HYPERSIMPLICITY AND SEMICOMPUTABILITY IN THE WEAK TRUTH TABLE DEGREES

GEORGE BARMPALIAS

ABSTRACT. We study the classes of hypersimple and semicomputable sets as well as their intersection in the weak truth table degrees. We construct degrees that are not bounded by hypersimple degrees outside any non-trivial upper cone of Turing degrees and show that the hypersimple-free c.e. wtt degrees are downwards dense in the c.e. wtt degrees. Moreover, we consider the sets that are both hypersimple and semicomputable, characterize them as the (bi-infinite) c.e. cuts of computable orderings of $\mathbb{N}$ of order type $\omega + \omega^*$ and study their wtt degrees. We show that there are hypersimple degrees that are not bounded by any hypersimple semicomputable degree, investigate relationships with the join and look for maximal and minimal elements of related classes.

1. Introduction

We are interested in how hypersimplicity and semicomputability (in the sense of Jocku sch [4]) relate to the weak truth table degrees. Hypersimple sets where invented by Post as a solution to his problem (now called Post's problem) for the structure of truth table degrees. Then they where shown to be a natural solution to Post's problem for the weak truth table degrees as well. So it is interesting to know the distribution of these natural solutions in the weak truth table degrees. Moreover, weak truth table reducibility is the most appropriate for the study of hypersimplicity given that its essense is the existense of computable bounds (in the use of the relative computation) and hypersimplicity of a set A is based on the same notion: a computable sequence of bounds f(n) below which we get (strictly) more and more elements outside A. This connection becomes even more clear if we note that elements outside A below computable bounds are also important in a weak truth table reducibility since only elements not yet in A and below the use can rectify computations.

It is known (Jockusch[4]) that every c.e. wtt degree has a c.e. semicomputable member while an old theorem of Post asserts that the complete wtt degree contains no hypersimple set. The latter proof makes full use of the completeness of the halting problem. In the next section we show that the c.e. wtt degrees which are bounded by no hypersimple degree (a property of the complete degree) are quite common. In particular, they occur outside

Date: March 2004.
Key words and phrases. Hypersimple, Semicomputable, Weak Truth Table Degrees.

any non-trivial cone of Turing degrees. The existence of such sets can be intuitively justified as hypersimple sets have quite 'sparse' compliments while a wtt reduction $A \leq_{\text{wtt}} W$ in general requires numbers of *fixed* segments of $\mathbb{N}$ to stay outside W (in order to be used for the rectification of the functional we are building, if needed).

Next, we show that the hypersimple-free c.e. wtt degrees are downwards dense in the c.e. wtt degrees; i.e. every non-zero c.e. wtt degree bounds a non-zero hypersimple-free c.e. wtt degree. We ask whether this can be extended to full density and we conjecture a negative answer.Furthermore, we show that for every hypersimple wtt degree there is one strictly above it.

In the final section we study the wtt degrees which contain sets that are both hypersimple and semicomputable. We characterize this class as the c.e. cuts of computable linear orderings of $\mathbb{N}$ of order type $\omega + \omega^*$ (where ω^* is the inverse of ω). This characterization will help a lot in the constructions involving such sets as we only have to deal with linear orderings with the *finite predecessor-or-successor* property (that is, each number has either finitely many predecessors or finitely many successors) and not with a conjunction of hypersimplicity and semicomputability.

Using this, we point out that the wtt *degrees of approximation representations for c.e. reals* studied in Barmpalias[2, 3] are exactly the wtt degrees of hypersimple semicomputable sets (in fact the actual classes of sets coincide) and so some of the results there can be stated in terms of the present paper and contribute to our study. For example, *there is a hypersimple wtt degree which is bounded by no hypersimple semicomputable degree* (a corollary of a result in [3]). Moreover, we can consider the c.e. wtt degrees decomposed into two classes: the ones that are bounded by a hypersimple semicomputable degree and the ones that are not. Since the first one is downwards closed and the second is upwards closed we can think of them as the bottom and upper part of the c.e. wtt degrees (with respect to this decomposition). The two classes are non-trivial (as it follows from [3]) and two very interesting questions are

(a) Are there minimal elements of the upper class?
(b) Are there maximal elements of the bottom class?

A positive answer to question (a) would mean the existence of a bottom of a hypersimple semicomputable free upper cone in the c.e. wtt degrees which bounds only elements of the first class. A positive answer to question (b) would mean the existence of maximal hypersimple semicomputable wtt degrees (in the sense that no degree above them is hypersimple semicomputable). In the last section we prove that there is no maximum hypersimple semicomputable wtt degree (theorem 5). Moreover we construct two degrees of the bottom class whose join belongs to the upper class. This shows that the bottom class is not an ideal and the hypersimple semicomputable wtt degrees are not closed under join. We wish to note that most of the proofs in this paper do not rely on classical strategies for the satisfaction of the requirements. For example, in theorem 5 we are building a set avoiding a given initial segment in the c.e. wtt degrees but the usual Sacks coding cannot be applied because of the nature of the sets we are dealing with. So we needed to design a strategy based on the fact that we are dealing with hypersimple semicomputable sets.

In the following we use standard notation and when we describe a construction we assume a *current value* (corresponding to the current stage) for each of the various parameters involved. All the degrees will be c.e. and $A \le_{\mathrm{wtt}} B$ is indicated as $\Phi^B = A;\phi$ when we wish to make the algorithm (functional) Φ and the computable use ϕ of the reduction explicit. Finally, we use ℓ to denote the length of agreement of a potential reduction e.g. $\ell(\Phi^W = A;\phi)$ is the length of agreement of $A \le_{\mathrm{wtt}} W$ via the functional Φ and with use bounded by the partial computable function ϕ.

2. Wtt c.e. degrees that are not bounded by hypersimple wtt degrees

In this section we look at wtt c.e. degrees that are not bounded by hypersimple wtt degrees. These are degrees containing c.e. sets that cannot be wtt-coded into hypersimple sets. In other words they are bottoms of hypersimple-free upper cones in the wtt degrees.

Theorem 1. *Wtt c.e. degrees that are not bounded by hypersimple wtt degrees occur outside any non-trivial upper cone of c.e. Turing degrees. Formally, if B is c.e. and non-computable then there exists $A \not\ge_T B$ such that the upper cone $\{\mathbf{w} \mid \mathbf{w} \ge \mathbf{a}\}$ in the c.e. wtt degrees is hypersimple-free.*

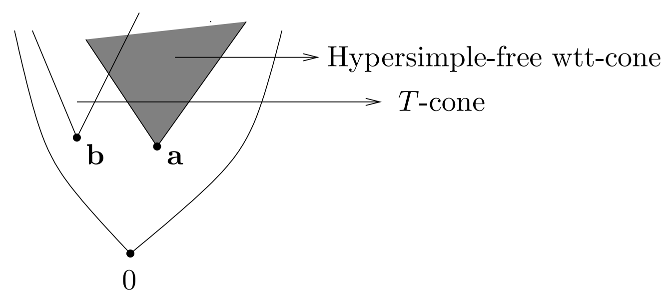

Figure 1: Theorem 1.

Proof. Apart from $A \not\ge_T B$ which can be achieved in a standard way (via Sacks restraints) the requirements we have to satisfy are

$$\mathcal{Q}_{\Phi,W} : \Phi^W = A;\phi \Rightarrow \left\{ \begin{array}{l} \exists(D_n)((D_n) \text{ sequence of consecutive segments of} \\ \mathbb{N} \wedge \ \forall n(\overline{W} \cap D_n \neq \emptyset)) \end{array} \right.$$

As usual, we can assume that ϕ is strictly monotone. The effective sequence (D_n) above will serve as a disjoint array witnessing that W is not hypersimple. If $\forall n(\overline{W} \cap D_n \neq \emptyset)$ fails, we will be able to diagonalize successfully against $\Phi^W = A; \phi$; this will be achieved via a ripple of diagonalizations, the last of which is successful (i.e. is not rectified).

The strategy for $\mathcal{Q}_{\Phi,W}$ consists of steps A_n, B_n, $n \in \mathbb{N}$. The family $(A_n)_{n\in\mathbb{N}}$ enumerates (D_n). If at some stage we find that (D_n) does not fulfil the purpose of its construction (i.e. $D_n \subseteq W$ occurs for some n) we turn to step B_n (for that particular n that witnessed the failure). This D_n-failure step will start a ripple of diagonalizations, succeeding $\Phi^W \neq A; \phi$. Hence, either all A_n are performed (thus satisfying $\mathcal{Q}$ via its second clause) and no B_n is activated, or finitely many A_n are performed, until a single B_n-step is activated which (eventually) ends $\mathcal{Q}$'s activity (satisfying it through the negation of its first clause). $\mathcal{Q}$ will choose the witnesses for its diagonalizations from a special set $U \subseteq \mathbb{N}$ disjoint from the special sets of other requirements (e.g. $U = \mathbb{N}^{[e]}$, the e-th column of $\mathbb{N}$ where e is the index of $\mathcal{Q}$ under an effective ordering of the requirements). Let $a_1 = 1$ and $I_0 = \emptyset$. The A_n, B_n steps are as follows:

A_n *(D_n definition)*
  (1) Define I_n as the set of the next a_n unused (i.e. not in $\cup_{i<n} I_i$) elements in U. *This is the set of witnesses (agitators) of step A_n. They have the potential to be used by B_n after an A_n-failure. Their number $|I_n| = a_n$ is defined by the previous step A_{n-1}.*
  (2) Restrain I_n (from A) and wait until $\ell(\Phi^W = A; \phi) > t$, for all $t \in I_n$.
  (3) Define $D_n := \{\max D_{n-1} + 1, \dots, \max_{i\in I_n} \phi(i) - 1\}$ and $a_{n+1} := |\cup_{i\leq n} D_i| + 1$ $(= \max \phi(\cup_{i\leq n} I_i) + 1)$.

B_n *(D_n-failure diagonalization loop)*
  (a) Wait for a Φ-expansionary stage.
  (b) Put the least element of $I_n \cap \overline{A}$ into A and go to (a).

The $\mathcal{Q}$-module operates as follows: it executes $A_1, A_2, \dots$ but before moving to A_n it checks whether $D_i \cap \overline{W} \neq \emptyset$ for all $i < n$. If this holds, it proceeds to A_n, otherwise it proceeds to B_k for the least k with $D_k \cap \overline{W} = \emptyset$. When the $\mathcal{Q}$ module is called, it starts operating from where it last stopped, until it meets a 'wait' condition which is not fulfilled *or* it finishes an A_n step (in which case it stops at the beginning of A_{n+1}). We start from A_1.

Now that we have defined the operation of $\mathcal{Q}$, we explain why this strategy works. First of all note that $D_1, D_2, \dots$ are consecutive segments of $\mathbb{N}$ and $I_1, I_2, \dots$ are consecutive segments of U (the use-set of $\mathcal{Q}$). The restraints set on U are potentially infinite, but this is no problem as numbers in U are only used by $\mathcal{Q}$. The outcomes are as follows:

(1) when $\mathcal{Q}$ executes all A_n. Then, according to the module (D_i) is an infinite disjoint array with $D_i \cap \overline{W} \neq \emptyset$ for all i. Indeed, in order to proceed to A_{n+1} we must make sure that $D_i \cap \overline{W} \neq \emptyset$ for all $i \leq n$.

(2) when we are permanently stuck in a 'wait' instruction in some A_n step. In this case it is obvious that $\Phi^W \neq A; \phi$ and $\mathcal{Q}$ is satisfied.
(3) when the above fail, and so the $\mathcal{Q}$-module passes control to some B_n step. This must happen after $D_n \cap \overline{W} = \emptyset$ (i.e. $D_n \subseteq W$) has been noticed by the module.

In the third outcome, B_n will start a ripple of at most $|I_n|$ diagonalizations and we claim that the last one will be impossible to rectify. In other words that $\Phi^W \neq A; \phi$ is a certain final outcome. Indeed, the only rectification codes (i.e. numbers that can rectify Φ^W computations) for any agitator in I_n are in $\mathbb{N} \upharpoonright \max \phi(I_n)$ and so they are not more than $\max \phi(I_n)$. But $\mathbb{N} \upharpoonright \max \phi(I_n) = \cup_{i \leq n} D_i$ and since $D_n \subseteq W$ any rectification code (for witnesses in I_n) is in $\cup_{i<n} D_i = \mathbb{N} \upharpoonright \max \phi(I_{n-1})$. So if R_n is the set of these codes,

$$|R_n| = \max \phi(I_{n-1}) < \max \phi(I_{n-1}) + 1 = a_n = |I_n|.$$

Since for each I_n-enumeration (into A, at a Φ-expansionary stage) at least one R_n-enumeration (into W) is needed for a new expansionary stage to come, there will be a (final) I_n diagonalization which is not rectified. This means that the module will be stuck on (a) of B_n unable to obtain an expansionary stage. So $\Phi^W \neq A; \phi$ and the third outcome satisfies $\mathcal{Q}$. Hence the module is successful.

The *construction* for the satisfaction of the $\mathcal{Q}$ requirements is: at stage s run successively the modules of $\mathcal{Q}_0, \dots, \mathcal{Q}_s$. The satisfaction of the requirements follows by the analysis of outcomes we discussed above. In particular, there is no injury. If we wish to add the requirement $A \not\geq_T B$ for some given c.e. non-computable B we just need to attach the $\mathcal{Q}$ requirements in the usual Sacks-restraint argument (e.g. on a tree) for the satisfaction of:

$$\mathcal{N}_\Phi : B \neq \Phi^A.$$

There is no non-trivial interaction of strategies apart from those discussed above and those in the classical Sacks argument. Each $\mathcal{Q}$ strategy will occupy a (1-branching) node on the tree and will only be asked to respect a finite amount of A-restraint. So the only modification in its strategy is to choose I-witnesses larger than this finite (or at least with $\liminf < \infty$) restraint. The verification of this construction follows the lines of the classical Sacks argument and our analysis of outcomes for the $\mathcal{Q}$-requirements. □

In [3] we constructed a hypersimple set which is not wtt-bounded by any cut any computable ordering of $\mathbb{N}$ of order type $\omega + \omega^*$. By theorem 4 of section 5 this implies (in fact, is equivalent to)

Corollary 1. . *There is a hypersimple set which is $\leq_{wtt}$-bounded by no set which is both hypersimple and semicomputable.*

We would like to make an interesting comparison between the $\mathcal{Q}$-strategy in the proof of theorem 1 with the strategy employed in [3] in order to prove the previously mentioned version of corollary 1. The crucial difference is that in the latter, the A-restraint on columns of $\mathbb{N}$ (imposed by a fixed requirement) is only finite; and this is what allows us to make A hypersimple. Here is how we achieve this: our typical requirement is

$$\mathcal{Q}'_{\Phi,W,\psi} : \Phi^W = A; \phi \Rightarrow \begin{cases} W \text{ is not the left cut of the computable} \\ \text{ordering of } \mathbb{N} \text{ of order type } \omega+\omega^* \text{ defined by } \psi \end{cases}$$

Here ψ is the function possibly defining such an ordering $\prec$ on $\mathbb{N}$ (in the sense that $\psi(n,m) = 1 \iff n \prec m$) with left cut W; Φ runs over the partial computable functionals, ϕ, ψ over the partial computable functions and W over the c.e. sets. In a family of steps (A_n) (similar to the ones we used above) we enumerate a set D (instead of an array as in the above argument) intended to be $\overline{W}$. If at some point $D \cap W \neq \emptyset$ we are able to diagonalize through a B_t step in a way analogous to the above proof. This way we are able to satisfy the following requirements:

$$\mathcal{Q}''_{\Phi,W,\psi} : \Phi^W = A; \phi \Rightarrow \begin{cases} W \text{ is not the left cut of the computable} \\ \text{ordering of } \mathbb{N} \text{ of order type } \omega+\omega^* \text{ defined by } \psi \\ \text{or } \overline{W} = D \text{ (so } W \text{ is computable).} \end{cases}$$

The satisfaction of all $\mathcal{Q}''$ imply that A is non-computable. Using this, $\mathcal{Q}''$ implies $\mathcal{Q}'$. Moreover, the only way to have infinite restraints on $\mathcal{Q}'$'s column is to let the sequence (A_n) act forever. According to that construction, this implies that $D = \overline{W}$ and so W is computable. It also implies that $\Phi^W = A; \phi$ and hence the outcome '$D = \overline{W}$' is never realized (so we call it pseudo-outcome). Hence no sequence (A_n) acts forever and the restraint on columns of $\mathbb{N}$ imposed by any fixed requirement is only finite. Using this fact we are able to show that the hypersimplicity requirements are satisfied as well.

So the point is that in [3], due to the special nature of the requirements we were able to force a stop on the (A_n) routine (and so, the restraint it imposes to the lower hypersimplicity requirements) whereas in the proof of theorem 1 we are not.

3. Hypersimple-free c.e. wtt-degrees

The next result shows that the c.e. wtt hypersimple-free degrees are more common than the ones studied in the previous section. In fact, we show their downward density in the c.e. wtt degrees.

Theorem 2. *The hypersimple-free c.e. wtt-degrees are downwards dense in the c.e. wtt-degrees. That is, if* $\mathbf{c} > \mathbf{0}$ *then there is a c.e. hypersimple-free* $\mathbf{a}$ *such that* $\mathbf{0} < \mathbf{a} < \mathbf{c}$.

Proof. By the density of the c.e. wtt-degrees it is enough to show that for every $\mathbf{c} > \mathbf{0}$ there is a hypersimple free $\mathbf{a}$ with $\mathbf{0} < \mathbf{a} \leq \mathbf{c}$. Suppose a non-computable c.e. set C. We are going to construct a non-computable c.e.

$A \leq_{\mathrm{wtt}} C$ and equivalent to no hypersimple set. The requirements (apart from the permitting $A \leq_{\mathrm{wtt}} C$) are:

$$\mathcal{Q}_{\Phi,\Psi,W} : \Phi^W = A; \phi \text{ and } \Psi^A = W; \psi \Rightarrow \begin{cases} \exists (D_n)((D_n) \text{ sequence of} \\ \text{consecutive segments of } \mathbb{N} \wedge \\ \forall n(\overline{W} \cap D_n \neq \emptyset)) \end{cases}$$

We also have the non-computability requirements

$$\mathcal{P}_\Phi : A \neq \Phi.$$

We start off with the following atomic module for $\mathcal{Q}_{\Phi,\Psi,W}$. The idea behind this strategy is similar to the one of theorem 1: assuming $\Phi^W = A; \phi$ we enumerate a strong array (D_n) and try to achieve $\overline{W} \cap D_n \neq \emptyset$ for all n. The definition of each D_n is such that if we all of its elements appear in W later on (giving $\overline{W} \cap D_n = \emptyset$) then we are able to ensure $\Phi^W \neq A; \phi$ by diagonalizing. But since we want $A \leq_{\mathrm{wtt}} C$ such diagonalization must be C-permitted. So since C is arbitrary, in general it will not allow the number of diagonalizations that steps B_n performed in theorem 1. To avoid this difficulty we modify the enumeration of (D_n) using the additional hypothesis $\Psi^A = W; \psi$ that we are given and we make sure that if $D_n \subseteq W$ occurs then we are able to destroy $\Phi^W = A; \phi$ with *a single diagonalization*. Hence, every D_n definition is associated with a diagonalization witness a which will be used if and when $D_n \subseteq W$ occurs.

The C-permitting is represented formally by a function (in other words, a functional with empty oracle) Δ which tries to compute C. Let U an infinite computable set especially for the use of $\mathcal{Q}$ strategy. Below, s is the current stage and any parameters mentioned in the construction are supposed to have a current value.

A_n *(n-attack setup)* Find a least $a < s$ such that $a \in U - A$ and
- $\ell(\Phi^W = A; \phi) > a$
- for all $x \in \cup_{i<n} D_i (\psi(x) < a)$

and define $a_n = a$ and $D_n = \{\max \cup_{i<n} D_i + 1, \ldots, \phi(a_n)\}$.

When A_n is run D_i for $i < n$ are already defined. If $D_n \subseteq W$ later on, we will be able to diagonalize successfully by $a_n \searrow A$ and imposing a finite restraint on A (in order to preserve a segment of W).

B_n *(D-failure step; in particular when the D-enumeration done in A_n has proved wrong)*

Consider a_n which was defined in step A_n.

(a) Wait until $\ell(W, \psi^A; \psi) > x$ for all $x \in \cup_{i<n} D_i$. Restrain $A \restriction v$ where v is the use of these computations.

(b) Express desire for $a_n \searrow A$: define the functional $\Delta \restriction a_n = C \restriction a_n$.

It will be $v < a_n$ as in step A_n. If we are permitted to put $a_n \searrow A$, this enumeration respects the A restraint we imposed in (a) above.

> *This diagonalization can only be rectified via a W-enumeration below $\Phi(a_n)$. But no such enumeration can happen with elements in $\cup_{i<n} D_i$ due to the A restraint we impose. Hence it must be with elements in D_n; however these are already in W (this made us start step B_n) and so the disagreement we create is permanent.*

The parts A_n, B_n above are only a piece of the whole $\mathcal{Q}$-strategy. We call them AB-routine. A recursive iteration of AB -routines $(AB(0), AB(1), \dots)$ constitutes the $\mathcal{Q}$-strategy. We explain how a single AB-routine works. It enumerates its own array (D_n), which is a sequence of consecutive segments of (and potentially covering) $\mathbb{N}$, while Δ belongs to all AB-routines. It starts by performing successively the steps $A_1, A_2, \dots$ and at each A_i it defines D_i. It also finds a suitable a_i which is a witness for a 'back-up diagonalization' planned in case $D_i \subseteq W$ later on, i.e. in case the guess made in A_i is wrong.

After that A_i has been completed and in order to pass to A_{i+1} we check whether $D_k \subseteq W$ holds for some $k \leq i$. In other words, whether all the $\overline{W}$-guesses we made so far (via (D_k)) look correct. If yes, then we can proceed to A_{i+1} in order to push (D_k) further. Otherwise for the least n with $D_n \subseteq W$ we pass control to B_n. No more steps apart from B_n will ever be performed in this AB-routine. B_n activates the back-up diagonalization prepared in A_n: it suggests (at some later suitable stage) a_n as a witness for $\Phi^W \neq A; \phi$ and it also restrains the W-use of the computation (even after a possible $a_n \searrow A$). Note that in the atomic module above there are 'wait' instructions. Taking into account that we may have to wait forever, the outcomes of an AB-routine are:

- $\boxed{1}^{AB}$ As we go through $A_1, A_2, \dots$ we get stuck in a 'wait' instruction of some A_i and stay there forever. According to the 'wait' conditions, this implies the satisfaction of $\mathcal{Q}$.
- $\boxed{2}^{AB}$ Before passing to a next A_i we collapse onto a B_n-step. This does not automatically imply satisfaction of $\mathcal{Q}$ but it advances the functional Δ (which belongs to all AB-routines). If the Δ-axioms enumerated by B_n are later shown to be wrong, C will permit a_n and $\mathcal{Q}$ will be permanently satisfied.
- $\boxed{3}^{AB}$ We go through $A_1, A_2, \dots$ with no permanent distraction. Under this outcome the AB-routine produces an infinite disjoint array (D_n) with $D_n \cap \overline{W} \neq \emptyset$ for all n, thus proving that W is not hypersimple (and $\mathcal{Q}$ is satisfied).

Turning to the whole $\mathcal{Q}$-strategy, we start executing $AB(1)$ (which is identical to the typical AB-routine described above) and continue as follows (in an inductive mode). If and when $AB(i)$ has come to an end (in the sense of outcome $\boxed{2}^{AB}$) *and* Δ looks correct, we start $AB(i+1)$ with the additional (but not essential) restriction that all the a_t-witnesses chosen during its A_t-steps are larger than the witnesses already suggested by the $AB(m)$ for $m < i$, when they terminated (this ensures that every time we pass to a higher AB-routine, Δ has grown longer). If Δ does not look correct, we finish with the $\mathcal{Q}$-*termination routine*:

(a) Let $n = \mu i[\Delta(i) \neq C(i)]$ and a the least witness $> n$ suggested at a previous AB-termination.
(b) Put $a \searrow A$ thus satisfying $\mathcal{Q}$

(the disagreement will be preserved as explained above). From the above, any enumeration into A is C-permitted and so $A \leq_{\text{wtt}} C$. Note that as we go through $AB(1), AB(2), \ldots$, we build on more and more restraints on A. If C is indeed non-computable, Δ must fail and so at some point the $\mathcal{Q}$-termination routine will satisfy $\mathcal{Q}$. The outcomes of the entire $\mathcal{Q}$-strategy are:

$\boxed{1}^{\mathcal{Q}}$ As we go through $AB(1), AB(2), \ldots$ we get stuck in a 'wait' instruction of some $AB(i)$ and stay there forever. Or some $AB(i)$ never stops running. Either case implies the satisfaction of $\mathcal{Q}$ as before, and also that the overall A-restraints that $\mathcal{Q}$ imposes are bounded (i.e. finite).

$\boxed{2}^{\mathcal{Q}}$ A Δ-check finds Δ wrong and we enter the $\mathcal{Q}$-termination routine. Again $\mathcal{Q}$ is satisfied as explained above.

$\boxed{3}^{\mathcal{Q}}$ We never stop running $AB(1), AB(2), \ldots$. This means that Δ is total and correct, so that C is computable.

These outcomes show that our strategy is successful. Moreover it is not difficult to see that all $\mathcal{Q}$ strategies can work together with only a finite injury effect. Whenever some $\mathcal{Q}$ act it initializes all lower requirements and probably increases its A-restraints. But according to the outcomes above it acts only finitely often (imposing a final finite A-restraint) and so it allows lower priority requirements (which respect the higher priority A-restraint) to be satisfied. This also shows that the $\mathcal{P}$ requirements can be added with the same finite injury effect. We reserve special sets P for the witnesses of each $\mathcal{P}$ and let them act according to the usual non-computability strategy: choose a witness larger than the restraints of higher priority $\mathcal{Q}$ requirements, wait until $\Phi(t)\downarrow= 0$ and put $t \searrow A$. When $\mathcal{P}$ acts it initializes all lower priority requirements. When itself is initialized, it starts anew (with a new witness).

Construction. At stage s let the highest $\mathcal{Q}$ or $\mathcal{P}$ requirement (with index $< s$) requiring attention act. A $\mathcal{Q}$ requires attention is one of its AB-routines requires attention; and this happens for $AB(i)$ if all higher AB routines have finished a B-step and itself is ready to move on a further step (after we successfully complete a Δ-correctness check, in case $AB(i)$ is at the beginning). Once an AB-routine ends up in a B-step it starts carrying the responsibility for the correctness of a segment of Δ (namely from the threshold marking the arguments on which the higher AB-routines have enumerated axioms, up to the largest argument for which $AB(i)$ enumerated computations). If a Δ-correctness check fails, we go back to the AB-routine which has the relevant responsibility, and in particular its B-step which enumerated the axioms.

This concludes the description of the construction. For the verification we note that (as explained above) any $\mathcal{Q}$ acts at most finitely many times and so all requirements can work together with the standard *finite injury* effect. The satisfaction of a single $\mathcal{Q}$ is already explained above and this is enough for the verification as there are no non-trivial interactions between the $\mathcal{Q}$ requirements. □

It is natural to ask whether downward density can be extended to full density of the hypersimple-free wtt degrees in the c.e. wtt degrees. If we start with an interval $B <_{\mathrm{wtt}} C$ (instead of just $\emptyset <_{\mathrm{wtt}} C$) one can see that the B-coding into A that we are constructing forces the need for multiple enumeration (similar to the diagonalization ripple of theorem 1) for the satisfaction of (the analogue of) $\mathcal{Q}$; and this requires multiple permitting by C which is not always available. So we conjecture that a non-density result may be possible.

4. Hypersimple Sets in the Wtt degrees: no maximal elements

The following theorem shows that there are no maximal hypersimple wtt degrees i.e. for every hypersimple wtt degree there is one strictly above it.

Theorem 3. *If W is hypersimple, there exists a hypersimple set A such that $W <_{wtt} A$.*

Proof. We have seen in the previous sections that there is a certain type of conflict when we try to construct a hypersimple set A above a given W, and sometimes this makes such a construction impossible. We show now that when we have the information that W is hypersimple, this conflict is managable and a construction is possible. If D_n is an effective enumeration of all finite sets and (Φ, ϕ) runs over an effective enumeration of all partial computable functionals/functions then the following requirements guarantee the result:

$$\begin{aligned} \mathcal{Q}: &\quad W \leq_{\mathrm{m}} A \\ \mathcal{P}_{\Phi\phi}: &\quad \Phi^W \neq A; \phi \\ \mathcal{R}_\phi: &\quad \exists n (D_{\phi(n)} \subseteq A) \ \vee \ D_\phi \text{ not a strong array.} \end{aligned}$$

We say that D_ϕ is a strong array if ϕ is computable and for $n \neq m$, $D_{\phi(n)} \cap D_{\phi(m)} = \emptyset$. Notice that $\mathcal{Q}$ asks for something stronger than we really need, namely m-reducibility instead of wtt. Fix a computable $c : \mathbb{N} \to \mathbb{N}$ which is 1-1 and such that $\mathbb{N} - c(\mathbb{N})$ is infinite (e.g. $c(n) = 2n + 1$). We will arrange that

$$n \in W \iff c(n) \in A$$

thus satisfying $\mathcal{Q}$. Assume a priority list where $\mathcal{Q}$ has highest priority and the infinitely many $\mathcal{P}_\Phi$, $\mathcal{R}_\phi$ follow in an effective way (based on the effective enumeration of (Φ, ϕ) that we assumed earlier). Each of $\mathcal{P}_\Phi$, $\mathcal{R}_\phi$ will be finitary (i.e. act finitely often) and any A-enumeration the do must not bring $\mathcal{Q}$ in difficult position. An A-enumeration affects $\mathcal{Q}$ only when it involves c-codes, i.e. elements in $c(\mathbb{N})$.

$\mathcal{P}_{\Phi\phi}$ **strategy**. As usual, we can assume that ϕ is strictly monotone. We are going to attack $\mathcal{P}$ by steping on the hypersimplicity of W: we construct a strong array (F_n) which tries to show that W is not hypersimple, in such a way that when it fails (i.e. $F_n \subseteq W$) we are able to diagonalize successfully (i.e. in a way that makes a final dissagreement unavoidable) against $\Phi^W = A; \phi$. This will be achieved via a ripple of diagonalizations, the last of which is successful (i.e. is not rectified).

The strategy consists of steps A_n, B_n, $n \in \mathbb{N}$. The family $(A_n)_{n\in\mathbb{N}}$ enumerates (F_n). If at some stage we find that (F_n) does not fulfil the purpose of its construction (i.e. $F_n \subseteq W$ occurs for some n) we turn to step B_n (for *that* particular n which witnessed the failure). This F_n-failure step will start a ripple of diagonalizations, succeeding $\Phi^W \neq A; \phi$. Since W is hypersimple, only finitely many A_n will be performed, and at some point a single B_n-step will be activated which (eventually) ends $\mathcal{Q}$'s activity leaving it satisfied. For the diagonalizations we will choose witnesses from $\mathbb{N} - c(\mathbb{N})$ so that we dont interfere with $\mathcal{Q}$. Let $a_1 = 1$, $I_0 = \emptyset$ and assume a constant restraint r from the higher priority requirements. The A_n, B_n steps are as follows:

A_n *(F_n definition)*

(1) Define I_n as the set of the next a_n unused (i.e. not in $\cup_{i<n} I_i$) elements in $\mathbb{N} - c(\mathbb{N})$, greater than r and not yet in A. *This is the set of witnesses (agitators) of step A_n. They have the potential to be used by B_n after an A_n-failure. Their number $|I_n| = a_n$ is defined by the previous step A_{n-1}.*

(2) Restrain I_n (from A) and wait until $\ell(\Phi^W = A; \phi) > t$, for all $t \in I_n$.

(3) Define $F_n := \{\max F_{n-1} + 1, \dots, \max_{i\in I_n} \phi(i) - 1\}$ and $a_{n+1} := |\cup_{i\leq n} F_i| + 1$ $(= \max \phi(\cup_{i\leq n} I_i) + 1)$.

B_n *(F_n-failure diagonalization loop)*

(a) Wait for a Φ-expansionary stage.

(b) Put the least element of $I_n \cap \overline{A}$ into A and go to (a).

The $\mathcal{P}$-module operates as follows: it executes $A_1, A_2, \dots$ but before moving to A_n it checks whether $D_i \cap \overline{W} \neq \emptyset$ for all $i < n$. If this holds, it proceeds to A_n, otherwise it proceeds to B_k for the least k with $D_k \cap \overline{W} = \emptyset$. When the $\mathcal{Q}$ module is called, it starts operating from where it last stopped, until it meets a 'wait' condition which is not fulfilled *or* it finishes an A_n step (in which case it stops at the beginning of A_{n+1}). We start from A_1.

Now that we have defined the operation of $\mathcal{P}$, we explain why this strategy works. First of all note that $D_1, D_2, \dots$ are consecutive segments of $\mathbb{N}$. The outcomes are as follows:

(1) when $\mathcal{P}$ executes all A_n. Then, according to the module (F_i) is an infinite disjoint array with $D_i \cap \overline{W} \neq \emptyset$ for all i. This is impossible since W is hypersimple.
(2) when we are permanently stuck in a 'wait' instruction in some A_n step. In this case it is obvious that $\Phi^W \neq A; \phi$ and $\mathcal{Q}$ is satisfied.
(3) when the above fail, and so the $\mathcal{Q}$-module passes control to some B_n step. This must happen after $D_n \cap \overline{W} = \emptyset$ (i.e. $D_n \subseteq W$) has been noticed by the module.

In the third outcome, B_n will start a ripple of at most $|I_n|$ diagonalizations and we claim that the last one will be impossible to rectify. In other words that $\Phi^W \neq A; \phi$ is a certain final outcome. Indeed, the only rectification codes (i.e. numbers that can rectify Φ^W computations) for any agitator in I_n are in $\mathbb{N} \restriction \max \phi(I_n)$ and so they are not more than $\max \phi(I_n)$. But $\mathbb{N} \restriction \max \phi(I_n) = \cup_{i \leq n} F_i$ and since $F_n \subseteq W$ any rectification code (for witnesses in I_n) is in $\cup_{i<n} F_i = \mathbb{N} \restriction \max \phi(I_{n-1})$. So if K_n is the set of these codes,

$$|K_n| = \max \phi(I_{n-1}) < \max \phi(I_{n-1}) + 1 = a_n = |I_n|.$$

Since for each I_n-enumeration (into A, at a Φ-expansionary stage) at least one K_n-enumeration (into W) is needed for a new expansionary stage to come, there will be a (final) I_n diagonalization which is not rectified. This means that the module will be stuck on (a) of B_n unable to obtain an expansionary stage. So $\Phi^W \neq A; \phi$ and the third outcome satisfies $\mathcal{P}$. Hence the module is successful. Also, note that in each of the two realizable outcomes above the restraints that $\mathcal{P}$ imposes (to lower priority requirements) are finite.
$\mathcal{R}_\phi$ **strategy**. Although we were able to find a strategy for $\mathcal{P}$ which does not interfere with $\mathcal{Q}$, it is not possible to do the same with $\mathcal{R}$, since hypersimplicity requirements can not afford to choose their witnesses from a pre-arranged computable set. So we have to allow them to enumerate into elements of $c(\mathbb{N})$ as well and to avoid the destruction of $\mathcal{Q}$ we will take advantage of the hypersimplicity of W once more. Based on the given strong array $(D_{\phi(n)})$ (which tries to show that A is not hypersimple) we will construct a strong array (G_n) which tries to show that W is not hypersimple. When (G_n) fails, i.e. $G_k \subseteq W$ for some k, we will cause a $(D_{\phi(n)})$ -failure (i.e. $D_{\phi(k)}$ for some k) without creating any potential problems to $\mathcal{Q}$. Note that (G_n) will definitly fail since W is given hypersimple. To be more specific, we simply define

$$G_n := \{k \mid c(k) \in D_{\phi(n)}\}.$$

Now since W is hypersimple, some $G_n \subseteq W$ at some stage. But then $\mathcal{R}_\phi$ can be satisfied by putting into A only the elements in $D_{\phi(n)} - c(\mathbb{N})$; indeed, $c(\mathbb{N}) \cap D_{\phi(n)}$ is already in A by $\mathcal{Q}$'s module and $G_n \subseteq W$. In other words we satisfy $\mathcal{R}$ without enumerating into A any c-codes (such an enumeration is left to $\mathcal{Q}$).

Construction. In order to let all the strategies work together we only need to make sure that lower priority requirements respect the restraint r set by higher ones. Note that only $\mathcal{P}$ impose restraints. Whenever a $\mathcal{P}$ or $\mathcal{R}$ receives attention we initialize all lower priority $\mathcal{P}$-requirements. Every $\mathcal{P}$ chooses witnesses greater than the restraint r and restrains them; $\mathcal{R}$ only enumerates into A a G_n with all members greater than r. A $\mathcal{P}$ requirement requires attention when its module is ready to move to the next step; and a $\mathcal{P}$ requirement when there is a G_n with $G_n \subseteq W$ and $\min G_n > r$. The *construction* is: at stage s

- For every n, if $n \in W$ (and $c(n) \notin A$) put $c(n) \searrow A$.
- Find the least $\mathcal{P}$ or $\mathcal{R}$ which requires attention in the first case run the relevant module (from where it last stopped) and in the latter find the least n with $G_n \subseteq W$, $\min G_n > r$ and enumerate the elements of $D_{\phi(n)}$ into A. Initialize the lower priority requirements.

The satisfaction of the requirements follows by the analysis of outcomes we discussed above and an application of the finite injury method. □

5. Hypersimple Semicomputable Sets in the Wtt degrees

In the previous sections we dealt with the notion of hypersimplicity and now we consider how semi-computability (in the sense of Jockusch[4]) relates to the wtt c.e. degrees along with hypersimplicity. We recall the following definition:

Definition 1 (Jockusch[4])**.** *A set A is semicomputable if there is a computable f such that*

- $f(x, y) \in \{x, y\}$
- $x \in A \ \vee \ y \in A \Rightarrow f(x, y) \in A$.

Semicomputable sets are known to be exactly the cuts of computable linear orderings of $\mathbb{N}$ and as Jockusch[4] points out,

Proposition 1 (Jockusch[4])**.** *Every c.e. wtt (and indeed tt) degree contains a c.e. semicomputable set.*

So it makes sense to study the wtt degrees of sets that are both hypersimple and semicomputable. First we provide a characterisation of the hypersimple semicomputable sets, which will give a better intuition in our constructions.

Theorem 4. *A set is hypersimple semicomputable iff it is the left c.e. non-computable cut of a computable ordering of $\mathbb{N}$ of type $\omega + \omega^*$.*

Proof. It will be clear that 'left' can be replaced by 'right'. As mentioned above, it is well known that semicomputable sets are exactly the cuts of computable orderings of $\mathbb{N}$. Also, it is not difficult to show that if a cut of a computable ordering of $\mathbb{N}$ of type $\omega + \omega^*$ is c.e. non-computable, then it is hypersimple (see [2][1]). Hence one direction of the theorem follows easily.

For the other, assume that A is semicomputable and hypersimple. Then it is the left cut of a computable ordering $\prec$ of $\mathbb{N}$. Assume an effective enumeration A_s of A (with $\max A_s < s$) and define the set B as follows:

stage s. If s lies on the $\prec$-left of some element in A_s, enumerate $s \searrow B$.

Obviously B is a computable subset of A. It is the set of elements which we know they belong to A, by the time they are enumerated in the standard enumeration of $\mathbb{N}$. We will define a new order $\prec_*$ of $\mathbb{N}$ which is of type $\omega + \omega^*$ and its left cut is A. In fact, $\prec$ and $\prec_*$ will only differ on B.

The intuition is that in order to transform the order type of $\prec$ to $\omega + \omega^*$ it is sufficient (and necessary) to ensure that every element has either finitely many predecessors or finitely many successors. Since A is infinite, any element of $\overline{A}$ has infinitely many $\prec$-predecessors and so we must ensure that it has only finitely many $\prec_*$-successors. Similarly, for the elements in A we must ensure that they have only finitely many predecessors, and we do this by reordering some of them.

We view the construction of $\prec_*$ as mapping (placing) natural numbers into a dense line like $\mathbb{Q}$. The order of the rationals induces $\prec_*$ via the mapping. In fact, we already have such a mapping with respect to $\prec$. Thus we only have to *move* some naturals on the line, and this re-placement will define $\prec_*$. At stage s it is enough to specify the position of s with respect to the numbers in $\mathbb{N} \upharpoonright s$. Here is the construction. Run the construction of B as above and at stage s, if $s \searrow B$ we place s between the two $\prec$-largest elements in $A_s - B$ (and larger than every B-element currently in there). If not, we leave it in its old position.

Note that $\prec$ is a (computable) order; also, we only move elements in the left cut A of $\prec$ and the new positions remain in A. So A is a left cut of $\prec_*$ as well. Now if there was an element in $\overline{A}$ with infinitely many $\prec_*$-successors, there would be an infinite c.e. subset of $\overline{A}$. This contradicts the hypersimplicity assumption. The only thing left to show is that any element of A has finitely many $\prec_*$-predecessors. Indeed, by induction every element in A has a $\prec_*$-successor in A. If $t \in A$ and $t \prec_* k$, every $s > \max\{t, k\}$ will be $\prec_*$-greater than t. This concludes the proof. □

What we did in the above construction is to spot elements which are 'born' too low (i.e. too left on the line) and lift them as much as possible within the least initial segment of the line which contain all elements of A (we

[1]in terms of that paper, theorem 4 can be restated as 'semicomputable hypersimple sets are exactly the approximation representations (of c.e. reals)'.

sometimes call this 'black area'). The set B contains the 'low-born' elements (of the black area). Theorem 4 shows that the *approximation representations* (or simply *representations*) for c.e. reals studied in Barmpalias[2] are just the subsets of $\mathbb{N}$ that are both hypersimple and semicomputable. Below we will often say 'representation' instead of 'hypersimple semicomputable set'. Also, a representation or hypersimple semicomputable wtt degree is one that contains representations. The reason why we use this term is that technically speaking we do not see these sets as a combination of the two classical notions but rather as sets with an easily identifiable and intuitively clear structure (as cuts of computable orderings of a special type). The building of a representation will be a construction of a computable linear ordering of type $\omega + \omega^*$ with a simultaneous infinite enumeration of its left cut (roughly as in the proof of theorem 4).

5.1. **No greatest element for the hypersimple semicomputable wtt degrees.** It is natural to ask whether there is a 'universal' c.e. cut of a $\omega + \omega^*$ computable ordering of $\mathbb{N}$ in the sense that it $\leq_{\text{wtt}}$-bounds every other of the same kind. We give a negative answer by showing that there is no maximum hypersimple semicomputable wtt degree, i.e. one that bounds all the others.

Theorem 5. *There is no greatest hypersimple semicomputable wtt degree.*

For the proof we assume we are given a representation A and we construct a representation B such that $B \not\leq_{\text{wtt}} A$. We want to satisfy the following:

$$\mathcal{Q}_\Phi : \Phi^A \neq B; \phi$$

Here Φ runs over the partial computable functionals and ϕ over the partial computable functions (intended as the use of Φ). The plan is to diagonalize against $\Phi^A = B; \phi$ in a way that is impossible (for the opponent) to rectify (by A-enumeration). For this we will need to diagonalize a number of times, of which the first one (with witness b in the module below) has a special role. We choose b along with a finite number of other witnesses so that after the rectification of $b \searrow B$ the A-enumeration triggered (the set D in the module below) has left less rectification points with respect to our other witnesses than the number of these very witnesses. This guarantees that when we start successive diagonalizations with the other witnesses (at expansionary stages) at least one of them will be impossible to rectify. For this plan the fact that A is a representation is crucial.

We view A as the left bi-infinite cut of a computable $\omega + \omega^*$ ordering $<_A$ of $\mathbb{N}$; so we are given $<_A$ and an enumeration of A. The construction will define a computable ordering $<_B$ of $\mathbb{N}$ of the same type and simultaneously enumerate its unique left bi-infinite cut in B. We view the definition of $<_A, <_B$ as taking place on an A-line and B-line respectively (since they are linear). The enumeration of a cut is represented graphically by a c.e. *black area* (see figure 2) which is continuously expanding and eventually covers the part of the line that contains elements in the cut. The elements of $\mathbb{N}$ are also called points when they are mentioned in relation to the A or B-line. Another way to say $m <_A n$ for two numbers m, n is that m is on the left of n (or n on the right of m) on the A-line (see figure 2). At any stage only a finite segment of $\mathbb{N}$ is $<_A$ (or $<_B$) -ordered and so, as we say, the numbers in this segment have a position on the corresponding line.

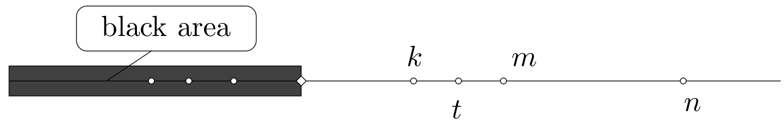

Figure 2: Construction of a c.e. cut of a $\omega + \omega^*$ computable ordering.

In the module below we use the symbols ∞_A, ∞_B which refer to the A and B lines respectively. These have the properties $n >_A \infty_A$, $n \not\leq_A \infty_A$ for all n, $\infty_A \notin A$ and similarly for B. Intuitively they are a non-standard point on the corresponding line, on the left of any standard one and we use them just to make our description simpler and more uniform. To save space, we talk about the strategy as if there is potential for $\mathcal{Q}$ to work with other similar requirements. However it can also be seen as the module of $\mathcal{Q}$ working in isolation. We use the *origins* o_A, o_B (as parameters of $\mathcal{Q}$) on the A, B lines respectively which are the points defining the segments on these lines involved in higher priority requirements' activity (if any). So $\mathcal{Q}$ uses points on the left of the origins and it also assumes that $o_A \notin A$, i.e. that no point on the A-line related to a higher requirement enters A. If this assumption is false it will be initialized.

5.1.1. $\mathcal{Q}_\Phi$*-module.*

(1) Define the origin o_B of this requirement on the B-line as the leftmost point currently outside B (if such doesn't exist, let $o_B = \infty_B$). Put the next number b without a position on the B-line, to the left of o_B. Set $I_B = \{b\}$.

Define the origin o_A of this requirement on the A-line as the leftmost point in the I_A-intervals of higher requirements (if they don't exist or they are empty, let $o_A = \infty_A$). Dynamically define the set I_A for this requirement as:

$$I_A = \{i \mid i < \max_{k \in I_B} \phi(k) \wedge i <_A o_A\} - D - A$$

where D is dynamically defined $= \{i \mid i \leq_A \min_A\{t < \phi(b) \mid t \notin A\}\}$; $\min_A$ is the $<_A$-minimum and $\min_A \emptyset = \infty_A$.

The dynamic definition of a parameter means that whenever mentioned it is (re)defined by applying the definition using the current values of any parameters involved. The set I_B contains the agitators that we plan to use for our diagonalizations. D is the set of numbers *that will enter A if the diagonalization $b \searrow B$ of the next step is rectified. So I_A is the set of elements that can rectify I_B-diagonalizations after $b \searrow B$ has been rectified.*

(2) Wait until $\ell(\Phi^A = B; \phi)$ is greater than all elements of I_B and ask: is $|I_B| > |I_A|$?
   - Yes: Put $b \searrow B$ and redefine dynamically $I_A := I_A - A$ (the right hand side I_A having the value it was last assigned); go to step 4.
   - No: go to step 3.

   If the 'yes' clause holds, then we can start the diagonalization ripple of step 5 and I_A indeed contains the only rectification codes we have to deal with. After $b \searrow B$, D plays no role in the definition of I_A and so we fix the latter. The redefinition $I_A = I_A - A$ is just a way to express that whenever a point of I_A enters A, then it exits I_A (not being a rectification point anymore).

(3) Put the least number m not having a position on the B-line, on the right of b and on the left of any other point currently on the line and $>_B b$ (where $>_B$ is the ordering of $\mathbb{N}$ associated with the representation B we are constructing). Define

$$I_B := I_B \cup \{m\}$$

   and go to step 2.

(4) *(diagonalization loop)*
   (a) Set $I_B = I_B - B$ and wait until the next expansionary stage.
   (b) Put the leftmost point of I_B into B (and expand the black area up to that point) and go to (a).

Note that in contrast to I_A's dynamic definition, I_B has the value it was last assigned. I_A is not necessarily an interval in the A-line (in the sense of the set of points contained in between two points) but it is an interval restricted to numbers in an initial segment of $\mathbb{N}$. However I_B is an interval (on the B-line).

5.1.2. *Analysis of Outcomes.* By its definition, I_A will only be finite in the long run (due to the fact that A is given as a representation); and since we keep putting elements into I_B, at some point (having enough expansionary stages) we will exit step 2 through the 'yes' clause. After step 4(a) every rectification point for any of our I_B witnesses will be in I_A. And since $|I_B| > |I_A|$ (which will always hold since a I_B-diagonalization can only be rectified by a I_A-enumeration) the loop of step 4 will terminate on (a) due to the lack of an expansionary stage. So $\mathcal{Q}$ is satisfied.

5.1.3. *Construction.* For every requirement, if its origin o_A has entered A, initialize all higher priority requirements (i.e. initialize their module and enumerate their I_B set into B). Otherwise consider the highest priority $\mathcal{Q}$

requiring attention (i.e. ready to perform the next step) and activate it; initialize all the lower priority requirements.

5.1.4. *Verification.* We prove by induction that every $\mathcal{Q}$ ceases to require attention and is satisfied with I_A, I_B ending up either undefined (if $\Phi;\phi$ partial) or fixed finite sets. Suppose that all higher priority requirements than $\mathcal{Q}$ are satisfied in this way (and beyond a least stage s_0 they don't require attention). This means that o_A of $\mathcal{Q}$ is eventually settled on a final value outside A.

For a contradiction suppose that $\Phi^A = B;\phi$. Step 1 will run and we claim that the loop of steps 2-3-2 will produce what we call a *saturation state* i.e. a stage that $|I_B| > |I_A|$. Indeed, consider the final value of $t = \min_A\{t < \phi(b) \mid t \notin A\}$. If it is ∞_A then $I_A = \emptyset$ and the inequality holds. Otherwise, only finitely many points can be in $[t, \infty_A)$ since A is representation. So I_A is finite and by successively adding elements in I_B we will eventually get $|I_B| > |I_A|$.

From now on every diagonalization using elements of I_B requires A-enumeration of elements in I_A; indeed, it holds for the diagonalization with b, and any other point below the use of some I_B computation either belongs to D (as it was defined before executing the 'yes' clause of step 3) or I_A, or it is $\geq_A o_A$. But by the time the b-diagonalization is rectified, $D \subset A$ and by hypothesis no element $\geq_A o_A$ is ever going to enter A. This means that any subsequent rectification must be done with elements in I_A. Now that we found a suitable I_A, we fix it except for the fact that we update it by deleting points that have entered A (and so, are useless for rectification).

The (a)-(b) loop of step 4 will keep on reducing I_B ensuring that for each I_B enumeration at least one I_A-enumeration happens. And elements in I_B can only be enumerated by $\mathcal{Q}$ (at expansionary stages) due to the hypothesis (the rest of higher priority requirements) and the fact that lower priority requirements choose I_B-witnesses on the left of those of higher ones. So at some point $|I_A| = 0$ while $I_B \neq \emptyset$ and a further diagonalization with an I_B witness will be impossible to rectify. So $\Phi^A \neq B;\phi$, a contradiction. Since $\Phi^A \neq B;\phi$, after a certain stage there will be no more expansionary stages. So I_B will stabilize and I_A as well (by its definition).

5.2. **Hypersimple Semicomputable wtt degrees and the join.** Since we are studying the class of hypersimple semicomputable wtt-degrees it is natural to ask whether they are closed under join. We show that they are not; moreover, we construct two hypersimple semicomputable sets such that any set which can wtt-compute both of them, is not hypersimple semicomputable.

Theorem 6. *There are hypersimple semicomputable A,B such that no W $\geq_{wtt} A \oplus B$ is hypersimple semicomputable.*

Let $A \oplus B = \{\langle a, b\rangle \mid a \in A \wedge b \in B\}$ where $\langle ., . \rangle$ is a standard pairing function. We want to satisfy the following:

$$\mathcal{Q}_{\Phi,W,\theta} : W \text{ is representation via } \theta \ \Rightarrow \Phi^W \neq A \oplus B; \phi$$

Here Φ runs over the partial computable functionals, W over the c.e. sets and θ over the partial computable functions. The phrase 'W is a representation via θ' means that W is the left cut of the computable ordering of $\mathbb{N}$ determined by θ (in the sense that $n \prec_\theta m \iff \theta(n,m) = 1$) and this ordering has type $\omega + \omega^*$ (ω^* is the inverse of ω). Here we use the fact that representations are exactly the left cuts of such orderings in order to test this property over the list of c.e. sets. In the following when we talk about a particular requirement, θ will only be implicit; i.e. we will talk about a point (i.e number) t being 'on the left' of another k (on the W-line) meaning that $t \prec_\theta k$ (and analogously for 'on the right').

Of course if we only had one representation instead of A, B above, the satisfaction of the requirements would be impossible (and it is instructive to see why). The problems in that situation can be solved if we share our diagonalization witnesses between two sets. The strategy is to gather enough suitable witnesses so that if W is indeed a representation (via θ) and we put each witness into $A \oplus B$ in successive (Φ-) expansionary stages, the W-enumeration we will cause (needed for rectification of Φ^W) is enough to guarantee impossibility of rectification by the time we enumerate the last witness. If W is a representation, we can trigger massive enumerations into W with just one diagonalization since if a point enters W, all points on its left enter W as well; and if $t \notin W$ almost all points are on the left of t. For this plan, the first of our witnesses is the one which triggers a massive W-enumeration and the others just need a usual W-enumeration (i.e. one element below the use). Since we definitely want to diagonalize with a particular witness t before all the others and the sets we are building must be representations, we should either

(1) enumerate t in one of A, B and the rest in the other; or
(2) all in the same set but in this case t must be on the left of all the other witnesses (because otherwise its enumeration will cause other witnesses to be enumerated as well, before they are used).

If W is not a representation (a fact that we cannot predict effectively) the above plan does not work, simply because we may not find suitable witnesses, able to trigger desired W-enumerations (but $\mathcal{Q}$ is satisfied in a trivial way). However, this situation may induce an infinite search for witnesses, and if we choose to act as in (2) we may destroy the representation structure of A or B. So we choose to follow (1) and this is why we need to use diagonalization witnesses from two sets (A and B) instead of one.

We use A for our initial witness and B for the rest ones. In this situation we do restrain our A-witnesses but we don't restrain the B ones unless we are sure we have got enough (to start the diagonalization ripple). So the 'infinite search' described above will have no significant effect in the construction (e.g. in terms of restraints). This approach assumes that B is co-infinite (so that we are able to find arbitrarily many potential witnesses) before we are able to show the satisfaction of the requirements. This assumption is justified (i.e. can be proved) by allowing $\mathcal{Q}_n$ to use B-witnesses only beyond (in particular, to the left of) a certain point $p(n)$—the n-th point outside B counting from right to left—which takes a final value in the course of the construction.

Its time to turn this informal discussion into a formal strategy for a single requirement, the $\mathcal{Q}_{\Phi,W,\theta}$ module described below. To save space, we present it as the module of $\mathcal{Q}_n$ (assuming that $\mathcal{Q}_{\Phi,W,\theta}$ is the n-th requirement in an effective list $\mathcal{Q}_0, \mathcal{Q}_1, \dots$ of all requirements); this does not affect the clarity of the presentation since we can easily get the atomic module (i.e. $\mathcal{Q}_{\Phi,W,\theta}$ working in isolation) by fixing n and considering r (the restraint imposed by higher priority requirements) to be 0. The length of agreement of $\Phi^W = A \oplus B; \phi$ is $\ell(\Phi^W = A \oplus B; \phi)$. By convention we assume that $\Phi^W(t)\downarrow$ implies that all the numbers below the use of the computation have been ordered by θ.

Recall the intuition we built in the proof of theorem 5 on constructing a representation: here we also have A and B lines and a black area for each of these (see figure 2). The current value of $\overline{B}$ is the set of elements having been assigned a position on the B-line and being currently outside B. At each stage s the construction (stated later) will order s on the left of any point outside A on the A-line, and similarly for B. This can be seen as building the orderings of $\mathbb{N}$ associated with the representations A, B. To be consistent with their representation nature, whenever an action enumerates a point into A or B, we assume that all points on its left are also enumerated into the same set (in our terminology, we expand the black area of the corresponding set up to that point).

5.2.1. *$\mathcal{Q}_{\Phi,W,\theta}$-module.*

(1) Choose an A-agitator $a \in \overline{A}$ on the left of any (current) A-agitator of a higher requirement.
(2) Wait until $\ell(\Phi^W = A \oplus B; \phi) > \langle 0, a\rangle$.
(3) Wait until
   (a) $|\overline{B} - R| > |E|$
   (b) $\ell(\Phi^W; A \oplus B) > \langle 1, b\rangle$ for all $b \in I$

   where
   - $p(n)$ is the n-th point (from right to left) on the B-line, outside B.
   - $R = \{t \in \overline{B} \mid t \geq_\theta p(n) \text{ or } \exists k \leq r(t \geq_\theta k \ \wedge k \notin B)\}$. *These are the restrained points.*
   - $E = \{t \mid t >_\theta \min_\theta(\overline{W} \restriction \phi(\langle o, a\rangle))\}$ *(it includes the rectification codes against our planned diagonalizations, at any stage after $a \searrow A$);* $\min_\theta$ is the minimum with respect to θ and by convention $\min_\theta \emptyset = \infty_\theta$, a symbol with the properties $\infty_\theta \notin W$ and for all n, $n <_\theta \infty_\theta$ and $\infty_\theta \not\leq_\theta n$.

- I is the set of the first $|E|+1$ points on the B-line outside B and after (i.e. on the right of) any element of R. *It is the set of B-witnesses for our future diagonalizations and is defined provided that the first condition is satisfied.*

Note that if there are less than n elements on the B-line outside B, $p(n)$ is undefined. R is the set of restrained elements; the component r comes from the higher priority requirements and the component $p(n)$ comes from our intention to make sure that $\overline{B}$ is eventually infinite. E contains the codes that can rectify the B-motivated diagonalizations we plan to do (for which we are searching witnesses in this step) except the ones which are on the left of the leftmost rectification code for $\Phi^W \neq A \oplus B; \phi$ on $\langle 0, a\rangle$ that will be created on step 4. These additional codes will vanish after step 4 and so we need not take them into account. The symbol ∞_θ is analogous to ∞_A or ∞_B that we used in the proof of theorem 5.

The first condition asks for a number of points on the B-line outside B and outside the restrained segment R, greater than the number of elements which can rectify the diagonalizations that can be performed using the former as witnesses. If it is satisfied, we are guaranteed a successful diagonalization. Conversely, if indeed W is a representation via θ, E will be (eventually) finite and since $\overline{B}$ is infinite the condition will be satisfied. Finally, the second condition, if satisfied, makes sure that all rectification codes for our potential diagonalizations have been taken into account in E. Note that every parameter has a current value; e.g. E considers only points (numbers t)) that are currently defined on the W-line.

(4) Restrain I and put $a \searrow A$. Dynamically redefine $E = E - W$. *Once we find suitable B-witnesses we restrain them from B for later use. Note that this restraint is for the lower priority requirements, not $\mathcal{Q}_{\Phi,W,\theta}$ itself (or the higher ones). The enumeration of our A-witness into A triggers the ripple of diagonalizations that are going to follow (as long as we get Φ-expansionary stages). It makes sure that after the next expansionary stage E (as it was defined just before we enter this step) will indeed contain every possible rectification code (and so the plan is sound). Moreover we fix E to its last value (which is what we were looking for), with the exception that elements that enter W are deleted form E as they have no rectification potential; this way, at any time after this step, E will indeed be the set of rectification codes against our diagonalizations.*

(5) *(diagonalization loop)*
   (a) Wait until the next expansionary stage.
   (b) Put the leftmost point of $I \cap \overline{B}$ into B (and expand the black area up to that point) and go to (a).

5.2.2. *Analysis of Outcomes.* Requirement $\mathcal{Q}$ works on the assumption that the higher requirements have ceased to require attention (i.e. have rested). If this is false, it will be initialized. From the module described above it follows that every requirement eventually rests (since there are no infinite loops—I is finite) and so in this analysis of outcomes we can assume that all higher requirements have rested (or that we work with a single requirement in isolation).

If we don't have the chance to perform step 1 it will be because of the lack of expansionary stages and so $\mathcal{Q}$ is satisfied in a very trivial way. The rest of the outcomes are listed below:

$\boxed{w_1}$: we wait in step 2 forever. Then $\Phi^W; \phi$ is partial and $\mathcal{Q}$ is satisfied.

$\boxed{w_2}$: we wait in step 3 forever. Then either we cease getting expansionary stages ($\mathcal{Q}$ satisfied) or each time we get them one of the conditions in step 3 fails. Since $\overline{B}$ is infinite (this is a working hypothesis which will be the first thing to prove in the verification and it does not depend on this analysis),

$$|\overline{B} - R| \to \infty \text{ as } s \to \infty$$

and so $|E| \to \infty$ as $s \to \infty$. But this means that $\min_\theta(\overline{W} \upharpoonright \phi(\langle 0, a\rangle))$ is a point on the W-line (and not ∞_θ) and so W is not a representation. Hence $\mathcal{Q}$ is satisfied and no B-restraints are imposed.

$\boxed{w_3}$: we wait in step 5(a) forever. Again, $\Phi^W; \phi$ partial and $\mathcal{Q}$ is satisfied.

Finally there is a possibility that we are in 5(b) and unable to execute it because $I \cap \overline{B} = \emptyset$. We show that this cannot happen; indeed when we leave step 4 we hold (in I) $|E| + 1$ elements of $\overline{B}$ and these will not enter B unless $\mathcal{Q}$ instructs so (since they are restrained). An enumeration of any of them at an expansionary stage will require W-rectification.

Before leaving step 4 we also put $a \searrow A$ currently being at an expansionary stage, which means that before running step 5 some $t \in \overline{W} \upharpoonright \phi(\langle 0, a\rangle)$ must enter W (for the diagonalization to be rectified). After this W enumeration any point that can rectify an I-diagonalization is in E: indeed, it had a position on the W-line when we left step 3 and at that time it was $>_\theta \min_\theta(\overline{W} \upharpoonright \phi(\langle 0, a\rangle))$ (otherwise it would have entered W by now). Now every time we return to 5(a), $|E \cap \overline{W}|$ will be (at least) one less than it was before; and since $|I| = |E| + 1$ (here E is as it was defined when we left step 3) when we spend our last I-diagonalization, $E \cap \overline{W} = \emptyset$ already and a rectification (and so, leaving (a)) will be impossible.

5.2.3. *Construction.* At stage s put s on the A, B lines (outside the black area) on the left of any existing point outside the black area. Consider the least $\mathcal{Q}$ requiring attention (i.e. ready to perform the next step) and run the corresponding module. Initialize all lower priority $\mathcal{Q}$ requirements.

5.2.4. *Verification.* First we verify our working hypothesis.

Lemma 1. *$\overline{B}$ is infinite.*

Proof. Suppose not, i.e. that $p(n) \to \infty$ as $s \to \infty$ for a least n. By the $\mathcal{Q}$ module, no Q_i, $i \geq n$ can act enumerating (some value of) $p(n) \searrow B$. And since $p(n)$ is (enumerated and) redefined infinitely often, there must be a least $\mathcal{Q}_i, i < n$ which enumerates values of $p(n)$ into B infinitely often. But this is not possible since each $\mathcal{Q}$ only requires attention finitely often (given the finitary nature of the module—there are no infinite loops since I is finite). □

And now, by an adaptation of the analysis of outcomes discussed earlier we can show that each $\mathcal{Q}$ is satisfied. Suppose that $\mathcal{Q}_i, i < n$ have stopped requiring attention. After the last time they received attention, $\mathcal{Q}_n$ will start anew. If it does not execute step 3 (and so 4) its satisfaction follows as in the analysis of outcomes. Otherwise steps 3,4 run and any rectification point on the W line (at any stage) is either in E (as it was defined when step 3 run) or on the left of the leftmost point $< \phi(\langle 0, a \rangle)$ on the W-line outside W. So, since $|I| > |E|$ the loop in step 5 has to stop at some point due to the lack of expansionary stages, thus satisfying $\mathcal{Q}_n$.

References

[1] G. Barmpalias, The approximation structure of a computably approximable real, *J. Symbolic Logic* **68**, no.3 (2003) pp.885–922

[2] G. Barmpalias, Approximation Representations for Δ_2 Reals, *to appear in the Archive for Mathematical Logic.*

[3] G. Barmpalias, Approximation Representations for Reals and their wtt-Degrees, *to appear in Mathematical Logic Quarterly.*

[4] C. Jockusch, Semirecursive sets and positive reducibility, *Trans. Am. Math. Soc.* **131** (1968)

[5] P. Odifreddi, "Classical recursion theory", Amsterdam Oxford: North-Holand, 1989

[6] P. Odifreddi, "Classical recursion theory Vol.II", Amsterdam Oxford: North-Holand, 1999

[7] R. I. Soare, "Recursively enumerable sets and degrees", Berlin London: Springer-Verlag, 1987

School of Mathematics, University of Leeds, Leeds LS2 9JT, U.K.
E-mail address: `georgeb@maths.leeds.ac.uk`