbf{a}^{(n+1)} = \mathbf{z}^{(n+1)}\}$ *and* $High_n^{\mathbf{z}} = \{\mathbf{a} \in \mathcal{R}^{\mathbf{z}} | \, \mathbf{a}^{(n)} = \mathbf{z}^{(n+1)}\}$ *are definable in* $\mathcal{R}^{\mathbf{z}}$ *for* $n \geq 2$. $\square$

Corollary 3.7. *For every degree* $\mathbf{z}$*, the jump class* $High_1^{\mathbf{z}} = \{\mathbf{a} \in \mathcal{R}^{\mathbf{z}} | \, \mathbf{a}' = \mathbf{z}''\}$ *is definable in* $\mathcal{R}^{\mathbf{z}}$. $\square$

On the other hand, the proofs of the last part of Theorem 2.16 and of Corollary 2.19 do not relativize. Indeed, any attempt at talking about maps from degrees to indices or even any form of unique codes for sets of given degrees is doomed to failure as any function definable in $\mathcal{R}^{\mathbf{z}}$(and so arithmetic) taking degrees $\mathbf{d}$ to (unique) representatives of $\mathbf{d}$ would contradict arithmetic determinacy. The same is true even if we try to associate degrees (REA in $\mathbf{z}$) with integers (in the standard model of arithmetic defined in $\mathcal{R}^{\mathbf{z}}$) up to any jump:

Theorem 3.8. *There are degrees* $\mathbf{z}$ *such that there is no* $k \in \omega$ *and no map* f *from* $\mathcal{R}^{\mathbf{z}}$ *to* $\mathbf{N}^{\mathbf{z}}$*, the isomorphic copy of* $\mathcal{N}$ *definable in* $\mathcal{R}^{\mathbf{z}}$*, which is definable in* $\mathcal{R}^{\mathbf{z}}$*such that* $f(\mathbf{a}) = f(\mathbf{b})$ *implies that* $\mathbf{a}^k \equiv_T \mathbf{b}^k$.

Proof. If not, there would be a k and a well-ordering of $\mathcal{R}^{\mathbf{z}^{(k)}}$ which is Borel definable from $\mathbf{z}$. Then using the coding of reals by parameters in $\mathcal{R}^{\mathbf{z}^{(k)}}$, one could map the degree $\mathbf{z}$ to a real X in a uniformly Borel way so that this map is not constant on any cone. For example, order the reals coded in $\mathcal{R}^{\mathbf{z}^{(k)}}$ by the ordering of the parameters which code them and choose the first real X such that every real coded in $\mathcal{R}^{\mathbf{z}}$ is similarly coded in $\mathcal{R}^{\mathbf{x}}$. The existence of such a map contradicts Borel determinacy. $\square$

Thus, in general, no analog of the second part of Theorem 2.16 is possible for $\mathcal{R}^{\mathbf{z}}$. A similar argument shows that the analog of Corollary 2.19 also fails:

Theorem 3.9. *There are degrees* $\mathbf{z}$ *and* $\mathbf{c}$ *with* $\mathbf{c}$ *REA in* $\mathbf{z}''$*, such that the set of degrees in* $\mathcal{R}^{\mathbf{z}}$ *with double jump* $\mathbf{c}$ *is not definable in* $\mathcal{R}^{\mathbf{z}}$*.*

We can, in fact, use the relativized results that do hold for every $\mathbf{z}$ to improve the nonisomorphism results derived above to nonelementary equivalence for certain $\mathbf{z}$ and $\mathbf{w}$. A subset Z of the natural numbers $\mathcal{N}$ is *implicitly definable* if, for some formula ψ_Z in the language $L(+, \times, P)$ with a symbol P for a unary predicate, Z is the unique set satisfying the description. Formally, for each $X \subseteq \omega$

$$(\mathbf{N}, X) \models \psi_Z \;\Leftrightarrow\; Z = X.$$

In the literature, such sets Z are also called "arithmetical singletons". For example, each arithmetical set and $\emptyset^{(\omega)}$ are implicitly definable. It is easy to verify that the class of implicitly definable sets is closed under the equivalence relation of having the same arithmetical degree.

Theorem 3.10. *If* Z *is implicitly definable (for example,* $Z = \emptyset$*) and* $\mathcal{R}^{\mathbf{z}} \equiv \mathcal{R}^{\mathbf{w}}$ *then* $\mathbf{z}'' = \mathbf{w}''$*.*

Proof. It is easy to see that there is sentence θ in $L(+, \times, P)$ such that

$$(\mathcal{N}, X) \models \theta \Leftrightarrow Z^{(2)} =_T X^{(2)}.$$

Of course, in $\mathcal{R}^{\mathbf{z}}$ the sentence saying that there is a (code for a) set in $\mathbf{N}^{\mathbf{z}}$ satisfying θ and that this set is in $\mathbf{L}_2^{\mathbf{z}}$ is true. The only way this sentence can hold in some other $\mathcal{R}^{\mathbf{w}}$ is for $\mathbf{z}$ and $\mathbf{w}$ to have the same double jump. $\square$

Note that some condition on $\mathbf{z}$ and $\mathbf{w}$ is needed in this theorem since, by Borel determinacy, there is a degree $\mathbf{z}$ such that $\mathcal{R}^{\mathbf{z}} \equiv \mathcal{R}^{\mathbf{w}}$ for every $\mathbf{w} \geq \mathbf{z}$.

Turning next to $\mathcal{D}(\leq \mathbf{0}')$, the degrees below $\mathbf{0}'$, we briefly explain how the coding procedures here can be used to improve Shore's invariance and definability results in [29] by one jump to get results for $\mathcal{D}(\leq \mathbf{0}')$ analogous to the ones provided in §2 for $\mathcal{R}$. (References in the discussion below refer to [29].) Effective standard models of arithmetic are coded by intervals of degrees which are lattices using Lerman's results [17, XII] on lattice initial segments of $\mathcal{D}(\leq \mathbf{0}')$. The crucial change needed is simply to require (as is part of the definition of Slaman-Woodin sets) that the elements of the model M coded below some $\mathbf{e}$ also join some $\mathbf{p} < \mathbf{e}$ above some other $\mathbf{q} < \mathbf{e}$. This makes the collection of (indices for) the elements of M a Σ_3^E set rather than one recursive in $\mathbf{e}^{(3)}$. Thus the function h (of Theorem 1.1) enumerating these indices is recursive in $\mathbf{e}''$ rather than $\mathbf{e}^{(3)}$. We can now argue that the structure of the degrees below any $\mathbf{a} \leq \mathbf{0}'$ determines $\mathbf{a}''$ (rather than just $\mathbf{a}^{(3)}$) and so the double jump is invariant in $\mathcal{D}(\leq \mathbf{0}')$ as it is in $\mathcal{R}$.

By Lemma 1.3, if $\mathbf{f}$ is r.e. in and strictly above $\mathbf{e}$, any Σ_3^E set S can be coded in M by a pair $\mathbf{x}, \mathbf{y} < \mathbf{f}$ (with $i \in S \Leftrightarrow i^M \leq \mathbf{x}, \mathbf{y}$). Consider then any $\mathbf{a} \leq \mathbf{0}'$. If $\mathbf{a}'' = \mathbf{0}''$ then only Σ_3 sets can be coded by pairs below $\mathbf{a}$ in effective standard models which are segments $[\mathbf{b}, \mathbf{e}]$ below $\mathbf{a}$. On the other hand, if $\mathbf{a}$ is not Low_2 then we argue that the sets coded by pairs below $\mathbf{a}$ in models given by segments below $\mathbf{a}$ determine $\mathbf{a}''$: For $\mathbf{g} < \mathbf{a}$ let

$$H(\mathbf{g}) = lub\{\deg(X)| \text{ Every set coded by a pair below } \mathbf{g} \text{ in an effective standard model given by a segment below } \mathbf{g} \text{ is } \Sigma_1^X\}.$$

As every set in $H(\mathbf{g})$ is Σ_3^G, $H(\mathbf{g}) \leq \mathbf{g}'' \leq \mathbf{a}''$. Thus it suffices to show that every $\mathbf{x} \geq \mathbf{0}''$ which is r.e. in $\mathbf{a}'$ is actually $H(\mathbf{g})$ for some $\mathbf{g} < \mathbf{a}$ as the least upper bound of such $\mathbf{x}$'s is $\mathbf{a}''$ by standard density type results on r.e. degrees (relativized to $\mathbf{a}'$). This then implies that $\mathbf{a}'' = lub\{H(\mathbf{g})|\mathbf{g} < \mathbf{a}\}$ as required to determine $\mathbf{a}''$ from $\mathcal{D}(\leq \mathbf{a})$. Now consider any such $\mathbf{x}$. The proof of Theorem 2.1 shows that there are degrees $\mathbf{k} < \mathbf{b} < \mathbf{e} < \mathbf{g} < \mathbf{a}$ such that $\mathbf{k}'' = \mathbf{x}$; $\mathbf{g}$ is 1-generic over $\mathbf{k}$ and r.e. in $\mathbf{b}$; $[\mathbf{e}, \mathbf{b}]$ codes an effective standard model of arithmetic; and there are $\mathbf{x}, \mathbf{y} < \mathbf{g}$ coding each Σ_3^E set in M. As $\mathbf{g}$ is 1-generic over $\mathbf{k}$ and $\mathbf{k} < \mathbf{e} < \mathbf{g}$, $\mathbf{g}'' = \mathbf{x}$ and $\Sigma_3^E = \Sigma_3^G = \Sigma_3^K$. Thus $\mathbf{x} = H(\mathbf{g})$ as required.

This proves the invariance of the double jump in $\mathcal{D}(\leq \mathbf{0}')$. By using the language of arithmetic in the coded models, the invariance result can be converted into definitions of the jump classes Low_n and High_n for $n \geq 2$ (rather than $n \geq 3$). Now, an argument similar to that for our Theorem 2.21 above shows that High_1 is definable in $\mathcal{D}(\leq \mathbf{0}')$ by the same formula used in $\mathcal{R}$. The only change is that the final application of the RJIT is replaced by the jump interpolation theorem given in Theorem 1.6(b) and due to Jockusch and Posner [13]. We have thus proved the following results:

Theorem 3.11. *i) Any relation on $\mathcal{D}(\leq \mathbf{0}')$ invariant under the double jump is invariant.*
ii) The jump classes Low_{n+1} and $High_n$ for $n \geq 1$ are definable in $\mathcal{D}(\leq \mathbf{0}')$.
iii) For each $\mathbf{c}$ REA in $\mathbf{0}''$, the set of degrees $\mathbf{x} \leq \mathbf{0}''$ with $\mathbf{x}'' = \mathbf{c}$ is definable in $\mathcal{D}(\leq \mathbf{0}')$.

The general definability results for arithmetical relations invariant under the double jump in $\mathcal{D}(\leq \mathbf{0}')$ can then be derived (as mentioned in [29]) using comparison maps between models via the methods of Slaman and Woodin [34] as explicated in [25]. One can then derive results on relativizations to structures $[\mathbf{z}, \mathbf{z}']$ as we did above for $\mathcal{R}^{\mathbf{z}}$.

Assuming the definability of the jump (as now proved by Cooper [5]) Nerode and Shore [19] showed that every degree above $\mathbf{0}^{(3)}$ is fixed under every automorphism of $\mathcal{D}$. Slaman and Woodin improved this result by replacing $\mathbf{0}^{(3)}$ with $\mathbf{0}''$. As the last application of our results we note that the relativization of the characterization of $\mathbf{a}''$ from $\mathcal{D}(\leq \mathbf{a})$ supplies another proof of this result:

Theorem 3.12. (Slaman and Woodin)*: Every automorphism of $\mathcal{D}$ is fixed on every $\mathbf{x} \geq \mathbf{0}''$.*

Proof. If $\mathbf{x} \geq \mathbf{0}''$ then there is (by the Friedberg jump theorem) a degree $\mathbf{z}$ such that $\mathbf{z}'' = \mathbf{x}$. As the jump is definable in $\mathcal{D}$ by [5], this relation is definable from $\mathbf{x}$ as is $\mathbf{z}'$ from $\mathbf{z}$. Our arguments now show that $\mathbf{x}$ is determined by the sets coded as above in such structures $[\mathbf{z}, \mathbf{z}']$ and so is fixed by all automorphisms of $\mathcal{D}$. $\Box$

In the above proof our constructions show that we can actually definably get a representative of the degree $\mathbf{x}$ in the coded model. Moreover, in $\mathcal{D}$ we can definably map one such model to any other. Thus we can definably choose a representative of any degree up to double jump and for example prove the following:

Theorem 3.13. *Any relation on $\mathcal{D}$ definable in second order arithmetic and invariant under the double jump is definable in $\mathcal{D}$.*

Of course, as we see from the codings described in §2, there is no need for the degrees coding the model of arithmetic to determine an entire interval of degrees and so require full initial segment type arguments. Thus simpler constructions (if done from scratch) would suffice to construct such codings that would be sufficient to prove these results for $\mathcal{D}(\leq \mathbf{0}')$ and $\mathcal{D}$.

4. Slaman-Woodin Sets

We begin the discussion of our constructions with a description of Slaman and Woodin's basic scheme for constructing a countable set of incomparable degrees $\mathbf{g}_i$ which are definable in $\mathcal{R}$ from finitely many parameters $\mathbf{p}, \mathbf{q}, \mathbf{r}$ as the minimal degrees $\mathbf{x}$ below $\mathbf{r}$ such that $\mathbf{q} \leq_T \mathbf{x} \vee \mathbf{p}$.

Theorem 4.1. (Slaman and Woodin)*: There are r.e. sets P, Q, R and G_i (for $i \in \omega$) with $R = \oplus G_i$ such that*

1. *$(T) : G_i \oplus P \geq_T Q$.*
2. *$(D) : G_i \mid_T G_j$ for $i \neq j$.*
3. *(M) : If W is r.e., recursive in R and $W \oplus P \geq_T Q$, then there is a $j \in \omega$ such that $G_j \leq_T W$.*

We first give a preliminary description of the basic strategies for insuring each of the requirements (1)-(3) above.

4.1. The individual requirements

1. T_i: $G_i \oplus P \geq_T Q$.

This requirement is met by building a functional Γ_i such that $\Gamma_i(G_i \oplus P) = Q$. It introduces the set G_i in the construction and so no requirement preceding it can involve G_i. A crucial goal of the construction will be to arrange that a number x is put into Q to satisfy some requirement of lower priority than T_i only when we can safely put the associated marker $\gamma_i(x)$ (which is the use of $\Gamma_i(G_i \oplus P)$) into P. Of course, we can move this marker if we so desire when G_i (or P) has changed on the use of the functional at x. The typical situation will be that, before x can go into Q, x or some smaller number must enter G_i. At that point, we will redefine $\gamma_i(x)$ to be large enough so that, when x goes into Q, we can put $\gamma_i(x)$ into P without injuring the P preservation associated with the requirement putting x

into Q. We must also somewhere in the construction maintain this ability to put x into Q by keeping $\gamma_i(x)$ above this P restraint or cancel the possibility that x might enter Q. (Requirements of type 4 below will play a role here.)

2. $D_{i,j,e}$: $\Psi_e(G_i) \neq G_j$. ($\langle \Psi_e | \, e \in \omega \rangle$ is a listing of all partial recursive functionals.)

This requirement is met by a Friedberg-Muchnik type of argument. We begin by choosing a *suitable* follower y which is larger than any number used so far in the construction. The conditions on suitability of followers are intimately connected with the strategy for minimality of the G_i (3) and cannot be precisely defined until we describe the strategy for those requirements. Suffice it to say that maintaining the suitability of followers will consist of imposing P restraint and will be $D_{i,j,e}$'s major task in the construction. We wait for a stage s at which the (still suitable) follower is *realized*, $\Psi_e(G_i; y) = 0\,[s]$. We then put y into G_j and preserve G_i on the appropriate use $\psi_e(y)$ by initializing all lower priority requirements.

3. M_i: If $\Theta_i(R) = W_i$ and $\Phi_i(W_i \oplus P) = Q$, then $G_j \leq_T W_i$ for some $j \in \omega$. (Here $\langle W_i, \Theta_i, \Phi_i | \, i \in \omega \rangle$ is a listing of all triples consisting of an r.e. set W and two partial recursive functionals Θ and Φ.)

These requirements, which insure minimality, are by far the most complicated ones. Suppose, for example, that the sets $G_0, \dots G_n$ have been introduced by the requirements T_j for $j \leq n$ of higher priority than M_i. Our first task is to see if the hypotheses of M_i seem satisfied. Thus we first see if we have a new *i-expansionary stage*: We define a length of agreement function $l(i,s) =$

$$\mu z\{\neg[\Phi_i(W_i \oplus P; z) \downarrow= Q(z)\,[s] \,\wedge\, (\forall w \leq \phi_i(z))(\Theta_i(R; w) \downarrow= W_i(w)\,[s]\,)]\,\}.$$

We say that a stage s of our construction is *i-expansionary* if $l(i,s) > l(i,t)$ for every stage $t < s$.

If there are infinitely many expansionary stages for the functionals in the hypothesis of M_i, we will, for each $j \leq n$, build functionals $\Delta_{i,j}$. Moreover, we will guarantee that, for some $j \leq n$ and some m, $\Delta_{i,j}(W_i; x) = G_j(x)$ for every $x > m$. (We could build distinct functionals $\Delta_{i,j,m}$ so that one of them would work for all x but it is technically more convenient to build just one functional that works almost everywhere. Of course this suffices to get $G_j \leq_T W_i$ as desired.) If, on the other hand, the last functional, $\Delta_{i,n}$, is seen to fail, we will kill off the requirement M_i by putting some number x into Q while preserving the Φ_i use $\phi_i(x)$ to guarantee that $\Phi_i(W_i \oplus P) \neq Q$. Thus there will be only finitely many i-expansionary stages. The preservation of this win on requirement M_i has two components. We preserve first $P^- \phi_i(x)$ and then $W_i^- \phi_i(x)$ by preserving $R^- \theta_i \phi_i(x)$, the amount of R needed to compute $W_i^- \phi_i(x)$ via Θ_i, by initializing all lower priority requirements.

Our main goal in setting up the situation to produce this win on M_i is to insure that when we put x into Q, we can correct all the functionals $\Gamma_j(x)$ of higher priority than M_i that are supposed to be computing Q from $G_k \oplus P$ for some k. We expect to be able to do this by putting the number $\gamma_k(x)$ into P. The crucial point is then that we must insure that each $\gamma_k(x)$ is greater than $\phi_i(x)$ as our diagonalization against Q depends on preserving $P^- \phi_i(x)$. The basic idea for getting $\gamma_k(x)$ to be greater than $\phi_i(x)$ is that before x can go into Q it, or some smaller number, must go into each such G_k. When that happens we can increase $\gamma_k(x)$ to maker it larger than $\phi_i(x)$. Some restraint will also be imposed in the usual way (as, for example, in the minimal pair argument) between expansionary stages to preserve $P^- \phi(x)$ until the next expansionary stage even after the associated number has gone into G_k. It is dropped, and so may be violated (even by lower priority requirements), at expansionary stages. Other preservation will be indirectly imposed (by type 4 requirements) that will prevent this from happening infinitely often. In addition, we must know that no W_i change can later increase $\phi_i(x)$ without allowing us to restore the computations $\Delta_{i,j}(W_i; x)$ whose failure prompted us to consider putting x into Q. This condition translates into the requirement that $\delta_{i,j}(x) \geq \phi_i(x)$ as, in this case, any W_i change on $\phi_i(x)$ would allow us to redefine $\Delta_{i,j}(W_i; x)$ to once again equal $G_j(x)$. Thus insuring such inequalities as $\delta_{i,j}(x) \geq \phi_i(x)$ will be an important, if only implicit, concern in the construction.

Although it is a considerable oversimplification, it may be instructive to first consider the case that $n = 0$. Here, there is only one set, G_0, that we can compute from W_i to satisfy the requirement and only one functional $\Delta_i = \Delta_{i,0}$ is constructed.

Now, our concern is that some requirement S of lower priority may want to put x into G_0 (and so destroy the correctness of Δ_i). If such a requirement appoints x as a follower, it will be its responsibility to preserve the suitability of x (until x enters G_0) by imposing restraint on $P^- \phi_i(x)$. As far as the requirement M_i is concerned, the suitability of x will be equivalent to being able to insure that, if and when needed, we can make $\gamma_0(x) > \phi_i(x)$. An additional point to notice is that even though S is preserving $P^- \phi_i(x)$, it is not preserving R. Hence W_i may change on $\phi_i(x)$ before we put x into G_0. As mentioned above, when this happens we redefine the axiom for Δ to make $\delta_i(x) \geq \phi_i(x)$. In addition, S now imposes restraint on P up to the new $\Phi_i(x)$ use. (A more precise description of this restraint will be given when we formally define the action of the Friedberg-Muchnik type diagonalization requirements. We will also explain below how we add new requirements (type 4) to the construction to prevent $\phi_i(x)$, and so the P restraint imposed by S, from becoming unbounded.)

On the other hand, if s is i-expansionary and there is a point at which Δ_i has *failed* (to correctly compute G_0) let x be the least number such that $\Delta_i(W_i; x) \downarrow \neq G_0(x)\,[s]$. Consider now which requirement S put x into G_0 at a stage $t < s$. On general principles, S must have lower priority than M_i or we would simply cancel M_i and start it over. In accordance with the above provisions, S then imposed restraint on $P \upharpoonright \phi_i(x)\,[v]$ as soon as $l(i, v) \geq x$.

Note that, by our definition of Δ_i, $\delta_i(x)[v] = v > \phi_i(x)[v]$. As long as x is not cancelled, this restraint will not be violated. Thus the only way $\phi_i(x)$ can change between v and t is for $W_i \upharpoonright \phi_i(x)$ to change. When this happens we redefine Δ_i at i-expansionary stages so as to keep $\delta_i(x) \geq \phi_i(x)$.

Now, before we return to M_i at an i-expansionary stage there is a stage r, $t < r \leq s$, at which we deal with the requirement T_0. At stage r, we must redefine $\Gamma_0(G_0 \oplus P; x)$. Of course, x has not yet gone into Q and so we will set $\Gamma_0(G_0 \oplus P; x) = 0$; but, we will define the functional by an axiom of length $\gamma_0(x) = r > \phi_i(x)\,[t]$. When we now finally return to M_i at the first i-expansionary stage $s > t$ and see that $\Delta_i(W_i; x) \downarrow \neq G_0(x) = 1\ [s]$, we are ready to kill M_i. Note that $W_i \upharpoonright \phi_i(x)[t]$ has not changed or we would not have $\Delta_i(W_i; x) \downarrow$ and a failure at x. (The point here is again that $\delta_i(x)[t] \geq \phi_i(x)[t]$ by our maintaining this inequality whenever $W_i \upharpoonright \phi_i(x)$ changes. A change in P would cause us to cancel x.) We put x into Q, r into P and preserve $R \upharpoonright \theta_i\phi_i(x)[s]$ and $P \upharpoonright \phi_i(x)[s]$ by initializing lower priority requirements. As long as this restraint is not violated $l(i, v) < x$ and there are no more i-expansionary stages. (As $R \upharpoonright \theta_i\phi_i(x)$ does not change, $W_i \upharpoonright \phi_i(x)$ cannot change without forever preventing $l(i, v)$ from getting above x because of the clause requiring that $\Theta_i(R; w) \downarrow = W_i(w)$ for every $w \leq \phi_i(x)[s]$. Moreover, no numbers enter $P \upharpoonright \phi_i(x)[s]$ by our initialization procedure and so if $W_i \upharpoonright \phi_i(x)[s]$ also does not change, $\Phi_i(W_i \oplus P; x)$ remains constant at its value at t which was necessarily 0 as x was not in Q at that time. Thus our preservation guarantees that either $\Phi_i(W_i \oplus P; x) \neq Q(x)$ or $\Theta_i(R; w) \neq W_i(w)$ for some $w \leq \phi_i(x)$, i.e. there are no more i-expansionary stages.)

Finally, we can correctly redefine $\Gamma_0(x)$ at the next stage because we put $r < \gamma_0(x)$ into P. Thus if Δ_i ever fails to correctly compute G_0 from W_i, we can kill off the requirement M_i and still correct the functionals of higher priority. Of course, we can only follow this plan of action if no requirement of higher priority than M_i restrains r from entering P.

To describe the action for a requirement M_i in general, suppose that the sets $G_0, \ldots G_n$ ($n > 0$) are the ones that have been introduced by requirements T_j for $j \leq n$ of higher priority than M_i. As mentioned above we will be constructing various functionals $\Delta_{i,j}$ and attempting to see that, for some j, $\Delta_{i,j}(W_i) =^* G_j$. We say that a functional $\Delta = \Delta_{i,j}$ *fails* at a number x at stage s if $\Delta_{i,j}(W_i; x) \downarrow \neq G_j(x)\,[s]$. This *failure* remains *active* at stages $t > s$ as long as there is no change in $W_i \restriction \delta_{i,j}(x)[s]$.

At stages s which are not i-expansionary, we take no action for M_i but we do maintain restraint on $P \restriction u$ where u is the last i-expansionary stage. At i-expansionary stages we drop M_i's P restraint and proceed for $j = 0$ by trying to build a functional $\Delta_{i,0}$ such that $\Delta_{i,0}(W_i) =^* G_0$. Suppose s is an i-expansionary stage. We begin the construction with a *chit* for each number y. We will use these chits to keep track of the failures of functionals $\Delta_{i,j}$. We first find the least $x < l(i,s)$ such that $\Delta_{i,0}(W_i; x) \uparrow$ and the least $y < l(i,s)$ such that the *chit* for y has not been assigned to any $\Delta_{i,0}$ computation and has not been cancelled. Assign the chit for y to $\Delta_{i,0}(W_i; x)$ and set $\Delta_{i,0}(W_i; x) = G_0(x)[s]$ with use $\phi_i(y)[s]$. This chit for y can be assigned to various computations $\Delta_{i,j}(W_i; z)$ subject to the provisos that at any given stage it is assigned to at most one such computation for any single j and it cannot be assigned to one for $j+1$ until the one for j to which it is assigned fails. The chit will be cancelled if the failure of $\Delta_{i,0}$ at x becomes inactive. Moreover, we will insure that the use $\delta_{i,j}(z)[t]$ of any convergent computation to which it is assigned at t is $\phi_i(y)[t]$ for any $t \geq s$ until y enters the set for which it is targeted. Thus if the chit becomes inactive, any computation associated with a chit for x becomes undefined (and so can be corrected at the next stage if necessary).

If $n > 1$, we proceed for $j = 1$ much as we did above for $j = 0$ except that we must restrict our actions by the availability of chits. If there is no active uncancelled chit which is assigned to some failure of $\Delta_{i,0}$ but not to any $\Delta_{i,1}$ computation, we terminate this substage of stage s. If there is such a chit, we assign the least such chit to the computation $\Delta_{i,1}(W_i; x)$ where x is least such that $\Delta_{i,1}(W_i; x) \uparrow$. If there is now a failure of $\Delta_{i,1}$ whose chit (for some number y) has not been assigned to any $\Delta_{i,2}$ computation, we continue on with $j = 2$. Note that if we assign the chit for y to $\Delta_{i,1}(W_i; x)$, then $y > x$. Thus, for example, $x < \gamma_1(y)$ and so when we put x into G_1 we can change $\gamma_1(y)$ as well as $\gamma_1(x)$ to be larger than $\phi_i(y)$.

At any point we may cancel the chit for a number y because of certain changes in P that would interfere with our correcting higher priority Γ_k should we put y into Q. If some x enters G_j at a stage t when the chit for y is assigned to a computation $\Delta_{i,j}(W_i; x)$ and at a later i-expansionary stage $s > t$, before y has gone into Q, $\phi_i(y)[s] > t$ then we cancel the chit for y (and so it can never be assigned to any other computation and y can never enter Q).

If we reach a point where there are no unassigned active chits to pass on to the next level of functionals, we terminate this substage of stage s. Otherwise, we reach the functional $\Delta_{i,n}$. If there is a failure of this functional at x and the (uncancelled, active) chit for y is assigned to $\Delta_{i,n}(W_i; x)$ and our action would not violate any higher priority P restraint, then we act to kill M_i. We put y into Q, $\gamma_j(y)$ into P for each $j \leq n$ and preserve $R\upharpoonright\theta_i\phi_i(y)$ and $P\upharpoonright\phi_i(y)$ by initialization. The point to verify is that $\gamma_j(y) > \phi_i(y)$ for each $j \leq n$.

4. $K_{e,x}$: If there are infinitely many s such that $\Xi_e(R \oplus P; x) \downarrow [s]$, then $\Xi_e(R \oplus P; x) \downarrow$.

Here $\{\Xi_e\}$ lists all partial recursive functionals and includes various ones appearing in the construction with the approximations given in the construction. In particular, it includes $\Delta_{i,j}(W_i; x)$ and the function of i, j, n which converges when there are n failures of the functional $\Delta_{i,j}$ at an i- expansionary stage. When we consider such functionals as included in the Ξ_e we intend their uses (from R) to include the Θ_i uses needed to compute the relevant Φ_i uses on W_i. These requirements are not explicit in the statement of the theorem but they are used, for example, to prevent the uses $\phi_i(y)$ of the chit assigned to the computation (and so the associated P restraint) from going to infinity. They also play a role in maintaining conditions of the form $\gamma_k(x) > \phi_i(y)$ discussed above and making the functionals Γ_k total. The requirement $K_{e,x}$ simply attempts to preserve $P\upharpoonright\xi_e(y)$ and $R\upharpoonright\xi_e(y)$ by initialization.

We now turn to the formal construction of our sets L, P, Q, R and G_i.

4.2. Construction

We begin by listing all the requirements $T_i, D_{i,j,e}, M_i, K_{e,x}$ in a priority list of order type ω such that T_i and T_j appear on the list before any $D_{i,j,e}$. Each stage s of our construction will have at most s many substages n at which we may deal with requirement n. Each requirement may put numbers into some sets, axioms into a functional and impose restraint on various sets in the usual fashion. Whenever a requirement puts numbers into a set we initialize all requirements of lower priority. Initialization of a requirement at stage s means that all followers, chits and all restraint associated with the initialized requirement are cancelled and no further attempts are made at maintaining the correctness at numbers less than s of any functionals it is constructing. (As we only care about the functional being correct almost everywhere, this will not cause problems as long as the requirement is initialized only finitely often.) As new followers are always chosen larger than any number previously used in the construction, initialization also acts to preserve various computations. At other times we may cancel chits because the situation that made them usable has been destroyed. We now describe our actions at substage n of stage s of the construction according to the type of the nth requirement on our list. We let r be the last stage at which the requirement being considered at substage n was injured ($r = 0$ if there is no such stage). Material enclosed in double brackets [[]] is motivational only and not part of the formal construction. When we choose a "*large*" number at a stage s we mean a number larger than any used in the construction before s. We use the style of i-expansionary stage restraint on P as in the minimal pair construction in [35, IX.1].

1. $T_i : G_i \oplus P \geq_T Q$.

Find the least $x > r$ such that $\Gamma_i(G_i \oplus P; x) \uparrow [s]$ and set $\Gamma_i(G_i \oplus P; x) = Q(x)[s]$ with large use $\gamma_i(x)$. [[That Γ_i will be defined almost everywhere will follow from the type 4 requirements below. That it will be correct almost everywhere is tied to the action of the type 3 requirements which put numbers into Q and will require proof below.]] We now continue on to the next substage of stage s.

2. $D_{i,j,e} : \Psi_e(G_i) \neq G_j$.

If this is the first stage at which we deal with this requirement since r, we choose a new large follower x for $D_{i,j,e}$ from the column $\omega^{\langle 0,i,j,e\rangle}$ associated with $D_{i,j,e}$. At every stage $t \geq s$ until x is cancelled or enters G_j, $D_{i,j,e}$ imposes restraint $r(D, i, j, e, t)$ on P where

$$r(D, i, j, e, t) = \max\{\delta_{k,j}(W_k; x)[t] |\, M_k \text{ has higher priority than } D_{i,j,e}\}.$$

[[Our definition of $\Delta_{i,j}$ will make this equivalent to $\max\{\phi_k(z)[t] |\, M_k$ has higher priority than $D_{i,j,e}$ and there is a computation of $\Delta_{k,j}(W_k; x) \downarrow [t]$ which is associated with a chit for z.$\}$]] This restraint is dropped when x is cancelled by some higher priority requirement or enters G_j. [[This restraint can be violated only when we act for a type 3 requirement of higher priority in which case we would initialize $D_{i,j,e}$ and all lower priority requirements.]]

If $D_{i,j,e}$ has an uncancelled follower x and $\Psi_e(G_i; x) \downarrow= 0$, we say *the follower is realized.* If the realized follower is not yet in G_j, we put it into G_j, initialize all lower priority requirements and terminate stage s. [[This has the effect of preserving $G_i \upharpoonright \psi_e(x)$ with priority $D_{i,j,e}$.]] Otherwise, we continue on to the next substage of stage s.

3. M_i: If $\Theta_i(R) = W_i$ and $\Phi_i(W_i \oplus P) = Q$, then $G_j \leq_T W_i$ for some j such that T_j has higher priority than M_i.

We first define a length of agreement function and i-expansionary stages as above:

$$\begin{aligned} l(i,u) &= \mu z\{\neg[\Phi_i(W_i \oplus P; z) \downarrow= Q(z)\ [u] \\ \wedge (\forall w &\leq \phi_i(z))(\Theta_i(R; w) \downarrow= W_i(w)\ [u]\,)]\,\}. \end{aligned}$$

We then define the notion of an *e-expansionary* stage and the *restraint* $r(M,e,u)$ imposed on P by M_e by induction on $e \leq i$ as in [37, IX.1]. If $u = 0$ or $l(0,u)$ is larger than $l(0,t)$ for every $t < u$, the stage u is 0-expansionary and $r(M,0,u) = 0$. Otherwise, u is not 0-expansionary and $r(M,0,u)$ is the largest number used in the construction up through the last 0-expansionary stage. The stage u is $(e+1)$-expansionary if $l(e+1,u) > l(e+1,t)$ for every $t < u$ for which $r(M,e,t) = r(M,e,u)$. We let $r(M,e+1,u)$ be the maximum of $r(M,e,u)$, any number used at a stage $t < u$ at which $r(M,e,t) < r(M,e,u)$ and, if u is not $(e+1)$-expansionary, any number used at any $(e+1)$-expansionary stage $t < u$ for which $r(M,e,t) = r(M,e,u)$. Now we can describe our action for M_i at stage s.

If s is not i-expansionary or M_i has put a number into Q since r, we go on to the next substage of stage s. Otherwise, suppose that $T_0, \ldots, T_n$ are the requirements of type 1 of higher priority than M_i (n is really n_i but we drop the subscript for this description). We are constructing functionals $\Delta_{i,j}$ with the expectation that, for some $j \leq n$, $\Delta_{i,j}(W_i) =^* G_j$. We say that a functional $\Delta_{i,j}$ *fails* at a number x at stage s if $\Delta_{i,j}(W_i; x) \downarrow\neq G_j(x)\,[s]$. This *failure* remains *active* at stages $t > s$ as long as there is no change in $W_i \upharpoonright \delta_{i,j}(x)[s]$. At the start of the construction, there is a *chit* (for M_i) for each number y. We will assign these chits to various computations of the functionals $\Delta_{i,j}$ and use them to keep track of the failures of these functionals.

We begin with $j = 0$ by finding the least $x > r$ such that $\Delta_{i,0}(W_i; x) \uparrow$ [[necessarily $x < l(i,s)$]]. Let y be the least number $< l(i,s)$ such that the chit for y is not assigned to any computation of $\Delta_{i,0}$ and has not been cancelled. (If there is no such y we go on to the next substage of stage s.) We assign the chit for y to $\Delta_{i,0}(W_i; x)$ and set $\Delta_{i,0}(W_i; x) = G_0(x)[s]$ with use $\phi_i(y)[s]$. This chit for y can be assigned to various computations $\Delta_{i,j}(W_i; z)$, $0 < j \leq n$, subject to the provisos that at any given stage it is assigned to at most one such computation for any single j and it cannot be assigned to one for $j+1$ until the one for j to which it is assigned fails. The chit will be cancelled if the failure of $\Delta_{i,0}$ at x becomes inactive or M_i is initialized. It may also be cancelled at stages where certain changes occur in P on computations associate with the chit. [[Moreover, we will insure that, unless cancelled, the use $\delta_{i,j}(z)[t]$ of any convergent computation to which the chit is assigned at t is $\phi_i(y)[t]$ for any $t \geq s$. Thus if the chit for y becomes inactive any computation at a number larger than the last initialization of M_i to which the chit for y is assigned becomes undefined, and so can be corrected at the next stage.]]

If $n > 1$, we proceed for $j > 0$ much as we did above for $j = 0$ except that we must restrict our actions by the availability of chits. If every uncancelled chit which is assigned to some failed computation of $\Delta_{i,j-1}$ has been assigned to some $\Delta_{i,j}$ computation, we terminate this substage of stage s. If there is such an unassigned chit, we assign the least such chit to the computation $\Delta_{i,j}(W_i; x)$ where x is least such that $\Delta_{i,j}(W_i; x) \uparrow$. If there is now a failure of $\Delta_{i,j}$ whose chit (for some number y) has not been assigned to any $\Delta_{i,j+1}$ computation, we continue on with $j+1$. [[Note that if we assign the chit for y to $\Delta_{i,j}(W_i; x)$, then $y > x$. Thus, for example $x < \gamma_1(y)$ and so when we put x into G_1 we can change $\gamma_1(y)$ as well as $\gamma_1(x)$ to be larger than $\phi_i(y)$.]]

If we assign the chit for y to a computation $\Delta_{i,j}(W_i; x)$ at s, x enters G_j at $t \geq s$ and we return to M_i at an i-expansionary stage $u \geq t$ such that $\phi_i(y)[u] > t$, then we cancel the chit for y [[and so y will never enter Q]].

If we reach a point where there are no appropriate active chits to pass on to the next level of functionals, we terminate this substage of stage s. Otherwise, we reach the functional $\Delta_{i,n}$. If there is a failure of this functional at x and the uncancelled chit for y is assigned to $\Delta_{i,n}(W_i; x)$ and y is larger than any restraint of higher priority on P (i.e. $y > r(D,k,l,m,s), r(M,k,s)$ for every $D_{k,l,m}$ and M_k of higher priority than M_i), then we act to kill M_i: We put y into Q, $\gamma_j(y)$ into P for each $j \leq n$ and preserve $R \upharpoonright \theta_i\phi_i(y)$ and $P \upharpoonright \phi_i(y)$ by initializing all requirements of lower priority than M_i and terminate stage s. [[The point to verify will be that $\gamma_j(y) \geq \phi_i(y)$ for each $j \leq n$.]]

4. $K_{e,x}$: If there are infinitely many s such that $\Xi_e(R \oplus P; x) \downarrow [s]$, then $\Xi_e(R \oplus P; x) \downarrow$.

Here $\{\Xi_e\}$ lists all partial recursive functionals and includes various ones appearing in the construction with the approximations given in the construction. In particular, it includes the ones $\Gamma_i(G_i \oplus P; x)$, $\Delta_{i,j}(W_i; x)$ and the function of i, j, n which converges when there are n failures of the functional $\Delta_{i,j}$ at an i-expansionary stage associated with uncancelled chits. When we consider such functionals as included in the Ξ_e we intend their uses (from R) to include the Θ_i uses needed to compute the relevant Φ_i uses on W_i as well as the information directly used about P, R and any G_k viewed as a component of R.

If $\Xi_e(R \oplus P; x) \downarrow$ for the first time since $K_{e,x}$ was last initialized, we initialize all requirements of lower priority. [[This has the effect of imposing restraint on $P \upharpoonright \xi_e(x)[s]$ and on $R \upharpoonright \xi_e(x)[s]$ and hence, for example, on $W_i \upharpoonright \delta_{i,j}(x)[s]$ for the appropriate e.]] In any case, continue on to the next substage of stage s.

At the end of substage s, we terminate stage s (if it has not been terminated before). When stage s is terminated, we go on to stage $s + 1$.

4.3. Verifications

First, note that (other than through initialization) the only restraint imposed in the construction is on P. Moreover, by definition of the e-expansionary stages and the restraint functions $r(M, e, u)$, the restraint $r(M, i, s)$ imposed on P by requirements M_j of priority greater than or equal to that of M_i have a finite $\liminf$ which is realized on the i-expansionary stages if there are infinitely many such stages and, if not, on almost all the stages at which $r(M, i-1, s)$ realizes its $\liminf$. We are proving by induction that each requirement succeeds and that those of type 2 impose at most finite restraint. (Of course, requirements of type 1 never impose any restraint.) It is immediate from the definition of their actions that if a requirement of type 2,3 or 4 is never initialized after stage r, it acts at most once after r to put numbers into sets and to initialize lower priority requirements. As the other requirements never initialize anything, it is obvious by induction that each requirement has a stage after which it is never initialized. Let us consider the fate of any requirement which is last initialized at stage r.

4. If $\Xi_e(R \oplus P; x)$ never converges after stage r, we never act for $K_{e,x}$ and

it is satisfied. If $\Xi_e(R \oplus P; x)$ converges for the first time at $s > r$, it remains convergent at every later stage. The point is that $K_{e,x}$ initializes all requirements of lower priority and so none can put any elements below the use into R or P. No requirement of higher priority can put a number into R or P as that would initialize $K_{e,x}$ contrary to our choice of r. Thus the $K_{e,x}$ succeed.

In particular, $\lim \delta_{i,j}(x)$ exists and is finite (possibly 0 if $\Delta_{i,j}(W_i; x) \uparrow$) for every i, j, x. Thus the restraint imposed by any particular follower x of a requirement $D_{m,n,e}$ is bounded by $\lim \max\{\delta_{k,l}(x)[t] | \ M_k$ has higher priority than $D_{m,n,e}\}$. (Note that the set of relevant $\delta_{k,l}$ is finite as $l \leq n_k$ and each M_k must have higher priority than $D_{m,n,e}$.) As only one follower is ever appointed for such a requirement after it is last initialized, the restraints of this type are bounded for each such requirement.

2. $D_{i,j,e}$: At stage $r+1$, $D_{i,j,e}$ appoints some large follower x. If the follower x is never realized, the requirement is satisfied. If it is realized at a stage t, we put x into G_j and initialize all lower priority requirements. Now, no higher priority requirement can put a number below $\psi_e(x)[t]$ into G_i as this would initialize $D_{i,j,e}$ contrary to our assumption. No lower priority argument can put any number below $\psi_e(x)[t]$ into G_i after stage t as all such numbers are appointed as followers after stage t and so are larger than $\psi_e(x)[t]$. (The only requirements which can put numbers into G_i are of type 2 and these always choose new followers larger than any number previously used in the construction.) Thus $D_{i,j,e}$ is satisfied. The argument in the proceeding paragraph shows that the P restraint imposed by $D_{i,j,e}$ now goes to a limit.

1. T_i: Consider each $x > r$ in turn. It is clear from the construction that we build a functional Γ_i such that $\Gamma_i(G_i \oplus P; x) \downarrow [t] = Q(x)[t]$ for infinitely many t. The success of the requirements $K_{e,x}$ now guarantees that Γ_i is defined for all $x > r$. Now, only requirements of type 3 put numbers into Q and when they put some y into Q they put $\gamma_k(y)$ into P for every T_k of higher priority. Thus at any later stage we can correctly redefine $\Gamma_k(G_k \oplus P; y)$. At all other stages this functional is obviously correct when defined and so $G_i \oplus P \geq_T Q$ as required.

3. M_i: If the hypotheses of M_i ($\Theta_i(R) = W_i$ and $\Phi_i(W_i \oplus P) = Q$) are satisfied, there are infinitely many i-expansionary stages at which the P restraint of higher priority is constant. If the hypotheses are not satisfied, the success of the requirements $K_{e,x}$ guarantee that $\lim l(i, s) < \infty$ and so we eventually stop

all action for M_i and in particular, the restraint it imposes is eventually constant.

We begin by supposing that the hypotheses are satisfied and prove a series of lemmas.

Lemma 4.2. *$\Delta_{i,0}(W_i; x)$ is defined for every $x > r$.*

Proof. If not, let x be the least counterexample and s a stage by which $\Delta_{i,0}$ has settled down for every z, $r < z < x$. It is clear from the construction and our hypotheses that $\Delta_{i,0}(W_i; x)[t] \downarrow$ for infinitely many $t > s$. As this corresponds to some $\Xi_e(R \oplus P; x)[t] \downarrow$, the success of the requirements of type 4 guarantees that $\Delta_{i,0}(W_i; x) \downarrow$. $\Box$

Lemma 4.3. *Suppose $\Delta_{i,j}(W_i; x)$ is defined for almost every x. If there are infinitely many x such that $\Delta_{i,j}(W_i; x) \neq G_j(x)$, then there are infinitely many x such that the computation of the failure of $\Delta_{i,j}$ at x is associated with a chit which is never cancelled and so, if $j < n_i$, there are infinitely many chits which are never cancelled and are available for assignment to $\Delta_{i,j+1}$.*

Proof. We proceed by induction on n and so assume there are $n - 1$ many failures of $\Delta_{i,j}$ with computations associated with chits which are never cancelled. Let x be a larger failure point for $\Delta_{i,j}$. Consider the stage t at which x enters G_j. Suppose $\Delta_{i,j}(W_i; x)$ was last defined at (a necessarily i-expansionary) $s < t$ with $\delta_{i,j}(x)[s] = \phi_i(y)[s]$ where the chit for y was assigned to this computation of $\Delta_{i,j}(W_i; x)$. We can later get our failure at x only if x is already a follower of some $D_{k,j,e}$ of lower priority than M_i. Thus the restraint associated with this requirement after stage s is at least $\delta_{i,j}(x)[s] = \phi_i(y)[s]$. If P changes below this restraint before x enters G_j, x would be cancelled (as P can change only if a requirement of higher priority puts a number in and so initializes $D_{i,j,e}$) and so would never enter G_j and never produce a failure of $\Delta_{i,j}$. Thus when x enters G_j at $t \geq s$, $P \upharpoonright \phi_i(y)[s]$ has not changed since the last stage at which $\Delta_{i,j}(W_i; x)$ was defined. Now no number enters P at t because only one requirement can put numbers into $R \oplus P$ at any stage. Let u be the first i-expansionary stage after t. No number below $\phi_i(y)[s]$ can enter P at any v between t and u by the definition of the restraint $r(M, i, v)$ as M_i can no longer be initialized. Thus when we return to M_i at stage u, we have a failure of $\Delta_{i,j}$ at x which has not been cancelled and so seem to have $W_i \oplus P$ correct on the use $\phi_i(y)[u]$. Our assumption guarantees that this happens for infinitely many x. Consider now the functional $\Xi_e(R \oplus P; k)$

which gets defined at s if there are k many failed computations of $\Delta_{i,j}$ which are associated with chits whose computations are $R \oplus P$ correct at s. Once again, the success of requirements of type 4 and our assumptions guarantee that this functional is defined at n. As (after M_i is never initialized) a chit for y such that $R \upharpoonright \theta_i\phi_i(y)$ and $P \upharpoonright \phi_i(y)$ are correct is never cancelled, we have the existence of the desired nth instance of a failed computations of $\Delta_{i,j}$ associated with a chit which is never cancelled. □

Lemma 4.4. *If $j < n_i$ and there are infinitely many chits which are never cancelled and available for assignment to $\Delta_{i,j+1}$, then $\Delta_{i,j+1}$ is defined for every $x > r$.*

Proof. The argument is like the one above for $\Delta_{i,0}$. By construction, we must infinitely often define $\Delta_{i,j+1}$ at some x as there are new chits available infinitely often. Thus, for each x at which we can try to define $\Delta_{i,j+1}$ infinitely often (i.e. each $x > r$), the functional is actually defined by the success of requirements of type 4. □

Lemma 4.5. *If $\Delta_{i,j}(W_i) \neq^* G_j$ for every $j \leq n_i$ then M_i eventually puts a number into Q.*

Proof. Arguing by induction and using the above Lemmas, we see that Δ_{i,n_i} is defined for every $x > r$. If there were infinitely many failures of Δ_{i,n_i}, there would be infinitely many whose associated chit is never cancelled. So, in particular, there would be an x associated with such a chit y which is larger than the $\liminf$ of the restraint of higher priority than M_i imposed on P. It is clear by construction that we would then eventually put one such y into Q. □

Finally, we conclude the verification of the success of M_i with the following lemma.

Lemma 4.6. *If M_i ever puts a number y into Q after stage r, then $\lim l(i,s) < \infty$ and so M_i is satisfied. (Of course, in this case there are only finitely many i-expansionary stages and the restraint imposed by M_i is constant after the last one.)*

Proof. Suppose it does so at stage s. We first claim that no number used in the computations associated with y at s can ever be put into R or P. As no higher priority requirement ever puts a number into $R \oplus P$ by our choice of r and no lower priority one puts one in which is used in the computations at s by initialization, it suffices to prove that each $\gamma_k(y)[s]$ put into P at s is larger than these uses from P. In order for us to put y into Q at s, we must have a chit for y which is uncancelled and associated with failed computations $\Delta_{i,j}(W_i; x_j) = 0 \neq G_j(x)$ for each $j \leq n_i$. Consider the stage $s_j < s$ at which x_j was put into G_j by some requirement of type 2 of lower priority than M_i (as otherwise $s_j < r$ and so the chit for y would have been cancelled at r). Suppose $\Delta_{i,j}(W_i; x_j)$ was last defined before s_j at the i-expansionary stage t_j with a chit for y and use $\phi_i(y)[t_j]$. At some stage $u_j > s_j$ we redefine $\Gamma_j(W_j \oplus P; y)$ with a large use and so one larger than $\phi_i(y)[t_j], t_j$. If there were any later i-expansionary stage (including s) at which $\phi_i(y)$ increased above this Γ use, we would cancel the chit for y by construction and so it could not later go into Q as assumed. Of course, as time goes by $\gamma_j(y)$ is nondecreasing. Thus when we put y into Q, $\gamma_j(y) > \phi_i(y)$ as required. □

This completes the proof that each requirement succeeds and so we have constructed the required sets. □

4.4. Codings

We now describe the additional requirements needed to code enough relations on the definable set of degrees $\langle \mathbf{g}_i \rangle$ constructed above to get a definition of arithmetic on the defined set of degrees and so the undecidability of $\mathcal{R}$. Although other codings are possible (such as graphs), we follow Slaman and Woodin's original plan and code an arbitrary recursive partial ordering $\mathcal{P} = \langle \omega, \preceq \rangle$ on the $\mathbf{g}_i$ by constructing an additional set L and adding requirements that guarantee that $G_j \leq_T G_i \oplus L \Leftrightarrow j \preceq i$. For the "only if" direction, we adjust the construction so that any number x put into G_j is also simultaneously put into G_i or L. In particular, when a requirement $D_{i,j,e}$ puts a number x into G_j it also puts x into L. For the "if" direction, we add in requirements of type 5.

5. $N_{i,j,e}$: If i ffl j then $\Psi_e(G_i \oplus L) \neq G_j$.

This requirement is met by a Friedberg-Muchnik type strategy like that used for requirements $D_{i,j,e}$. The only additional features are the ones necessitated by the coding for the positive ordering facts as described above. When a new large follower x is appointed, $N_{i,j,e}$ imposes restraint $r(N, i, j, e, t)$ on P that takes into account the fact that a witness x targeted for G_j will also have to be put into G_l for $l \succeq j$:

$$r(N, i, j, e, t) = \max\{\delta_{k,l}(W_k; x)[t] |\ M_k \text{ has higher priority than } N_{i,j,e} \text{ and } l \succeq j\}.$$

[[Our definition of $\Delta_{i,j}$ will make this equivalent to $\max\{\phi_k(z)[t] |\ M_k$ has higher priority than $N_{i,j,e}$, $l \succeq j$ and there is a computation of $\Delta_{k,l}(W_k; x) \downarrow [t]$ which is associated with a chit for $z.\}$]] This restraint is dropped when x is cancelled by some higher priority requirement or enters G_j. If $N_{i,j,e}$ has an uncancelled follower x and $\Psi_e(G_i; x) \downarrow = 0$, we say the follower is realized. If the realized follower is not yet in G_j, we put it into G_l for every $l \succeq j$, initialize all lower priority requirements and terminate stage s. The argument that these requirements are satisfied and impose only finite restraint on the rest of the construction is again exactly as for the $D_{i,j,e}$. (A more detailed version of the treatment of such requirements is presented in the construction for, and verification of, Theorem 5.1.)

4.5. Permitting

The next extensions of the basic construction with the coding apparatuses described above involve permitting. First, we can make the sets G_i and L constructed recursive in any given nonrecursive r.e. set B. The requirements $D_{i,j,e}$ and $N_{i,j,e}$ now wait for a realized follower x which is permitted by B. While all followers are realized but none has been permitted, they appoint new followers (initializing all lower priority requirements) and wait for them to be realized (again initializing all lower priority requirements). As B is nonrecursive, they either get a follower which is never realized or eventually put a realized follower into the appropriate sets. In any case they act only finitely often and are eventually satisfied. (Again a more detailed argument for a similar requirement is presented in Theorem 5.1.)

The situation for the M_i requirements and so for the sets P and Q is somewhat different. The restraint imposed on P by higher priority M_j only has finite lim inf. Thus coordinating action by M_i to put numbers into Q and so P with permitting by an arbitrary nonrecursive r.e. B requires B permission on expansionary stages. There is no reason to believe that this is possible. Indeed, in the analogous situation for minimal pairs it is known that there are nonrecursive r.e. B below which it is not possible to construct a minimal pair. Thus we view the problem of constructing Slaman-Woodin definable sets as in Theorem 4.1 below an arbitrary r.e. degree as an open question.

If, however, the given set recursively in which we wish to construct P and Q is a promptly simple set S rather than an arbitrary one B, the usual arguments (as in [3]) work. Thus if S is promptly simple, we can let M_i put a number y into Q (and so various $\gamma_j(y)$ into P) only when y is permitted by S by using the function p witnessing the prompt simplicity of B. As we saw in the proof of Lemma 4.5, if we are reduced to the case that all the $\Delta_{i,j}$ fail then there are infinitely many y's which remain permanently eligible to enter Q once S permits. We let M_i impose restraint to preserve the eligibility of such numbers as they become candidates for M_i to put into Q. We then enumerate these numbers into an auxiliary set when we next have an M_i expansionary stage. At each such stage s we enumerate S up to stage $p(t)$ to see if it permits the number to go into Q. (Here t is the stage at which these numbers are enumerated in a standard enumeration of the auxiliary set given by the recursion theorem.) As usual, since S is promptly simple, one will eventually be allowed to enter Q. Of course, the numbers $\gamma_j(y)$ going into P are larger than y and so both P and Q are permitted by S as required.

We can combine all of these modifications to construct a Slaman-Woodin set below any nonrecursive B along with the coding apparatus for a partial ordering. Moreover, if S is promptly simple, we can get the auxiliary sets P and Q needed to define the Slaman-Woodin set to be below S as well:

Theorem 4.7. *Given a recursive partial ordering $\mathcal{P} = \langle \omega, \preceq \rangle$, a nonrecursive r.e. set B and a promptly simple one S, there are r.e. sets L, P, Q, R and G_i for $k < n$ and $i \in \omega$) with $R = \oplus G_i, L \leq B$ and $P, Q \leq_T S$ with the following properties:*

1. *$(T) : G_i \oplus P \geq_T Q$.*
2. *$(D) : G_i \; \Lambda_T \; G_j$ for $i \neq j$.*
3. *(M) : If W is r.e., recursive in R and $W \oplus P \geq_T Q$, then there is a $j \in \omega$ such that $G_j \leq_T W$.*
4. *$R \oplus P$ is low. (Indeed we can just as easily make the join of all the sets constructed low.)*
5. *$(N) : i$ ﬄ $j \Leftrightarrow G_i \oplus L \; \Lambda_T \; G_j$. $\square$*

5. Cone Avoiding and Comparisons

The next modification we wish to consider is cone avoiding combined with permitting. More specifically, given various $U_i \; \Theta_T \; V_j$ with V_j low, we wish (under certain other assumptions) to make $G_i \leq_T U_i$ and $G_i \,\Theta_T\, V_j$. We first briefly describe the necessary additions to the basic construction for Theorem 4.1 to handle one triple G_i, U_i, V_i. To begin, we will only allow numbers x to enter G_i when permitted by U_i and so will follow the procedure described above for all requirements $D_{k,i,e}$ putting numbers into G_i so that they are permitted by U_i. The new requirements of type 6 are as follows:

6. $Z_{i,e}$: $\Psi_e(V_j) \neq G_i$.

The basic plan is as for the diagonalization requirements D of type 2 with permitting. We plan to appoint suitable followers x; preserve their suitability by imposing P restraint; and look for one which is realized ($\Psi_e(V_j; x) = 0$ at the current stage) that we are permitted to put into G_i. To guarantee that we eventually put in a follower such that $\Psi_e(V_j; x)$ really is 0, our diagonalization procedure will include a two step guessing procedure exploiting the lowness of V_j and taking into account that we need U_i permission to put the desired V_j-correctly realized follower into G_i.

To hope to be able to handle a list of such triples G_i, U_i, V_j we have some obvious restrictions that we want to impose. First of all, as our notation indicates, we do not want to have to build a single G_i below more than one prescribed set U_i. The next issue arises because we want to combine these requirements with the coding apparatus (for partial orderings) described above. The problem here is that the coding procedures require that when we put some x into a G_i for a diagonalization requirement, we must also put it into L (which we do not need to permit) or into G_k for $k \succeq i$. If we are required to put it into another G_k which must be permitted by some U_k we have hit the first (essentially insurmountable) problem. Thus we restrict our attention to codings of partial orderings and to requirements of type 6 for G_i which are minimal element of the given ordering. In this situation, no requirements other than those directly diagonalizing against G_i put numbers into G_i and so we have no worries about multiple permitting. We state and prove a theorem of which Lemma 2.6 (needed to define the comparison maps) is clearly a special case.

Theorem 5.1. *Suppose $\mathcal{P} = \langle \omega, \preceq \rangle$ is a recursive partial order; H is a recursive set of minimal elements of $\mathcal{P}$; $\langle U_i, V_{i,j} \rangle_{i \in H}$ is a uniformly r.e. array of pairs of sets such that the $V_{i,j}$ are uniformly low (i.e. we can recursively in i, j calculate an index for computing $V'_{i,j}$ from $\emptyset'$); and, for every $i \in H$, $U_i \,\Theta_T\, V_{i,j}$ (and so in particular $U_i > 0$ for $i \in H$). Then there are r.e. sets L, P, Q, R, and G_i (for $i \in \omega$) with $R = \oplus G_i$ such that*

1. $(T) : G_i \oplus P \geq_T Q$.
2. $(D) : G_i \Lambda_T G_j$ *for* $i \neq j$.
3. (M) : *If* W *is r.e., recursive in* R *and* $W \oplus P \geq_T Q$, *then there is a* $j \in \omega$ *such that* $G_j \leq_T W$.
4. $(K) : R \oplus P$ *is low.*
5. $(O) : i \succeq j \Rightarrow G_i \oplus L \geq_T G_j$.
6. $(N) : i$ ﬄ $j \Rightarrow G_i \oplus L \Lambda_T G_j$.
7. $(Z) : G_i \Theta_T V_{i,j}$ *for* $i \in H$.
8. $(Q) : G_i \leq_T U_i$ *for* $i \in H$.

We have already discussed the procedures for all the requirements and so directly give the construction and verification.

5.1. Construction

The basic terminology and procedures are that same as for Theorem 4.1 with the addition of the requirements $N_{i,j,e}$ for i ﬄ j and $Z_{i,j,e}$ for $i \notin H$ to the priority listing of type ω. There are no direct actions for properties 5 (O) and 8 (Q) as these are satisfied by putting numbers entering various G_i into other sets (various G_k or L) and by permitting, respectively. The actions for T_i, M_i and $K_{e,x}$ are exactly the same as in the basic construction. The only change for the $D_{i,j,e}$ with $j \notin H$ is that when one puts an x into G_j one also puts the same x into L. We describe the action for the other requirements at the appropriate substage of stage s. (Remember, r is the stage at which the requirement being considered was last initialized and 0 if there is no such stage.)

2. $D_{i,j,e} : \Psi_e(G_i) \neq G_j$ for $j \in H$.

If we have put a number into G_j since stage r for requirement $D_{i,j,e}$, we continue on to the next substage of stage s. If not, we see if we have a realized follower x of $D_{i,j,e}$ which is permitted by U_j (i.e. some $z \leq x$ entered U_k at s). If so, we put x into G_j and L, initialize all lower priority requirements and terminate stage s. If not, and every uncancelled follower $x_0, \ldots, x_{n-1}$ of $D_{i,j,e}$ is realized, (if there are no followers, $n = 0$), we appoint a new large follower $x_n \in \omega^{\langle 0,i,j,e \rangle}$ of $D_{i,j,e}$, initialize all lower priority requirements and impose P restraint at all $t \geq s$ equal to

$$r(D, i, j, e, x_n, t) = \max\{\delta_{k,j}(W_k; x_n)[t] | \, M_k \text{ has higher priority than } D_{i,j,e}\}.$$

As before, this P restraint is dropped if x_n is cancelled or some follower of $D_{i,j,e}$ enters G_j. At any stage t (until a number is put into G_j for $D_{i,j,e}$) the P restraint imposed by $D_{i,j,e}$ is

$$r(D,i,j,e,t) = \max\{r(D,i,j,e,x_n,t) |\ x_n \text{ is a follower of } D_{i,j,e} \text{ at } t\}.$$

6. $N_{i,j,e}$: If i ffl j then $\Psi_e(G_i \oplus L) \neq G_j$. Case 1: $j \notin H$.

If this is the first stage at which we deal with this requirement since r, we choose a new large follower x for $N_{i,j,e}$ from the column $\omega^{\langle 1,i,j,e\rangle}$ associated with it. As long as x is not cancelled or put into G_j, $N_{i,j,e}$ imposes restraint $r(N,i,j,e,t)$ on P as described above:

$$r(N,i,j,e,t) = \max\{\delta_{k,l}(W_k;x)[t] |\ M_k \text{ has higher priority than } N_{i,j,e} \text{ and } l \succeq j\}.$$

This restraint is dropped when x is cancelled by some higher priority requirement or enters G_j. If $N_{i,j,e}$ has an uncancelled follower x and $\Psi_e(G_i;x)\downarrow= 0$, we say the follower is *realized*. If the realized follower is not yet in G_j, we put it into G_l for every $l \succeq j$, initialize all lower priority requirements and terminate stage s. Otherwise, we continue on to the next substage of stage s.

Case 2: $j \in H$.

If we have put a number into G_j since stage r for requirement $N_{i,j,e}$, we continue on to the next substage of stage s. If not, we see if we have a realized follower x of $N_{i,j,e}$ which is permitted by U_j (i.e. some $z \leq x$ entered U_j at s). If so, we put x into G_l for every $l \succeq j$ (but not into L) initialize all lower priority requirements and terminate stage s. If there is no realized follower which is permitted and every uncancelled follower $x_0, \ldots, x_{n-1}$ of $N_{i,j,e}$ is realized (if there are no followers, $n = 0$), we appoint a new large follower $x_n \in \omega^{\langle 1,i,j,e\rangle}$ of $N_{i,j,e}$, initialize all lower priority requirements and impose P restraint at all $t \geq s$ equal to

$$r(N,i,j,e,x_n,t) = \max\{\delta_{k,l}(W_k;x_n)[t] |\ M_k \text{ has higher priority than } N_{i,j,e} \text{ and } l \succeq j\}.$$

As before, this P restraint is dropped if x_n is cancelled or some follower of $N_{i,j,e}$ enters G_j. At any stage t (until a number is put into G_j for $N_{i,j,e}$) the P restraint imposed by $N_{i,j,e}$ is

$$r(N,i,j,e,t) = \max\{r(N,i,j,e,x_n,t) |\ x_n \text{ is a follower of } N_{i,j,e} \text{ at } t\}.$$

7. $Z_{i,,j,e}$: $\Psi_e(V_{i,j}) \neq G_i$ (for $i \in H$).

We first note that, by the lowness of $V_{i,j}$ and the recursion theorem, we have recursive functions $f(i,j,e,s)$ and $g(i,j,e,s)$ such that

- $\lim_{s\to\infty} f(i,j,e,s)$ and $\lim_{s\to\infty} g(i,j,e,s)$ are each 0 or 1 for every i,j,e with $i \in H$;
- $\lim_{s\to\infty} f(i,j,e,s) = 1$ iff there is a stage t of our construction at which there is a realized follower z of $Z_{i,j,e}$ such that $\Psi_e(V_{i,j}; z) \downarrow = 0$ $[t]$ via a $V_{i,j}$ correct computation and $z \in G_i$; and
- $\lim_{s\to\infty} g(i,j,e,s) = 1$ iff there is a stage t of our construction at which we do not terminate the substage of t dealing with $Z_{i,j,e}$ because of our analysis of the realized followers already in G_i and there is an uncancelled follower z of $Z_{i,j,e}$ such that $\Psi_e(V_{i,j}; z) \downarrow = 0$ $[t]$ via a $V_{i,j}$ correct computation and some $x \leq z$ enters U_i at t.

A follower y of $Z_{i,j,e}$ is realized at s if $\Psi_e(V_{i,j}; y) = 0$ $[s]$. If there is a realized follower y of $Z_{i,j,e}$ in G_i, we find the first $u \geq s$ such that $f(i,e,u) = 1$ or $V_{i,j} \upharpoonright \psi_e(y)[u] \neq V_{i,j} \upharpoonright \psi_e(y)[s]$. If $f(i,e,u) = 1$, we terminate this substage of stage s and go on to the next substage. If $f(i,e,u) \neq 1$, we first check to see if there is a realized follower y of $Z_{i,j,e}$ which is permitted by U_i, i.e. some number $\leq y$ has entered U_i at s. If so we find the least $u \geq s$ such that either $g(i,e,u) = 1$ or $V_{i,j} \upharpoonright \psi_e(y)[u] \neq V_{i,j} \upharpoonright \psi_e(y)[s]$. If $g(i,e,u) = 1$, we put every realized follower y permitted by U_i into G_i, initialize all requirements of lower priority and terminate stage s. If $g(i,e,u) \neq 1$, and this is the first time we have dealt with requirement $Z_{i,j,e}$ since it was last initialized at r (or all followers, $y_0, \ldots, y_{n-1}$, appointed since r are realized or in G_i), we appoint a new large follower y_0 $(y_n) \in \omega^{\langle 2,i,e\rangle}$ of $Z_{i,j,e}$. Whether we appoint a new follower or not, we now go on to the next substage of stage s.

The P restraint imposed by $Z_{i,j,e}$ at stage $t \geq s$ is given by

$$r(Z,i,j,e,t) = \max\{\delta_{l,i}(x)[t] \mid M_l \text{ has higher priority than } Z_{i,j,e} \text{ and } x \text{ is a follower of } Z_{i,j,e} \text{ at } t\}$$

5.2. Verifications

The success of the positive coding requirements (O) is immediate by the construction as is that of the permitting condition (Q). The general format of the rest of the verifications is as for Theorem 4.1 and we are proving the same claims by induction. The arguments for $K_{e,x}, T_i, D_{i,j,e}$ for $j \notin H$ and M_i are exactly as for Theorem 4.1. We thus just give the analyses for the other requirements.

Lemma 5.2. *($D_{i,j,e}$ for $j \in H$): If $D_{i,j,e}$ is never initialized after stage r, there is a stage after which $D_{i,j,e}$ never puts any more numbers into G_j and so never initializes other requirements. Moreover, $D_{i,j,e}$ is satisfied, appoints only finitely many followers and the P restraint it imposes, $r(D,i,j,e,t)$, is eventually constant.*

Proof. If we eventually put a number into G_j for $D_{i,j,e}$ after stage r, then we argue as before that the requirement succeeds and never acts again. Suppose then that we never put a number into G_j for $D_{i,j,e}$. If there is a last follower x_n ever appointed, say at $t > r$, then for one of the followers x appointed after r, $\neg(\Psi_e(G_i; x_k) = 0)$; for if all these computations converged correctly to 0, we would eventually appoint another follower by construction. In this case, $D_{i,j,e}$ is satisfied and certainly appoints only finitely many followers. Now suppose that we appoint infinitely many followers x_n. Each time we appoint a new one, x_n, every previous one y is realized by construction and we initialize all lower priority requirements. In particular, no number $z < \psi_e(y)$ can ever be put into G_i and so all the previous followers remain realized at every later stage. Were one of them ever later permitted by U_j, it would go into G_j contrary to our assumption. Thus we can compute U_j recursively for a contradiction.

Finally, we argue that $\lim r(D,i,j,e) < \infty$. In every case, only finitely many followers are ever appointed. The P restraint $r(D,i,j,e,x_n,t)$ which is just $\max\{\delta_{k,j}(W_k; x_n)[t] |\, M_k$ has higher priority than $D_{i,j,e}\}$ associated with each such x_n is eventually constant by the success of the requirements of type 4 as in the verification of Theorem 4.1. □

Lemma 5.3. *($N_{i,j,e}$): If $N_{i,j,e}$ is never initialized after stage r, there is a stage after which $N_{i,j,e}$ never puts any more numbers into G_j and so never initializes other requirements. Moreover, $N_{i,j,e}$ is satisfied, appoints only finitely many followers and the P restraint it imposes, $r(N,i,j,e,t)$, is eventually constant.*

Proof. We argue as for $D_{i,j,e}$. □

Lemma 5.4. *($Z_{i,j,e}$): If $Z_{i,j,e}$ is never initialized after stage r, there is a stage after which $Z_{i,j,e}$ never puts any more numbers into G_i and so never initializes other requirements. Moreover, $Z_{i,j,e}$ is satisfied.*

Proof. If $\lim_{s\to\infty} f(i,j,e,s) = 1$, we eventually have a realized follower y with a $V_{i,j}$ correct computation of $\Psi_e(V_{i,j};y) = 0$ which is in G_i. In this case, it is clear from the construction that $Z_{i,j,e}$ is satisfied and eventually stops acting entirely. Suppose then that $\lim_{s\to\infty} f(i,e,t) = 0$ and (for the sake of a contradiction to the first claim of the lemma) that infinitely often we put a realized follower of $Z_{i,j,e}$ into G_i. By construction, this can happen at a stage s only if there is a $u \geq s$ such that $g(i,e,u) = 1$. Thus it can happen infinitely often only if $\lim g(i,e,u) = 1$. In this case, there is a stage t at which we do not terminate the substages of t dealing with $Z_{i,j,e}$ because of our analysis of the realized followers already in G_i and there is an uncancelled follower z of $Z_{i,j,e}$ such that $\Psi_e(V_{i,j};z)\downarrow = 0$ $[t]$ via a $V_{i,j}$ correct computation and some $x \leq z$ enters U_i at t. At such a stage t, we would put z into G_i by construction. As the $V_{i,j}$ computation is correct, z will remain a realized follower with a correct computation which is in G_i. As this contradicts our assumption that $\lim f(i,j,e,s) = 0$, we have proved the first assertions of the lemma.

Finally, we argue that $Z_{i,j,e}$ is satisfied if $\lim f(i,j,e,s) = 0 = \lim g(i,j,e,s)$. If there is a follower $y \notin G_i$ such that $\neg(\Psi_e(V_{i,j};y) = 0)$, the requirement is satisfied. So suppose there is no such y. Under these hypotheses, we would appoint infinitely many followers z of $Z_{i,j,e}$ and, for almost all of them, $\Psi_e(V_{i,j};y) = 0$. Let t_0 be a stage by which $f(i,j,e,s)$ and $g(i,j,e,s)$ have reached their limits (0) and such that $\Psi_e(V_{i,j};y) = 1$ for every follower $y > t_0$ of Z. Consider the sequence of followers $y_n, y_{n+1}, \ldots$ appointed after t_0. Recursively in $V_{i,j}$ we can find the stages t_k by which $\Psi_e(V_{i,j};y_k) = 0$ by $V_{i,j}$ correct computations. If $U_i {}^{-} y_k$ ever changed at some $t > t_k$ we would contradict the assumption that $\lim g(i,j,e,s) = 0$ by providing a stage t as required to make $\lim g(i,j,e,s) = 1$ by our choice of g. On the other hand, if there is never such a change in U_i, $U_i \leq_T V_{i,j}$ for the desired final contradiction. □

6. Lattice Embeddings and Effective Successor Models

We need to combine the basic construction of a Slaman-Woodin set (§4) with the lattice embedding properties necessary to construct an effective SMA below $(\mathbf{a}, \mathbf{u})$ for any given nonzero r.e. degree $\mathbf{a}$ and promptly simple $\mathbf{u}$. As $\mathbf{u}$ is promptly simple and so noncappable, there is always a degree below both $\mathbf{a}$ and $\mathbf{u}$. Thus, without loss of generality, we may assume that $\mathbf{a} \leq \mathbf{u}$ and so state a theorem sufficient to establish Theorem 2.14 as follows:

Theorem 6.1. *Given any $0 <_T A \leq_T U$ with U promptly simple and recursive partial ordering $\mathcal{P} = \langle \omega, \preceq \rangle$ with a specified infinite recursive set $H = \{h_i | i \in \omega\}$ of minimal elements, there are r.e. sets $E_0, E_1, F_0, F_1, B, P, Q, R$ and G_i (for $i \in \omega$) with $R = \oplus G_i, F_0 = \oplus G_{h_{2i}}, F_1 = \oplus G_{h_{2i+1}}$ and $B = G_i^{[0]}$ (for each $i \in \omega$) such that $P, Q \leq_T U$, all the other sets constructed are recursive in A and*

1. $(T) : G_i \oplus P \geq_T Q$.
2. $(D) : G_i \,|_T\, G_j$ *for* $i \neq j$.
3. (M) : *If B is recursive in an r.e. W which is recursive in R and $W \oplus P \geq_T Q$, then there is a $j \in \omega$ such that $G_j \leq_T W$.*
4. (K) : $R \oplus P$ *is low.*
5. $(O) : i \succeq j \Rightarrow G_i \oplus L \geq_T G_j$.
6. $(N) : i$ ffl $j \Rightarrow G_i \oplus L \,|_T\, G_j$.
7. (Y) : *For each* $i \in \omega$, $\deg(G_{h_{2i}} \oplus E_1) \wedge \deg(F_1) = \deg(G_{h_{2i+1}})$ *and* $\deg\left(G_{h_{2i+1}} \oplus E_0\right) \wedge F_0 = \deg(G_{h_{2i+2}})$.

The individual requirements of the types previously considered are indexed as in the basic construction with the exception that in M_i we replace W_i by $W_i \oplus B$ to take into account the fact that we are only constructing a Slaman-Woodin set above B. The new infimum requirements Y are indexed as follows:

7. $Y_{f,2m+l}$: If $\Phi_f(G_{h_{2m+l-1}} \oplus E_l) = \Phi_f(F_l) = h$ then $h \leq_T G_{h_{2m+l}}$ for the pair $m = 0$ and $l = 1$ as well as all $m > 0$ and $l \in \{0, 1\}$.

We want to use the pinball model due to Lerman [16] to satisfy the lattice preservation properties Y. If we consider for the moment, only the lattice type requirements (D for $j \in H$ and Y), the construction is like that for the lattice N_5 (the pentagon) in that the tracing procedure is finitary and the construction falls into the class of partial lattices (the TPP partial lattices) considered and embedded in the r.e degrees by Lerman, Shore and Soare [18]. A somewhat new ingredient, even in the lattice case, is the permitting required to get all the relevant sets recursive in A. Although it is generally known that this can be done (for example Ambos-Spies and Fejer [2] prove this for the particular lattice N_5 by an indirect argument; Ambos-Spies, Lempp and Soare have announced [4] the much more difficult result that every TPP partial lattice can be embedded in every nontrivial interval of r.e. degrees) and an argument showing that M_5 the basic nondistributive lattice not in TPP can be embedded below any nonlow$_2$ r.e. degree is given in [7], no proof combining ordinary permitting with TPP type lattice embedding has appeared in the literature. Thus we first explain (for the reader familiar with some pinball machine argument with traces (see, for example, [7]) how to satisfy all the requirements except (T) and (M) and only then how to combine them with the basic construction of a Slaman-Woodin set.

All the positive ordering facts needed are guaranteed by direct coding as specified in the theorem except for the requirements that $G_{h_{2i+1}} \leq_T G_{h_{2i}} \oplus E_1$ and $G_{h_{2i+2}} \leq_T G_{h_{2i+1}} \oplus E_0$. These reductions are constructed by the usual tracing procedure. Whenever we have a ball (number) $x^{h_{2i+1}}(x^{h_{2i+2}})$ targeted for $G_{h_{2i+1}}(G_{h_{2i+2}})$ on the machine, it will have a trace t targeted for either $G_{h_{2i}}(G_{h_{2i+1}})$ or $E_1(E_0)$. The rules of the construction will guarantee that if t does not enter the set for which it is targeted, then $x^{h_{2i+1}}(x^{h_{2i+2}})$ cannot enter its set either. If t does enter its set then either $x^{h_{2i+1}}(x^{h_{2i+2}})$ enters its set at the same stage or we appoint a new trace t'. As we will also guarantee that every ball targeted for some G_j, for $j \in H$, gets a last trace appointed, this procedure gives us the required reductions of $G_{h_{2i+1}}(G_{h_{2i+2}})$ to $G_{h_{2i}} \oplus E_1(G_{h_{2i+1}} \oplus E_0)$.

The infimum requirements, $Y_{f,2m+l}$ are met by a procedure that in essence makes (almost) every ball targeted for some G_j with $j \in H$ (or some smaller number associated with it) pass a gate devoted to the infimum requirement that, roughly speaking, does not allow balls targeted for both sides of the pair with the specified infimum to pass simultaneously unless they are targeted for the infimum itself and forces later balls (larger numbers) to wait until the relevant length of agreement has recovered before letting new balls go by that might injure the other side.

The diagonalization requirements D and N for $j \in H$ are met by appointing balls in the "holes" of the pinball machine as followers of the requirement associated with the hole. for example, the hole $H_{i,j,e}$ devoted to the requirement $D_{i,j,e}$ that $\Psi_e(G_i) \neq G_j$ appoints a follower $x = x^j_{i,e,n}$ (the nth follower appointed for this requirement) which is targeted for G_j. When appointed, the ball immediately gets an appropriate sequence of *traces* associated with it which may change over time. The set of balls so associated with a particular follower is called the *entourage* of the follower. When the follower is realized ($\Psi_e(G_i; x) \downarrow = 0$), it and its entourage are released from the hole and roll down the machine until they come to the first unoccupied gate. The last ball in the entourage rolls out to the gate and receives its own sequence of traces (if needed). The rest of the entourage waits in the corral associated with this gate. The balls waiting at the gate wait until the length of agreement associated with it reaches a new maximum. The gate then opens and the balls waiting at the gate are released to roll down the machine in the same way. When a set of balls passes the last gate they enter the permitting bin. The last ball in the entourage is given a trace targeted for B and the set of balls waits for permission from A. When they are permitted by a number less than the original follower with which they are associated being enumerated into A, each one of them is put into the set for which it is targeted. If the follower itself enters its target set, the requirement is satisfied and, if not injured by the action of higher priority requirements, requires no further attention. If not, the previous ball in the entourage (which is necessarily in the corral of an unoccupied gate) moves out to the gate in whose coral it has been waiting, gets an appropriate sequence of traces (if necessary) and waits for the gate to open. Whenever a ball is placed on the machine or moves, all lower priority balls are cancelled and removed from the machine. In order to precisely describe the priority listing we must first define the tracing procedure.

6.1. Tracing and Priorities

Only balls targeted for some G_j for $j \in H$, $j \neq h_0$ need traces. Whenever we have a ball $x^{h_{2i+1}}(x^{h_{2i+2}})$ targeted for $G_{h_{2i+1}}(G_{h_{2i+2}})$ on the machine, it will have a trace t targeted for either $G_{h_{2i}}(G_{h_{2i+1}})$ or $E_1(E_0)$. When a follower $x = x^j_{i,e,n}$ is first appointed in the hole associated with a diagonalization requirement $D_{i,j,e}$ or $N_{i,j,e}$ it is targeted for G_j. If $j = h_{2i+k}$ ($k \in \{0, 1\}$ and $2i + k > 0$), x is given a trace t targeted for E_k. When a ball x of the form $x^{h_{2i+k}}$ rolls out to a gate for some infimum requirement $Y_{f,2m+l}$, we assign traces as follows:

a) If $2i + k = 2m + l$ no trace is necessary.

b) If $l = k$ (but $i \neq m$), then x gets a trace t targeted for $G_{h_{2i+k-1}}$ and t gets a trace r targeted for $E_l - k$.

c) If $l \neq k$ then x gets a trace targeted for E_k.

When a sequence of balls (necessarily the tail end of some entourage) rolls past the last gate, the entire sequence is put into the permitting bin and the last ball in the sequence is given a trace targeted for B. (These balls then wait for A to permit the follower which is the first ball in their entire entourage and then all go simultaneously into the sets for which they are separately targeted.)

The crucial point for this construction is that we can clearly calculate a bound on the number of balls that can ever be assigned to the entourage of a follower of a requirement $D_{i,j,e}$ or $N_{i,j,e}$. Indeed, the calculation of such a bound can be made solely as a function of the number of gates of higher priority than $D_{i,j,e}(N_{i,j,e})$. We denote such a bound by $g(i,j,e)$ and use it to assign priorities to followers of this requirement. (For technical convenience, we have $D_{i,j,e}$ and $N_{i,j,e}$ between the same gates and so have a single function $g(i,j,e)$ for both.) The idea is that $g(i,j,e)$ is a bound on the number of permissions needed from A (at particular stages) to eventually get the follower into its target set. Contrary to the usual practice of assigning priorities to followers in the order in which they are appointed, we prefer those that have received more permissions. To be precise, each ball x is assigned a triple $\langle a,b,c\rangle$ that determines its priority by the lexicographic ordering of triples from ω. The first entry is the index of the requirement $D_{i,j,e}(N_{i,j,e})$ of which the first ball in x's entourage is a follower. The second entry is $g(i,j,e)-n$ where n is the number of times a ball in x's entourage has been permitted to leave the permitting bin and enter its target set. The third entry is simply x itself. (This last coordinate orders otherwise similar balls by the point at which they are placed on the machine since new balls are always chosen larger than any already on the machine.)

The construction of the required partial lattice and the verification that it succeeds are now fairly standard with some additional care being need because of the permitting conditions and the associated change in priority ordering. Before giving a formal description of the full construction for Theorem 6.1, we will informally describe the modifications needed to combine the lattice embedding argument with the construction of a Slaman-Woodin set. The constructions should be viewed almost as two distinct ones with certain interactions. The diagonalization requirements for $j\in H$ are shared between the two constructions and mediate the interactions. (Note that, as every $j\in H$ is a minimal element of $\mathcal{P}$, no other requirements put numbers into such G_j.) These requirements now sit in the appropriate hole of the pinball machine but need to appoint followers that are suitable in the sense of the Slaman-Woodin construction and impose P restraint to maintain their suitability as in that construction. Once appointed, however, they get traces assigned, wait for realization and then roll down the machine as for the pinball argument. The only interaction that they then have with the rest of the Slaman-Woodin construction is that they might initialize lower priority requirements or be cancelled by actions of higher priority requirements. Such cancellation by requirements not involved in the pinball construction, however, is finitary and so no worse than the cancelation by higher priority requirements already inherent in this construction. The gates of the pinball machine act as before and so the infimum requirements are satisfied as for the pinball argument alone. The minimality requirements M_i of the Slaman-Woodin construction proceed as before to assign chits, define functionals and, perhaps in the end, put a number into Q (but now only when promptly permitted by U) to kill off the requirement. The only change caused by the pinball argument is again the possibility of extra finitary cancelation of followers by actions of higher priority. The other diagonalization requirements $D_{i,j,e}$ and $N_{i,j,e}$ for $j \notin H$, the lowness conditions K and functional constructions T are similarly affected only by additional finitary initializations. The coding requirements O are handled exactly as before. (Again noting that the minimality within $\mathcal{P}$ of all $j \in H$ guarantees that this procedure does not put any numbers into sets involved with the pinball construction.) Thus there is no real difficulty in combining the two constructions.

6.2. Construction

We begin by listing all the requirements $T_i, D_{i,j,e}, N_{i,j,e}, M_i, K_{e,x}, Y_{f,2m+l}$ in a priority list of order type ω such that T_i and T_j appear on the list before any $D_{i,j,e}$ and such that each $N_{i,j,e}$ immediately follows $D_{i,j,e}$ and so lies between the same $Y_{f,2m+l}$ requirements. The pinball machine is the usual one with gates corresponding to the $Y_{f,2m+l}$ requirements and holes to the $D_{i,j,e}$ and $N_{i,j,e}$ ones for $j \in H$. The other requirements make no direct appearance on the machine. Each stage s of our construction will have at most s many substages n at which we may deal with requirement n. Each requirement may move balls (numbers) on the pinball machine, put numbers (balls) into some sets, axioms into a functional and impose restraint on various sets in the usual fashion.

We act to satisfy the direct coding requirements in the obvious way. Whenever a ball (number) x is put into G_{2i+k}, $\langle 2i+k, x\rangle$ is put into R and F_k. Whenever a ball (number) x is put into B, $\langle 0, x\rangle$ is put into every G_i. Whenever action for a requirement $D_{i,j,e}$ puts a number x into some G_k the same number x is put into L. When action for a requirement $N_{i,j,e}$ puts a number x other than the original follower of the requirement into some G_k the same number x is put into L [[these numbers will be larger than the computation from $G_i \oplus L$ that $N_{i,j,e}$ needs to preserve]] ; when the follower x itself of $N_{i,j,e}$ goes into G_j it also goes into all G_k with $k \succeq j$. Whenever any action is taken for some ball we cancel all balls of lower priority (i. e. remove them from the machine) and initialize all requirements of lower priority than the one associated with the ball for which we acted. Initialization of a requirement at stage s means that all balls, chits and all restraint associated with the initialized requirement are cancelled and no further attempts are made at maintaining the correctness at numbers less than s of any functionals it is constructing. (As we only care about the functional being correct almost everywhere, this will not cause problems as long as the requirement is initialized only finitely often.) As new followers are always chosen larger than any number previously used in the construction, initialization also acts to preserve various computations. At other times we may cancel chits because the situation that made them usable has been destroyed.

We now describe our actions at substage n of stage s of the construction according to the type of the nth requirement on our list. Note that there is no direct action by the infimum requirements $Y_{f,2m+l}$ other than the restraint imposed by the usual action of the corresponding gates. This effect is implicit in the rules of motion for the balls on the machine and so there are no explicit steps in the construction devoted to the $Y_{f,2m+l}$ requirements. We let r be the last stage at which the requirement being considered at substage n was last initialized ($r = 0$ if there is no such stage). When we choose a "*large*" number at a stage s we mean a number larger than any used in the construction before s.

1. $T_i : G_i \oplus P \geq_T Q$.
As in the basic construction.

2. $D_{i,j,e} : \Psi_e(G_i) \neq G_j$.
If $j \notin H$ the procedure is exactly the same as in the previous constructions so we describe the new procedure if $j \in H$. If we have put a follower of $D_{i,j,e}$ into G_j since stage r, we go on to the next substage. If not, we find the ball x of highest priority (as defined above) associated with a follower of $D_{i,j,e}$ that requires attention according to one the following cases cancel all balls of lower priority, initialize all requirements of lower priority and then act as indicated in each of the cases before going on to the next substage.

a) x is a previously unrealized follower of $D_{i,j,e}$ sitting in the associated hole and it is now *realized*, i. e. $\Psi_e(G_i; x)\downarrow= 0$. Release x and its trace t from the hole and let them fall down to the first unoccupied gate for a requirement Y. The trace t for x rolls out to the gate (which it now *occupies*) and x is put into the corral for this gate. If there is no such gate below them they roll into the permitting bin and t gets a new large trace targeted for B.

b) x is occupying a gate for $Y_{f,2m+l}$ and the associated length of agreement reaches a new maximum, i. e. $\ell(f, 2m+l, s) = \mu z\{\neg(\Phi_f(G_{h_{2m+l-1}} \oplus E_l; z) \downarrow = \Phi_f(F_l; z)\,[s])\}$ is greater than $\ell(f, 2m+l, t)$ for every $t < s$. Let x and the later elements in its entourage [[the traces also occupying the same gate]] roll down the machine leaving the gate at which it now resides *unoccupied.* They roll to the first unoccupied gate. The last element of the entourage rolls out to the gate which it now *occupies* and the others are put into the associated corral. If there is no such gate, the balls roll into the permitting bin and the last of them is given a new large trace targeted for B.

c) x is in the permitting bin and a number less than the follower in x's entourage is enumerated in A. We put x and the later elements of its entourage [[which are larger and also sitting in the permitting bin]] into the sets for which they are targeted. Moreover, if x is not itself the follower, we find the last element y of its entourage [[which is necessarily in some corral whose gate is unoccupied]], move it out to the gate associated with the corral in which it is waiting and appoint traces in accordance with the requirement associated with this gate as described above and cancel all balls of lower priority than y which, together with its traces, now *occupies* this gate. [[Note that we cancel balls again as the priority of y is now higher than that of x before our action since one more permission has been acted upon.]]

If there is no such ball that requires attention and there is no unrealized follower sitting in the hole for $D_{i,j,e}$, all requirements of lower priority are initialized and we choose a new large follower x for $D_{i,j,e}$ from the column $\omega^{\langle 0,i,j,e\rangle}$ associated with $D_{i,j,e}$. [[There are no balls associated with this requirement with lower priority than x.]] We also appoint a new large trace t for x as described above and put both t and x in the hole for $D_{i,j,e}$. At every stage $t \geq s$ until x is cancelled or enters G_j, $D_{i,j,e}$ imposes restraint $r(D, i, j, e, x, t)$ on P for each element y of the entourage of x which is targeted for some G_n as in the basic construction. Thus we define the restraint as follows:

- $r(D, i, j, e, t) = \max\{\delta_{k,n}(W_k; y)[t] | \, y$ is in the entourage of some (uncancelled) follower x of $D_{i,j,e}$ and targeted for G_n and M_k has higher priority than $D_{i,j,e}\}$.

This restraint is dropped when some follower of D enters G_j. Because the action of $D_{i,j,e}$ is not obviously finitary we must also act to stabilize this restraint: If a y in the entourage of a current follower of $D_{i,j,e}$ is targeted for G_n, M_k has higher priority than $D_{i,j,e}$ and some $\Delta_{k,n}(W_k; y)$ has just converged at s then we initialize all requirements of lower priority than $D_{i,j,e}$ and terminate stage s.

If none of the above conditions are satisfied, we simply move on to the next substage.

3. M_i: If $\Theta_i(R) = W_i \oplus B$ and $\Phi_i(W_i \oplus B \oplus P) = Q$, then $G_j \leq_T W_i \oplus B$ for some j such that T_j has higher priority than M_i.

Our action here is the same as in the basic construction until we reach a point at which we would have killed the requirement by putting some particular y into Q and various $\gamma_j(y)$ into P. At such a point we use the function p witnessing the prompt simplicity of U in the usual way to determine if U will permit us to put y into Q. If so, we proceed as in the basic construction and terminate stage s. If not, we just move on to the next substage. (We will never consider this same y again for killing M_i.)

[[Note that we do not have to worry about the suitability of numbers x targeted for B as any Δ computation that might be invalidated by the corresponding code entering some G_i is immediately repaired by x's entering B and destroying any previously defined computation of $\Delta(W_i \oplus B; \langle i, x\rangle)$.]]

4. $K_{e,x}$: If there are infinitely many s such that $\Xi_e(R \oplus P; x) \downarrow [s]$, then $\Xi_e(R \oplus P; x) \downarrow$.

We proceed exactly as in the basic construction.

At the end of substage s, we terminate stage s (if it has not been terminated before). When stage s is terminated, we go on to stage $s+1$.

5. The procedure for $N_{i,j,e}$ for $j \notin H$ is as in §4.4. For $j \in H$, it is like that of the $D_{i,j,e}$ with $j \in H$ modified with the same modification that relates $N_{i,,j,e}$ to $D_{i,j,e}$ in the basic construction. To be specific, we adjust the restraint $r(N, i, j, e, x)$ imposed by $N_{i,j,e}$ on P at stage t to be the following:

- $r(N, i, j, e, x) = \max\{\delta_{k,l}(W_k; y)[t] \mid y$ is in the entourage of some (uncancelled) follower x of $N_{i,j,e}$ and targeted for G_n, $l \succeq n$ and M_k has higher priority than $D_{i,j,e}\}$.

6.3. Verifications

The verification procedure is somewhat more complicated than in the previous arguments. Of course, we must now verify that the infimum requirements are met but the primary new source of complications in the rest of the argument is the action for the diagonalization requirements. It is no longer true that a single requirement $D_{i,j,e}$ or $N_{i,j,e}$ for $j \in H$ acts at most once after it is last initialized. Indeed, it is not obvious that it acts at most finitely often. Thus we cannot argue *ab initio* that the lowness requirements of type 4 automatically succeed.

It is still true that (other than through initialization) the only restraint imposed in the construction is on P and the restraint $r(M, i, s)$ imposed on P by requirements M_j of priority greater than or equal to that of M_i have a finite lim inf which is realized on the i-expansionary stages if there are infinitely many such stages and, if not, on almost all the stages at which $r(M, i-1, s)$ realizes its lim inf. As for the action of the gates, for now we simply note that it is immediate from the construction that if there is a ball which permanently occupies a gate then there are finitely many (indeed, at most two) balls which are permanently at that gate and once they occupy the gate all other balls roll past that gate when they reach it in their movement down the machine.

Once again it is immediate from the definition of their actions that if a requirement M_i or $K_{e,x}$ of type 2 or 4 is never initialized after stage r, it acts at most once after r to put numbers into sets (for M_i) and to initialize lower priority requirements. The only other requirements that initialize anything are the $D_{i,j,e}$ and $N_{i,j,e}$. We prove a lemma that shows by induction that these types of requirements eventually never initialize any other requirement and in other ways have finite effect on the construction and are satisfied.

Lemma 6.2. *($D_{i,j,e}/N_{i,j,e}$): If $D_{i,j,e}$ is never initialized after stage r, there is a stage after which $D_{i,j,e}$ never puts any more numbers into G_j and never initializes other requirements. Moreover, $D_{i,j,e}$ is satisfied, appoints only finitely many followers and the P restraint it imposes, $r(D, i, j, e, t)$, is eventually constant. The same facts are also true about the requirements $N_{i,j,e}$.*

Proof. The arguments for $D_{i,j,e}$ and $N_{i,j,e}$ are essentially the same (just replace $\Delta_{k,n}(W_k; y)$ by $\Delta_{k,l}(W_k; y)$ for $l \succeq n$) and for simplicity we present only that for $D_{i,j,e}$. If $j \notin H$ it is obvious that the requirement can act at most once after stage r and as in the basic construction must be satisfied. Thus we assume that $j \in H$. We can assume that any gate of higher priority than $D_{i,j,e}$ that gets a permanent occupant already has it by stage r. Thus no members of the entourage of any later follower of $D_{i,j,e}$ ever stop at these gates. Moreover, they can temporarily stop at other gates but must then either be released by the gate opening or cancelled by actions for a ball of higher priority also in the entourage of some follower of $D_{i,j,e}$ (any other cancelation would contradict our choice of r). As there can be only finitely many actions taken for the balls in the entourage of a single follower, the only situations that could prevent the appointment of infinitely many followers are one of them entering G_j or one remaining in the hole for $D_{i,j,e}$ forever unrealized. In either of these cases it is clear from the construction that we never act for $D_{i,j,e}$ again. Moreover, $D_{i,j,e}$ is then satisfied by the same argument as in the basic construction.

We therefore suppose that infinitely many followers are appointed and argue that A is recursive for a contradiction. Consider then the follower x_1 of $D_{i,j,e}$ that attains the highest priority of any follower appointed after r. By our choice of r and the rules of the construction x_1 can never be cancelled. Since it must eventually be realized but never enters G_j by our assumptions, some member of its entourage must be permanently stuck in the permitting bin. Let $x_2, x_3, \ldots$ be the successively next highest priority ball to reach this same state. Remember that the priorities of these balls are given by triples the first coordinate of which is simply the global priority of $D_{i,j,e}$. The second coordinate is $g(i,j,e)$ minus the number of times balls in the entourage have been permitted to enter their target sets. Thus this coordinate must eventually stabilize as well, say at n. Consider now the balls x_m for $m \geq m_0$ which all have the second coordinate of their priorities equal to n. It is clear that once such a ball is in the permitting bin, it can never be permitted to enter its target set as that would increase the priority of the remaining balls in the entourage contrary to the choice of n. Moreover, it can be cancelled only by some smaller ball with the same first two coordinates of its priority triple the same as its own moving. This can happen only when the smaller ball was stopped at one of the gates that are not permanently occupied when the larger one entered the permitting bin. As we know which these gates are and that any occupants are only temporary, we can tell when any ball in the final priority state (as far as the first two coordinates are concerned) enters the permitting bin if it will ever be cancelled. (It will be cancelled if and only if there is a higher priority ball at one of these gates in which case one of those balls must eventually enter the permitting bin.) Thus we can recursively list the x_m which enter the permitting bin in this final state and are never cancelled. As usual this listing provides a way of recursively computing A for the desired contradiction.

Finally, we argue that $\lim r(D,i,j,e) < \infty$. Consider any $y > r$ targeted for G_n which is in the entourage of one of the finitely many followers of $D_{i,j,e}$. If some $\Delta_{k,n}(W_k;y)$ converges for the first time at a stage t after x is appointed, we initialize all lower priority requirements. Thus no number below $\theta_k\delta_{k,n}(y)$ can ever enter R after stage t and so none below $\delta_{k,n}(y)$ can enter W_k. Thus $\delta_{k,n}(y)$ is fixed from now on and so can never again cause initialization by its convergence. $\square$

Thus we know that for each requirement there is a stage r after which it is never initialized. The argument that the requirements $K_{e,x}$ are met is now routine and exactly as in the basic construction. Given the success of these requirements and the fact that initializations for each requirement are bounded, we can now argue for the success of the M_i requirements. As in the basic construction we see that if all of the functionals $\Delta_{i,j}$ fail infinitely often then there are infinitely many x for which we reach a point in the construction when we would put x into Q if permitted by U at that particular stage. The prompt simplicity of U then guarantees that we actually put one of these numbers into Q. The argument of the basic construction then shows that M_i never acts again, imposes only a finite, eventually constant restraint and is satisfied.

All that remains is to verify that the infimum requirements are met. We first prove a technical lemma about the priority ordering.

Lemma 6.3. *If balls $x < y$ are both on the machine at stage s then x has higher priority than y.*

Proof. Suppose for the sake of a contradiction that y has higher priority than x. As y is larger than x it must have been placed on the machine after x. If it had higher priority than x when initially placed on the machine then x would have been cancelled by construction. On the other hand, if y's priority improved at some stage to make it higher than that of x, then x would have been cancelled at that stage according to clause (c) of item 2 (or 5) of the construction. $\square$

Lemma 6.4. *If $\Phi_f(G_{h_{2m+l-1}} \oplus E_l) = \Phi_f(F_l) = h$ then $h \leq_T G_{h_{2m+l}}$ for the pair $m = 0$ and $l = 1$ as well as all $m > 0$ and $l \in \{0,1\}$.*

Proof. Let r be a stage after which the corresponding requirement $Y_{f,2m+l}$ is never initialized and every gate of higher priority that has a permanent occupant already has one. We claim that $h(x)$ can be correctly computed by finding a stage $s > r$ at which $\ell(f, 2m+l, s) > x$, the computations are $G_{h_{2m+l}}$ correct (i.e. $G_{h_{2m+l}} \upharpoonright u[s] = G_{h_{2m+l}} \upharpoonright u$ where u is the use of the relevant computations at x) and only balls that never move again are at gates, corral or bins at or below the gate for $Y_{f,2m+l}$. Now, we can recognize such a stage recursively in $G_{h_{2m+l}}$ as there are only finitely many balls permanently at gates or corrals below the gate for $Y_{f,2m+l}$ while ones in the permitting bin are there permanently if and only if the trace targeted for B at the end of their entourage is not in B, a fact recursive in $B = G^{[0]}_{h_{2m+l}}$ and, of course, $G_{h_{2m+l}}$-correctness is recursive in $G_{h_{2m+l}}$. By the hypothesis of the Lemma there is a $t > r$ such that at every $s > t$ the length of agreement is larger than x and $G_{h_{2m+l}}$-correct . Thus all that remains to be shown is that there is one at which only balls that will permanently remain in their current positions are on the machine at or below the gate for $Y_{f,2m+l}$ and that the value computed at such a stage s is the correct one.

As for the first claim, consider any stage $s > t$ at which some cancellation or initialization is caused by a requirement and no requirement of the same or higher priority ever causes any cancellation or initialization again. (Clearly such stages exist by the arguments above.) It is clear from the definition of the construction that any ball now on the machine at or below the gate for $Y_{f,2m+l}$ must remain in its current location forever. Thus any such stage s is as required and there are obviously infinitely many such.

We now argue by induction that the computation at x found at a stage s as described above remains constant and is $G_{h_{2m+l}}$-correct on at least one side of the infimum at every later stage. Now at s any ball that can injure the computation at x must already be on the machine (ones appointed later are larger than the use) and above the gate for $Y_{f,2m+l}$ (by hypothesis the associated length of agreement goes to infinity and so no ball permanently occupies the gate for $Y_{f,2m+l}$). Let z be the first ball below the use which enters one side of the infimum (it cannot enter both as the computation is assumed to be $G_{f,2m+l}$-correct). We claim that z occupies the gate for $Y_{f,2m+l}$ at some stage after s_0 and before it enters the set for which it is targeted. If not, then by the construction at the stage at which it rolls past this gate, the gate must be occupied by a ball of higher priority. (If it were occupied by one of lower priority that ball would be cancelled when we are about to move z and so the gate would become unoccupied in which case z would have to stop at the gate or the corral. In the latter case, it would have to move out to the gate before going past it.) Thus z was not on the machine when the ball occupying the gate reached it (for it would then have been cancelled). The gate has not opened since the ball of higher priority than z reached it (or the ball would have left the gate) and so z is larger than the use of the computation at x

for a contradiction.

Suppose x occupies the gate for $Y_{f,2m+l}$ and is about to leave it at stage t. It leaves the gate because a new maximum length of agreement has been reached. As no number less than the use at s has entered either side of the infimum, the value is the same as at s. We claim that the computation from the side of the infimum for which x is not targeted now becomes $G_{h_{2m+l}}$-correct. Of course, any ball larger than x is cancelled when x moves by the rules of the construction and Lemma 6.3. On the other hand, any ball smaller than x is within the range already assumed to be G_{2m+1}-correct. To continue the inductive hypotheses we now only need to argue that all balls less than the use at t that might enter this (now certified as $G_{h_{2m+l}}$-correct) side of the infimum are above the gate for $Y_{f,2m+l}$. Any such ball below the gate must have passed the gate at some stage after s (as the ones below at s never move and no balls are placed on the machine by requirements of higher priority than $Y_{f,2m+l}$ by our choice of r). If the ball has lower priority than x then it would have been cancelled at t and so no longer be on the machine. If it had higher priority than x it would have cancelled x when attempting to pass the gate contrary to our assumption that x is still at the gate at t. Similarly, if it had passed the gate with lower priority than x but gotten higher priority afterwards then it would have cancelled x when it got its higher priority. We can now continue by induction to prove that at every stage when a ball that would injure the previously certified $G_{h_{2m+l}}$-correct side leaves the gate associated with $Y_{f,2m+l}$ that the computation from one side of the infimum is made $G_{h_{2m+l}}$-correct and gives the same output as that given at s. $\square$

7. Coding Σ_3 Sets

Our goal in this section is to prove the coding results needed in §2 as given by Theorem 2.15 which we recall as follows:

Theorem 7.1. *If $\langle \mathbf{g}_i \rangle$ is a u.r.e. antichain, $\oplus_i \mathbf{g}_i$ is low and $(\forall i)[\mathbf{a} \mathbin{\Theta} \mathbf{g}_i]$, then for each $\Sigma^0_3(A)$–set S there are $\mathbf{c}, \mathbf{d} \leq \mathbf{a}$ such that $S = \{i : \mathbf{c} \leq \mathbf{g}_i \vee \mathbf{d}\}$.*

We begin by giving an effective approximation to S, using methods similar to ones of Nies in [20] and [23].

Lemma 7.2. *There is a u.r.e. sequence $X_{\langle i,n,m \rangle}$ of initial segments of ω such that*

$$i \in S \Rightarrow \text{ a.e. } n, m\ X_{\langle i,n,m \rangle} \textit{ is finite and} \tag{7.1}$$

$$i \notin S \Rightarrow \forall n\ \exists m(X_{\langle i,n,m\rangle} = \omega). \tag{7.2}$$

Moreover, there is a partial A–recursive function χ_X for the array X_k, which is defined at $k = \langle i,n,m\rangle$ iff X_k is finite, in which case X_k equals $X_{k,\chi_X(k)}$.

Proof. Since S is $\Sigma^0_3(A)$, there is an r.e. set W such that

$$i \in S \Leftrightarrow \exists n\ \forall m(\langle i,n,m\rangle \in W^A).$$

Let $Y_k = \{s : (\exists t \geq s)(k \notin W^A[t])\}$. Then this sequence is u.r.e., each Y_k is an initial segment of ω and $k \in W^A \Leftrightarrow Y_k$ is finite and so (7.2) holds for $\langle Y_k\rangle$. Moreover, the second claim of the lemma holds for the sequence: To calculate the partial function $\chi_Y(k)$ required for the array Y_k, search for an s such that $k \in W^A[s]$ by an A–correct computation.

We now modify $\langle Y_k\rangle$ to satisfy (7.1) by first reducing the number of sets which equal ω. By the row $\langle i,n\rangle$ (of $Y = \{Y_k | k \in \omega\}$) we mean the collection of sets $\{Y_{\langle i,n,m\rangle} | m \in \omega\}$. First we replace $Y_{\langle i,n,m\rangle}$ by $Y'_{\langle i,n,m\rangle} = Y_{\langle i,0,m\rangle} \cap \ldots \cap Y_{\langle i,n,m\rangle}$. Then, if the row $\langle i,n\rangle$ (of Y') contains only finite sets, so do all rows $\langle i,n'\rangle$, $n' > n$. To compute the partial function $\chi_{Y'}$ required for the array Y'_k as in the Lemma proceed as follows: Given $\langle i,n,m\rangle$ start computations with oracle A for $\chi_Y(\langle i,j,m\rangle)$, $j \leq n$ in parallel; if any of these computations converge, continue computing and enumerating all the sets $Y_{\langle i,j,m\rangle}$ until we have, for each $j \leq n$, either $\chi_Y(\langle i,j,m\rangle)$ converges or $Y_{\langle i,j,m\rangle}$ contains all the elements in the smallest of the $Y_{\langle i,j',m\rangle}$ for which $\chi_Y(\langle i,j',m\rangle)$ has converged.

Next, we modify Y' once more to produce the desired sequence $\langle X_k\rangle$ by guaranteeing that each row has at most one infinite member: Let $C_m = Y'_{\langle i,n,m\rangle}$. We replace this set in Y' by a class of sets $D_{\langle m,g\rangle} = X_{\langle i,n,\langle m,g\rangle\rangle}$ where g is thought of as a guess about $|\bigcup_{j<m} C_j|$. As long as the guess is correct, $D_{\langle m,g\rangle}$ is allowed to copy C_m. Formally, let $D_{p,0} = \emptyset$ and, for $s > 0$ and $p = \langle m,g\rangle$, if $g = |\cup_{j<m} C_{j,s}|$, let $D_{p,s} = C_{m,s}$, otherwise let $D_{p,s} = D_{p,s-1}$. Clearly, there is at most one $\langle m,g\rangle$ such that $D_{\langle m,g\rangle} = \omega$. (It is the pair for which m is least such that $Y'_{\langle i,n,m\rangle}$ is infinite and $g = |\cup_{j<m} C_{j,s}|$.)

We now show how to compute the required partial function $\chi = \chi_X$ for the array $\langle X_k\rangle$ that we have obtained. Suppose $k = \langle i,n,p\rangle$. As before, let $C_m = Y'_{\langle i,n,m\rangle}$ and let $p = \langle m,g\rangle$. To compute $\chi_X(k)$, we find an s such that either $|\bigcup_{j<m} C_{j,s}| > g$, in which case we give this s as output, or all computations $\chi_{Y'}(\langle i,n,j\rangle)$ have converged at stage s for $j < m$. If $|\bigcup_{j<m} C_{j,s}| < g$, again we give s as output, but if it equals g, we start simulating the computation of $\chi_{Y'}(\langle i,n,m\rangle)$. $\square$

To prove Theorem 7.1, we want to build r.e. sets $C, D \leq_T A$ such that

$$i \in S \Rightarrow C \leq_T G_i \oplus D \tag{7.3}$$

and

$$i \notin S \Rightarrow \forall n\ (C \neq \Psi_n^{G_i \oplus D}), \tag{7.4}$$

where $\langle \Psi_n \rangle$ is an effective listing of all Turing functionals such that, to compute the value at input x, we first compute all values for smaller inputs (this will be technically convenient).

Intuitively, to satisfy (7.4) we view the number n in (7.2) as an index for a T–reduction. For each m, a requirement $Q_{\langle i,n,m \rangle}$ tries to guarantee that $C \neq \Psi_n^{G_i \oplus D}$. If $i \notin S$, then the first requirement such that $X_{\langle i,n,m \rangle} = \omega$ will succeed. If $X_{\langle i,n,m \rangle}$ is finite, $Q_{\langle i,n,m \rangle}$ acts only finitely often. Thus the diagonalization requirements are

$$Q_{\langle i,n,m \rangle} : X_{\langle i,n,m \rangle} = \omega \Rightarrow C \neq \Psi_n^{G_i \oplus D}.$$

We write $i_Q(k) = i$ if Q_k works on (7.4) for i, i.e., if $k = \langle i, n, m \rangle$ for some pair n, m.

For each i, there are infinitely many coding requirements C_k ($k = \langle i, n \rangle$), whose goal is to build a functional Δ_k such that $C = \Delta_k^{G_i \oplus D}$. We write $i_C(k) = i$ if $k = \langle i, n \rangle$ for some n. Fix $C_0 < Q_0 < C_1 < Q_1 \ldots$ as the priority ordering of the requirements. C_k may be affected by *lower* priority requirements Q_p such that $i_C(k) = i_Q(p)$. However, each such Q_p can affect the functional Δ_k at only finitely many inputs (which are distinct for distinct p) and, if X_p is finite, only finitely often at each of those inputs. Thus, if it needs to succeed, the coding requirement relies on the hypothesis that each single requirement Q_p affects it only finitely often:

$$C_k : (\forall p \geq k)[i_Q(p) = i_C(k) \Rightarrow X_p \text{ finite}] \Rightarrow C = \Delta_k^{G_i \oplus D}.$$

If $i \in S$ then, since infinitely many coding requirements work on (7.3), we can, by (7.1), choose k with $i_C(k) = i$ such that the hypothesis is correct. Therefore $C \leq_T G_i \oplus D$.

We now describe the strategies for the requirements. C_k tries to build the functional Δ_k. For an input x, let $\delta_k(x)$ be the use of this functional. Whenever $G_i \oplus D \upharpoonright \delta_k(x)+1$ changes, we have a chance to declare $\Delta_k^{G_i \oplus D}(x)$ to be undefined, and C_k will have to redefine it at a later stage t to the correct value $C_t(x)$, with a new use $\delta_t(x)$ (normally a large number). We can only enumerate x into C while $\Delta_k^{G_i \oplus D}(x)$ is undefined. To make $\Delta_k^{G_i \oplus D}$ total, we have to make sure that, for each x, $\Delta_k^{G_i \oplus D}(x)$ is defined at some stage and is redefined only finitely often.

We now explain the Q_p strategy, and describe when to correct the functionals Δ_k. Let $p = \langle i, n, m \rangle$ and suppose that $X_p = \omega$. The goal of the Q_p–strategy is to diagonalize against $C = \psi_n^{G_i \oplus D}$. To do so, Q_p has to enumerate, at some stage s, a follower y such that $\Psi_n^{G_i \oplus D}(y) = 0$ into C where $\psi_n^{G_i \oplus D}(y)$ does not change later. To ensure that $C \leq_T A$, Q_p also requires an A–permission, i.e., $A_s \upharpoonright y \neq A_{s-1} \upharpoonright y$, before enumerating y into C at stage s. Now, the computation $\Psi_n^{G_i \oplus D}(y) = 0$ could later be destroyed by a G_i–change not under our control. However, using the lowness of G_i, we can prevent Q_p from acting infinitely often because of this kind of injury. On the other hand, the computation could also be destroyed by an enumeration of uses $\delta_k(z)$ into D $(k \leq p)$. The main case to worry about is the necessity of enumerating $\delta_k(y)$ into D for any $k \leq p$ for which $\Delta_k(y)$ is already defined, since we need to make $\Delta_k(y)$ undefined before enumerating y into C. To avoid the destruction of $\psi_n^{G_i \oplus D}(y)$ by such an enumeration, Q_p tries to ensure that $\psi_n^{G_i \oplus D}(y)$ is k*–cleared* for each $k \leq p$, i.e.,

$$\delta_k(y) > \psi_n^{G_i \oplus D}(y) \tag{7.5}$$

(where $\psi_n^{G_i \oplus D}(y)$ is, as usual, the use of $\Psi_n^{G_i \oplus D}(y)$). If $i_C(k) = i_Q(p)$ the k-clearing is done in an active way: First, Q_p picks a "killing point" x where it intends to make $\Delta_k(x)$ undefined in the limit if X_p is infinite. The followers y of Q_p will all be chosen to be $\geq x$. If X_p increases, then $\delta_k(x)$ is enumerated into D and $\Delta_k(x)$ is declared to be undefined. Since $\delta_k(x)$ will be redefined to be a large number, eventually we reach the desired inequality (7.5) (provided that $\Psi_n^{G_i \oplus D}(y)$ converges). The second claim in Lemma 7.2 is used to show that A can calculate when such enumerations have ceased and so to prove that $D \leq_T A$. For technical reasons, we will actually require that $\psi_n^{G_i \oplus D}(y) \leq \max X_p$ whenever we consider putting a follower y into D in this way. We also ensure that $\delta_k(x) > \max X_p$ whenever $k \leq p \leq x$ and $i_Q(p) = i_C(k)$.

Now we discuss clearing the computation $\Psi_n^{G_i \oplus D}(y)$ of markers $\delta_k(y)$ when $k < p$ and $i_Q(p) \neq i_C(k) = j$. In this case, Q_p relies on G_j–changes to ensure that $\Psi_n^{G_i \oplus D}(y)$ will be k–cleared. Q_p maintains a whole list of followers $y_{n,s}$ at stage s such that $\Psi_n^{G_i \oplus D}(y_r) = 0[s]$ with use below $\max(X_{p,s})$. If y is appointed as a new follower, Q_p restrains $D \upharpoonright \psi_n^{G_i \oplus D}(y)$ and requests that $\psi_n^{G_i \oplus D}(y)$ be k-cleared (i.e., that $\delta_k(y)$ be moved above at the next stage where $G_j \upharpoonright \delta_k(y)$ changes). Since $G_j \, \Theta_T \, G_i$, (if the requirement is not met in some trivial way) there will be infinitely many r with stages s such that $\psi_n^{G_i \oplus D}(y_r) = 0[s]$ by a G_i–correct computation and $G_j \upharpoonright y_r$ changes at s. Since $\delta_k(x) \geq x$ for any x, this allows Q_p's request for clearing to be carried out.

Altogether a Q_p strategy is confronted with $p+1$ coding strategies of higher priority, C_k $(k \leq p)$. The above actions are carried out for each k in parallel. In the end we can argue that potentially infinitely many G_i–correct computations $\Psi_n^{G_i \oplus D}(y)$ are cleared for all $k \leq p$: By the D–enumeration of Q_p all computations we consider are eventually k–cleared if $i_C(k) = i_Q(p)$. For the other numbers $k \leq p$, we argue by induction on descending k: If $k \leq p$ the above argument can be carried out to k–clear computations which are already k'-cleared for all k' with $k < k' \leq p$, because we can (by induction, G_i–recursively enumerate infinitely many G_i–correct computations $\Psi_n^{G_i \oplus D}(y)$ which are already so cleared. Thus, infinitely many G_i–correct computations will be k–cleared for all $k \leq p$. As $A\ \Theta_T\ G_i$, one such will have its input y permitted by A. At this point we will put y into C successfully diagonalizing and the action for Q_p will cease.

There is one final problem: For $p' < p$, the D–enumeration of the higher priority $Q_{p'}$ due to $X_{p'}$–increase may destroy computations $\Psi_n^{G_i \oplus D}(y)$ on which Q_p relies. If $X_{p'}$ is finite, Q_p can assume this side effect has ceased to occur, but if $X_{p'} = \omega$, Q_p must choose followers in a way such that the associated uses are $\leq \max X_{p'}$ (as it did for X_p itself). So we have to equip Q_p with a guess at which sets $X_{p'} = \omega$ for $p' < p$. This is done in a standard way using a priority tree. We use $2^{<\omega}$ as our tree of strategies. If $s > 0$, we define the sequence δ_s of accessible nodes by induction on k. Of course, $\delta_s(0) = \emptyset$. For $1 \leq k \leq s$, we define $\delta_s(k)$ as follows: Let $t < s$ be the greatest stage such that $t = 0$ or $\delta_s\ \bar{}\ k \subseteq \delta_t$, let $\delta_s(k) = 0$ if $|X_{k,t}| < |X_{k,s}|$ and $\delta_s(k) = 1$ otherwise. For $\alpha \in 2^{<\omega}$, s is an α–stage if $s = 0$ or $\alpha \subseteq \delta_s$. Since the tree is finitely branching there exists a true path f (as defined in [35]), and

$$X_p = \omega \Leftrightarrow f(p) = 0.$$

If $\alpha \in 2^{<\omega}$, $|\alpha| = k$, we say that α has a *version* of the C_k–strategy and $\alpha\frown 0$ is has a version of the Q_k–strategy (recall that Q_k only has to be active if $X_k = \omega$, i.e., if $f(k) = 0$). We adopt the notation above in the obvious way. For instance, we write $i_C(\alpha)$ instead of $i_C(|\alpha|)$, etc. As in [35], $\omega^{[\alpha]}$ stands for $\omega^{[n(\alpha)]}$, where $\alpha \mapsto n(\alpha)$ is some effective numbering of strings.

If $k = \langle i, n, m \rangle$, let $\ell g(p,s) = \max\{x : (\forall z < x)(\psi_n^{G_i \oplus D}(z) = C(z)[s])\}$ be the length of agreement associated with Q_p. To avoid the problem explained above, $\alpha\frown 0$'s version of the Q_p strategy only considers numbers y as followers at stage s which are *eligible* for $\alpha\frown 0$, i.e.,

$$\begin{aligned} y \in \omega^{[\alpha]} \wedge \ell g(p,s) > y \wedge \Psi_n^{G_i \oplus D}(y) = 0[s] \wedge \\ (\forall \beta)[\beta\frown 0 \subseteq \alpha\frown 0 \Rightarrow \max X_{|\beta|} > \psi_n^{G_1 \oplus D}(y)[s]]. \end{aligned}$$

Note that, unlike the case of more involved tree constructions, the sequence $\langle \delta_s \rangle$ and hence the true path are predetermined by $\langle X_k \rangle$. This is the reason why we can prove that $C, D \leq_T A$: A can enumerate those nodes which are to the left of the true path (this follows from Lemma 7.2) and hence can give an upper bound on the stages at which a Q–strategy $\alpha^\frown 0$ can possibly enumerate a number into D.

We now present the formal construction and verification needed to prove the theorem.

Construction:

Stage 0. Let $C_0 = D_0 = \emptyset$. Initialize all strategies.
Stage s, $s > 0$. Initialize all strategies $\beta >_L \delta_s$. For each $k \leq s$ in order, carry out the substage k.
Substage k. Let $\alpha = \delta_s \upharpoonright k$, and let $t < s$ be the last α–stage. Let $i = i_C(k)$.

Actions for the C_k–strategy α. Firstly, if there is an x such that $G_{i,t} \upharpoonright \delta_{\alpha,t}(x) \neq G_{i,s} \upharpoonright \delta_{\alpha,t}(x)$, and some Q–strategy β, $\alpha \subset \beta$, is requesting clearing of $\Delta_\alpha(x)$, now let x be minimal such, declare $\Delta_\alpha(x')$ undefined for each $x' \geq x_n$ and cancel all clearing requests for such x'.

Secondly, for all x such that $\Delta_0(x)$ is now undefined, redefine it with value $C_s(x)$, and a large use $\delta_\alpha(x)$, so that δ_α is monotonic.

Suppose $k < s$ and $\alpha^\frown 0 \subseteq \delta_s$. We describe the actions for the Q_k–strategy $\alpha^\frown 0$. The strategy has as parameters a "killing point" $x > k$ and a chain $y_0 < \cdots < y_{n-1}$ of followers, where $x < y_0$.

Suppose $k = \langle i, n, m \rangle$.

Step 1. (Killing higher priority C requirements) If the killing point x is undefined, assign a large number as its value. If $\delta_\beta(x)$ is defined, where $\beta \subseteq \alpha$ and $i_C(\beta) = i_Q(\alpha)$, enumerate $\delta_\beta(x)$ into D and declare $\Delta_\beta(x')$ undefined for $x' \geq x$. [We continue this action even if it seems that Q_k has already been met.]

Step 2. (Cancelling followers) Let $t < s$ be the last $\alpha^\frown 0$–stage. If a follower y_p was already appointed at stage t, but G_i changed below, the use $\psi_n^{G_i \oplus D}(y_p)$, then cancel y_p and the clearing request associated with y_p.

Step 3. (Appointing a new follower) Let $\widetilde{y}$ be the maximal follower which is still uncancelled (and $\widetilde{y} = 0$ if no such follower exists), and let $\widetilde{s}$ be the last stage at which the Q_k–strategy $\alpha\hat{}0$ appointed $\widetilde{y}$ (and $\widetilde{s} = 0$ if $\widetilde{y} = 0$). Since $\oplus G_i$ is low, there is, by the recursion theorem, a total recursive function $g(\alpha, \widetilde{y}, \widetilde{s}, t)$ such that $\lim_t g(\alpha, \widetilde{y}, \widetilde{s}, t)$ exists, equals 0 or 1, and the limit is 1 iff the following $\Sigma^0_1(G_i)$–question has an affirmative answer:

$$(\exists s > \widetilde{s})(\exists y > \widetilde{y})[\text{at stage } s \text{ of our construction}$$
$$y \text{ is eligible for } \alpha\hat{}0 \text{ via a } G_i\text{–correct computation}]. \tag{7.6}$$

Now, at the actual substage k of stage s of the construction, look for a minimal number $y > \widetilde{y}$ which is eligible for $\alpha\hat{}0$. Find the least number $s' \geq s$ such that either G_i changes below the use $\psi_n^{G_i \oplus D}(y)[s]$ by stage s' or $f(\alpha, \widetilde{y}, \widetilde{s}, s') = 1$. In the second case, we view y as certified and therefore appoint y as a new follower. To restrain C and D, initialize all strategies β such that $\alpha\hat{}0 \subset \beta$ or $\alpha\hat{}0 <_L \beta$.

Step 4. (Diagonalization) If there is a follower y which was already appointed at the last $\alpha\hat{}0$ stage $t < s$, remained uncancelled since t, is β cleared for all $\beta \subseteq \alpha$, and moreover, $A_t \upharpoonright y \neq A_s \upharpoonright y$, then let y be the least such follower. To diagonalize, enumerate y into C. For each $\beta \subseteq \alpha$, enumerate $\delta_\beta(y)$ (if defined) into D and declare $\Delta_\beta(y')$ undefined for $y' \geq y$.

Verification:

Lemma 7.3. *Suppose that $\alpha\hat{}0$ is a Q_k–strategy on the true path. Then the action of $\alpha\hat{}0$ in Steps 2, 3 and 4 is finitary, and Q_k is met.*

Proof. Let $k = \langle i, n, m \rangle = |\alpha|$.

By induction, let s_0 be the least $\alpha\hat{}0$ stage such that $\alpha\hat{}0$ is not initialized at any stage $s \geq s_0$. Assume for a contradiction that

$$\text{for each } p,\ \alpha\hat{}0 \text{ eventually appoints a follower } y_p \text{ at stage } s \geq s_0$$
$$\text{and this follower is never cancelled.} \tag{7.7}$$

We call such a follower *stable*. Since $\alpha\hat{}0$ has highest priority from s_0 on, stability can be recognized recursively in G_i.

Claim: Assume (7.7). Then there are infinitely many stable followers of $\alpha\frown 0$ which are eventually β–cleared for all $\beta \subseteq \alpha$.
Proof of Claim: Since the follower is automatically β-cleared if $i_C(\beta) = i_Q(\alpha)$ by the definition of eligibility, we only need to consider those β such that $i_C(\beta) \neq i_Q(\alpha)$. For these β, we argue by downward induction on the length of β. Suppose the claim is true for all β', $\beta \subset \beta' \subseteq \alpha$, and $j = i_C(\beta) \neq i_Q(\alpha)$. If after stage $s_1 \geq s_0$ no more stable followers which are already β' cleared for $\beta \subset \beta'$ become β–cleared, then $G_j \leq_T G_i$: Given an input z, G_i–recursively find an $\alpha\frown 0$ stage $s \geq s_1$, such that a stable $y > z$ has been appointed which is β'–cleared for all $\beta' \supset \beta$. Since $\alpha\frown 0$ requests clearing of $\Delta_\beta(y)$, any $G_j \restriction y$ change would lead to β–clearing of $\Psi_n^{G_i \oplus D}(y)$. So $G_{j,s}(z) = G_j(z)$.

To obtain a contradiction from (7.7) we now argue that $\alpha\frown 0$ could successfully diagonalize in Step 4 through a stable completely cleared y. Since $A\ \Theta_T\ G_i$, by a similar argument as above, there must be an $\alpha\frown 0$–stage $s \geq s_0$ where a stable y which is cleared for all $\beta \subseteq \alpha$ has been appointed such that $A \restriction y$ changes by the next $\alpha\frown 0$ stage. So in Step 4, $\alpha\frown 0$ diagonalizes through the least such y_1 which causes the length of agreement associated with Q_k to be permanently below y, contrary to the assumption (7.7) .

Since (7.7) fails, there is a pair $(\tilde{y}, \tilde{s})$ such that $\tilde{y}$ is the maximum stable follower ever appointed after stage s_0, and it is appointed at the $\alpha\frown 0$–stage, $\tilde{s}$ for the last time (let $\tilde{y} = 0$, $\tilde{s} = s_0$ if no stable follower exists at all). We claim that there can be no pair of witnesses y, s for (7.6) as otherwise $\lim_s f(\alpha, \tilde{y}, \tilde{s}, s) = 1$. Choose a minimal $y > \tilde{y}$ which witnesses (7.6) , and let $s > \tilde{s}$ be the first $\alpha\frown 0$ stage where $\Psi_n^{G_i \oplus D}(y)$ converges and is G_i–correct. Then we would appoint y as the next follower $> \tilde{y}$ at stage s: by the properties on the function Ψ_n described in the beginning, if we had appointed some $y' > y$ at the stage s', $\tilde{s} < s' < s$ and it were uncancelled, we would have seen the computation $\Psi_n^{G_i \oplus D}(y)[s]$ already at the stage s'. Moreover, if some y', $\tilde{y} < y' < y$ were appointed at s, then $\Psi_n^{G_i \oplus D}(y')$ would also be G_i–correct, contrary to the minimality of y. Finally, $G_i \restriction \psi_{n,s}(y)$ is stable, so we can appoint y at s. A new stable follower has been found, contrary to the definition of $\tilde{y}$.

Since there is no witness to (7.6), $\lim g_t(\alpha, \tilde{y}, \tilde{s}, t) = 0$, so $\alpha\frown 0$ will stop all actions through Steps 2, 3 (and hence through 4) at some point. To finish the proof of Lemma 1, we show that Q_k is met: If $\Psi_n^{G_i \oplus D} = C$, there is a number $y \in \omega^{[\alpha]}$ above all numbers in $\omega^{[\alpha]}$ that are ever appointed by $\alpha\frown 0$ such that $\Psi_n^{G_i \oplus D}(y) = 0$. But then the answer to (7.6) is "yes"! □

Lemma 7.4. $i \in S \Leftrightarrow C \leq_T G_i \oplus D$.

Proof. For one direction suppose that $i \notin S$. For arbitrary n, we show that $C \neq \Psi_n^{G_i \oplus D}$. Let m be the number such that $X_{\langle i,n,m\rangle} = \omega$. Then, for some α, $|\alpha| = \langle i, , m\rangle$, $\alpha\hat{\ }0$ is on the true path. So, by Lemma 7.3 , $Q_{\langle i,n,m\rangle}$ is met.

Now suppose that $i \in S$. The, by (7.1) , there exists a k, $i_C(k) = i$, such that the hypothesis of C_k is correct. Let α on the true path have a version of the C_k–strategy. Suppose α is not initialized after s_0. Since a Q–strategy $\beta\hat{\ }0$, $\alpha \subseteq \beta$, enumerates $\delta_\alpha(y)$ into D whenever it enumerates y into C, after stage s_0 the functional Δ_α can always be redefined correctly. It remains to verify that Δ_α is not partial, i.e., $\Delta_\alpha(y)$ is only declared undefined finitely often for each y. $\Delta_\alpha(y)$ can be declared undefined for two reasons:
(1) Some Q–strategy $\beta\hat{\ }0$, $\alpha \subseteq \beta$, requests clearing and its killing point is $\leq y$. This can happen only finitely often, by initialization if $\beta\hat{\ }0$ is to the right of the true path, and by Lemma 7.3 if $\beta\hat{\ }0$ is on the true path.
(2) If $i_Q(\beta) = i$ for some β, $\alpha \subseteq \beta$, the killing point of the Q–strategy $\beta\hat{\ }0$ is $\leq y$ (hence $|\beta| \leq y$) and $X_{|\beta|}$ increases, i.e., $\beta\hat{\ }0 \subseteq \delta_t$. By the choice of k, this can only happen finitely often. □

Lemma 7.5. *$C, D \leq_T A$*

Proof. We first analyze what A knows about the true path.

Claim. There is an A–recursive function $p(\alpha, t_0)$ such that, for each t_0, $p(\alpha, t_0)$ is the first stage $t \geq t_0$ such that one of the following happens:

(a) $\alpha \subseteq \delta_t$, or

(b) $\delta_t <_L \alpha$, or

(c) $\alpha <_L \delta_{t'}$ for all $t' > t$.

We define p by recursion on $k = |\alpha|$. Let $p(\emptyset, t_0) = t_0$, and, for each t, let $p(\alpha\hat{\ }1, t_0) = p(\alpha, t_0)$ (in these cases, (a) through (c) are satisfied). Now let

$$p(\alpha\hat{\ }0, t_0) = p(\alpha, \tilde{t}),$$

where $\tilde{t}$ is the first stage $\geq t_0$ where $|X_k|$ increases or we have determined in $\tilde{t}$ stages, using the computation procedure from Lemma 7.3 with A as an oracle, that $X_k = X_{k,\chi(k)}$ (in which case $\chi(k) < \tilde{t}$, so also $X_k = X_{k,\tilde{t}}$).

To verify the required properties, if $\alpha <_L \delta_t$, for $t > p(\alpha, \tilde{t})$, then also $\alpha\hat{\ }0 <_L \delta'_t$. If $\delta_{p(\alpha,\tilde{t})} <_L \alpha$, then $\delta_t <_L \alpha\hat{\ }0$. Finally if (a) holds for α, i.e., $\alpha \subseteq \delta_{p(\alpha,\tilde{t})}$, then we consider two cases: if $\tilde{t}$ is the first stage where $|X_k|$ increases, then $\alpha\hat{\ }0 \subseteq \delta_{p(\alpha,\tilde{t})}$. If we found out that $X_k = X_{k,\tilde{t}}$, then $p(\alpha, \tilde{t})$ is an upper bound for the last stage t

where a previous increase of $|X_k|$ can possibly lead to $\alpha^\frown 0 \subseteq \delta_t$, so we are in case (c) for $\alpha^\frown 0$.

Now to prove that $C \leq_T A$, consider an input $y \in \omega^{[\alpha]}$. If y ever enters C it must enter at an $\alpha^\frown 0$–stage, where $A \upharpoonright y$ changed since the last $\alpha^\frown 0$–stage . Let t_0 be such that $A \upharpoonright y = A_{t_0} \upharpoonright y$, and evaluate $t = p(\alpha, t_0)$ recursively on A. Then t is the last stage where this C–enumeration can take place. So $C \leq_T A$.

Finally, to prove that $D \leq_T A$, first apply a similar argument to the D–enumeration of $y = \delta_\beta(z)$ along with a diagonalization of a Q–strategy through z (note that $y \geq z$). If y is enumerated because $y = \delta_\beta(x)$ and x is the killing point of some Q–strategy $\alpha^\frown 0$, $\beta \subseteq \alpha$, this strategy can be identified by stage y. So $p(\alpha, y)$ is an upper bound for any stage at which such an enumeration of y into D can take place. Λ

8. References

1. K. Ambos-Spies [ta], Automorphism bases of the r.e. degrees, preprint, University of Heidelberg .

2. K. Ambos-Spies and P. Fejer [1988], Degree theoretical splitting properties of the recursively enumerable sets, *J. Symb. Logic* **53**, 1110-1137.

3. K. Ambos-Spies, C. G. Jockusch, R. A. Shore and R. I. Soare [1984], An algebraic decomposition of the recursively enumerable degrees and the coincidence of several degree classes with the promptly simple degrees, *Trans. Amer. Math. Soc.* **281**, 109-128

4. K. Ambos-Spies, L. Lempp and R. I. Soare [ta], Intervals of recursively enumerable degrees: Lattice embeddings and non-$\aleph_0$-categoricity of the partial order, in *Proceedings of Logic Colloquium '97*, S. B. Cooper et al., eds., to appear.

5. S. B. Cooper [1993], On a conjecture of Kleene and Post, *University of Leeds, Department of Pure Mathematics Preprint Series* No. 7.

6. S. B. Cooper [1996], Beyond Gödel's theorem: the failure to capture information content, *University of Leeds, Department of Pure Mathematics Preprint Series*, No. 4.

7. R. G. Downey and Shore, R. A. [1995], Degree theoretic definitions of the low_2 recursively enumerable sets, *J. Symb. Logic* **60** (1995), 727-756.

8. R. G. Downey and Shore, R. A. [1996], Lattice embeddings below a nonlow_2 recursively enumerable degree, *Israel J. Math.* **94**, 221-246.

9. L. Harrington and A. Nies [ta], Coding in the lattice of recursively enumerable sets, *Advances in Mathematics*, to appear.

10. L. Harrington and S. Shelah [1982], The undecidability of the recursively recursively enumerable degrees (research announcement), *Bull. Amer. Math. Soc.* **6**, 79-80.

11. W. Hodges [1993], *Model Theory*, Cambridge University Press, Cambridge, England.

12. W. Hodges and A. Nies [ta], Interpreting infinite linear orders, in *Proceedings of Logic Colloquium '95*, J. Makowsky ed., Springer-Verlag, to appear.

13. Jockusch, Jr., C.G. and D. Posner [1978], Double jumps of minimal degrees, *J. Symb. Logic* **43**, 715-724.

14. Jockusch, Jr., C.G. and Simpson, S. G. [1976], A degree theoretic definition of the ramified analytic hierarchy, *Ann. Math. Logic* **10**, 1-32

15. S. Lempp, A. Nies and T. Slaman [1996], The Π_3-theory of the partial order of r. e. T-degrees is undecidable, *Trans. Amer. Math. Soc.*, to appear.

16. M. Lerman [1973], Admissible ordinals and prority arguments, in *Cambridge Summer School in Mathematical Logic*, LNMS **37**, Springer-Verlag, Berlin, 311-344.

17. M. Lerman [1983], *Degrees of Unsolvability*, Springer-Verlag, Berlin.

18. M. Lerman, R. A. Shore and R. I. Soare [1984], The Elementary Theory of the Recursively Enumerable Degrees is not $\aleph_0$-categorical, *Adv. Math.* **3**, 301-320.

19. A. Nerode and R. A. Shore [1980], Reducibility orderings: theories, definability and automorphisms, *Ann. Math. Logic* **18**, 61- 89.

20. A. Nies [1992], *Definability and Undecidability in Recursion Theoretic Semilattices*, Ph.D. Thesis, Universität Heidelberg.

21. A. Nies [1994], The last question on recursively enumerable many one degrees, *Algebra i Logika* **33**(5), 550-563.

22. A Nies [1995], Undecidable fragments of elementary theories, *Algebra Universalis* **35**, 8-33.

23. A. Nies [1996], On a uniformity in degree structures, in *Complexity, Logic and Recursion Theory*,A.Sorbi, ed., Lecture Notes in Pure and Applied Mathematics, **187**, (1997), 261-276.

24. A. Nies [ta], Flexible coding of standard models of arithmetic, preprint, University of Chicago.

25. G. Odifreddi and R. A. Shore [1991], Global properties of local structures of degrees, *Bul. U. Mat. Ital.*, **7**, 97-120.

26. R. A. Shore [1981], The theory of the degrees below $\mathbf{0}'$, *J. London Math. Soc* (2) **24**, 1-14.

27. R. A. Shore [1982], On homogeneity and definability in the Turing degrees, *J. Symb. Logic* **47**, 8-16.

28. R. A. Shore [1982], Finitely generated codings and the degrees r. e. in a degree $\mathbf{d}$, *Proc. Am. Math. Soc.* **84**, 256-263.

29. R. A. Shore [1988], Defining jump classes in the degrees below $\mathbf{0}'$, *Proc. Am. Math. Soc.* **104**, 287-292.

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

31. R. A. Shore and Slaman T. A. [1993], Working below a high recursively enumerable degree, *J. Symb. Logic* **58**, 824-859.

32. S. G. Simpson [1977], First order theory of the degrees of recursive unsolvability, *Ann. Math.* (2), **105**, 121-139.

33. T. A. Slaman [1991], Degree structures, in *Proc. Int. Cong. Math., Kyoto 1990*, Springer-Verlag, Tokyo, 303-316.

34. T. A. Slaman and H. W. Woodin [1986], Definability in the Turing degrees, *Illinois J. Math.* **30**, 320-334.

35. R. Soare [1987], *Recursively enumerable sets and degrees*, Springer-Verlag, Berlin.

36. R. Soare and M. Stob [1982], Relative recursive enumerability, in *Procceedings of the Herbrand Symposium: Logic Colloqium '81*, J. Stern ed., North-Holland, Amsterdam, 299-324.