The n-r.e. degrees: undecidability and Σ_1 substructures

Mingzhong Cai †
Department of Mathematics
Cornell University
Ithaca NY 14853

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

Theodore A. Slaman‡
Department of Mathematics
University of Califonia, Berkeley

August 11, 2010

1 Introduction

Turing reducibility (introduced in Turing [1939]) captures the intuitive notion of one set $A \subseteq \mathbb{N}$ being computable from another B, We write $A \leq_T B$, A is *Turing reducible* to or *computable from* B to mean that there is a Turing machine (program) Φ that can compute A if given access to an "oracle" for B in the sense that the computing machine is augmented by a procedure that allows it to ask for any number n it computes if $n \in B$ and to receive the correct answer. This reducibility naturally induces a partial order $\leq_T$ on the set $\mathcal{D}$ of equivalence classes (called *Turing degrees* or simply *degrees*) $\mathbf{a} = \{B | A \leq_T B \,\&\, B \leq_T A\}$. The structure of $\mathcal{D}$ then captures that of relative complexity of computation of sets and functions (on $\mathbb{N}$). The study of this relation on all sets (functions), and on many important subclasses of sets has been a major occupation of recursion (computability) theory ever since its introduction.

In addition to the full structure, $\mathcal{D}$, the most important substructures studied have been those of the recursively enumerable degrees, $\mathcal{R}$, and $\mathcal{D}(\leq \mathbf{0}')$, the degrees below

†Partially supported by NSF Grant DMS-0852811.
*Partially supported by NSF Grants DMS-0554855, DMS-0852811 and John Templeton Foundation Grant 13408.
‡Partially supported by NSF Grant DMS-1001551 and by the John Templeton Foundation.

the halting problem, $K = \{e|\Phi_e(e) \text{ converges}\}$ whose degree is denoted by $\mathbf{0}'$. The recursively enumerable sets are those which can be enumerated (listed) by a recursive (computable) function. They can also be seen as those sets A for which there is a very simple approximation procedure, a recursive function $f(x,s)$ to the characteristic function $A(x)$ of A such that $\forall x(f(x,0) = 0 \ \& \ \lim f(x,s) = A(x))$ that changes its mind about membership in A at most once, i.e. there is at most one s such that $f(x,s) \neq f(x,s+1)$. Shoenfield's Limit Lemma [1959] says that the sets (or functions) computable from the halting problem $0'$ are precisely those with some convergent recursive approximation, i.e. the sets A such that there is a recursive function $f(x,s)$ such that $\forall x(f(x,0) = 0 \ \& \ \lim f(x,s) = A(x))$. So, while for each x there are only finitely many changes, the number of such changes over all x may be unbounded.

In this paper we study a natural hierarchy of intermediate classes of sets and degrees. The n-r.e. sets are those for which there is a recursive approximation $f(x,s)$ as above for which there are at most n changes of value at each x. The corresponding degree structures are denoted $\mathcal{D}_n$, the degrees of the n-r.e. sets. (So $\mathcal{D}_1 = \mathcal{R}$ the r.e. degrees.) This hierarchy was introduced by Putnam [1965] and Gold [1965]. It was extended into the transfinite by Ershov [1968, 1968a, 1970] who proved that the sets in the transfinite hierarchy he defined are precisely those computable from $0'$.

The early work on degree theories began with the investigation of local algebraic or order-theoretic properties of the structures. This work continues in full force to this day. In the past three decades or so, a more global approach has emerged as well. Here one studies issues such as the decidability or, more generally, the complexity of the theories of degree structures as well as related questions about definability in, and possible automorphisms of, these structures.

For the first couple of decades, a major motivating idea was that (at least some of) these structures should be simple and characterizable by basic algebraic properties. Shoenfield's conjecture [1965] would have been such a complete characterization of $\mathcal{R}$ analogous to that of the rationals as the countable dense linear order without endpoints. Even after the conjecture had been refuted by Lachlan [1966] and Yates [1966], Sacks [1966] still conjectured that the r.e. degrees were decidable. More recent results have produced a dramatically different prevailing paradigm for $\mathcal{D}$, $\mathcal{D}(\leq_T \mathbf{0}')$ and $\mathcal{R}$ as well as many degree structures for other notions of reducibility. Rather than seeing the complexity of the structures as an obstacle to characterization, it suggests that a sufficiently strong proof of complexity would completely characterize each structure. Instead of expecting the structures to be decidable and homogeneous with many automorphisms (like the rationals), one looks to prove that the theories are as complicated as possible, there are definable degrees and that the structure has few automorphisms.

Typical results include the following:

Theorem 1.1. *$\mathcal{D}$, $\mathcal{D}(\leq_T \mathbf{0}')$ and $\mathcal{R}$ are each undecidable by Lachlan [1968]; Epstein [1979] and Lerman [1983]; and Harrington and Shelah [1982], respectively.*

Theorem 1.2. *The theories of $\mathcal{D}$, $\mathcal{D}(\leq_T \mathbf{0}')$ and $\mathcal{R}$ are as complicated as possible,* *i.e. recursively isomorphic to true second order arithmetic for $\mathcal{D}$ and to true first order arithmetic for $\mathcal{D}(\leq_T \mathbf{0}')$ and $\mathcal{R}$ by Simpson [1977]; Shore [1981]; and Harrington and Slaman and then Slaman and Woodin (both unpublished) (see Nies, Shore and Slaman [1998] for a proof and stronger results), respectively.*

Theorem 1.3. *All relations invariant under the double jump that are definable in arithmetic are definable in $\mathcal{D}$, $\mathcal{D}(\leq_T \mathbf{0}')$ and $\mathcal{R}$ where for $\mathcal{D}$ we mean second order arithmetic and for the others first order by Slaman and Woodin [2001] (see Slaman [1991] for an announcement and Shore [2007] for a quite different proof that applies to various substructures of $\mathcal{D}$ as well), essentially Shore [1988] (but see also Nies, Shore and Slaman [1998, Theorem 3.11 and the remarks following it])and Nies, Shore and Slaman [1998], respectively. (The converse holds by the definability of these degree structures in arithmetic.)*

A survey paper for this area is Shore [2006].

In this paper we take the first steps on this road for the structures $\mathcal{D}_n$ by proving that they are all undecidable. We conjecture that our work can be extended along the lines of Nies, Shore and Slaman [1998] to show that their theories are also all recursively isomorphic to that of true arithmetic. Perhaps one can even prove definability results as done there for $\mathcal{R}$. Basic survey papers on the structure of the $\mathcal{D}_n$ are Arslanov [2009, 2010] and Stephan, Yang and Yu [2009].

Another important theme in the study of these degree structures has been delimiting the similarities and explicating the differences among them. While it is relatively easy to distinguish among $\mathcal{D}$, $\mathcal{D}(\leq_T \mathbf{0}')$ and $\mathcal{R}$ in many way the issue becomes particularly compelling when we turn to the $\mathcal{D}_n$. It is easy to imagine, and was proved early on, that moving from $\mathcal{R}$ to all sets or even to the unlimited approximations characterizing those below $0'$ introduces many differences. For the $\mathcal{D}_n$, however, the question is what does the ability to change precisely one more time buy us in terms of additional degrees, algebraic structure and complexity.

Of course, the first question is are the $\mathcal{D}_n$ actually distinct. Indeed, there are, for each n, $(n+1)$-r.e. degrees which are not n-r.e. ([Cooper [1971] with the stronger result that they can be found not even n-rea in Jockusch and Shore [1984]]). While the one quantifier theory of all the degree structures from $\mathcal{R}$ to $\mathcal{D}$ are the same since one can embed all finite (even countable) partial orderings into $\mathcal{R}$ (and so all the rest as well), there were many early results establishing elementary differences between $\mathcal{R}$ and the other $\mathcal{D}_n$ with cupping, density and lattice embedding properties playing the featured role (as in, for example, Arslanov [1985] Cooper et al. [1991], Downey [1989], respectively). Differences between any of the other $\mathcal{D}_n$, however, seemed hard to find. Downey [1989] even conjectured that they might all be elementarily equivalent, i.e. all sentences (in the first order language with $\leq$) true in any $\mathcal{D}_n$ for $n \geq 2$ is true in all of them. This conjecture was not refuted until quite recently. Arslanov, Kalimullin and Lempp [2010] provide an elementary difference between $\mathcal{D}_2$ and $\mathcal{D}_3$. In fact, the sentence they exhibit on which the structures differ is at the smallest possible level: two quantifiers ($\forall\exists$). They conjecture (as one would now expect) that the $\mathcal{D}_n$ are pairwise not elementarily equivalent. They also conjecture that this level of difference ($\forall\exists$) is as small as possible in the strong sense that every $\exists\forall$ sentence true in any $\mathcal{D}_n$ is true in every $\mathcal{D}_m$ for $m \geq n$.

Now an $\exists\forall$ sentence is true if there are choices (parameters substitutable) for the existentially quantified variables such that the resulting universal sentence is true of these parameters. The strongest way that their conjecture could be true is for the same parameters to work in both structures. This view brings to mind a much earlier question raised about other pairs of our degree structures. Are any Σ_1 substructures of any others. ($\mathcal{M}$ is a Σ_1 *substructure* of $\mathcal{N}$, $\mathcal{M} \preceq_1 \mathcal{N}$, if for any Σ_1 formula $\exists \bar{y}\varphi(\bar{x}, \bar{y})$ where φ is quantifier free and any choice of elements $\bar{a}$ from $\mathcal{M}$, $\mathcal{M} \vDash \exists \bar{y}\varphi(\bar{a}, \bar{y}) \Leftrightarrow \mathcal{N} \vDash \exists \bar{y}\varphi(\bar{a}, \bar{y})$.)

Slaman ([1983]) proved early on that this fails at the extreme ends: $\mathcal{R} \npreceq_1 \mathcal{D}(\leq \mathbf{0}')$ (and so, *a fortiori*, $\mathcal{D}_n \npreceq_1 \mathcal{D}(\leq \mathbf{0}')$ for any $n \geq 1$. Slaman and then others raised the natural question of whether it could be that $\mathcal{D}_n \preceq_1 \mathcal{D}_m$ for any $n < m$. Yang and Yu [2006] provided a negative answer for $n = 1$ and $m = 2$ (and so for any $m \geq 2$). We complete the picture by showing that $\mathcal{D}_n \npreceq_1 \mathcal{D}_m$ for any $n < m$. (We have just heard that Arslanov and Yamaleev are preparing a different proof for the case $n = 2$.)

Turning now to our proofs, we begin with undecidability. As usual (see for essentially our situation §2 of Nies, Shore and Slaman [1998] or for a more general model theoretic treatment Hodges [1993, 5.3]), we have a formula $\varphi_D(x, \bar{p})$ which, for each choice of parameters $\bar{p}$, defines a subset D of our structure $\mathcal{D}_n$ and another formula $\varphi_R(x, y, \bar{p})$ which defines a binary relation R on D. To prove undecidability it suffices to show that, as the parameters vary over $\mathcal{D}_n$, a sufficiently rich class of structures (D, R) are coded in this way. In our case, we code partial orders. As the (r.e.) set of theorems of the theory of partial orders is recursively inseparable from the (r.e.) set of sentences (of the language of partial orders) that are false in some finite partial order (Taitslin [1962]), it suffices to code any collection of relations containing all finite partial orders. The point here is that if $\mathcal{D}_n$ were decidable then the set of sentences true in every partial order coded by φ_D and φ_R as the parameters $\bar{p}$ range over all elements of $\mathcal{D}_n$ would be recursive. Of course, it contains the theorems of the theory of partial orders and, if we code all finite ones, is disjoint from the set of sentences with finite counterexamples. As it turns out, it is no more difficult to prove that one can code all recursive partial orders than all finite ones. This is what we do explicitly in our proof of Theorem 1.4. We use the basic idea of the domain being maximal degrees $\mathbf{g} \leq_{\mathbf{T}} \mathbf{a}$ not joining some other degree $\mathbf{p}$ above $\mathbf{q}$ from Harrington and Shelah [1982] and build on their work.

Theorem 1.4. *Given a recursive partial order* $(\omega, \leq_*)$ *and an* $n \geq 1$, *there exist uniformly n-r.e. sets* G_i *for each* $i \in \omega$, *an n-r.e. set* L *and r.e. sets* P *and* Q *such that:*

1. *Each* $\mathbf{g}_i$ *is a maximal n-r.e. degree below* $\mathbf{a}$ *such that* $\mathbf{q} \nleq \mathbf{g}_i \vee \mathbf{p}$ *where* $A = \bigoplus_i G_i$.
2. $\mathbf{g}_i \leq \mathbf{g}_j \vee \mathbf{l}$ *if and only if* $i \leq_* j$.

Thus the required formulas φ_D and φ_R defining our domains and order relations have parameters $\mathbf{a}$, $\mathbf{p}$ and $\mathbf{q}$. The first says that $\mathbf{x}$ is a maximal degree below $\mathbf{a}$ such that $\mathbf{q} \not\leq \mathbf{x} \vee \mathbf{p}$. The second says that $\mathbf{x} \leq \mathbf{y} \vee \mathbf{l}$. so we have the desired result.

Theorem 1.5. *The theories of $\mathcal{D}_n$ are undecidable for every n.*

If instead of recursive inseparability, we wanted to rely only on the undecidability of the theory of partial orders, we should code all partial orders recursive in $0'$ as every sentence which is not a theorem (of the theory) has a counterexample recursive in $0'$ by the effective version of the completeness theorem.

One can with only minor modifications not affecting the structure of our proof handle partial orders recursive in $0'$. We precisely describe the modifications needed in 6.5. With some additional work and a serious reorganization of the priority tree, one can get all partial orders recursive in $0''$. One puts in a new type of node which guesses in a Δ_3 procedure at each bit of information about this partial order and bases later work on these guesses. The added complexity is considerable without much gain for applications. It seems that one can even get any Σ_3 partial order by a slightly more complicated procedure. We briefly describe this procedure in 6.5 as well.

It is worth remarking that our proof works for $n = 1$ as well as all larger n. Indeed, it can be significantly simplified for $n = 1$ by omitting all items that consider the possibility that the G_i and W_i (the list of n-r.e. sets recursive in A) are not r.e. This gives a considerably simplified proof of the undecidability of $\mathcal{R}$ along the lines suggested in Harrington and Shelah but with a simpler statement using fewer parameters and a significantly easier construction. We do not believe any proof even for $\mathcal{R}$ along these lines has been published before.

We next turn to Σ_1 substructures.

Theorem 1.6. *$\mathcal{D}_n \not\preceq_1 \mathcal{D}_m$ for $n < m$.*

The technical result needed here is the following generalization of Theorem 1.12 in Yang and Yue [2006] who do the case $n = 1$:

Theorem 1.7. *For any $n \geq 1$, there are r.e. degrees $\mathbf{g}, \mathbf{p}, \mathbf{q}$, an n-r.e. degree $\mathbf{a}$ and an $n+1$-r.e. degree $\mathbf{d}$ such that:*

1. *For every n-r.e. degree $\mathbf{w} \leq \mathbf{a}$, either $\mathbf{q} \leq \mathbf{w} \vee \mathbf{p}$, or $\mathbf{w} \leq \mathbf{g}$.*
2. *$\mathbf{d} \leq \mathbf{a}$, $\mathbf{q} \not\leq \mathbf{d} \vee \mathbf{p}$, and $\mathbf{d} \not\leq \mathbf{g}$.*

This theorem shows directly that $\mathcal{D}_n$ is not a Σ_1 elementary substructure of $\mathcal{D}_{n+1}$ in the language with $\vee$ as well as $\leq$: In $\mathcal{D}_n$, no $\mathbf{w}$ below $\mathbf{a}$ has the property that $\mathbf{q} \not\leq \mathbf{w} \vee \mathbf{p}$ and $\mathbf{w} \not\leq \mathbf{g}$ while in $\mathcal{R}_{n+1}$, $\mathbf{d} \leq_T \mathbf{a}$ has both properties. We can eliminate $\vee$ by rephrasing the property of $\mathbf{w}$ as $\exists \mathbf{z}(\mathbf{w}, \mathbf{p} \leq \mathbf{z} \;\&\; \mathbf{q} \not\leq \mathbf{z}) \;\&\; \mathbf{w} \not\leq \mathbf{g}$ which is Σ_1 in just $\leq$ and so the existence of a $\mathbf{w}$ with this property is true in $\mathcal{D}_{n+1}$ (i.e. $\mathbf{d}$) but false in $\mathcal{D}_n$. Of course, as $\mathbf{d}$ is in $\mathcal{D}_m$ for every $m \geq n+1$, $\mathcal{D}_n \not\preceq_1 \mathcal{D}_m$ as well.

Much of the construction and verification is the same for Theorems 1.4 and 1.7. We treat the first theorem as primary. In §2 where we cover basic notions and conventions common to both, we use curly brackets {} to indicate changes (usually alphabetic only at this stage) for the second theorem. The rest of the paper is divided into two parts, one for each of the theorems. Each part describes first the requirements (§3, 7), then the priority tree (§4, 8), the construction (§5, 9) and finally the verifications that the construction succeeds (§6, 10). We describe everything in full detail for the first theorem and then for the second describe only the changes needed. In our descriptions of the constructions, material enclosed in square brackets [] is meant to convey intuition or describe aspects of the construction that will only be verified later. It is not part of the formal definition of the construction procedures.

As might be expected from the types of requirements, both constructions are $0'''$ arguments even for the case $n = 1$. As these constructions go, however, ours are at the simpler end: the priority tree is finitely branching, there is no backtracking and only one type of requirement is injured along the true path. The key idea for carrying the arguments from the r.e. case ($n = 1$) to the n-r.e. one ($n > 1$) in Theorem 1.4 is what we call *shuffling* (§5.2). Roughly speaking, at the crucial $0'''$ determined nodes, we are attempting to construct functionals Δ that, to working towards the maximality of the $\mathbf{g}_i$, try to compute some given n-r.e. set $W = \Phi(A)$ from one G_i that we are building over the full construction. The most delicate part of the verification of the first construction is the correctness of these functionals (Lemma 6.8). We argue that the cause of an incorrect computation, say of $\Delta(u)$, must be that some number z entered A for the first time and allowed $W(u)$ to change. Another delicate argument shows that if W also changed for the first time, we could correct the functional Δ (or see that we are not on the true path). If the change in W was not that u entered for the first time, we argue that we can shuffle A between two past values (giving, via Φ, two different values for $W(u)$) by repeatedly taking z out and putting it back in as necessary so as to eventually show that $W \neq \Phi(A)$.The point here is that z has entered A for the first time while the change in W is not a first change. Thus as W_l can make no more than n changes overall, it can make no more than $n-2$ additional changes. On the other hand, as z has entered A for the first time, we can make $n-1$ more changes in A and so eventually guarantee that $W \neq \Phi(A)$.

In the second construction (§9.2.1), the correctness of the functionals Δ becomes immediate as we simply change G when necessary. The crucial problem then becomes guaranteeing the correctness of computations from G diagonalizing against D (§10). Here we take advantage of the fact that D can change one more time than any other set by using a procedure like one used in Yang and Yu [2006] to remove a number (that entered for the first time) from D. In our case it allows us to either cure some problem we are facing or start a shuffling procedure for A diagonalizing against the offending W.

2 Basic Notions and Conventions

Given a set A, let $A \upharpoonright u$ be the initial segment of A of length u.

We use upper case Greek letters to denote Turing functionals. For any Turing functional Δ, the *use* of a convergent $\Delta(A;x)$ is defined as the least number u such that $\Delta(A \upharpoonright u; x) \downarrow$. We use lower case Greek letters corresponding to the Turing functional to denote the use, e.g. $\delta(A;x)$ denotes the use of Δ at x. More importantly, we injure the computation by adding $\delta(A;x)-1$ into A, but not by adding $\delta(A;x)$ into A. If it doesn't cause confusion, we may omit A and write $\delta(x)$.

We will have families Ψ, Π, Θ and Φ which specify standard enumerations of all the Turing functionals. We follow the usual conventions for such standard enumerations such as the approximations to these functionals for any (approximation to an) oracle set at stage s asks questions about (makes use of) only numbers less than s and converges only at inputs less than s. We also assume, without loss of generality, that for the standard enumerations with two oracles such as $\Theta(G \oplus P; x)$ the uses on both are always the same and we denote it by $\theta(x)$.

We will also construct two families of Turing functionals $\Delta(G)$ and $\Gamma(W \oplus P)$. For the ones with two oracles, we *do not* require that the uses of W and P are the same. Hence we can write $\gamma(W;x)$ and $\gamma(P;x)$ to denote the W and P parts of the use, respectively. Although for simplicity we generally work as if we are specifically defining these oracles at each individual x with the associated uses, we really are assuming that the uses are monotonic in x and make all changes to keep them that way, usually without explicit mention. As Q is r.e., when we are computing it from $W \oplus P$ by Γ, except for this monotonicity condition, we only need to produce computations (axioms) that at x give output 0 when $x \notin Q$. These may be injured and new ones put into Γ (perhaps with larger use). In the case that $x \notin Q$ and we are expecting Γ to compute Q, we must eventually settle on a convergent computation (axiom) applying to $W_i \oplus P$. If $x \in Q$, when x enters Q it suffices to kill any current computation of 0 from $W_i \oplus P$. We do this by putting a number less than the P-use into P. We can then simply keep the value of Γ at 1 without changing the use (remembering that P is r.e.).

In our two constructions, we specify priority trees which grow downward. At each stage s of the construction, we build a path of length s {at most s} of *accessible* nodes along the priority tree. Our convention is that, the nodes to the left of, or above, a node α have higher priority. We always preserve the information used at previous stages by the nodes that are to the left of the accessible ones by initializing the nodes that are to the right of the accessible ones, i.e., remove all information from previous stages such as witness numbers, defined functionals and imposed restraints.

Nodes can impose two types of restraint: a permanent one or an alternating one. Permanent restraint means that no node of lower priority can act so as to injure the restraint by changing a set where restrained. By convention permanent restraint imposed at stage s restrains the initial segments of length s of L and all the G_i {A, D and G}. Any permanent restraint on P must be mentioned specifically. [We never need to restrain Q.] Alternating restraints are caused by the announcements of A-stages or P-stages which we describe later in the construction. Basically, during A-stages, we remove the alternating restraint for L and the G_i $\{A$ and $D\}$ allowing numbers to enter (or leave) these sets and we impose an alternating restraint on P and Q $\{$and $G\}$ so that no numbers can enter P or Q $\{$or $G\}$ at this stage. During P-stages, we do the opposite (except that no numbers ever leave the r.e. sets P or Q $\{$or $G\}$).

3 Requirements I

We now begin the proof of our main technical result.

Theorem 1.4. *Given a recursive partial order $(\omega, \leq_*)$ and an $n \geq 1$, there exist uniformly n-r.e. sets G_i for each $i \in \omega$, an n-r.e. set L and r.e. sets P and Q such that:*

1. *Each $\mathbf{g}_i$ is a maximal n-r.e. degree below $\mathbf{a}$ such that $\mathbf{q} \not\leq \mathbf{g}_i \vee \mathbf{p}$ where $A = \bigoplus_i G_i$.*
2. *$\mathbf{g}_i \leq \mathbf{g}_j \vee \mathbf{l}$ if and only if $i \leq_* j$.*

First, for the negative order facts, we have requirements for each pair $i \not\leq_* j$ and each e:

$$\Psi_{e,i,j}: \ \Psi_e(L \oplus G_j) \neq G_i.$$

Similarly for each triple (i, j, e) with $i \neq j$, we also want:

$$\Pi_{e,i,j}: \ \Pi_e(G_j) \neq G_i,$$

i.e., the G_i's are pairwise incomparable.

Then for each pair (i, e) we need:

$$\Theta_{e,i}: \ \Theta_e(G_i \oplus P) \neq Q.$$

We also need the main requirements that each $\mathbf{g}_i$ is a maximal n-r.e. degree $\mathbf{g} \leq_{\mathbf{T}} \mathbf{a}$ such that $\mathbf{q} \not\leq \mathbf{g} \vee \mathbf{p}$. We let W_i be an effective list of all the n-r.e. sets.

$$\Phi_{e,i}: \ \Phi_e(A) = W_i \rightarrow [\exists \Gamma(\Gamma(W_i \oplus P) = Q) \vee (\exists k(W_i \leq_T G_k))].$$

- Note that these Φ requirements by themselves do not ensure that each $\mathbf{g}_i$ is maximal. That is why we need the Π requirements to make all the G_i's pairwise incomparable. The Φ requirements then do guarantee that the $\mathbf{g}_i$ are maximal.

If it does not cause confusion, we may omit the subscripts of the requirements and sets in our argument to simplify the notation.

Finally, we have to deal with the positive order facts, i.e., $G_i \leq_T L \oplus G_j$ for $i <_* j$. We will guarantee that, for $x > i, j$, $x \in G_i \Leftrightarrow x \in L$ or $x \in G_j$. Putting numbers into a G_i is initiated only by a Ψ or Π requirement. For Π action, we simply put a witness x that is going into G_i (for diagonalization) into L as well. When action is initiated for diagonalization by Ψ at stage s, we put x into G_i and also into each G_l with $l >_* i$ for each $l < x$. As, in this case, $i \not\leq_* j$, this action does not add elements to G_j and so it does not injure the Ψ computation initiating the action. We say that each witness x (for a Ψ or Π requirement) has an *associated block of sets* (the G_l such that $l < x$ and $i \leq l$ or G_i and L, respectively). During the construction x moves into or out of all the sets in its block simultaneously.

4 Priority Tree I

We put all the Ψ, Π, Θ and Φ requirements into one priority list. Our *priority tree* consists of *nodes* and *branches*. Each node is associated with a requirement in the list and each branch leaving a node is assigned an outcome. We label each node with its associated requirement and each branch with the assigned outcome. When we list outcomes of a node we do so in a left to right order that specifies the left to right order on the priority tree of the branches leaving that node

A Ψ or Π node has two outcomes: d and w, which stand for "diagonalization" and "wait" respectively.

A Φ node has outcomes s_{n-1}, s_{n-2}, ..., s_1, i and w. Outcome s_i stands for "shuffle" for the i-th time. We will explain what this means in detail in the construction. Roughly, it means that we expect to shuffle between two versions of A (by removing numbers from A and then possibly putting them back in) as we cycle back to this node. The expected result of this shuffling is to guarantee that $\Phi(A) \neq W$ by a diagonalization. Outcome i stands for "infinite" agreement between $\Phi(A)$ and W and outcome w stands for "wait".

A Θ node β has outcomes d, g_{α_1}, g_{α_2},...,g_{α_k} and w. As usual, d and w stand for "diagonalization" and "wait" respectively. Each α_i is a Φ node above β which has outcome i along β. If γ is a node below β extending the g_{α_i} branch from β, then we say that γ *sees* an $\alpha_i - \beta$ pair. [The intuition here is that γ believes that α_i and its associated requirement is satisfied by β.]

A Φ node α is *active* at $\gamma \supset \alpha$ if α has outcome i along γ and γ does not see an $\alpha' - \beta'$ pair such that $\alpha' \subseteq \alpha \subset \beta' \subset \gamma$. For there to be a g_{α_i} outcome of a Θ node β, we also require that α_i be active at β. We order these g_{α_i}'s from left to right in descending order going down the tree to β, i.e., $\alpha_1 \subset \alpha_2 \subset ... \subset \alpha_k$. [This choice of left to right order comes into play at the very end of the proof of Lemma 6.16 and is discussed in §**??**.]

The priority tree is defined recursively as follows: suppose τ is an immediate extension of σ, we associate τ with the highest priority requirement among all requirements which either have not appeared above τ or are Φ requirements that, above τ, have appeared only at nodes δ with outcome i such that τ sees an $\alpha - \beta$ pair with $\alpha \subset \delta \subset \beta$. [So δ looks inactive but not really satisfied, i.e. if satisfied at some earlier point it has since been "captured" by some other pair.] Then we add the corresponding number of branches (outcomes) below τ. It is easy to see that this tree is recursive.

5 Construction I

At stage s of the construction, we build a path of length s of the *accessible nodes* along the priority tree. It is possible that at some accessible node we will *announce* that s is an A-stage or a P-stage. All later nodes accessible at s must respect this announcement by acting according the the rules governing A-stages or P-stages: no changes in A can occur once a P-stage has been announced and none in P or Q once an A-stage has been announced. In the construction, we will make sure that the first accessible Θ node with a type g outcome (if any) makes the announcement for the stage s. An over-riding rule is that permanent restraint imposed by a node (not since initialized) is not violated by action at any node of lower priority (i.e. below it or to its right). If any instruction below leads to any such situation, we do not carry it out and instead go to outcome w [and do nothing].

In this section, we first describe the construction at stage s for each node when there has been as yet no announcement for the stage and then specify the modifications for when there has already been one.

5.1 no announcement, Ψ or Π node

The actions at Ψ and Π nodes are quite standard: If it is the first time we come to this node (after it was last initialized), then we pick a witness number x which is *fresh*, i.e., larger than any number we have seen by this point in the construction. In general, at a Ψ ($\Psi(L \oplus G_j) \neq G_i$) or a Π ($\Pi(G_j) \neq G_i$) node with a witness x already assigned (and not yet canceled by initialization), we check whether the computation at x converges to 0. If it diverges or converges to a nonzero number, then we do nothing and go to the w outcome. If it converges to 0 and $x \notin G_i$, then we do a diagonalization: put x into G_i and into all the other sets in its block as described in Section 3, impose permanent restraint [to preserve the use of the computation] and go to the d outcome. If x is already in G_i, then we (again) go to outcome d [and keep the restraint already imposed].

5.2 no announcement, Φ node

At a Φ node α ($\Phi(A) = W$) if we have not yet had a type s outcome (since α was last initialized) let t be the last stage at which α was accessible (since last initialized). If there is a such stage and a $u < l_\alpha(t), l_\alpha(s)$ such that $\Phi(u)$ (and so $W(u)$) differ at t and s with the difference not being that u has entered W for the first time and the only change in $A \upharpoonright \phi(u)$ at t is that some z has entered its block of sets for the first time because of the action of a node extending α then we *initiate a shuffle on* z by removing z from its block of sets, impose permanent restraint and go to outcome s_1. We call this shuffle strategy *Plan S* with *shuffle points* $sp1(= t) < sp2(= s)$. [Note that these shuffle points have the property that $A_{sp1}(= A_{sp1} \upharpoonright sp1)$ and $A_{sp2}(= A_{sp2} \upharpoonright sp2)$ differ below $sp1$ only in that z is in its block of sets in A_{sp2} and out of them in A_{sp1}. More crucially, they produce different values for Φ at some u, i.e. $\Phi(A_{sp1}; u) \neq \Phi(A_{sp2}; u)$.] If we had an outcome of type s at the last stage t at which α was accessible, we check whether $W(u)$ is different at s than at t. If so, restore the initial segment of A to the version of A which is different from the current one (by putting z into or taking it out of its block of sets), impose permanent and let the outcome be s_{i+1}. If not, we stay at the s_i outcome. [This maintains any previously imposed permanent restraint.]

If we haven't initiated shuffling, let $l_\alpha(s)$ be the length of agreement between the current versions of $\Phi(A)$ and W. Note that whenever we initialize this node α, we also initialize the values of this function to be 0. If this is the first time that $l_\alpha(s) > 0$ after it has last been initialized, or $l_\alpha(s) > l_\alpha(t)$ where t is the last stage when α had an i outcome, then we go to the i outcome; otherwise we go to the w outcome.

If we go to the i outcome, we continue to define a functional Γ [aiming to make $\Gamma(W \oplus P) = Q$]. At this point, we enumerate a new axiom making $\Gamma(W \upharpoonright l_\alpha(s) \oplus P \upharpoonright v; w) = Q(w)$, where v is a fresh number and w is one more than the largest number where we have previously defined Γ (since it was last initialized). If P has changed on its Γ-use at some $x < w$ and the change was caused by the action of a node $\beta\hat{\ }g_\alpha$ [necessarily extending $\alpha\hat{\ }i$] with witness x as in §5.4.2, then we redefine $\Gamma(x)$ to be the current value of $Q(x)$ with W-use $l_\alpha(s)$ and fresh P-use. [As P is r.e. this change permanently invalidates the previous axiom for $\Gamma(x)$.] Similarly, if W has changed on its use u_1 (where its old P-use is v_1) so as to make $\Gamma(x)$ divergent but x has not entered Q, we see if x is currently the witness for some Θ node β below α (for G). If so, we look at the last stage t at which β was accessible and see if its outcome was g_α. If G has not changed on $\theta(x)$ as defined at the point of stage t at which β was reached and the change in W includes one at some u making it different from the common value of $\Delta(u)$ and $W(u)$ at t, then we redefine $\Gamma(x)$ with W-use $l_\alpha(s) = u_2$ and fresh P-use. In all other cases of a W or P change on $\gamma(W; x)$ or $\gamma(P; x)$, respectively, that makes $\Gamma(x)$ divergent we redefine $\Gamma(x)$ with the same uses as it last had but for the new values of W and P (subject, of course, to our monotonicity requirements on the use).

5.3 no announcement, Θ node

At a Θ node β accessible for first time after it has been last initialized, we pick a fresh witness x for diagonalizing $\Theta(G_i \oplus P) \neq Q$. In general, if we have a witness x already assigned (and not yet canceled by initialization), we check whether the computation converges at the witness x. [As usual when there are higher priority requirements that are expected to put infinitely many numbers into a set, we restrict our attention to computations that are consistent with our beliefs as prescribed by our actions in §5.4.2. Here this means the following:] We also require that the computation be *believable*, i.e. for every requirement $\hat{\Theta}$ assigned to a node α with witness $\hat{x}$ and $\alpha\hat{\ } g_{\hat{\alpha}} \subseteq \beta$ for some $\hat{\alpha}$, $\theta(x) < \gamma_{\hat{\alpha}}(P;\hat{x})$ and if $\gamma_{\hat{\alpha}}(P;\hat{x})$ has been previously increased by a $\hat{W}$ change (as described at the end of §5.2) from say u_1 to u_2 and $v_1 - 1$ is not yet in P then $\theta(x) < v_1$ as well. If $\Theta(x)$ does not converge with a believable computation or so converges to a nonzero number, then we go to the w outcome and do nothing.

[If the believable computation $\Theta(G_i \oplus P; x)$ converges to 0 with P-use $\theta(x)$, then we would like to diagonalize, i.e., put x into Q and preserve the P and G_i use of the computation. However, we must worry about whether doing so injures some already defined Γ computation at a node above β. For example, if there is a such a $\Gamma(W \oplus P) = Q$ which computes $Q(x) = 0$ with $\gamma(P;x) \leq \theta(x)$, then our desired diagonalization would falsify this computation of Q while correcting the Γ computation (by putting its use into P and redefining the functional) would injure our Θ computation for diagonalization. Our plans must be more subtle.] If $\Theta(x)$ converges with a believable computation we proceed as follows:

Let $\alpha_1 \subset \alpha_2 \subset \ldots \subset \alpha_k$ be all the active nodes above β with each α_j defining its functional $\Gamma_j(W_{l_j} \oplus P)$. Let $\gamma_j(P;x)$ be the P-use of Γ_j at x, if it has already been defined. [See §**??** for some comments on the left-right ordering indicted here for these nodes and the associated outcomes g_{α_j} below.]

5.3.1 Plan D: diagonalization

If $\theta(x) < \gamma_j(P;x)$ for all j for which $\gamma_j(P;x)$ is defined, we do a modified diagonalization: We enumerate x into Q and also enumerate $\gamma_{\alpha_j}(P;x) - 1$ into P for each j. [This allows us to correct the $\Gamma_j(x)$ when $\alpha_j\hat{\ } i$ is next accessible.] We now impose the usual permanent restraint but also one on $P \restriction \theta(x)$ [to preserve the Θ computation] and go to outcome d. Until β is initialized, it has outcome d at every later stage at which it is accessible.

5.4 Stage Announcements

If we cannot follow Plan D, i.e., $\theta(x) \geq \gamma_j(P;x)$ for some j, then we take the largest j such that $\theta(x) \geq \gamma_j(P;x)$ [and are likely to go to outcome g_{α_j} where we build a functional Δ computing W_{l_j} from G_i]. [The choice of j is relevant at the very end of the proof of Lemma 6.8 but our choice of the largest j (rather than say the smallest) doesn't make any difference in this construction. It does, however, matter in the at the end of the proof of Lemma 10.1 for our second theorem.]

5.4.1 Plan A: A-stage announcement

If this is the first time (since the last initialization) that we would go to the g_{α_j} outcome or the last time we went there we announced a P-stage then we go to outcome g_{α_j} and announce an A-stage [and so allow elements to be enumerated into or taken out of A].

Otherwise, let t be the last stage when $\beta\hat{\ }g_{\alpha_j}$ was accessible. By our construction and case assumption, t must have been announced as an A-stage at $\beta\hat{\ }g_{\alpha_j}$. (If some node to the right or left of $\beta\hat{\ }g_{\alpha_j}$ made an announcement at stage t then β would not have been accessible at t. If some node α above β made an announcement at t then one would also have to be made above β at s contrary to our case assumption that no announcement has been made at this stage before we reached β.)

5.4.2 Plan P: P-stage announcement

We now go to the g_{α_j} outcome and extend Δ by adding axioms computing $W_{l_j}(u)$ from G_i with fresh use for any $u < l_{\alpha_j}(s)$ for which Δ has not previously been defined. In addition, we put $\gamma_j(P;x) - 1$ into P to injure the current Θ computation (since $\gamma_j(P;x) \leq \theta(x)$). [This kills the current computation of $\Gamma(x)$ and as P is r.e. it can never apply to $W \oplus P$ again.] Moreover, if $\gamma_j(P;x)$ has been previously increased by a $\hat{W}$ change (as described at the end of §5.2 from say u_1 to u_2 and $v_1 - 1$ is not yet in P then we also put $v_1 - 1$ into P. [This kills the old computation of $\Gamma_j(x)$ as well as and guarantees that it too will never again apply to $W \oplus P$.] [We will redefine Γ_j with axioms using the new version of P with a fresh P-use and W_{l_j} use $l_{\alpha_j}(v)$ when we next get to $\alpha_j\hat{\ }i$ at v. The result of this action is that we increase the Γ_j use from P and W_{l_j} and so the next time when this β is accessible with the g_{α_j} outcome, the use $\theta(x)$ must be larger than that of this stage. If this happens infinitely often $\Gamma_j(x)$ diverges but we expect to satisfy the associated Φ requirement by building $\Delta(G_i) = W_{l_j}$ at $\beta\hat{\ }g_{\alpha_j}$. We then also satisfy the Θ requirement associated with β as $\Theta(x)$ diverges as well.] We now announce that the current stage is a P-stage.

If there has been a change in G_i that leaves $\Delta(u)$ undefined where it had previously been defined, we put in a new axiom computing the current value of $W_{l_j}(u)$ with the old use.

[We shall argue for β on the true path with true outcome g_{α_j} that we build Δ consistently and correctly compute W_{l_j} at each stage (Lemma 6.8). Typically, it turns out that, along the true path, if W_{l_j} has changed where previously computed, then G_i must have changed at the corresponding part used in the computation.]

5.5 modifications with a stage announcement

When there is has already been a stage announcement before we reach β, the node β has to obey the appropriate rules. For a Ψ, Π or Θ node, we see what we would have done if there had been no stage announcement as yet. If that action is compatible with the current stage announcement (no announcement of an A-stage or change in A if a P-stage; no change in P or Q and no announcement of a P-stage if an A-stage), we proceed as if there had been no announcement. If not, we do nothing and go to outcome w.

For a Φ node, the modification is slightly trickier. [Later we will need the fact that each node along the true path passes down alternating A and P restraints in the construction.] Here in order to go to the i outcome, we need to wait (with outcome w) for a stage when the stage announcement is different from the last stage t when we had an i outcome, and also the length of agreement is longer than its last value. [In this way, the Φ node passes down alternating A and P restraints along the i outcome.] When we have already initiated shuffling, we act as before at A-stages and at P-stages we go to outcome w. [This maintains the permanent restraint imposed when we initiated shuffling or last shuffled as the nodes that imposed it are now to our left.]

6 Verification I

6.1 True path and true outcome

First of all, as in usual priority tree arguments, there is a leftmost path accessible infinitely often. (Each node has only finitely many outcomes.) This is the *true path* and the outcomes along it the *true outcomes*.

Lemma 6.1. *Numbers enter or leave A or L only when permanent restraint is imposed by a Π, Ψ or Φ node. When such nodes β impose permanent restraint, we move to the left of any previous outcome that has been accessible since β was last initialized.*

Proof. By inspection of the construction. □

Lemma 6.2. *At most one node acts to change A at any stage s.*

Proof. If we first act at α to change A at s then we move to an outcome to the left of all previously accessible ones (since α was last initialized) by Lemma 6.1. So all later nodes accessible at s that can change A are accessible for the first time since last initialized and so at most appoint fresh witnesses or (for Φ nodes) begin their construction of Γ anew. None of these witnesses can go in at s as no convergences can be seen at numbers larger than s. No shuffling can been initiated for any of the Φ nodes by construction. □

Lemma 6.3. *If a node α is initialized at stage s then it never later acts to change any set below s.*

Proof. If α is a Π, Ψ or Θ node it only acts to put numbers at least as large as its witness x into A or P and any witness appointed after s is larger than s. For Π and Ψ nodes this is immediate. For Θ, its action puts numbers of the form $\gamma(P;x)-1$ into P and by construction $\gamma(P;x)>x$. For Φ nodes, the only action changing sets is shuffling. This shuffling only involves numbers appointed below α at stages when α was accessible since it was last initialized. □

Recursively along the true path, we now determine the actions of the nodes on it after no node to their left is ever accessible and prove that all the requirements are satisfied along it. For any node β on the true path we let $s(\beta)$ be the first stage at which β is accessible but after which no node to its left is ever accessible again.

Lemma 6.4. *Any permanent restraint imposed by a node β at any $s \geq s(\beta)$ is never injured by any other node.*

Proof. The only actions that can injure such restraint after $s(\beta)$ are ones by nodes above it on the true path. None can change A or L by Lemma 6.1. As for P, the only permanent restraint imposed on P is by Θ nodes when we go to outcome d and restrain $P \upharpoonright \theta(x)$. Now nodes $\hat{\beta}$ above β of type Θ with outcomes $g_{\hat{\alpha}}$ may put numbers into P but they only put in ones of the form $\gamma_{\hat{\alpha}}(P;\hat{x})$ and our believability requirement on the computation of $\Theta(x)$ guarantees that all of the current values of these $\gamma_{\hat{\alpha}}(P;\hat{x})$ are larger than $\theta(x)$. The only way one of them could decrease is if it had previously been increased from u_1 to u_2 by a change in W as described at the end of §5.2 and then W changes back to the old value before the old computation is killed by $v_1 - 1$ going into P. However, our believability condition also requires that $\theta(x)$ is less than these v_1 as well. Any later change increases the use above the previous values Thus no changes every occur in P below $\theta(x)$. □

Lemma 6.5. *The final witness for any node β chosen at $s(\beta)$ is larger than any permanent restraint of higher priority than β.*

Proof. By construction the witness is chosen fresh and so larger than anything previously seen. The only nodes of higher priority that can impose permanent restraint later are ones above β. None of type Π, Ψ or Φ can do so by Lemma 6.1. One of type Θ also imposes permanent restraint only when it moves left to outcome d contradicting our definition of $s(\beta)$. □

Before we show that the requirements are satisfied we analyze the alternating restraint.

6.2 Alternating A and P-stages

Lemma 6.6. *Every node along the true path above the first Θ node β with type g outcome on the true path never sees or makes a stage announcement (imposes alternating restraint) when accessible. For the other nodes α on the true path, their true outcomes, o, are accessible at infinitely many A and P-stages. Indeed, after $s(\alpha\hat{\ }o)$, the stages at which $\alpha\hat{\ }o$ is accessible alternate between A and P ones (the node passes down alternating A and P restraints along its true outcome).*

Proof. For any node above β the claim is immediate from the rules of the construction. For a Π or Ψ node below β, it is immediate from the construction that after $s(\beta)$ either we always have outcome w or, whenever we are at β after the first time we have outcome d we also have outcome d. So for these nodes the Lemma is obvious. For a Φ node below β, either the true outcome is shuffling (s), or waiting (w), or infinitary (i). In the two former cases, the outcome is again eventually constant: Once we move to a type s outcome, the construction guarantees that we can move only to the left to another type s outcome. Thus the outcome is eventually constant at some type s outcome. As for outcome w, any rightmost outcome that is the true outcome of a node β on the true path is the outcome at almost every stage at which β is accessible. In the third case, our construction in §5.5 ensures that it passes down alternating A and P restraint along the outcome i as required.

The Θ node β which first makes the announcements along the true path must have true outcome some g_α. Then according to our construction, if it announced a P-stage the last time it was accessible, we follow Plan A and make an A-stage announcement. If it last announced an A-stage, then we follow Plan P and announce a P-stage.

Finally for any other Θ node β' on the true path, the claim is also immediate for true outcome d or w as above. In the case of a g_α outcome, the construction automatically guarantees that it always waits for an alternation in the type of restraint to move again to the true g_α outcome. □

We next analyze the functionals Δ and Γ that we construct.

6.3 The functionals are well-defined and correct

Lemma 6.7. *The functionals Γ built at nodes $\alpha\hat{\ }i$ on the true path starting at $s(\alpha\hat{\ }i)$ are well-defined, i.e., we do not add contradictory axioms in the construction and when defined give the correct current value for Q. They are defined on arbitrarily large initial segments of ω and so, if convergent at every x, they correctly compute the desired sets.*

Proof. It is clear by construction that we define $\Gamma(x)$ at least once for each x: As $\alpha\hat{\ }i$ is on the true path, $l_\alpha(s)$ is going to infinity on the stages at which $\alpha\hat{\ }i$ is accessible. On each of these stages we define Γ on a new number. Once defined $\Gamma(x)$ is then defined at every stage at which $\alpha\hat{\ }i$ is accessible by construction. As for consistency and correctness, note first that it is immediate from the construction that at any stage at most one axiom in Γ applies to the current value of $W \oplus P$. Now, the only way any problem could arise is if $Q(x)$ has changed for some x but W and P have not changed on the corresponding use or they change and then W reverts back to a previous value that applies to some older computation (with value 0). However, in our construction, $Q(x)$ can change only when we implement Plan D to put x into Q at a some stage v for a Θ node $\hat{\beta}$. Any number x put into Q by such nodes to the right of $\alpha\hat{\ }i$ must be both appointed fresh and then put in while we are to the right of $\alpha\hat{\ }i$ without $\alpha\hat{\ }i$ becoming accessible in between. Thus when $\alpha\hat{\ }i$ is again accessible Γ has not been defined at x and so when we define $\Gamma(x)$ we set it equal to 1 with the first axiom and all later ones as well. Any x put in by nodes to the left of $\alpha\hat{\ }i$ are in by $s(\alpha\hat{\ }i)$ and so we define Γ correctly on them as well.

Thus we can assume that $\hat{\beta}$ is below $\alpha\hat{\ }i$. If α is active at $\hat{\beta}$, then when x enters Q at v we put $\gamma(P;x)-1$ into P by construction and so kill the current computation and put in a new one giving the correct answer when we next reach $\alpha\hat{\ }i$. If α is not active at $\hat{\beta}$, there must be a first $\beta' \subset \hat{\beta}$ with $\beta'\hat{\ }\hat{\alpha} \subset \hat{\beta}$ for some $\hat{\alpha} \subseteq \alpha$. So in particular, α is active at β'. Now β' has a witness x' necessarily less than x (as x is appointed later) and at stage v when we reached $\beta'\hat{\ }g_{\hat{\alpha}}$ we put $\gamma(x') < \gamma(x)$ into P and so kill the current computation of $\Gamma(x)$ and correct it when we next reach $\alpha\hat{\ }i$. Note that both P and Q are r.e. so $\gamma(P;x)-1$ has never been in P before and x will never leave Q. As we impose permanent restraint when we go to outcome d and diagonalize, no later change in W can return us to any old computation. □

Lemma 6.8. *The functional Δ defined at the true g_α outcome of a Θ node β (for G_i) on the true path starting at $s(\beta\hat{\ }g_\alpha)$ is well-defined. When $\Delta(u)$ is convergent while we are at $\beta\hat{\ }g_\alpha$, it always give the current value of $W(u)$ on an initial segment of ω. Indeed, $\Delta(G_i) = W$ as desired. (For notational convenience we assume that α is assigned the Φ requirement for W which is constructing the functional Γ at its i outcome.)*

Proof. As for the final claim, note first that by construction if $\Delta(u)$ is ever defined it is defined at every $u' < u$ and then it is defined there at every later P-stage at which $\beta\hat{\ }g_\alpha$ is accessible. Moreover, at these stages we extend its domain of definition to $l_\alpha(s)$ which is going to infinity since $\alpha\hat{\ }i \subseteq \beta$ is on the true path. Once $\Delta(u)$ is defined, its use never changes by construction and so if, at every stage when defined at $\beta\hat{\ }g_\alpha$, it correctly computes $W(u)$, it does so in the end.

For correctness, we argue by induction on the stages at which $\beta\hat{\ }g_\alpha$ is accessible beginning at $s(\beta\hat{\ }g_\alpha)$ that $\Delta(u) = W(u)$ at every u at which Δ is defined. This is obviously true by construction if s is the first stage at which $\Delta(u)$ is defined. Suppose it is true at s and the next stage at which $\beta\hat{\ }g_\alpha$ is accessible is t and the problem occurs at u.

Note that no change in $A \upharpoonright s$ or $P \upharpoonright \theta(x)$ can occur between s and t. No node to the right of $\beta\hat{\ }g_\alpha$ can make such a change by Lemma 6.3. No node to its left can do so as they are never accessible after $s(\beta\hat{\ }g_\alpha)$. No node above $\beta\hat{\ }g_\alpha$ can change A at all by Lemma 6.1. Nodes above $\beta\hat{\ }g_\alpha$ may act to put numbers of the form $\gamma_{\hat{\alpha}}(P;\hat{x})$ into P via other Θ requirements $\hat{\beta}$ above β with outcome $g_{\hat{\alpha}}$ but by our believability condition on the $\Theta(x)$ computation, at stage s none of them are below $\theta(x)$ at s. The only way one of them could decrease is if it had previously been increased from u_1 to u_2 by a change in W as described at the end of §5.2 and then W changes back to the old value before the old computation is killed by $v_1 - 1$ going into P. However, our believability condition requires that $\theta(x)$ is also less than these v_1.

If s was a P-stage then no change occurred in $A \upharpoonright s$ at s so none has by stage t. Thus if $W(u)$ at t is different from its value at s it would be different from $\Phi(A;u)$ at t since that is the same as it was at s. This would move us to outcome w at α and so $\beta\hat{\ }g_\alpha$ would not be accessible for a contradiction.

Suppose then that s was an A-stage. No change in $P \upharpoonright \theta(x)$ occurs at s as no node above the one announcing the A-stage can change P without declaring a P-stage or moving left and none after it can because it is an A-stage. Moreover, none can occur before t as above. If no change occurs in A at s then, as none occurs before t, $\Phi(u)$ and $\Delta(u)$ would be the same at t as at s. If this value is not that of $W(u)$ at t then $\alpha\hat{\ }i$ would again not be accessible at t for a contradiction. Thus there has been some change in A at s. By Lemma 6.2, there is precisely one z that entered or left its block of sets at s. By Lemmas 6.1 and 6.3, the node σ causing the change must extend $\beta\hat{\ }g_\alpha$ as we are after $s(\beta\hat{\ }g_\alpha)$ and $\beta\hat{\ }g_\alpha$ is accessible. Now if there is no change in $W(u)$ between s and t, then the only way we could have a disagreement with $\Delta(u)$ at t (so the old axiom for $\Delta(u)$ at s is no longer valid but we cannot simply put in a new one with the same value) is that the change for z returns us to a previous computation of $\Delta(u)$ giving a different value. However, such a change in z can be caused only by σ shuffling z. Such a shuffle returns $A \upharpoonright v$ to its value at a previous stage v. If $\Delta(u)$ was defined at v then by induction it would have the same value as $\Phi(u)$ and $W(u)$ and no change can have happened in any of these when we reach t. Thus we would still have agreement at t as required. So we may also assume that $W(u)$ is different at s and t.

Suppose first that $z < \theta(x)$ and G_i is in its block of sets. Next, suppose the change occurred because of some shuffling procedure at σ. If $\Delta(u)$ was first defined before the shuffling began at σ, then we would have a contradiction as above.

If $\Delta(u)$ was first defined after the shuffling began, say at $v \leq s$ with, for the sake of definiteness, $z \in G$, then its use is fresh at v and so larger than z. Let $v' \geq v$ be the next stage after v at which we shuffle z at σ. If $v' = s$ then z has been in G_i from v to s and is removed at stage s by our action at $\sigma \supset \beta\hat{\ }g_\alpha$. Thus when we next return to $\beta\hat{\ }g_\alpha$ at t we have $z \notin G_i$ for the first time since $\Delta(u)$ was defined and we redefine it to be the current value of $W(u)$ as required. So we may assume that $v' < s$ and at stage v' we have $\Phi(u) = W(u) = \Delta(u)$ at $\beta\hat{\ }g_\alpha$ by induction. When we reach σ at v' we remove z from its block of sets including G_i and impose permanent restraint. When $\beta\hat{\ }g_\alpha$ is next accessible, say at $v'' \leq s$, we redefine $\Delta(u) = W(u) = \Phi(u)$ with $z \notin G_i$ (and its block of sets) but with the rest of $A \upharpoonright v'$ the same as it was at v' (because of the permanent restraint imposed at v' which, by Lemma 6.1, could be violated only by moving to the left of σ which could then not cause our problem at s). If $v'' = s$ then at stage s we shuffle z back into G_i (and its block of sets) at σ and impose permanent restraint. We next return to $\beta\hat{\ }g_a$ at t and have $\Phi(u)$ and $\Delta(u)$ and so $W(u)$ the same as they were at stage v' at $\beta\hat{\ }g_\alpha$, i.e. they all agree as required. Finally, if $v'' < s$ then later at stage s we shuffle at σ between the values for all of these sets and functionals that we had at v' and v''. Thus once again when we reach $\beta\hat{\ }g_\alpha$ at t all agree.

Thus the change that has occurred is that z entered G_i for the first time at s. If the change in $W(u)$ is not that u has entered for the first time, then we would have initiated a shuffling procedure at α at stage t and so move to $\alpha\hat{\ }s_1$ hence to the left of $\beta\hat{\ }g_\alpha \supset \alpha\hat{\ }i$ for a contradiction.

Thus through stage s, $W(u) = 0$. Suppose $\Delta(u)$ was first defined at s', of course with value 0 and fresh use q. We claim that $z < q$ and so its entry into G allows us to correct $\Delta(u)$ at t as desired. If not, it was chosen fresh as a witness for σ at a point in the construction during a stage $s'' \geq s'$ after $\Delta(u)$ was defined at s' but before s. In this case, however, $z > s''$ and so $z > \phi(u)$ at s''. Note that $\Phi(u)$ is defined at s'' because z is appointed at $\sigma \supset \beta \supset \alpha\hat{\ }i$. Now from the point of stage s'' at which $\alpha\hat{\ }i$ is accessible to stage s any change in $A \upharpoonright s''$ would initialize σ and so z could not enter A at s. (At s'' no node between $\alpha\hat{\ }i$ and σ can change A without moving left of σ. Then at σ all nodes to the right of σ are initialized and so cannot make any changes below s'' by Lemma 6.3. As σ appointed a witness at s'', this is the first stage at which σ has been accessible since it was last initialized so all A action by nodes below σ also involve only numbers larger than s''. Finally, any A action after s'' by a node of higher priority than σ would also initialize it by Lemma 6.1.) Thus at s, $\phi(u)$ and $A \upharpoonright \phi(u)$ are the same as they were at s'', i.e. $\Phi(A, u) = 0 = W(u)$ at s with the same computations as at s''. Now the only changes in A during stage s is that z enters its block of sets but $z > s''$ and then no changes occur in $A \upharpoonright s$ before stage t. Thus at stage t we also have $\Phi(A, u) = 0$ and so if $W(u) = 1$ at t, the outcome of α would not be i, for a contradiction.

Finally, suppose $z \geq \theta(x)$ or G_i is not in its block so no change occurs in $G_i \upharpoonright \theta(x)$ at s and so none before t. When $\Delta(u)$ was defined at the P-stage v (necessarily before the A-stage s), $u < l_\alpha(v)$ and so after v, $u < \gamma(W; x)$ whenever it is defined. In fact, at each P-stage during which we reach $\beta\hat{\ }g_\alpha$ (starting with v) we put $\gamma(P; x)$ into P and subsequently increase $\gamma(W; x)$ to $l_\alpha(v') > l_\alpha(v)$ when we are next at $\alpha\hat{\ }i$ (at $v' > v$). Each computation of $\Gamma(x)$ killed in this way can never to reapply to $W \oplus P$ as P is r.e. As we cannot reach $\beta\hat{\ }d$, the only other way $\gamma(x)$ can change requires a W change that causes a difference between the previously computed common values of Δ and W. By our induction assumption this cannot have occurred before s. So all axioms for $\Gamma(x)$ provided before s are invalid by the end of stage s.

As $W(u)$ has different values at s and t, the change in W introduces a value of $W(u)$ that we see at $\alpha\hat{\ }i$ at some first stage v'' after s but no later than t. As we have argued, no old computation of $\Gamma(x)$ is still valid at v''. Thus by construction we would increase $\gamma(P; x)$ at v'' to a fresh value larger than $\theta(x)$. When we return to $\beta\hat{\ }g_\alpha$ at t, $\gamma(P; x)$ is now larger than $\theta(x)$ which has not changed since s. Thus by construction, g_α cannot be the outcome of β at t for a contradiction. □

6.4 All requirements are satisfied

Finally we want to show that all requirements are satisfied.

The positive order requirements are easily verified by our construction.

Lemma 6.9. *If $i <_* j$ then $G_i \leq_T L \oplus G_j$.*

Proof. Consider an $x > i, j$. To decide if $x \in G_i$ go to stage x of the construction and see if x has been appointed as a witness for some Π or Ψ requirement with G_i in its block. If not, then $x \notin G_i$. (Indeed x is not in any G_k.) If it is in the block for a Π requirement then L is also in its block. If for a Ψ requirement then G_j is in the block. In any case, as once appointed x moves into or out of all sets in its block during the entire construction, $x \in G_i \Leftrightarrow x \in L$ in the Π case and $x \in G_i \Leftrightarrow x \in G_j$ in the Ψ case. □

We now move to the negative (diagonalization) requirements.

Lemma 6.10. *The Π and Ψ requirements are satisfied.*

Proof. Suppose the requirement is assigned to the node β on the true path. If, after $s(\beta)$, we ever go to outcome d and so diagonalize, the result is immediate from the construction and Lemma 6.4. If not, it must be that the outcome is always w after after $s(\beta)$. If the relevant computation converged to 0 the correct computation would be available from some point on and so by Lemma 6.6 we would eventually see it at an A-stage and so move to outcome d by Lemma 6.5. If not, then x never enters G_i and we also satisfy the requirement as desired. □

We next consider the Θ requirements.

Lemma 6.11. *If a Θ node β on the true path has true outcome d then the associated requirement is satisfied.*

Proof. Consider the stage $s(\beta\hat{\ }d)$ when β has outcome d (and is never again initialized). We put the witness x into Q and impose permanent restraint to preserve the computations $\Theta(G_i \oplus P; x) = 0$. Lemma 6.4 shows that this computation is preserved. □

Lemma 6.12. *If a Θ node β on the true path has true outcome g_α, then its requirement is satisfied. Indeed, for x the final witness for β, both $\theta(x)$ and $\gamma_\alpha(x)$ go to infinity on the stages when $\beta\hat{\ }g_\alpha$ is accessible. Moreover, any time we increase $\gamma_\alpha(x)$ because of a W change as at the end of §5.2, we later kill the P-use of the old computation as well by putting v_1 into P.*

Proof. In this case, by our construction (and Lemmas 6.5 and 6.6), we infinitely often put numbers $(\gamma(P; x) - 1)$ into P and redefine $\Gamma(W_i \oplus P)$ with a fresh P-use. (The first of these Lemmas implies that the numbers we want to put into P are larger than any permanent restraint as they are of the form $\gamma(P; x)$ which is larger than the witness x for β.) So by our criteria for going to outcome g_α, we infinitely often see $\theta(x) > \gamma(P; x)$, so $\theta(x)$ must go to infinity (when $\beta\hat{\ }g_\alpha$ is accessible) along with $\gamma(P; x)$ and the computation $\Theta(G_i \oplus P; x)$ diverges. As for any increase in $\gamma(x)$ because of W change as in §5.2, the next time we are at $\beta\hat{\ }g_\alpha$ we put the associated v_1 into P by construction. □

Lemma 6.13. *If a Θ node β on the true path has true outcome w then the associated requirement is satisfied.*

Proof. Note that if the outcome of β were d at any stage after $s(\beta)$ then d would be the true outcome by construction. Thus in our case, we never put the final witness x for β into G_i. So our only concern is that $\Theta(G_i \oplus P; x) = 0$. In this case, there is a stage after which it always converges to 0 and with a fixed use. By the previous Lemma this computation is believable at almost every stage when we are at β. Thus by construction and Lemmas 6.5 and 6.6, as in Lemma 6.12, we would eventually have outcome d for a contradiction. □

We now turn to the Φ requirements.

Lemma 6.14. *If a Φ node β on the true path has true outcome of type s, then the associated requirement is satisfied.*

Proof. The nature of the shuffling points guarantees that, at every stage with outcome of type s, $\Phi(A; x) \downarrow \neq W(x)$ and so this is true at the end of the construction as well and the requirement is satisfied. The crucial point here is that the permanent restraint imposed by β which are increasing as we move left among the type s outcomes can never be injured (other than by the shuffling done by β itself) by Lemma 6.4. □

Lemma 6.15. *If a Φ node β on the true path has true true outcome w then the associated requirement is satisfied.*

Proof. If $\Phi(A) = W$ then the length of agreement would go to infinity and so, by construction and Lemma 6.6, we would eventually move to outcome i after $s(\beta\hat{\ }w)$ for a contradiction. □

Finally, we have to deal with the case that every Φ node on the true path has true outcome i.

Lemma 6.16. *Every Φ requirement is satisfied.*

Proof. As usual in a $0'''$ priority tree argument, we want to consider the last node α along the true path assigned to a given Φ requirement. To see that there is such a node, argue by induction on the Φ requirements. The point here is that any Φ requirement, once assigned to a node that is never again initialized, can return to the list of requirements from which we draw to make assignments of requirements to nodes (along the true path) only when a strictly higher priority node becomes inactive. So once no node with a higher priority Φ requirement assigned ever becomes inactive again, the next node β assigned to Φ either becomes inactive once along the true path (by being satisfied by action at a lower Θ node on the true path) and then remains inactive or it never becomes inactive on the true path. In either case, Φ is never assigned to a node below β by the definition of the priority tree.

Let α be the last node along the true path assigned to the Φ requirement $(\Phi(A) = W_i)$. By the previous two Lemmas, we may assume that its true outcome is i. If there is an

Θ node β ($\Theta(G_k \oplus P) \neq Q$) on the true path with true outcome g_α, then we have built a functional Δ at $\beta\hat{\ }g_\alpha$ that computes W_i from G_k by Lemma 6.8.

If there is no such Θ node β, then we claim that we have successfully built $\Gamma(W_i \oplus P) = Q$ starting at $s(\alpha\hat{\ }i)$. By Lemma 6.7, we only have to verify that $\gamma(x)$ is eventually constant for each x. We begin to define our Γ at $s(\alpha\hat{\ }i)$. Assume inductively that $\gamma(\hat{x})$ has stabilized for $\hat{x} < x$. Thereafter, once $\Gamma(x)$ is defined, our construction allows $\gamma(x)$ to increase because of a change in P at most once for each time some $\beta\hat{\ }g_\alpha$ below $\alpha\hat{\ }i$ is accessible and the Θ requirement assigned to β has witness x or once when $\beta\hat{\ }d$ is accessible (again with witness x for β). It can increase because of a W change at most finitely often for each such $\beta\hat{\ }g_\alpha$ and stage. (At worst only when W changes on the domain of Δ at that stage.) At most one β has x assigned as a witness. If β is not on the true path, it can be accessible with witness x at most finitely often. If β is on the true path, once $\beta\hat{\ }d$ is accessible, $\beta\hat{\ }g_\alpha$ cannot be accessible again unless β is initialized and so chooses a new witness. If $\beta\hat{\ }g_\alpha$ is accessible infinitely often, then some $g_{\hat{\alpha}}$ (possibly to the left of g_α) would be its true outcome and so $\hat{\alpha} \subseteq \alpha$. If $\alpha = \hat{\alpha}$ we contradict our case assumption. If $\hat{\alpha} \subset \alpha$ then α would be come inactive and Φ would be reassigned later to a node below $\beta\hat{\ }g_\alpha$ on the true path contradicting our choice of α. Thus $\Gamma(x)$ can change at most finitely often as required. □

6.5 Δ^0_2 and Δ^0_3 partial orders

To handle partial orders recursive in $0'$ we make the following changes in the construction:

We begin with a recursive approximation $f(i,j,s)$ to the (characteristic function of the) relation $i \leq_* j$. We now have requirements $\Psi_{e,i,j}$ for every e,i,j with a new additional leftmost outcome n. At stage s at a node α for $\Psi_{e,i,j}$, if $f(i,j,,s) = 1$ (so we think we do not want to diagonalize) we go to outcome n and do nothing. If $f(i,j,s) = 0$ we act as before with a new definition of the block for our witness x. When x (necessarily larger than i and j) is appointed as a witness, we determine its block by calculating $f(k,l,t)$ for each $k,l < x$ and $t > x$ until we reach a t at which either $f(i,j,t) = 1$ or the relation on numbers $k,l < x$ defined by $f(k,l,t)$ is a partial order $\preceq$. In the first case, the outcome is again n and we do nothing. In the second case, we put G_k into the block for x if and only if $i \preceq k$ (i.e. $f(i,k,t) = 1$). Note that $f(i,j,t) = 0$ by our case assumption and so G_j is not in the block.

To see that this modification works, note that if $i \nleq_* j$ then from some point on $f(i,j,t) = 0$ and so we never again have an outcome n for $\Psi_{e,i,j}$ and so satisfy the negative order requirements as before. For the positive ones, suppose $k \leq_* l$ and for $t \geq t_0$, $f(k,l,t) = 1$. For any witness $x \geq k,l,t_0$, if its block does not contain k then, of course, $x \notin k$. If it does contain k it also contains l and so $x \in G_k$ if and only if $x \in G_l$. (The case for Π requirements is as before.)

The modifications needed for partial orders recursive in $0''$ are more complicated. For each i,j we insert a requirement into the priority order used for the Δ^0_2 case and so on each path of the priority tree a node ε that guesses in a Δ_3^0 way whether $i \leq_* j$, i.e. the node has infinitely many outcomes $\langle x, k \rangle$ with $x \in w$ and $k \in \{0, 1\}$ ordered lexicographically. We organize determining the outcome of ε at each stage s so that if $\langle x, k \rangle$ is the leftmost outcome accessible infinitely often then x is the least witness to the Σ_3 formula which says that $i \leq_* j$ if $k = 1$ and the least witness to the Σ_3 formula which says that $i \not\leq_* j$ if $k = 0$. In addition we coordinate this guessing with the stage announcements so that the true outcome passes on alternating restraint as before. We then act at nodes as in the Δ_2^0 case but using at each node only the information about the ordering coded on the outcomes of the ε type nodes above it. So for for a Ψ type requirement, if the relation given in this way is not a partial order $\preceq$ or says that $i \preceq j$, we go to outcome n. (We put the ε nodes on the tree so that any node for a requirement $\Psi_{e,i,j}$ has an ε type node above it assigned to $i \leq_* j$.) If it is does specify a partial order with $i \not\preceq j$, then we act as before but now the block of sets for a witness x consists of all G_k with $i \preceq k$. One can now verify that the construction works. The argument for the positive order relations runs as follows: If $i \leq_* j$, find the node σ on the true path by which that fact has been decided. Nodes to the left of σ put only finitely many x into G_i and can be ignored. For nodes to its right that appoint any witness x (necessarily before stage x), we can wait for the node to be initialized to see if x enters G_i. For nodes below σ in the tree assigned to any Ψ requirement, any witness x that puts G_i in its block also puts G_j and so for those x, $x \in G_i \Leftrightarrow x \in G_j$. Of course, for witness x for $\Pi_{e,i,k}$ type nodes, $x \in G_i \Leftrightarrow x \in L$ as before. Of course, for any x not appointed as a witness for one of these type nodes, $x \notin G_i$.

If the partial order is only Σ_3, then one adjust the previous procedure by instead of single nodes with Δ_3 guessing at $i \leq_* j$ putting in individual nodes for each i and j guessing that a particular number is the (least) witness to the Σ_3 fact that $i \leq_* j$. Along a path with the Π_2^0 outcome that the witness is correct, one follows a coding stratgey incorporating this individual fact. If it is true, then some node σ on the true path has the correct witness and all nodes below it obey the required coding strategy. Nodes not below this one, are handled as above. For each node guessing a witness for the Σ_3 fact, where we see that it is false, i.e. along a path with the Σ_2^0 outcome, we put in one more Ψ requirement for $i \not\leq_* j$. So if $i \not\leq_* j$, then along the true path, we will put in $\Psi_{e,i,j}$ requirements for every e and so satisfy the requirement.

7 Requirements II

We now turn to our second technical result:

Theorem 1.7. *For any $n \geq 1$, there are r.e. degrees $\mathbf{g}, \mathbf{p}, \mathbf{q}$, an n-r.e. degree $\mathbf{a}$ and an $n+1$-r.e. degree $\mathbf{d}$ such that:*

1. *For every n-r.e. degree $\mathbf{w} \leq \mathbf{a}$, either $\mathbf{q} \leq \mathbf{w} \vee \mathbf{p}$, or $\mathbf{w} \leq \mathbf{g}$.*

2. $\mathbf{d} \leq \mathbf{a}$, $\mathbf{q} \not\leq \mathbf{d} \vee \mathbf{p}$, *and* $\mathbf{d} \not\leq \mathbf{g}$.

Our list of requirements is very similar to the one used for our first technical theorem:

1. $\Psi_e : \Psi_e(G) \neq D$;
2. $\Theta_e : \Theta_e(D \oplus P) \neq Q$;
3. $\Phi_{e,i} : (\Phi_e(A) = W_i) \rightarrow [\exists\Gamma(\Gamma(W_i \oplus P) = Q) \vee \exists\Delta(\Delta(G) = W_i)]$.

In addition, we need to make $D \leq_T A$. Note that we only add elements into D by diagonalization for Ψ requirements. Whenever we pick a witness x for D, x is fresh at that stage, and we reserve the pair $(x, x+1)$ for coding D into A. If x enters D for the first time, then we also put x into A. If x leaves D later, we either take x out of A or put $x+1$ into A. In the first case, we may shuffle x into and out of A and D simultaneously but allowing at most n changes. In the second case, we may shuffle $x+1$ into and out of A and D simultaneously again allowing at most n changes in A (but this may make for $n+1$ changes in D altogether). Therefore in the end x is in D if and only if x is in A and $x+1$ is not in A. No numbers other than these Ψ-witnesses enter or leave D in our construction, and so D is recursive in A, A is n-r.e. and D is $(n+1)$-r.e.

8 Priority Tree II

Our priority tree here is almost the same as the one used in the first theorem. Of course, we do not have Π nodes. For any Θ node β, we put a new [temporary] outcome r_{α_j} to the left of each g_{α_j}. So the outcomes of β are d, r_{α_1}, g_{α_1}, r_{α_2},...,g_{α_k} and w. We do not add nodes below these type r outcomes. [So a stage s may terminate at such an outcome before we reach level s of the priority tree. We show, however, in Lemma 10.1 that no node of type r can be on the true path.] The notions of active Φ nodes, $\alpha - \beta$ pairs are defined in the same way as in Section 4.

9 Construction II

We only specify the construction at stage s when there is no stage announcement. In the case when there is a stage announcement, we act as in §5.5. Note that in this construction we only change G during P stages. The default permanent restraint is on A, D and G while for P it must be specifically mentioned.

9.1 Ψ node and Φ node

At a Ψ node, the action is the same as the one in §5.1 with G for $L \oplus G_i$ and D for G_i.

At a Φ node α, we follow almost the same procedure as we did in §5.2. The only difference is in how we revise the computations from old Γ-axioms. As in the first construction, if P has changed and the change was caused by some β with g_α outcome by putting the old use into P, then we increase the W-use to $l_s(\alpha)$ and P-use to be fresh. If a W change caused some $\Gamma(x)$ to be undefined, then we check whether x is a diagonalization witness for some β below α, if so, we also check whether D has changed by putting in some number for the first time at the previous stage when β was accessible (and β has not been initialized since). If so, we then redefine $\Gamma(x)$ with W-use up to $l_s(\alpha)$ and fresh P-use. In all other cases we redefine the axiom without changing the uses.

9.2 Θ node

[At a Θ node, the obvious difference from the first construction is that we use D in our Θ computation but use G in our Δ computation. So the arguments in 6.3 are no longer valid. In the case that W changes, we have no reason to expect a G change. In fact, so far we have no requirements or procedures that put numbers into G. Here we actively put numbers into G to correct Δ computations. We will make full use of the fact that D is $n+1$-r.e., i.e., it has one more chance to change than A and the W_i. We may remove a number from D while leaving it in A but putting $z+1$ into A. This will afford us the opportunity to produce a situation in which we may initiate shuffling.]

At a Θ node β accessible for first time after it has been last initialized, we pick a fresh witness x for diagonalizing $\Theta(D \oplus P) \neq Q$. If we have a witness x already assigned (and not yet canceled by initialization) at β, we check whether the Θ computation converges at the witness x. If we do not have a believable (defined in the same way as in our first theorem) computation $\Theta(D \oplus P; x)$, we go to outcome w. If we do, we follow Plan D as in §5.3.1 if we can. If not, we have a planned outcome g_{α_j} as in §5.4.1 and check whether we have not been at this outcome since β was last initialized or whether the previous stage t when we went to this outcome was a P-stage. If so we announce an A-stage and continue the construction below β.

Otherwise, we have two possibilities.

9.2.1 Plan R: removal

Let t be the last stage at which β was accessible. If there is a $y < \gamma(W; x)$ such that $W(y)$ has different values at t and s and the only change in $A \upharpoonright t$ is that some element z entered D and A for the first time at stage t because of the action of a node below β [necessarily a Ψ node], we remove z from D and add $z+1$ into A [so we restore the version of D at stage t up to the θ use]. We go to the r_{α_j} outcome and terminate the current stage of the construction. We call the least such y the *key witness* for the removal plan.

[The idea here is that, before β can become accessible again without being initialized, we would see at α_j if y has left W. If so we would initiate a shuffle there on $z+1$ and initialize β. If not, we will argue that we must go to the left of g_{α_j} and r_{α_j}. Roughly, the idea is that the computation $\Theta(D \oplus P; x)$ will be the same as that at stage t while $\gamma_j(P; x)$ will have been increased above $\theta(x)$ by y entering W.]

9.2.2 Plan G: change G

If we satisfy none of the above criteria, we go to the outcome g_{α_j} and continue to build Δ consistently. For each u, if $\Delta(u)$ was defined at the last stage t at which $\beta\hat{\ } g_{\alpha_j}$ was accessible and $W(u)$ has not changed since then, we simply update the Δ axiom with the current version of G (if necessary) without changing the use. If $\Delta(u)$ was defined at t but $W(u)$ is now different, then let $\delta(u)$ be the use of G in the old Δ computation. We add $\delta(x) - 1$ into G and redefine Δ with a fresh use in G. [We preserve the consistency of Δ by doing this as G is r.e.] Then we also define a new computation $\Delta(u)$ for the next u which was undefined with fresh G-use. Finally, we follow Plan P to add γ uses into P as in Section 5.4.2 and announce a P-stage.

10 Verification II

We can go through most of Section 6 and show that we have a leftmost path visited infinitely often (that it is actually infinite follows from Lemma 10.1), and each node along the true path is passing down alternating A-stages and P-stages along the true path. There are obvious alphabetic changes needed In Lemmas 6.1-6.6 – no L or Π. Otherwise, note first that Plan R action for Ψ nodes are an exception to Lemma 6.1. Next, Lemma 6.2 applies to D as well as A and we have to remark that if we implement Plan R no node is even accessible thereafter, while Plan R action cannot be the second type to change A (or D) at s by the arguments given in the proof of Lemma 6.2. Finally, for the proof of 6.3 note that G-uses for Δ are also chosen fresh. It is then not difficult to see that functionals are well-defined and complete the job we assigned if they are along the true path. For the Γ's, use Lemma 6.7. For the Δ's, it is directly guaranteed by our construction. [The complicated argument for Lemma 6.8 is not needed but see the proof of Lemma 10.1 for some remnants of it.]

For the verification that all the requirements are satisfied we continue as in §6.4. The positive order requirements (Lemma 6.9) are simply replaced by the requirement that $D \leq_T A$. Our construction guarantees that x is in D if and only if x is in A and $x+1$ is not. Except for the satisfaction of the Ψ requirements (Lemma 6.10) in the case of d outcome all the other verifications proceed as in §6.4.

As for the Ψ requirements, the major issue is that here we add elements into G (when we construct Δ) and this action might, *a priori*, injure some apparently satisfied Ψ requirement of lower priority. To show that the Ψ requirements are all satisfied, we first need a few lemmas.

Lemma 10.1. *No outcome of type r can be on the true path.*

Proof. The argument is similar to the one in the end of the proof of Lemma 6.8. Consider any Θ node β on the true path and suppose, for the sake of a contradiction, that its true outcome is r_α.

Let $s = s(\beta\hat{\ } r_\alpha)$ and, as in the construction, let t be the previous stage at which β had outcome g_α; x, the diagonalization witness at β; y, the key witness for Plan R at s; and z, the unique element that entered D for the first time at t. We remove z from D at s following Plan R. Let s' be the next stage at which β is accessible with a believable Θ computation. If y was ever out of W between s and s' (when α was accessible) then, we would have initiated a shuffle plan at α and so moved left of $\beta \supseteq \alpha\hat{\ }i$ for a contradiction. (As we terminate stage s at $\beta\hat{\ }r_\alpha$ with no action, there is no change in A at s. Between s and s' Lemma 6.3 shows that $A \upharpoonright s$ is preserved. So if $W(y)$ changes we satisfy the conditions for shuffling at α.) Thus we also assume that y remains in W at every stage at which α is accessible through stage s'.

Now at stage s we restored the computation $\Theta(D \oplus P, x)$ of stage t by removing z from D: z is the only change to D up to $\theta(x)$ at t by Lemma 6.2, and P is preserved by the A-stage announcement at t. Between t and s, $D \upharpoonright t$ is preserved by Lemma 6.3 and $P \upharpoonright \theta(x)$ is not injured by nodes to the right. Finally, our believability condition guarantees that $P \upharpoonright \theta(x)$ is also not injured by nodes above β. Thus at stage s' we still have the same computation of $\Theta(x)$ as at stage t.

Now consider the Γ computation we build at $\alpha\hat{\ }i$. At stage s the conditions for Plan R guarantee that $y < \gamma(W; x)$ (at t) has entered W for the first time after t and by s. So when this happens and we are at α we see a change in the W part of $\Gamma(x)$ which makes $\Gamma(x)$ undefined. Therefore, by our construction, we add a new axiom with W-use the current length of agreement and P-use fresh, which is larger than $\theta(P, x)$ at t. This $\gamma(P, x)$ remains large since y remains in W, therefore at stage s' we will see that $\gamma(P, x) > \theta(P, x)$.

For other active α_l's between α and β, the corresponding $\gamma_l(W_l; x)$are larger than $\theta(P, x)$ at stage t by the rules for going to outcome g_α. The only way that anyone of these uses can decrease is by W_l changing at some stage $v \leq s'$ from its value on $\gamma_l(W_l; x)$ at t back to an older version. If this happens, then when α_l is accessible at $v \leq s'$ we initiate a shuffle at α_l by shuffling $z + 1$ which we added into A by Plan R at t with shuffle points t and v. This would move us to the left of β for a contradiction. Therefore these uses cannot decrease, and at stage s' they are still larger than $\theta(P, x)$. Hence at stage s' we will go to the left of the r_α outcome by our construction for the desired contradiction. $\square$

Lemma 10.2. *In the construction at a* Θ *node* β*, if we change* G *as in 9.2.2 at a stage* $s > s(\beta)$*, then it must be the case that at the previous stage* t *when* $\beta\hat{}g_\alpha$ *was accessible, we followed either Plan S or Plan R to change* D *at some node* σ *below* β*.*

Proof. By our construction, if we must change G by putting in some $\delta(u)-1$, then $W(u)$ and therefore $A \upharpoonright \phi(u)$ must be different at stage s from stage t. By Lemmas 6.3 (for nodes to the right), 6.2 (for nodes below), 6.1 (for nodes above not of type r) and 10.1 (to see that there are no type r nodes above), such a change in A can only happen at stage t. Thus we must have changed A and D at stage t below β since β was accessible.

We can change A and D in only three ways: diagonalization at a Ψ node, Plan S or Plan R. By Lemma 6.2 if we have applied diagonalization at a node below β, then it is the only change at stage t, so at stage s according to our construction we would want to apply Plan R to remove the element added into D at stage t [and restore the Θ computation]. Depending on the current stage announcement we would then have outcome either r or w. Since by assumption the outcome at s is g_{α_j}, we must have followed either Plan S or Plan R at t. □

Now we can prove that Ψ requirements are not injured by Plan G.

Lemma 10.3. *If* σ *is a* Ψ *node* ($\Psi(G) \neq D$) *on the true path and we go to outcome* d *after stage* $s(\sigma)$*, then the diagonalization will not be injured thereafter, i.e., the* Ψ*-use in* G *is preserved and the diagonalization witness* x *is not removed from* D *and so the* Ψ *requirement is satisfied.*

Proof. Suppose at stage $s > s(\sigma)$ we diagonalized at σ by putting x into D and so impose permanent restraint on D. First of all, x cannot be taken out of D at subsequent stages, since only nodes above σ could do so and then only if we follow either Plan S or Plan R at a node above σ. Plan S action would move us to the left for a contradiction and there are no outcomes r above σ by Lemma 10.1.

Next we claim that the G-use is also preserved. No node below or to the right of σ can add elements below this G-use into G after stage s, since their Δ-uses are defined to be fresh. The only worry is that some Θ node β with $\beta\hat{}g_\alpha$ above σ might follow Plan G (at stage $s_1 > s$) to add $\delta(y)-1$ into G for some changed u in W in order to correct some Δ axiom. The $\Delta(G;u)$ axiom being killed must have been enumerated before stage s, since its use was chosen fresh. So $W(u) = \Phi(A;u)$ at stage s.

By Lemma 10.2, the only circumstances under which we change G at stage s_1 in response to this change in W is that D has already changed by some other node τ below $\beta\hat{}g_\alpha$ following Plan S or Plan R at the previous stage t at which $\beta\hat{}g_\alpha$ was accessible. Such a τ cannot be above σ as we would then move to its left, so it must be to the right of, or below, $\sigma\hat{}d$.

Then there must be another Ψ' node σ' (below τ) which added a number into D after stage s (since at stage s we initialized all nodes to the right of τ, any change before s

cannot be used in shuffling or removal) and the number is taken out by τ. However, σ' cannot be below, or to the right of, σ, as its witness would then be larger than s and so could not affect the value of $W(u)$ which agrees with $\Phi(A;u)$ at stage s. So we get a contradiction.

In another words, once we apply diagonalization for σ, we (automatically) implement restraints for A and D and hence to W's up to the part that we have coded in. So we can guarantee that there can be no change in W which makes us change G on the uses we have already seen, and in particular, the Ψ-use of G is preserved. □

11 Bibliography

Arslanov, M. M. [1985], Lattice properties of the degrees below $\mathbf{0}'$, Doklady Ak. Nauk SSR **283**, 270-273.

Arslanov, M. M. [2009], Definability and elementary equivalence in the Ershov difference hierarchy in *Logic Colloquium 2006*, Lecture Notes in Logic, Association for Symbolic Logic and Cambridge University Press, New York, 1–17.

Arslanov, M. M. [2010], The Ershov hierarchy in *Computability in Context: Computation and Logic in the Real World*, Cooper, S. B. and Sorbi, A. eds., Imperial College Press, London.

Arslanov, M. M., Kalimullin, I. Sh. and Lempp, S. [2010], On Downey's Conjecture, *Journal of Symbolic Logic* **75**, 401-441.

Cooper, S. B. [1971], *Degrees of Unsolvability*, Ph.D. Thesis, Leicester University.

Cooper, S. B. Harrington, L., Lachlan, A. H.. Lempp S. and Soare, R. I.[1991], The d-r.e. degrees are not dense, *Annals of Pure and Applied Logic* **55**, 125-151.

Downey, R. G. [1989], D-r.e. degrees and the non-diamond theorem, *Bulletin London Mathematical Society* **21**, 43-50.

Epstein, R. L. [1979], *Degrees of Unsolvability: Structure and Theory*, **LNM** 759, Springer-Verlag, Berlin.

Ershov, Y. I. [1968], On a hierarchy of sets I, *Algebra and Logic* **7**, 25-43.

Ershov, Y. I. [1968], On a hierarchy of sets II, *Algebra and Logic* **7**, 212-232.

Ershov, Yu. I. [1970], On a hierarchy of sets III, *Algebra and Logic* **9**, 20-31.

Gold, E. M. [1965], Limiting recursion, *Journal of Symbolic Logic* **30**, 28-48.

Harrington, L. and Shelah, S. [1982], The undecidability of the recursively enumerable degrees (research announcement), *Bulletin of the American Mathematical. Society*, N.S. **6**, 79-80.

Hodges, W. [1993], *Model Theory*, Cambridge University Press, Cambridge U.K.

Jockusch, C. G. Jr. and Shore, R. A. [1984], Pseudo-jump operators II: transfinite iterations, hierarchies and minimal covers, *Journal of Symbolic Logic* **49**, 1205-1236.

Lachlan, A. H. [1966], Lower bounds for pairs of recursively enumerable degrees, *Proceedings of the London Mathematical. Society* (3) **16**, 537-569.

Lachlan, A. H. [1968], Distributive initial segments of the degrees of unsolvability, *Z. Math. Logik Grund. Math.* **14**, 457-472.

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

Nies, A., Shore, R. A. and Slaman, T. A. [1998], Interpretability and definability in the recursively enumerable degrees, *Proceedings of the London Mathematical. Society* (3) **77**, 241-291.

Putnam, H. [1965], Trial and error predicates and the solution to a problem of Mostowski, *Journal of Symbolic Logic* **30**, 44-50.

Sacks, G. E. [1966], *Degrees of unsolvability*, Annals of Math. Studies **55**, Princeton University Press, 2^{nd} ed., Princeton NJ.

Shoenfield, J. R., [1959], On degrees of unsolvability, *Annals of Mathematics* **69,** 644-653.

Shoenfield, J. R. [1965], An application of model theory to degrees of unsolvability, in *Symposium on the Theory of Models*, J. W. Addison, L. Henkin and A. Tarski eds., North-Holland, Amsterdam, 359-363.

Shore, R. A. [1981], The theory of the degrees below $0'$, *Journal of the London Mathematical Society* **24**, 1-14.

Shore, R. A. [1988], Defining jump classes in the degrees below $0'$, *Proceedings of the American Mathematical Society* **104**, 287-292.

Shore, R. A. [2006], Degree structures: Local and global investigations, *Bulletin of Symbolic Logic* **12**, 369-389.

Shore, R. A. [2007], Direct and local definitions of the Turing jump, *Journal of Mathematical Logic* **7**, 229-262.

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

Slaman, T. A. [1983] The recursively enumerable degrees as a substructure of the Δ_2^0 degrees, handwritten notes.

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

Slaman, T. A. and Woodin, H. [2001], *Definability in Degree Structures*, preprint.

Stephan, F. Yang, Y. and Yu, L. [2009], Turing degrees and the Ershov hierarchy in *Proceedings of the Tenth Asian Logic Conference, Kobe, Japan, 1-6 September 2008*, World Scientific, 300-321.

Taitslin, M. A. [1962], Effective inseparability of the sets of identically true and finitely refutable formulas of elementary lattice theory, *Algebra i Logika* **3**, 24-38.

Turing, A. M. [1939], Systems of logic based on ordinals, *Proceedings of the London Mathematical. Society* (3) **45**, 161-228.

Yang, Y. and Yu, L. [2006], $\mathcal{R}$ *is not a* Σ_1*-elementary substructure of* $\mathcal{D}_n$, *Journal of Symbolic Logic* **71**, 1223–1236.

Yates, C. E. M. [1966], A minimal pair of recursively enumerable degrees, *Journal of Symbolic Logic* **31**, 159-168.