2001

An Almost Deep Degree

Peter Cholak
University of Notre Dame

Marcia Groszek
Dartmouth College

Theodore Slaman
University of California - Berkeley

An Almost Deep Degree

Peter Cholak[*]
University of Notre Dame

Marcia Groszek
Dartmouth College

Theodore Slaman[†]
University of California – Berkeley

August 17, 1999

Abstract

We show there is a non-recursive r.e. set A such that if W is any low r.e. set, then the join $W \oplus A$ is also low. That is, A is "almost deep". This answers a question of Jockusch. The almost deep degrees form an definable ideal in the r.e. degrees (with jump.)

1 Introduction

Bickford and Mills in [1] defined an r.e. degree $\mathbf{a}$ to be *deep* in case, for all other r.e. degrees $\mathbf{b}$,

$$(\mathbf{b} \oplus \mathbf{a})' = \mathbf{b}'.$$

In other words, joining with $\mathbf{a}$ preserves the jump of every r.e. degree. They asked whether there are non-recursive deep degrees.

Part of the motivation for asking this question is an interest in finding definable ideals in the r.e. degrees. Such ideals can help in understanding the global structure of the r.e. degrees; in particular they can provide ways of understanding definability properties and automorphisms. We know of few

[*]Partially supported by NSF Grant DMS-96-3465.
[†]Partially supported by NSF Grant DMS-97-96121.

non-trivial examples of such ideals. One example is the ideal of halves of minimal pairs.

The deep degrees necessarily form an ideal in the r.e. degrees (in this case, definable using the jump.) However, this ideal turns out to be trivial. Lempp and Slaman in [3] showed that there are no non-recursive deep degrees: For any non-recursive r.e. degree $\mathbf{a}$, there is an r.e. degree $\mathbf{b}$ such that

$$(\mathbf{b} \oplus \mathbf{a})' >_T \mathbf{b}'.$$

Joining with $\mathbf{a}$ does not preserve the jump of $\mathbf{b}$.

One can try to salvage the idea of the deep degree by requiring only that joining with $\mathbf{a}$ preserves the jump on some subclass of r.e. degrees. However, there are severe limitations on this possibility: Analysis of the construction in [3] shows that the double jump of $\mathbf{b}$ can be controlled; in particular, $\mathbf{b}$ can be constructed to be low_2. Therefore there is no non-recursive r.e. degree $\mathbf{a}$ such that joining with $\mathbf{a}$ preserves the jump on all low_2 r.e. degrees, or on any other collection defined using the double jump.

This leaves open the possibility that there is a non-recursive r.e. degree $\mathbf{a}$ such that joining with $\mathbf{a}$ preserves the jump on low r.e. degrees; that is, such that joining with $\mathbf{a}$ preserves lowness. We call such a degree *almost deep.*

The almost deep degrees form an ideal in the r.e. degrees: Suppose that $\mathbf{a}_1$ and $\mathbf{a}_2$ are almost deep and $\mathbf{b}$ is low. Because $\mathbf{a}_1$ is almost deep, $\mathbf{b} \oplus \mathbf{a}_1$ is low; then because $\mathbf{a}_2$ is almost deep, $(\mathbf{b} \oplus \mathbf{a_1}) \oplus \mathbf{a}_2 = \mathbf{b} \oplus (\mathbf{a_1} \oplus \mathbf{a}_2)$ is low; this shows that $\mathbf{a}_1 \oplus \mathbf{a}_2$ is almost deep. The ideal of almost deep degrees is definable (using jump, or using the property of lowness.)

In this paper we show that this ideal is non-trivial: There exists a non-recursive almost deep degree. This answers an unpublished question of Jockusch.

Theorem 1 *There is a non-recursive r.e. set A such that if W is any low r.e. set, then the join $W \oplus A$ is also low.*[1]

In section 2 we outline the main ideas in the construction of A. In section 3 we give some notation and conventions. (The details of section 3 can be referred to in order to make the construction completely precise, but we believe that the reader familiar with priority arguments can read the rest of

[1]Both Harrington and Sui have announced and later withdrawn the same theorem.

the paper and fill in most missing details without reference to section 3.) In sections 4 and 5 we present the construction. In section 6 we prove that the construction does in fact produce an r.e. set A with the desired properties.

2 The Idea of the Construction

We enumerate A using a priority argument.

2.1 Requirements and strategies

There are two types of requirements. The simplest are non-recursiveness requirements. We guarantee that A is not recursive by making A simple: We make A coinfinite and, for each e, satisfy the requirement

> P_e: If W_e is infinite then $W_e \cap A$ is non-empty.

A strategy to satisfy this requirement waits for numbers greater than $2e$ to be enumerated into W_e, and then attempts to enumerate one such number into A. If such a strategy has infinitely many stages at which it can act, and the restraint higher priority strategies impose on A during those stages has finite lim inf, then the strategy will succeed in satisfying the requirement.

The remaining requirements guarantee that A is almost deep. We fix an enumeration (W_e, Δ_e) of all pairs of r.e. sets and possible recursive approximations to Δ_2 functions. For each e, we must satisfy the requirement

> R_e: If W_e is low and Δ_e approximates its jump as a Δ_2 function, then $W_e \oplus A$ is low and its jump is approximated as a Δ_2 function by some Φ_e (which we will build.)

We have subrequirements of each of these requirements. We fix an enumeration (Γ_i) of all recursive functionals. For each i we must satisfy the requirement

> $S_{e,i}$: Either Φ_e correctly predicts the convergence (or divergence) of $\Gamma_i(W_e \oplus A)$ (that is, Φ_e predicts whether $i \in (W_e \oplus A)')$, or Δ_e fails to correctly predict the convergence (or divergence) of some $\Theta_i(W_e)$ (which we will build.)

If all of these requirements are satisfied via the first clause, R_e is satisfied because Φ_e approximates $(W_e \oplus A)'$. If one is satisfied via the second clause, R_e is satisfied because Δ_e does not approximate W_e'.

We will include R_e as well as the $S_{e,i}$ in our priority ordering. The reasons for this will appear later. More or less, the priority assigned to R_e will be that of guaranteeing that Φ_e does approximate a Δ_2 function (that is, for each i, either Φ_e eventually predicts $\Gamma_i(W_e \oplus A)\downarrow$ or Φ_e eventually predicts $\Gamma_i(W_e \oplus A)\uparrow$), while the priority assigned to $S_{e,i}$ will be that of guaranteeing that Φ_e gives the correct prediction for the convergence of $\Gamma_i(W_e \oplus A)$.

Ignoring for the moment this complexity, here is the basic strategy for meeting requirement $S_{e,i}$:

(0.) Set Φ_e to predict $\Gamma_i(W_e \oplus A)\uparrow$.

(1.) Wait until a stage at which a computation $\Gamma_i(W_e \oplus A)\downarrow$ appears. Restrain the set A on the use of the computation $\Gamma_i(W_e \oplus A)\downarrow$.

(We are doing our part to preserve the computation. If we could preserve the computation, we could set Φ_e to predict convergence. However, we cannot guarantee convergence, because we are not in control of the enumeration of W_e. Instead of changing Φ_e we:)

Enumerate a computation $\Theta_i(W_e)\downarrow$ with the same use on W_e as the computation $\Gamma_i(W_e \oplus A)\downarrow$.

(Now we have two computations, $\Gamma_i(W_e \oplus A)$ and $\Theta_i(W_e)$, with the same use on W_e. As long as we are restraining A, these two computations will either both continue to converge or — if W_e changes on their common use — both diverge.)

(2.) Wait until a stage at which either $\Gamma_i(W_e \oplus A)\uparrow$ and $\Theta_i(W_e)\uparrow$ (due to a change in W_e), or else Δ_e predicts that $\Theta_i(W_e)\downarrow$. In the first case, drop the restraint on A and return to step (1). In the second, set Φ_e to predict $\Gamma_i(W_e \oplus A)\downarrow$ and go on.

(3.) Wait until a stage at which W_e changes below the common use of the computations $\Gamma_i(W_e \oplus A)$ and $\Theta_i(W_e)$ (so that now $\Gamma_i(W_e \oplus A)\uparrow$ and $\Theta_i(W_e)\uparrow$.) Drop the restraint on A.

(4.) Wait until a stage at which Δ_e predicts that $\Theta_i(W_e)\uparrow$. Set Φ_e to predict $\Gamma_i(W_e \oplus A)\uparrow$. Return to step (1).

Here is the idea behind this strategy. We are trying to satisfy the requirement $S_{e,i}$: If Δ_e correctly predicts the convergence of $\Theta_i(W_e)$ then Φ_e correctly predicts the convergence of $\Gamma_i(W_e \oplus A)$. We imagine an opponent enumerating the computations Γ_i and Δ_e and the set W_e. When a computation $\Gamma_i(W_e \oplus A) \downarrow$ appears, our opponent is challenging us to switch Φ_e to predict convergence. We don't respond to that challenge by immediately changing Φ_e, because if we always did that our opponent could keep Φ_e from settling down on a prediction just by issuing infinitely many challenges. Instead, by making $\Theta_i(W_e) \downarrow$ (with the same use as $\Gamma_i(W_e \oplus A)$), we in turn challenge our opponent to switch Δ_e to predict convergence. (We also restrain A to tie the convergence of $\Gamma_i(W_e \oplus A)$ to the convergence of $\Theta_i(W_e)$.)

Our opponent, who must make sure that Δ_e makes correct predictions, has two possible responses. One is to withdraw both challenges by changing W_e to make both computations diverge, in which case we have gained because Φ_e's prediction of divergence is once again valid and we haven't changed Φ_e. (Changing the prediction of Φ_e brings Φ_e one step closer to never settling down on a prediction.) Our opponent's second option is to change Δ_e to predict convergence, in which case we then change Φ_e to predict convergence; we have gained because Φ_e's prediction is once again valid, and although Φ_e is one step closer to never settling down, so is Δ_e.

The basic idea is to force our opponent into the same situations we are in, and then mimic our opponent's moves. The only way our opponent can lure us into changing Φ_e infinitely often is by changing Δ_e infinitely often. If our opponent succeeds in making Δ_e give a correct prediction then we succeed in making Φ_e give a correct prediction, and if our opponent does not succeed then the requirement is satisfied and we don't care about Φ_e's prediction.

Assuming (for the moment) that this strategy is always allowed to act and is never injured, there are two kinds of possible outcomes, finitary and infinitary.

> Finitary: Past some stage s, we are stuck forever waiting at some step. In this case, either Φ_e correctly predicts whether $\Gamma_i(W_e \oplus A)$ converges (if we wait forever at steps (1) or (3)), or Δ_e fails to correctly predict whether $\Theta_i(W_e)$ converges (if we wait forever at steps (2) or (4).) In these outcomes, the strategy imposes a finite permanent restraint on A.

> Infinitary: Either past some stage s, we alternate forever between steps (1) and (2), or we infinitely often cycle through all four steps. In the first case, Φ_e correctly predicts the divergence of $\Gamma_i(W_e \oplus A)$; in the second, Δ_e fails to correctly predict the divergence of $\Theta_i(W_e)$. (Δ_e fails to reach a limit, since it infinitely often changes its prediction.) In either case, the strategy infinitely often drops its restraint on A (whenever it returns to step (1).)

In every outcome, the requirement $S_{e,i}$ will be satisfied. Also in every outcome, the strategy creates an environment in which a lower priority P_c requirement can be satisfied: in the finitary outcomes, because the $S_{e,i}$ strategy eventually stops acting, imposing only finitely much restraint that P_c must respect; in the infinitary outcomes, because there are infinitely many stages at which $S_{e,i}$ is imposing no restraint at all.

2.2 Combining strategies

The difficulty in combining strategies for different requirements appears when we consider the case of strategies for requirements $S_{e,i}$ and $S_{d,j}$, each of which has an infinitary outcome. Infinitely often, the strategy for $S_{e,i}$ will drop all its restraint on A; the stages when this happens will be determined by changes in W_e. Subsequently it will impose new (perhaps higher) restraint on A. Meanwhile, the strategy for $S_{d,j}$ will act similarly, the stages at which it drops its restraint in A determined by changes in W_d. It is possible that the changes in W_e and W_d are staggered so that the two strategies never drop their restraint on A at the same time and, in fact, combine to produce restraint on A with infinite lim inf. This would prevent lower priority non-recursiveness requirements from being satisfied.

This is a completely standard sort of difficulty that appears in an infinite injury ($\emptyset''$ or Π_2) construction. We can get around it in the standard way, by having two versions of $S_{d,j}$.

One version, $S_{d,j}$, guesses that $S_{e,i}$'s restraint will have lim inf equal to zero (that is, $S_{e,i}$ has an infinitary outcome or a finitary outcome with zero restraint.) This strategy $S_{d,j}$ acts only when $S_{e,i}$ is imposing no restraint. At other stages, $S_{d,j}$ goes dormant, holding its restraint but taking no new action. This means that if $S_{d,j}$ is going to drop its restraint, it *must* do so when $S_{e,i}$ is imposing no restraint, because those are the only stages at which it can act. If $S_{e,i}$'s restraint does have zero lim inf, this strategy $S_{d,j}$ gets infinitely many chances to act (and its restraint is respected in the intervening stages); if $S_{e,i}$ has a finitary outcome with positive restraint, $S_{d,j}$ acts only finitely often so it has a finite effect on the construction. When $S_{e,i}$'s restraint has zero lim inf, the lim inf of $S_{d,j}$'s restraint is also the lim inf of the combined restraint.

The second strategy, $\hat{S}_{d,j}$, guesses that $S_{e,i}$ will have a finitary outcome with positive restraint, and acts only when $S_{e,i}$ is imposing restraint; at other stages, it is injured and drops all its restraint. This strategy $\hat{S}_{d,j}$ has no effect if $S_{e,i}$'s restraint has zero lim inf (since it is injured and has no effect during the critical stages when $S_{e,i}$ has dropped its restraint to zero), and if $S_{e,i}$ has a finitary outcome with positive restraint, it gets to act at every stage after $S_{e,i}$ has settled down. When $S_{e,i}$ has a finitary outcome with positive restraint, the lim inf of the combined restraint is the maximum of the lim inf of $\hat{S}_{d,j}$'s restraint and the final values of $S_{e,i}$'s restraint and $S_{d,j}$'s restraint.

Of course, our next strategy will have to guess at the outcomes of both $S_{e,i}$ and the relevant version of $S_{d,j}$, and so on. To keep track of all this, we build our construction on a tree; each node is a (version of a) strategy, and its immediate successors correspond to the possible outcomes of that strategy. During step s of the construction, we follow a path down to level s of the tree, at each level taking action according to the strategy we find on our path, and using its current restraint to tell us which version of the strategy on the next level we should go to.

There is another problem, caused by the solution we have just described. If $S_{e,i}$ has an infinitary outcome, the action of $S_{d,j}$ is supposed to guarantee that its requirement is satisfied. But the wrong-headed $\hat{S}_{d,j}$ gets infinitely many chances to act, during which it might switch the prediction of Φ_d on $\Gamma_j(W_d \oplus A)$. This will interfere with the action of $S_{d,j}$ if it happens infinitely often, because that would make Φ_d fail to settle down on a prediction regardless of what $S_{d,j}$ was doing.

This is the kind of difficulty that appears in a level 3 (Π_3 or $\emptyset'''$) construction. Broadly, the difficulty is that the Π_2-type requirements, the $S_{d,j}$, are not completely independent. They come in infinite families (subrequirements of a given R_d) whose actions must be coordinated; in our case, subrequirements of R_d affect the functional Φ_d in ways that might interfere with each other.

To address this problem, as usual, we put the requirement R_d on the tree above all the $S_{d,j}$'s. The role of R_d is to coordinate the action of its subrequirements to guarantee they don't interfere with each other. In this proof, the way the R_d strategy does this is by taking over steps (1) and (2) of the $S_{d,j}$ strategies and executing them in a coordinated way. In this construction only R_d can switch Φ_d to predict convergence. This prevents an individual $S_{d,j}$ from unilaterally switching the Φ_d prediction back and forth infinitely many times.

Of course, $\hat{S}_{d,j}$ (to return to our earlier example) can still switch Φ_d to predict divergence, but this is harmless to $S_{d,j}$: It won't happen when switching Φ_d conflicts with R_d, because R_d has higher priority. It won't happen when $S_{d,j}$ needs to wait at step (3), because if $\hat{S}_{d,j}$ makes this switch it is because $\Gamma_i(W_e \oplus A)$ actually does diverge at this stage, and $S_{d,j}$ will therefore move to step (4) at the next opportunity. And if it happens when $S_{d,j}$ is waiting at step (4) there is no problem, because if $S_{d,j}$ remains waiting at step (4) it is satisfied because Δ_e makes an incorrect prediction regardless of what Φ_d predicts, and if $S_{d,j}$ moves past step (4) it will itself switch Φ_d to predict divergence.

When R_d is carrying out steps (1) and (2) of an $S_{d,j}$ strategy, we will say that $S_{d,j}$ is *connected to* R_d. At a later stage $S_{d,j}$ may be disconnected either through the action of R_d or because $S_{d,j}$ or R_d is canceled. These connections are not the same as the links between nodes of a tree of strategies that are sometimes used in Π_3 or $\emptyset'''$ constructions; the "path of the construction" (the stagewise approximation to the true path) on a tree with links travels from the node at the top of a link directly down the link to the node at the bottom of the link, not visiting any of the intermediate nodes. This does not happen with the connections in our construction; the existence of a connection affects the action of the strategy R_d but not the path of the construction through the tree.

2.3 More requirements and strategies

We now have different versions of requirements R_e and $S_{e,i}$ that will be arranged on a tree. The version of requirement R_e placed at node α of the tree will be called R_e^α. Below it, for each i, will be a collection of subrequirements $S_{e,i}^\beta$ (which will also occur in different versions placed at different β's.) In order to prevent interference between different versions of the same requirement, insofar as possible, different versions will be building different functions and functionals. The strategy for R_e^α will be building a possible approximation Φ_e^α to a Δ_2 function, and will be trying to satisfy the version of requirement R_e

> R_e^α: If W_e is low and Δ_e approximates its jump as a Δ_2 function, then $W_e \oplus A$ is low and its jump is approximated as a Δ_2 function by Φ_e^α.

Its subrequirements $S_{e,i}^\beta$ will necessarily be concerned with the same Φ_e^α, but each will enumerate a different partial recursive functional Θ_i^β, and will be trying to satisfy the version of requirement $S_{e,i}$

> $S_{e,i}^\beta$: Either Φ_e^α correctly predicts the convergence (or divergence) of $\Gamma_i(W_e \oplus A)$, or Δ_e fails to correctly predict the convergence (or divergence) of $\Theta_i^\beta(W_e)$.

(This is why we have not bothered to index the Θ's as $\Theta_{i,e}$; in the formal construction we use Θ_i^β, and β will uniquely determine e.) For the rest of this section we will drop the superscripts α and β.

The strategy for R_e has the job of executing steps (1) and (2) of the earlier strategy for each $S_{e,i}$ in a coordinated way. Here is the basic strategy R_e will follow for each i. (The strategies for different i are independent of each other.)

> (0.) Wait for some requirement $S_{e,i}$ to ask R_e to begin step (1) of its strategy. We say $S_{e,i}$ connects to R_e.
>
> (1.) Wait until a stage at which a computation $\Gamma_i(W_e \oplus A)\downarrow$ appears. Restrain the set A on the use of the computation $\Gamma_i(W_e \oplus A)\downarrow$. Enumerate computations $\Theta_i(W_e)\downarrow$ with the same use on W_e as the computation $\Gamma_i(W_e \oplus A)\downarrow$ for every version of $S_{e,i}$ connected to R_e. (While R_e is waiting for $\Gamma_i(W_e \oplus A)\downarrow$ at this step, it can allow new versions of $S_{e,i}$ to connect.)
>
> (2.) Wait until a stage at which either $\Gamma_i(W_e \oplus A)\uparrow$ and $\Theta_i(W_e)\uparrow$ for all versions of $S_{e,i}$ connected to R_e (due to a change in W_e), or else Δ_e predicts that $\Theta_i(W_e)\downarrow$ for all versions of $S_{e,i}$ connected to R_e. In the first case, drop the restraint on A and return to step (1). In the second, set Φ_e to predict $\Gamma_i(W_e \oplus A)\downarrow$,

disconnect all the versions of $S_{e,i}$ and return to step (0). Drop the restraint on A. The $S_{e,i}$ now begin their strategy at step (3) with the restraint R_e was holding. (While R_e is waiting at this stage, it does not allow new versions of $S_{e,i}$ to connect.)

Assuming (for the moment) that this strategy is always allowed to act and is never injured, there are two kinds of possible outcomes, finitary and infinitary.

Finitary: Past some stage s, we are stuck forever waiting at some step. In this case, either Φ_e correctly predicts whether $\Gamma_i(W_e \oplus A)$ converges (if we wait forever at step (1)), or Δ_e fails to correctly predict whether some version $\Theta_i(W_e)$ converges (if we wait forever at step (2)), or some version of $S_{e,i}$ eventually waits forever at step (3) or (4) of the strategy insuring $S_{e,i}$ is satisfied (if we wait forever at step (0).) In the last case, our analysis of the $S_{e,i}$ strategy still applies. In these outcomes, the strategy imposes a finite permanent restraint on A.

(In the case that we wait forever at step (2), it is important to know that no new versions of $S_{e,i}$ will connect to R_e while we are waiting. This means that we are waiting for Δ_e to settle down on finitely many predictions; if it does not do so, there must be a single $\Theta_i(W_e)$ for which Δ_e does not settle down to predict convergence.)

Infinitary: Either past some stage s, we alternate forever between steps (1) and (2), or we infinitely often cycle through all three steps. In the first case, Φ_e correctly predicts the divergence of $\Gamma_i(W_e \oplus A)$; in the second, some version of $S_{e,i}$ is (with the help of R_e) cycling through all four steps of the strategy insuring $S_{e,i}$ is satisfied. In the second case, our analysis of the $S_{e,i}$ strategy still applies. In these outcomes, the strategy infinitely often drops all restraint on A.

There are only a couple of important ideas left for the detailed description of the construction. The first is a (standard) organization of the tree construction so that for all e and i: there is a version of R_e that after some stage gets to act infinitely often and is never injured, if the last clause of either of these R_e outcomes holds there is a single version of $S_{e,i}$ that after some stage gets to act infinitely often and is never injured, and the strategy for P_e has a version that gets infinitely many chances to act and has to respect only a finite restraint. The second is coordination of the action of R_e for all the subrequirements $S_{e,i}$ for different i. Without such coordination, an R_e that had infinitary outcomes for two different i could conceivably impose restraint with infinite lim inf by alternating which value of i it was imposing restraint for. We organize the construction so that we can show this doesn't happen by an analysis of true stages in the enumeration of W_e. The rest is in the details.

3 Notation and Conventions

3.1 Basics

We freely blur the distinction between ordered pairs, finite sets, etc. and their numerical codes.

We use the notation $[\sigma]$ to indicate approximations to r.e. sets and to computations at the beginning of stage σ of our construction. In particular, if W is an r.e. set, then $W[\sigma]$ is the set of elements enumerated into W before stage σ. $W \cap A = \emptyset[\sigma]$ indicates that $W[\sigma] \cap A[\sigma] = \emptyset$, and so on.

3.2 Partial recursive functionals and convergence

A *neighborhood condition* is a pair of disjoint finite sets (P, N). This neighborhood condition is *complete* in case $P \cup N$ is an initial segment of ω. It *applies to* a set W in case $P \subset W$ and $N \cap W = \emptyset$.

Because in approximating the jump our concern with partial recursive functionals is only with whether they converge or diverge, we will view a partial recursive functional Γ as an r.e. set of complete neighborhood conditions. For a set W, $\Gamma(W)$ converges ($\Gamma(W)\downarrow$) iff there is a neighborhood condition in Γ that applies to W. We may refer to this neighborhood condition as a *computation* in Γ witnessing $\Gamma(W)\downarrow$. The *use* of this computation is $max(P \cup N)$. We have required that neighborhood conditions be complete because we want to know that if a computation applies to an r.e. set

W at some stage, any later change of W below its use will guarantee that the computation no longer applies to W. There is no harm in this, because any neighborhood condition is equivalent to a finite collection of complete neighborhood conditions.

Consistent with the $[\sigma]$ notation, we say that $\Gamma(W) \downarrow [\sigma]$ just in case there is a neighborhood condition in $\Gamma[\sigma]$ that applies to $W[\sigma]$.

When considering whether a neighborhood condition applies to a recursive join $W \oplus A$, we may view (P, N) as (coding) $((P_0, P_1), (N_0, N_1))$, and say that (P, N) applies to $W \oplus A$ just in case (P_0, N_0) applies to W and (P_1, N_1) applies to A. If this is a computation witnessing that $\Gamma(W \oplus A) \downarrow$, its *use on W* (*W-use*) is $max(P_0 \cup N_0)$, and its *use on A* (*A-use*) is $max(P_1 \cup N_1)$.

We will be enumerating various partial recursive functionals Θ during the course of the construction. To "enumerate a computation $\Theta(W) \downarrow$" (at stage σ) is to enumerate into Θ a neighborhood condition that applies to $W[\sigma]$. In the context of the construction, we will have some computation $(P, N) = ((P_0, P_1), (N_0, N_1))$ witnessing $\Gamma(W \oplus A) \downarrow [\sigma]$. We will "enumerate a computation $\Theta(W) \downarrow$ with the same use on W as Γ", by which we will mean that we enumerate the neighborhood condition (P_0, N_0) into Θ.

3.3 Approximations to Δ_2 functions

The Δ_2 functions we are interested in are the (characteristic functions of) jumps of low r.e. sets W, that is, functions that identify the convergence or divergence of every partial recursive functional $\Gamma_i(W)$. For this reason, we define a possible recursive approximation to a Δ_2 function to be a total recursive function of two variables, Δ, that takes values from the set $\{\uparrow, \downarrow\}$. If we are considering Δ as a possible approximation to the jump of W, we say that Δ *predicts* $\Gamma_i(W)$ *converges at stage* σ just in case $\Delta(i, \sigma) = \downarrow$, and similarly for divergence. Δ approximates the jump of W just in case, for all i, Δ correctly predicts the convergence (or divergence) of $\Gamma_i(W)$:

$$\lim_{\sigma \to \infty} \Delta(i, \sigma) = \begin{cases} \downarrow & \text{if } \Gamma_i(W) \downarrow, \\ \\ \uparrow & \text{if } \Gamma_i(W) \uparrow. \end{cases}$$

There is a uniform enumeration of possible recursive approximations to Δ_2 functions (all of which are total) that includes correct approximations to the jump of every low r.e. set. (Begin with a uniform enumeration of partial recursive functions (f_e). To compute $\Delta_e(i,\sigma)$, compute in order the sequence $f_e(i,0)$, $f_e(i,1)$, $\ldots$, $f_e(i,\sigma)$, for σ many steps. If the computation of $f_e(i,0)$ has not yet converged after σ many steps, set $\Delta_e(i,\sigma) =\uparrow$; otherwise, set $\Delta_e(i,\sigma) = f_e(i,\tau)$ for the largest τ for which $f_e(i,\tau)$ did converge. It is not hard to see that if f_e approximates the jump of W, so does Δ_e; so if W is low, its jump is approximated by some Δ_e.)

During our construction we will be building potential approximations Φ to Δ_2 functions (specifically, the jumps of r.e. sets $W \oplus A$.) We do not explicitly state the value of every $\Phi(i,\sigma)$, instead, we say at each stage σ for which i this value changes: We start by setting $\Phi(i,0) =\uparrow$ for every i. At step σ of the construction we may switch Φ to predict $\Gamma_i(W \oplus A)$ converges (set $\Phi(i,\sigma+1) =\downarrow$) or switch Φ to predict $\Gamma_i(W \oplus A)$ diverges (set $\Phi(i,\sigma+1) =\uparrow$.) If we do not switch Φ then $\Phi(i,\sigma+1) = \Phi(i,\sigma)$. Instead of $\Phi(i,\sigma)$, we may write $\Phi(i)[\sigma]$, the prediction Φ makes at stage σ about the convergence of $\Gamma_i(W \oplus A)$.

3.4 The recursion theorem

The construction we are about to give in sections 4 and 5 involves a standard use of the recursion theorem.

During the construction, we enumerate various r.e. sets (partial recursive functionals) Θ_i^β. In fact, since when the requirement Q^β enumerating Θ_i^β is deactivated Θ_i^β is redefined to equal $\emptyset$, we are actually enumerating r.e. sets

$$\Theta_i^{\beta,s} = \{(P,N) \mid (P,N) \text{ is enumerated into } \Theta_i^\beta \text{at or after stage } s\}.$$

If Q^β is activated at stage s and never again deactivated, then at the end of the construction we have $\Theta_i^\beta = \Theta_i^{\beta,s}$.

At stage t we take action depending on what Δ_e predicts about the convergence of $\Theta_i^\beta(W_e)$. More correctly, if s was the last stage before t at which Q^β was activated, we take action depending on what Δ_e predicts about the convergence of $\Theta_i^{\beta,s}(W_e)$.

Since we are enumerating all these sets as part of a single recursive construction, given an index c for the construction we can recursively compute indices $f(c,i,\beta,s)$ for the sets $\Theta_i^{\beta,s}$ enumerated by the construction. The predictions of Δ_e about the convergence of $\Theta_i^{\beta,s}(W_e)$ on which the construction depends are given by the values

$$\Delta_e(f(c,i,\beta,s),t).$$

It seems as though we need to know the index c in order to determine the construction, but of course the recursion theorem solves the problem:

Given an index c for a construction, we can perform this construction using the values $\Delta_e(f(c,i,\beta,s),t)$ as the predictions of Δ_e at stage t about the convergence of $\Theta_i^{\beta(W_e),s}$. (I.e., we are pretending that c is the index of the construction we are performing.) This gives us a new construction, and its index can be recursively computed from c as $g(c)$. By the recursion theorem there is an index c such that c and $g(c)$ are indices for the same construction. This is the construction we want.

4 The Tree Construction

What follows in this section is a standard construction using a tree of strategies. Almost the only feature specific to this construction is the choice of outcomes for individual strategies, and even this is fairly standard given the nature of the strategies.

The tree of our construction will be the tree of finite sequences of natural numbers. The tree is ordered by end extension, as usual. To each node α of the tree we assign a strategy Q^α of the construction as follows:

$$|\alpha| = 2n \;\Rightarrow\; Q^\alpha = P_n^\alpha,$$

$$|\alpha| = 2n+1 \text{ and } n \text{ codes } (e,0) \;\Rightarrow\; Q^\alpha = R_e^\alpha,$$

$$|\alpha| = 2n+1 \text{ and } n \text{ codes } (e,i+1) \;\Rightarrow\; Q^\alpha = S_{e,i}^\alpha.$$

The intention is that the strategy Q^α is working on the requirement Q. Strategies assigned to nodes on the same level of the tree are working on the same requirement. The immediate successors of α correspond to the different possible outcomes of the strategy Q^α.

If $Q^\alpha = R_e^\alpha$, $Q^\beta = S_{e,i}^\beta$, $\alpha^\frown n \subset \beta$ and $n \neq 0$, then $S_{e,i}^\beta$ is *inert* and does nothing. This reflects the fact (which we will see) that if R_e^α has a non-zero outcome, corresponding to a non-zero permanent restraint, the requirement R_e is guaranteed to be satisfied by the action of the strategy R_e^α, and so there is no need for $S_{e,i}^\beta$ to take any action. (Recall that R_e^α is carrying out steps (1) and (2) of the $S_{e,j}$ strategies outlined above. Holding a permanent restraint corresponds to waiting forever at step (2). When this happens, the strategy is waiting forever for Δ_e to produce a correct prediction that some $\Theta_j(W_e)\downarrow$;

therefore, this means that Δ_e does not correctly predict the convergence of $\Theta_j(W_e)$, and requirement R_e is satisfied because Δ_e does not approximate the jump of W_e.)

If we let B be an infinite branch through the tree, we get a sequence of strategies $\left\langle Q^{B(i)} \,|\, i < \omega \right\rangle$ with the property that every requirement P_e, R_e and $S_{e,i}$ has an associated strategy in the sequence, and the strategy associated with a requirement R_e comes before the strategies associated with the subrequirements $S_{e,i}$.

We also have other orderings on our tree:

$$\alpha <_{left} \beta \;\Leftrightarrow\; \exists i\;[\;(\forall j < i)\; \alpha(j) = \beta(j) \text{ and } \alpha(i) < \beta(i)\;];$$

$$\alpha <_{lex} \beta \;\Leftrightarrow\; \alpha \subset \beta \text{ or } \alpha <_{left} \beta.$$

The strategy Q^α has *higher priority* than the strategy Q^β just in case $\alpha <_{lex} \beta$.

For convenience, we will index the stages of the construction by ordered pairs (s, t) with $t \leq s$. At stage (s, t) we will take action according to some strategy at level t of the tree; the choice of strategy will be determined by the actions taken at earlier stages. The successor of the stage $\sigma = (s, t)$ is defined by

$$\sigma^+ = \begin{cases} (s, t+1) & \text{if } t < s, \\ (s+1, 0) & \text{if } t = s. \end{cases}$$

In this section we will describe how we move through the tree. In section 5 we will detail the action to be taken for a given strategy at a given stage. This detail will define the *restraint* being held at the beginning of stage σ by strategy Q^α, which appears in the following definition as $\mathit{restraint}(Q^\alpha)[\sigma]$.

The *stage σ outcome* (the outcome at the beginning of stage σ) of strategy Q^α is defined to be

$$\mathit{outcome}(P_e^\alpha)[\sigma] = 1 \;\Leftrightarrow\; W_e \cap A = \emptyset[\sigma],$$

$$\mathit{outcome}(P_e^\alpha)[\sigma] = 0 \;\Leftrightarrow\; W_e \cap A \neq \emptyset[\sigma],$$

$$\mathit{outcome}(R_e^\alpha)[\sigma] = r \;\Leftrightarrow\; \mathit{restraint}(R_e^\alpha)[\sigma] = r,$$

$$\mathit{outcome}(S_{e,i}^\alpha)[\sigma] = r \;\Leftrightarrow\; \mathit{restraint}(S_{e,i}^\alpha)[\sigma] = r.$$

At stage $\sigma = (s, t)$ we *visit* the node of the tree $\mathit{node}(\sigma)$, which is defined as follows,

$$\sigma = (s, 0) \;\Rightarrow\; \mathit{node}(\sigma) = \langle\rangle\,,$$

$$\sigma = (s, t+1) = \rho^+ \Rightarrow node(\sigma) = node(\rho)^\frown outcome(Q^{node(\rho)})[\sigma],$$

we *cancel*, or *deactivate*, strategies Q^β for $node(\sigma) <_{left} \beta$, and we take action for the strategy $Q^{node(\sigma)}$. At stage σ the *current* or *stage σ path of the construction* is the finite branch of the tree terminating with $node(\sigma)$.

A note here: We take $[\sigma]$ to refer to the situation at the *beginning* of stage σ. In order to conform to this standard, the "stage σ outcome" of strategy Q^α, $outcome(Q^\alpha)[\sigma]$, is actually the outcome of Q^α at the beginning of stage σ. The outcome of action taken *during* stage σ is $outcome(Q^\alpha)[\sigma^+]$. Therefore at stage σ we visit the node $\alpha = node(\sigma)$, we take action for the strategy Q^α, and either we return to the root of the tree, $\langle\rangle$ (in case $\sigma = (s, s)$), or we go on to node $\alpha^\frown outcome(Q^\alpha)[\sigma^+]$.

We define the "true path of the construction" to be the leftmost path visited infinitely often. I.e., α is on the true path of the construction iff there are infinitely many stages σ for which $node(\sigma) = \alpha$, but only finitely many stages for which $node(\sigma) <_{left} \alpha$. Once the construction is defined, we will prove inductively that the true path of the construction is an infinite branch through the tree and the actions of the strategies along the true path guarantee the satisfaction of all the requirements.

In order to accomplish this last, we arrange the construction so that a strategy at any node respects the restraint imposed by strategies of higher priority, including those to its left in the tree. A strategy is injured and drops all its restraint (it is canceled or deactivated) every time the current path of the construction goes to its left. Therefore, a strategy along the true path will only have to respect finitely many higher priority strategies (those above it, and those to the left that are actually visited at some stage of the construction) and its restraint will not be injured by lower priority strategies (including those to the right.)

As a simple example, section 2.2 discussed two versions of one of the S requirements, $S_{d,j}$ and $\hat{S}_{d,j}$; $S_{d,j}$ acts when R_d is imposing no restraint and holds its restraint when R_d is imposing restraint, and $\hat{S}_{d,j}$ acts when R_d is imposing restraint and is canceled when R_d is not imposing restraint. We can think of $\hat{S}_{d,j}$ as being to the right of $S_{d,j}$ on the tree. When R_d is imposing no restraint, the path of the construction goes through $S_{d,j}$ (which gets to act) and $\hat{S}_{d,j}$ (to the right) is canceled. When R_d is imposing restraint, the path of the construction goes through $\hat{S}_{d,j}$ (which gets to act) and $S_{d,j}$ (to the left) is respected.

5 Action of Individual Strategies

Now we complete the construction by defining the action taken at each stage. Claims 1 through 8 at the end of this section isolate some easily checked properties of the construction. Section 6 will give the proof that the construction does produce a set having almost deep degree.

5.1 Formalities and conventions:

Suppose we are at stage σ. All sets and computations mentioned in the description of stage σ of the construction are taken to be as approximated at stage σ; for ease of reading, we eliminate the notation $[\sigma]$. For example, we will say (P, N) applies to W_e to mean (P, N) applies to $W_e[\sigma]$.

Various parameters are determined during each stage of the construction. At stage σ the construction determines which parameters will change their values; all others have the same value at the end of stage σ as at the beginning. We use the notation $[\sigma]$ to indicate the value of a parameter at the beginning of stage σ, but in the description of stage σ we drop the notation $[\sigma]$. For example, we will say R_e^α is in state (1) for i to mean $i\text{-}state(R_e^\alpha)[\sigma] = 1$.

Initial values of the parameters (at the beginning of stage 0) are:

For requirements R_e^α and indices i:

$i\text{-}state(R_e^\alpha) = inactive$,

$i\text{-}restraint(R_e^\alpha) = 0$,

R_e^α is not preserving any computation for i,

$\Phi_e^\alpha(i) =\uparrow$ (Φ_e^α predicts every computation diverges.)

We define $restraint(R_e^\alpha)$ to be the maximum of $i\text{-}restraint(R_e^\alpha)$.

For requirements $S_{e,i}^\beta$, $\alpha \subset \beta$:

$state(S_{e,i}^\beta) = inactive$,

$restraint(S_{e,i}^\beta) = 0$,

$S_{e,i}^\beta$ is not preserving any computation,

$S_{e,i}^\beta$ is not connected to R_e^α,

$\Theta_i^\beta = \emptyset$ (there are no computations in Θ_i^β.)

We use some less formal but intuitive terminology. For example, for a requirement Q^α to "impose restraint r" means to set $restraint(Q^\alpha) = r$, to "drop restraint" means to set $restraint(Q^\alpha) = 0$, to "be holding restraint r"

means that the value of $restraint(Q^\alpha)$ is r, and to "transfer restraint r to Q^β" means to set $restraint(Q^\beta) = r$.

The stage σ outcome of Q^α was defined in section 4. Except for the P_e^α, which have only two possible outcomes (determined by whether $W_e \cap A = \emptyset$), the outcome of a requirement is the restraint it is holding.

5.2 Action, activation and deactivation

At stage σ we deactivate Q^α for all α such that $node(\sigma) <_{left} \alpha$, and we take action for $Q^{node(\sigma)}$.

5.2.1 Deactivation and activation for $Q^\alpha = P_e^\alpha$:

Nothing needs to be done to activate or deactivate P_e^α.

5.2.2 Action for $Q^\alpha = P_e^\alpha$:

The restraint Q^α must respect during stage σ is the maximum R of the restraints

$$restraint(Q^\beta)[\sigma] \qquad \text{for } \beta <_{lex} \alpha.$$

If $W_e \cap A = \emptyset[\sigma]$ and there is some $x \in W_e[\sigma]$ such that $x > 2e$ and $x > R$, then enumerate one such x into A at stage σ. Otherwise, we take no action at stage σ.

5.2.3 Deactivation and activation for $Q^\alpha = R_e^\alpha$:

To deactivate R_e^α, set Φ_e^α to predict $\Gamma_i(W_e \oplus A) \uparrow$ for every i. Put R_e^α into state *inactive* for every i. Disconnect every $S_{e,i}^\beta$ from R_e^α. Set every i-restraint of R_e^α equal to 0.

To activate R_e^α, put R_e^α into state (0) for every i.

5.2.4 Action for $Q^\alpha = R_e^\alpha$:

Because R_e^α acts to preserve computations, and there may be several computations in Γ_i applying to $W_e \oplus A$ at a given stage, we order the computations so we can be sure none are overlooked. This ordering is defined in the context of given enumerations of Γ_i and $W_e \oplus A$ and its domain is all computations in Γ_i that apply to $W_e \oplus A[\sigma]$ at any stage σ:

Suppose $(P, N) \in \Gamma_i[\sigma]$ applies to $W_e \oplus A[\sigma]$, and σ is the least such stage. Suppose $(P', N') \in \Gamma_i[\sigma']$ applies to $W_e \oplus A[\sigma']$, and σ' is the least such stage. Then (P, N) is less than (P', N') in case:

(i.) $\sigma < \sigma'$,

(ii.) $\sigma = \sigma'$ and $use(P, N) < use(P', N')$,

(iii.) $\sigma = \sigma'$, $use(P, N) = use(P', N')$ and $code(P, N) < code(P', N')$.

(Clause (i.) is the key clause.)

Now we describe the action for R_e^α:

If R_e^α is inactive, activate it.

For all $i \leq s$ (where our current stage is $\sigma = (s, t)$):

If R_e^α is in state (0) for i:

> If no $S_{e,i}^\beta$ is connected to R_e^α, stay in state (0) for i, and end action for i. (The role of R_e^α is to carry out steps (1) and (2) of the strategies for subrequirements $S_{e,i}^\beta$. If no subrequirement is connected to R_e^α, that means no subrequirement needs to have one of these steps carried out at this stage.)
>
> If some $S_{e,i}^\beta$ is connected to R_e^α, then go to state (1) for i. (We will see from the rest of the construction that in this case Φ_e^α currently predicts that $\Gamma_i(W_e \oplus A) \uparrow$. See Claim 2.)

If R_e^α is in state (1) for i (R_e^α is carrying out step (1) of the strategies for connected subrequirements):

> If $\Gamma_i(W_e \oplus A) \uparrow$, end action for i.
>
> If $\Gamma_i(W_e \oplus A) \downarrow$, choose the least computation $(P, N) \in \Gamma_i$ that applies to $(W_e \oplus A)$. Let τ be the last stage before σ at which the construction visited α. (There must be such a stage, since R_e^α must have been activated at an earlier stage in order to be in state (1) at this stage.) If (P, N) did not apply to $W_e \oplus A$ at stage τ, do nothing. (For technical reasons we want to act only on computations that have persisted for at least two visits to R_e^α; it will allow us to insure that computations acted on at "true stages" are permanent computations.) Otherwise, for all $S_{e,i}^\beta$ connected to R_e^α, enumerate a computation $\Theta_i^\beta(W_e) \downarrow$ with the same use on W_e as $\Gamma_i(W_e \oplus A)$. Preserve the computation (P, N) by imposing i-restraint equal to A-$use(\Gamma_i(W_e \oplus A)) + 1$. Go to state (2) for i.

If R_e^α is in state (2) for i: R_e^α is preserving some computation (P, N) witnessing $\Gamma_i(W_e \oplus A)\downarrow$ by holding i-restraint equal to r.

> If that computation no longer applies to $W_e \oplus A$ (this will be due to a W_e change, so also all $\Theta_i^\beta(W_e)\uparrow$ for $S_{e,i}^\beta$ connected to R_e^α; see Claim 6), drop the i-restraint on A and go to state (1) for i.
>
> If (P, N) still applies to $W_e \oplus A$, and for some $S_{e,i}^\beta$ connected to R_e^α, Δ_e does not predict $\Theta_i^\beta(W_e)\downarrow$, stay in state (2) for i and end action for i. (We will see from the rest of the construction, no new $S_{e,i}^\beta$ can connect to R_e while R_e remains in state (2) and holding restraint on A, so we are waiting for finitely many Δ_e predictions; see Claim 2.)
>
> If (P, N) still applies to $W_e \oplus A$, and for all $S_{e,i}^\beta$ connected to R_e^α, Δ_e predicts $\Theta_i^\beta(W_e)\downarrow$, then: For all such $S_{e,i}^\beta$, disconnect $S_{e,i}^\beta$ from R_e, put $S_{e,i}^\beta$ into state (3) and transfer to $S_{e,i}^\beta$ the restraint r preserving the computation (P, N). Drop the i-restraint at R_e^α. Switch Φ_e^α to predict $\Gamma_i(W_e \oplus A)\downarrow$. Go to state (0) for i.

(This means strategies to the left of the path of the construction can increase their restraint even though they are not visited, as R_e on the path of the construction transfers A restraint downwards. However, for a given α on the true path, nodes to the left of α increase their A restraint in this way only finitely often; see Claim 1.)

The restraint held by R_e^α at the end of stage σ, $\mathit{restraint}(R_e^\alpha)[\sigma^+]$, is the maximum of the i-restraints it is holding for all $i \leq s$.

5.2.5 Deactivation and activation for $Q^\alpha = S_{e,i}^\alpha$:

By construction of the tree, there is a unique $\gamma \subset \alpha$ for which $Q^\gamma = R_e^\gamma$.

To deactivate $S_{e,i}^\alpha$, disconnect $S_{e,i}^\alpha$ from R_e^γ, set its restraint equal to 0, put $S_{e,i}^\alpha$ into state *inactive*, and set $\Theta_i^\alpha = \emptyset$ (in other words, remove any computations previously enumerated into Θ_i^α.)

To activate $S_{e,i}^\alpha$:

If $\gamma^\frown n \subset \alpha$ for $n \neq 0$, then $S_{e,i}^\alpha$ is inert; take no action. Otherwise:

If some other $S_{e,i}^\beta$ is in state (3) and is holding restraint equal to r to preserve a computation (P, N) witnessing $\Gamma_i(W_e \oplus A)\downarrow$ that still applies to

$(W_e \oplus A)$, then go into state (3), preserve the same computation (P, N) by imposing the same restraint r, and enumerate a computation $\Theta_i^\alpha(W_e) \downarrow$ with the same use on W_e. (Necessarily, in this case, $\beta <_{left} \alpha$. We will see that there is only one possible choice for the computation (P, N); see Claim 3.)

Otherwise, set Φ_e^α to predict $\Gamma_i(W_e \oplus A) \uparrow$, connect to R_e^γ, go into state *connected* and set the restraint of $S_{e,i}^\alpha$ equal to 0.

5.2.6 Action for $Q^\alpha = S_{e,i}^\alpha$:

There is a unique $\gamma \subset \alpha$ for which $Q^\gamma = R_e^\gamma$.

If $\gamma^\frown n \subset \alpha$ for $n \neq 0$, then $S_{e,i}^\alpha$ is inert; take no action. If $\gamma^\frown 0 \subset \alpha$, then:

> If $S_{e,i}^\alpha$ is inactive, activate it.
>
> If the strategy is connected to R_e^γ, do nothing.
>
> If the strategy is in state (3), then it is holding some restraint on A to preserve a computation (P, N) witnessing $\Gamma_i(W_e \oplus A) \downarrow$. Also, there is a computation in Θ_i^α with the same use on W_e. If (P, N) no longer applies to $W_e \oplus A$ (this will be due to a W_e change, so $\Theta_i^\beta(W_e) \uparrow$; see Claim 6), drop the restraint on A and go to state (4).
>
> If the strategy is in state (4), and Δ_e does not predict that $\Theta_i^\alpha(W_e) \uparrow$, do nothing. If Δ_e does predict that $\Theta_i^\alpha(W_e) \uparrow$, then change Φ_e^γ to predict that $\Gamma_i(W_e \oplus A) \uparrow$, go into state *connected* and connect to R_e^γ.

5.3 Properties of the construction

This completes the construction. Before going on, we state a few easily checked properties.

Claim 1 *If the path of the construction never goes to the left of α past stage σ, there is a stage $\tau > \sigma$ past which no requirement to the left of α ever changes its restraint.*

Once the path of the construction never goes to the left of α, no requirement to the left of α will ever act again. The only way in which a requirement can increase its restraint without acting is when an $S_{e,i}^\beta$ gets restraint transferred from an R_e^γ to which it was connected. But this can happen only finitely often, because only finitely many requirements to the left of α have been activated, and once one of them gains transferred restraint it is disconnected and will never be connected again (since it will never act again.)

Claim 2 *(i.) Whenever R_e^α is in state (2) for i, it is holding positive restraint.*

(ii.) Whenever R_e^α is in state (2) for i, no $S_{e,j}^\beta$ connects to R_e^α.

(iii.) Whenever R_e^α is in states (1) or (2) for i, Φ_e^α predicts $\Gamma_i(W_e \oplus A)$ diverges.

(i.) holds by the action of R_e^α.

(ii.) holds by (i) and the fact that all $S_{e,j}^\beta$ below non-zero outcomes for R_e^α are inert, so while R_e^α is holding non-zero restraint no $S_{e,j}^\beta$ can connect.

(iii.) holds because Φ_e^α is only set to predict $\Gamma_i(W_e \oplus A)$ converges when R_e^α leaves state (2) for state (0) and disconnects all $S_{e,j}^\beta$, and before R_e^α enters state (1) again some $S_{e,i}^\beta$ must connect to R_e^α and set Φ_e^α to predict $\Gamma_i(W_e \oplus A)$ diverges (which Φ_e^α will continue to predict until R_e^α leaves state (2) for state (0) again.)

Claim 3 *(i.) If (P, N) is the least computation in Γ_i that applies to $W_e \oplus A$ at stage σ, and (P, N) still applies to $W_e \oplus A$ at stage $\tau > \sigma$, then (P, N) is the least computation in Γ_i that applies to $W_e \oplus A$ at stage τ.*

(ii.) If any requirement is holding restraint to preserve a computation (P, N) witnessing $\Gamma_i(W_e \oplus A) \downarrow$ at stage σ, and (P, N) still applies to $W_e \oplus A$ at stage $\tau > \sigma$, then (P, N) is the least computation in Γ_i that applies to $W_e \oplus A$ at stage τ. (In particular, at any stage, no two requirements can be preserving different computations witnessing $\Gamma_i(W_e \oplus A) \downarrow$, both of which still apply to $W_e \oplus A$.)

(iii.) If $S_{e,i}^\beta$ is holding restraint at stage τ to preserve a computation (P, N) witnessing $\Gamma_i(W_e \oplus A) \downarrow$, and (P, N) still applies to $W_e \oplus A$ at stage τ, then no requirement switches Φ_e to predict $\Gamma_i(W_e \oplus A) \uparrow$ at stage τ.

(i.) holds by definition of the ordering on computations in Γ_i.

(ii.) holds by construction (when any computation is first preserved it is the least one that applies) and (i).

(iii.) holds because only some other $S_{e,i}^\gamma$ on the same level could switch Φ_e to predict $\Gamma_i(W_e \oplus A) \uparrow$ at stage τ. This γ couldn't be to the left of β, otherwise $S^{\beta}_{e,i}$ would have been canceled. If γ is to the right of β, $S^{\gamma}_{e,i}$ was activated after the construction last visited β, and by the rules for activating $S^{\gamma}_{e,i}$, $S^{\gamma}_{e,i}$ went into state (3) preserving the same computation (P, N). (Note that by (ii) the computation (P, N) was the only possible choice.) But then, since (P, N) still applies at stage τ, $S^{\gamma}_{e,i}$ remains in state (3) and does not switch the prediction of Φ^{α}_{e}.

Claim 4 *If a requirement Q^{α} is holding restraint r during stage σ, and a number less than r is enumerated into A during stage σ, then that requirement is canceled at stage σ^{+}.*

The requirement P^{β}_{e} enumerating this number into A cannot be to the left of Q^{α}, because in that case Q^{α} would already have been deactivated. It cannot be below or to the right of Q^{α}, because then it would respect the restraint of Q^{α}. Therefore it must be above Q^{α}; $\beta \subset \alpha$. The fact that it acts at stage σ means that $W_e \cap A = \emptyset[\sigma]$, and $W_e \cap A \neq \emptyset[\sigma^{+}]$. But this means that Q^{α} must be below $\beta^\frown 1$ (because it was activated before stage σ), and the outcome of P^{β}_{e} changes at stage σ, so $node(\sigma^{+}) = \beta^\frown 0 <_{left} \alpha$ and Q^{α} is deactivated at stage σ^{+}.

Claim 5 *Suppose Q^{α} is preserving (P, N) at the end of stage σ and (P, N) applies to $W_e \oplus A[\sigma]$ but not to $W_e \oplus A[\sigma^{+}]$. Then either $W_e[\sigma^{+}]$ differs from $W_e[\sigma]$ on the W_e-use of (P, N), or Q^{α} is deactivated at stage σ^{+}.*

This follows from Claim 4, since if Q^{α} is preserving (P, N) it is holding restraint greater than the A-use of (P, N), so if Q^{α} is not deactivated at stage σ^{+} it must have been a change in W_e that caused (P, N) to apply no longer.

Claim 6 *Suppose that at stage σ either $S^{\beta}_{e,i}$ is in state (4) or $S^{\beta}_{e,i}$ is connected to R^{α}_{e} and R^{α}_{e} is in state (1) for i. Then at stage σ, $\Theta^{\beta}_{i}(W_e)\uparrow$.*

Whenever a computation $\Theta^{\beta}_{i}(W_e)\downarrow$ is enumerated, we have $\Gamma_i(W_e \oplus A)\downarrow$ with the same W_e-use, and A is restrained by R^{α}_{e} in state (2) or $S^{\beta}_{e.i}$ in state (3). If R^{α}_{e} enters state (1) or $S^{\beta}_{e.i}$ enters state (4), we must have $\Gamma_i(W_e \oplus A)\uparrow$. By Claim 5, (since R^{α}_{e} or $S^{\beta}_{e,i}$ was not immediately canceled) the restraint on A was respected, so $\Gamma_i(W_e \oplus A)$ diverges because W_e changed below its W_e-use. This is also the use of $\Theta^{\beta}_{i}(W_e)$, so $\Theta^{\beta}_{i}(W_e)\uparrow$.

Claim 7 *If $S^{\beta}_{e,i}$ is never deactivated after some stage, then the final Θ^{β}_{i} is an r.e. set.*

If $S^{\beta}_{e,i}$ is last deactivated at stage σ, the final Θ^{β}_{i} is just all computations enumerated into Θ^{β}_{i} after stage σ.

Claim 8 *The set A enumerated by this construction is co-infinite.*

For each e, at most one number is enumerated into A by any version of P_e (since this will happen only at a stage when $W_e \cap A = \emptyset$), and that number is bigger than $2e$. No other numbers are enumerated into A.

6 The Rest of the Proof

Lemma 1 *If α is on the true path of the construction and $Q^{\alpha} = P^{\alpha}_{e}$, then α has an immediate successor on the true path of the construction, and P_e is satisfied.*

> P_e: If W_e is infinite then $W_e \cap A$ is non-empty.

If $W_e \cap A \neq \emptyset$, then for all stages σ large enough so that $W_e \cap A \neq \emptyset[\sigma]$, $outcome(Q^{\alpha})[\sigma] = 0$; $\alpha\frown 0$ is on the true path of the construction. In this case P_e is satisfied because $W_e \cap A \neq \emptyset$.

If $W_e \cap A = \emptyset$, then for all stages σ, $outcome(Q^{\alpha})[\sigma] = 1$; $\alpha\frown 1$ is on the true path of the construction. In this case, P_e is satisfied because W_e is finite:

Let σ be a stage past which the path of the construction is never to the left of α, and no Q^{β} for $\beta <_{left} \alpha$ ever changes its restraint. (There is such a stage by Claim 1.) Let R be the maximum of $restraint(Q^{\beta})[\sigma]$ for $\beta <_{left} \alpha$ and $\alpha(i)$ for $i < length(\alpha)$. (Recall that $\alpha(i)$ is the outcome of the strategy on level i when the path of the construction visits α, and in the relevant cases the outcome is the restraint that strategy is holding.) At every stage $\tau > \sigma$ for which $node(\tau) = \alpha$, the restraint P^{α}_{e} must respect at stage τ is R. If there were an $x > max(R, 2e)$ in W_e, then there would be one at a stage $\tau > \sigma$ for which $node(\tau) = \alpha$, and at that stage P^{α}_{e} would enumerate such an x into A. By assumption, $W_e \cap A = \emptyset$, so this does not happen. Therefore, W_e has no elements greater than $max(R, 2e)$.

Lemma 2 *If α is on the true path of the construction and $Q^\alpha = R^\alpha_e$, then:*

If $\alpha^\frown 0$ is on the true path of the construction, for each i one of:

(i.) From some stage on, R^α_e remains in state (0) for i. In this case, past this stage no $S^\beta_{e,i}$ is ever connected to R^α_e.

(ii.) From some stage on, R^α_e remains in states (1) and/or (2) for i. In this case, Φ^α_e correctly predicts the divergence of $\Gamma_i(W_e \oplus A)$. Therefore every $S^\beta_{e,i}$ for $\alpha \subset \beta$ is satisfied.

(iii.) There are infinitely many stages for which R^α_e is in state (0) for i, infinitely many for which it is in state (1), and infinitely many for which it is in state (2). In this case, every $S^\beta_{e,i}$ is disconnected from R^α_e during infinitely many stages.

If $\alpha^\frown 0$ is not on the true path of the construction, then $\alpha^\frown n$ is on the true path of the construction for some $n > 0$. In this case R^α_e is satisfied because Δ_e fails to correctly predict the convergence of some $\Theta^\beta_i(W_e)$.

In particular, α has an immediate successor on the true path of the construction.

> R^α_e: If W_e is low and Δ_e approximates its jump as a Δ_2 function, then $W_e \oplus A$ is low and its jump is approximated as a Δ_2 function by Φ^α_e.

Because α is on the true path of the construction, there is some stage ρ past which R^α_e remains active. First we suppose that for infinitely many stages $\tau > \rho$, $restraint(R^\alpha_e) = 0[\tau]$. Then $\alpha^\frown 0$ is on the true path of the construction. In this case, for a given i, there are several possible behaviors for R^α_e:

(i.) From some stage σ on, R^α_e remains in state (0) for i. By construction, whenever an $S^\beta_{e,i}$ connects to R^α_e (past stage ρ), R^α_e moves into state (1) the next time the construction visits R^α_e. Since R^α_e remains in state (0) from some stage on, this means that past stage σ no $S^\beta_{e,i}$ is ever connected to R^α_e.

(ii.) From some stage σ on, R^α_e remains in states (1) and/or (2). Infinitely often it is in state (1). (That last follows from the assumption that for infinitely many stages $\tau \supset \sigma$, $restraint(R^\alpha_e) = 0[\tau]$, because when R^α_e is in state (2) its restraint is positive.) In this case, from this stage on, Φ^α_e predicts that $\Gamma_i(W_e \oplus A) \uparrow$. (This is because whenever R^α_e is in states (1) or (2), Φ^α_e predicts $\Gamma_i(W_e \oplus A) \uparrow$, by Claim 2.) Furthermore, this prediction is correct:

Suppose not. Let (P, N) be the least computation in Γ_i that applies to $W_e \oplus A$. Choose a stage $\tau > \sigma$ such that (P, N) is the least computation in

$\Gamma_i[\tau]$ that applies to $W_e \oplus A[\tau]$, R_e^α is in state (1) at the beginning of stage τ, and $node(\tau) = \alpha$. When we take action for R_e^α at stage τ, R_e^α will go into state (2) and preserve the computation (P, N) by imposing restraint. Since we assume (P, N) actually applies to $W_e \oplus A$, $W_e \oplus A$ will never change on its use, and R_e^α will never move back into state (1), contradicting our assumptions about this case.

(iii.) The states of R_e^α cycle: There are infinitely many stages for which R_e^α is in state (0) for i, infinitely many for which it is in state (1), and infinitely many for which it is in state (2). In this case, every $S_{e,i}^\beta$ is disconnected from R_e^α during infinitely many stages, because every time R_e^α goes from state (2) to state (0), every $S_{e,i}^\beta$ is disconnected from R_e^α.

This proves the first part of the lemma.

It remains to analyze the case in which, past some stage, $restraint(R_e^\alpha) > 0$. Let σ be a stage past which R_e^α remains active and its restraint remains non-zero. At every stage past σ, then, R_e^α is in state (2) for at least one i.

At stage σ, only finitely many $S_{e,i}^\beta$ are connected to R_e^α. Past stage σ, no $S_{e,i}^\beta$ is ever connected to R_e^α (by Claim 2), but some may be disconnected as R_e^α moves into state (0) for that particular i. Choose σ large enough so that any $S_{e,i}^\beta$ that is going to be disconnected from R_e^α has already been disconnected by stage σ.

There are finitely many i for which some $S_{e,i}^\beta$ is connected to R_e^α at (and past) stage σ; for these i, R_e^α remains in states (1) and/or (2) from stage σ on. It may be that for some i there is a stage past which R_e^α remains in state (2); let I be the set of all such i, and choose σ large enough so that past stage σ, R_e^α remains in state (2) for all $i \in I$. (We will see shortly that $I \neq \emptyset$.)

Past stage σ, for each $i \in I$, R_e^α remains in state (2) for i, holding restraint r_i. Let R be the maximum of the r_i for $i \in I$. At every stage past σ, the restraint of R_e^α is at least R. We will show that at infinitely many stages it equals R, and therefore $\alpha^\frown R$ is on the true path of the construction. This will also show that $R > 0$ (since we are assuming that $\alpha^\frown 0$ is not on the true path), and so $I \neq \emptyset$.

Let $\{\sigma_n | n \in \omega\}$ be the set of stages past σ for which $node(\sigma_n) = \alpha$. Define σ_n for $n > 0$ to be a *true stage* in case

$$\exists x \in W_e[\sigma_n] - W_e[\sigma_{n-1}] \ \Big(W_e[\sigma_n] \cap \{0, \ldots, x\} = W_e \cap \{0, \ldots, , x\}\Big).$$

There are infinitely many true stages. (Let ρ be any stage, and x the least element of $W_e - W_e[\rho]$. Let σ_n be least such that $x \in W_e[\sigma_n] - W_e[\sigma_{n-1}]$. Then σ_n is a true stage greater than ρ.) If σ_n is a true stage, then for $i \notin I$ we show that

$$i\text{-}restraint(R_e^\alpha)[\sigma_n^+] = 0:$$

Recall, this is the restraint being held for i at the end of the stage σ_n. Let x be the least element of $W_e[\sigma_n] - W_e[\sigma_{n-1}]$. If R_e^α is holding i-restraint to preserve some (P, N) witnessing $\Gamma_i(W_e \oplus A)\downarrow$ at the beginning of stage σ_n, this computation applied to $W_e \oplus A$ at some earlier stage σ_m when this restraint was imposed, and if R_e^α does not drop that restraint during stage σ_n, that is because this computation still applies at the beginning of stage σ_n. If R_e^α imposes i-restraint to preserve some (P, N) witnessing $\Gamma_i(W_e \oplus A)\downarrow$ during stage σ_n, that is because this computation applied to $W_e \oplus A$ at stage σ_{n-1} and still applies at stage σ_n (since we act only on computations that have persisted through two visits to R_e^α.) In any case, if R_e^α is holding i-restraint at the end of stage σ_n, it is preserving a computation (P, N) that applied to $W_e \oplus A$ at stage σ_{n-1} and still applies at stage σ_n. This means x must be greater than the W_e-use of that computation, since x entered W_e between these two stages. But then, since W_e does not change below x after stage σ_n, this computation will always apply and so R_e^α will never switch back to state (1) for i, contradicting our assumption that $i \notin I$. Therefore,

$$\sigma_n \text{ is a true stage } \& \ i \notin I \ \Rightarrow \ i\text{-}restraint(R_e^\alpha)[\sigma_n^+] = 0;$$

$$\sigma_n \text{ is a true stage } \Rightarrow \ restraint(R_e^\alpha)[\sigma_n^+] = R.$$

Now, since $I \neq \emptyset$, there is at least one i for which R_e^α eventually remains in state (2) preserving a computation (P, N) witnessing $\Gamma_i(W_e \oplus A)\downarrow$. Since R_e^α never switches back to state (1), this computation continues to apply, so W_e never changes below its use. When the computation was first preserved, computations $\Theta_i^\beta(W_e)\downarrow$ with the same use on W_e were enumerated for all the finitely many β with $S_{e,i}^\beta$ connected to R_e^α, and these computations also continue to apply. Since R_e^α never switches back to state (0), there is at least one β for which Δ_e does not settle down on a prediction that $\Theta_i^\beta(W_e)\downarrow$. Therefore Δ_e does not correctly approximate the jump of W_e, and requirement R_e^α is satisfied.

Lemma 3 *If β is on the true path of the construction and $Q^\beta = S^\beta_{e,i}$, then β has an immediate successor on the true path of the construction. Either R^α_e is satisfied for the unique $\alpha \subset \beta$ with $Q^\alpha = R^\alpha_e$ or $S^\beta_{e,i}$ is satisfied.*

> $S^\beta_{e,i}$: Either Φ^α_e correctly predicts the convergence (or divergence) of $\Gamma_i(W_e \oplus A)$, or Δ_e fails to correctly predict the convergence (or divergence) of $\Theta^\beta_i(W_e)$.

If $S^\beta_{e,i}$ is on the true path, there is some stage ρ past which it is never canceled and it is visited at infinitely many stages. There are five possible results:

If the strategy is inert, $\alpha^\frown n \subset \beta$ for some $n > 0$. That means that $\alpha^\frown n$ is on the true path of the construction, so the lim inf of the restraint of R^α_e is $n > 0$. By Lemma 2, R^α_e is satisfied. The inert strategy never imposes any restraint, so $\beta^\frown 0$ is on the true path.

If the strategy is in state (3) from some stage σ on, it is preserving some computation (P, N) witnessing $\Gamma_i(W_e \oplus A)\downarrow[\sigma]$ by holding restraint r. This neighborhood condition must apply to $W_e \oplus A$ from σ on, since otherwise the strategy would switch to state (4) the next time it was visited, so in fact $\Gamma_i(W_e \oplus A)\downarrow$. Furthermore, Φ_e correctly predicts the convergence of $\Gamma_i(W_e \oplus A)$: When the strategy enters state (3) at stage σ, Φ_e predicts convergence, by construction; and by Claim 3, no strategy will change this prediction, since the computation continues to converge. Thus the requirement $S^\beta_{e,i}$ is satisfied. The restraint past stage σ is r, so $\beta^\frown r$ is on the true path.

If the strategy is in state (4) from some stage σ on, then Δ_e fails to correctly predict the divergence of $\Theta^\beta_i(W_e)$ ($\Theta^\beta_i(W_e)\uparrow$ by Claim 6), so $S^\beta_{e,i}$ is satisfied. No restraint is imposed on A past stage σ, so $\beta^\frown 0$ is on the true path.

If the strategy cycles through states (3), (4) and connection to R^α_e infinitely often, then Δ_e fails to converge on a prediction of the convergence of $\Theta_i(W_e)$. (Every time the strategy is disconnected from R^α_e and put into state (3), Δ_e predicts convergence, and every time the strategy leaves state (4) and connects to R^α_e, Δ_e predicts divergence.) Thus W_e is not low via Δ_e, and $S^\beta_{e,i}$ is satisfied. Restraint with lim inf equal to 0 is imposed on A, so $\beta^\frown 0$ is on the true path.

If the strategy remains connected to R^α_e from some stage σ on, then R^α_e remains in states (1) and/or (2) from some stage on, so by Lemma 2, requirement $S^{\beta}_{e,i}$ is satisfied. No restraint is imposed on A past stage σ, so $\beta^\frown 0$ is on the true path.

Lemma 4 *The set A enumerated by this construction is a non-recursive almost deep r.e. set.*

Because the construction is recursive, A is an r.e. set. By Lemmas 1, 2 and 3, applied inductively, the true path of the construction is an infinite branch through the tree of strategies. Every requirement P_e, R_e and $S_{e,i}$ has an associated strategy on the true path. By Claim 8, A is coinfinite, and by Lemma 1 every requirement P_e is satisfied, so A is non-recursive.

For any e, let R^{α}_e be on the true path above strategies $S^{\beta}_{e,i}$ on the true path. By Lemma 2, if R^{α}_e has a non-zero outcome, then R^{α}_e is satisfied because Δ_e fails to correctly predict the convergence of some $\Theta^{\gamma}_i(W_e)$, so Δ_e does not approximate the jump of W_e. Otherwise, by Lemma 3, either every $S^{\beta}_{e,i}$ is satisfied because Φ^{α}_e correctly predicts the convergence or divergence of $\Gamma_i(W_e \oplus A)$, in which case R^{α}_e is satisfied because Φ^{α}_e approximates the jump of $W_e \oplus A$, or some $S^{\beta}_{e,i}$ is satisfied because Δ_e does not predict the convergence or divergence of $\Theta^{\beta}_i(W_e)$, in which case R^{α}_e is satisfied because Δ_e does not approximate the jump of W_e.

Suppose W is any low r.e. set. Its jump is approximated by some Δ, and for some e we have $(W, \Delta) = (W_e, \Delta_e)$. Because (by the above) R^{α}_e is satisfied, Φ^{α}_e approximates the jump of $W_e \oplus A$; that is, $W \oplus A$ is low. Therefore A is almost deep.

7 Conclusion

We have shown that there is one way to capture a fragment of the definition of deep degree. But we have only just started to explore the relationship between the join and jump. Perhaps the most general question along this line is "Is the existential theory of the poset of r.e. degrees with join and n^{th} jump, for all n, decidable?". Lempp and Lerman in [2] have shown this is the case without join and claim this question has a positive answer in the case when 1 is not included in the language.

We will state and claim some results along these lines without proof. For every non-recursive r.e. set A there is a non-high r.e. set W such that

$A \oplus W$ is high. That is, joining with a non-recursive r.e. **a** cannot preserve the property of non-highness.

Call a r.e. degree **a** *n-deep* if for all r.e. **b**

$$(\mathbf{b} \oplus \mathbf{a})^{(n)} = \mathbf{b}^{(n)}.$$

For all n, there is no non-recursive n-deep degree. That is, joining with a non-recursive r.e. **a** cannot preserve the n^{th} jump.

However there are some possibly approachable open questions. While it is not possible to have a non-recursive r.e. degree **a** such that joining with **a** preserves the jump on all low_2 degrees, it is conceivable that joining with **a** could preserve the property of being low_2. The same question can be asked for other jump classes.

References

[1] M. Bickford and C.F. Mills. Lowness properties of r.e. sets. To appear.

[2] Steffen Lempp and Manuel Lerman. The decidability of the existential theory of the poset of recursively enumerable degrees with jump relations. *Adv. Math.*, 120(1):1–142, 1996.

[3] Steffen Lempp and Theodore A. Slaman. A limit on relative genericity in the recursively enumerable sets. *J. Symbolic Logic*, 54:376–395, 1989.