Electronic Notes in Theoretical Computer Science 167 (2007) 17–31 

[www.elsevier.com/locate/entcs](http://www.elsevier.com/locate/entcs)

A Cappable Almost Everywhere Dominating Computably Enumerable Degree

# George Barmpalias [1](#_bookmark0) ,[2](#_bookmark0)

*School of Mathematics University of Leeds Leeds LS2 9JT, U.K.*

# Antonio Montalba´n [3](#_bookmark0)

*Department of Mathematics* *University of Chicago*

*5734 S. University ave. Chicago, IL 60637, USA.*

Abstract

We show that there exists an almost everywhere (a.e.) dominating computably enumerable (c.e.) degree which is half of a minimal pair.

*Keywords:* Almost everywhere domination, minimal pairs, Turing degrees, computably enumerable.

# Introduction

Dobrinen and Simpson [[4](#_bookmark15)] defined the notion of *almost everywhere dominat- ing function* and started studying their Turing degrees. They also defined a uniform version of this property (the *almost everywhere uniformly dominating functions*) but this has been shown in [[1](#_bookmark13)] to be equivalent to the former notion. So let us settle on the following definition.

1 The first author was supported by EPSRC Research Grant number EP/C001389/1.

2 Email: [georgeb@maths.leeds.ac.uk](mailto:georgeb@maths.leeds.ac.uk)

3 Email: [antonio@math.uchicago.edu](mailto:antonio@math.uchicago.edu)

1571-0661 © 2007 Elsevier B.V. Open access under [CC BY-NC-ND license.](http://creativecommons.org/licenses/by-nc-nd/3.0/)

doi:10.1016/j.entcs.2006.08.005

Definition 1.1 Let *f, g* : *ω* → *ω*. The function *f* majorizes *g* if *f* (*n*) ≥ *g*(*n*) for all *n*. If *f* (*n*) ≥ *g*(*n*) for all but finitely *n* then *f* dominates *g*. We say that *f* is an almost everywhere (a.e.) dominating if the measure of reals *β* with the property

*g* ≤*T β* ⇒ *f* dominates *g*

is 1. A Turing degree is a.e. dominating if it contains an a.e. dominating function.

Kurtz [[5](#_bookmark17)] showed that 0' is a.e. dominating. The existence of incomplete

c.e. degrees with this property was shown in [[3](#_bookmark16)] and a high c.e. degree which is not a.e. dominating was constructed in [[2](#_bookmark14)]. In order to have a better picture of this class of c.e. degrees it is natural to compare it with known definable classes, the promptly simple degrees for instance. By well known results these are the non-cappable degrees, i.e. the c.e. degrees which are not halfs of minimal pairs in this structure. In this work we show that there are cappable c.e. degrees which are a.e. dominating. Thus a.e. dominating degrees lie in both parts of the decomposition of the c.e. degrees into promptly simple and cappable degrees.

Theorem 1.2 *There is an almost everywhere dominating c.e. degree which is a half of a minimal pair.* [4](#_bookmark2)

Before we start proving the theorem let us say more about what motivated this work. A central question in the study of a.e. dominating c.e. degrees is whether they are contained in a non-trivial upper cone of degrees. The construction in [[3](#_bookmark16)] of an incomplete a.e. dominating c.e. degree does not seem to combine with many other constructions in the theory of c.e. degrees and in particular the *cone avoidance*. To be more specific, that construction seems incompatible with the situation where *there is unpredictably large number of repeated restraints for a single requirement, in a way that when a new restraint is imposed the old one still remains in force*.

Although this is a very common feature in constructions of c.e. degrees (and in the cone avoidance strategy it seems unavoidable), by combining the construction in [[3](#_bookmark16)] with known techniques from the theory of promptly sim- ple degrees (in particular the use of the recursion theorem in order to trigger prompt diagonalizations) we were able to show that outside any upper cone of promptly simple degrees there is an a.e. dominating c.e. degree. Theorem

[1.2](#_bookmark1) implies this result but is also the first example of an infinite injury argu- ment which constructs a.e. dominating degrees. Indeed, the proof incorporates

4 After the submission of this article the first author was able to show the stronger result that there is an almost everywhere dominating computably enumerable degree which is non-cuppable in the c.e. degrees.

new ideas which help to control the restraint imposed on the a.e. dominat- ing function being constructed, although this may be infinitary. For further background, history, motivation and results related to this area we refer to [[2](#_bookmark14),[1](#_bookmark13),[3](#_bookmark16),[4](#_bookmark15)].

# Proof of theorem [1.2](#_bookmark1)

Following [[3](#_bookmark16)], there is a Turing functional Φ such that *f* is a.e. dominating iff it dominates Φ almost everywhere (where Φ is treated here as an operator from reals to reals). We may assume that all functionals considered here (including Φ) have the standard properties: if Γ*σ*(*n*)[*s*] ↓ then *n* and the (length of the) use of *σ* are less than *s*; also, if Γ*σ*(*n*)[*s*] ↓ then Γ*σ*(*i*)[*s*] ↓ for all *i < n*. We construct two functions *f, g* such that their degrees form a minimal pair. As in [[3](#_bookmark16)] *f* will be an a.e. dominating function which is approximated from below so that it has c.e. degree. In particular we construct a computable double sequence (*fs*(*n*)) such that *fs*(*n*) ≤ *fs*+1(*n*) for all *s, n*. Then we can take *A* = {⟨*n, m*⟩ | *m* ≤ *f* (*n*)} so that *A* is c.e. and Turing equivalent to

*f* . In the construction we often use *f* to mean *fs* where *s* is the current

stage. The function *g* will be the characteristic sequence of a c.e. set and in the construction it will be treated as a c.e. set. First of all we have the domination strategies which ensure that lim*s fs* = *f* exists (by restraining *f* ) *and* that *f* dominates Φ almost everywhere (by increasing *f* ). Then we got the minimal pair strategies. Both of these strategies can restrain *f* and so, as in [[3](#_bookmark16)], they need *backing strategies* which provide them with a good approximation of the measure of the domain of Φ (i.e. *μ*(domΦ)). Finally we got the simplicity strategies which ensure that *g* is not computable.

The construction will be a tree argument both because of the Π0 approx- imation of *μ*(domΦ) and the minimal pair strategies which are infinitary. As we want to make *f* a.e. dominating we will make sure that *at any stage no more than one restraint on f is in force for a single requirement*. According to the uniform labeling of the tree described below this means that at any stage no more than one restraint on *f* is in force at a single level of the tree. This is a principle which made the argument in [[3](#_bookmark16)] work. Interestingly, the proof of theorem [1.2](#_bookmark1) involves stretching the minimal pair construction rather than the construction in [[3](#_bookmark16)] of an a.e. dominating c.e. degree.

2

The tree argument in [[3](#_bookmark16)] naturally defined a 0'' computable sequence (*Ni*) which consisted of the permanent restraints which were imposed on the a.e. dominating sequence *f* being constructed. The property ([3](#_bookmark4)) which was verified for (*Ni*) was the main tool for showing that *f* is a.e. dominating. We will follow the same plan, only that (*Ni*) will be a bit more complex due to the

infinitary nature of the minimal pair requirements. Such a sequence is an important component of the construction of almost everywhere dominating reals. The following proposition says that every possible construction of an

a.e. dominating c.e. real defines such a sequence (*Ni*). Let

*Dn* = {*β* | Φ*β*(*n*) ↓} (1)

*D*[*n,m*)[*g*]= {*β* | (∀*k* ∈ [*n, m*)) Φ*β*(*k*)[*g*(*k*)] ↓}*.* (2)

Note that by the standard conventions on the functionals, if *β* ∈ *D*[*n,m*)[*g*] then *g* dominates Φ*β* in the interval [*n, m*).

Proposition 2.1 *If f is an a.e. dominating c.e. real then from a Turing functional* Φ *we can effectively get a* 0'' *computable sequence* (*Ni*) *such that*

*μ*(*dom*Φ − *D*[*N ,N* )[*f* ]) *<* 3−*i.* (3)

*i i*+1

We sketch a proof: the domain of a Turing functional as an operator from

the Cantor space to itself is a Π0

2

class. Also it is not hard to show the

following.

Lemma 2.2 *The measure of a* Π0

*n*

*or a* Σ0

*n*

*class is computable from* 0(*n*)*.*

*n*

*Moreover the measure of any ﬁnite intersection of* Π0

*n*

*and* Σ0

*classes is com-*

*putable from* 0(*n*)*.*

Now one can define (*Ni*) inductively by using 0'' as an oracle and using the fact that *f* a c.e. real. Note that the 3−*i* in proposition [2.1](#_bookmark3) can be replaced with 3−*g*(*i*) for any *g* ≤*T* 0''; however this is not particularly useful since 0'' is anyway used in order to define (*Ni*).

Requirements and tree.

We approximate functions *f, g* such that *f* is approximated from below and is a.e. dominating; *g* will be a c.e. set and the following requirements will be satisfied:

*Qe* : Φ*f* = Ψ*g* total ⇒ *he* = Φ*f* = Ψ*g*

*e e e e*

*Pe* : |*We*| = ∞⇒ *We* ∩ *g* /= ∅

where (Φ*e,* Ψ*e*) runs over all pairs of Turing functionals, (*We*) over all the c.e. sets and the functions *he* are partial computable and constructed by us. The strategy for *Qe* will be an adaptation of the minimal pair strategy and its outcomes will be inf *<*∗ fin (i.e. *inﬁnite* and *ﬁnite*). The strategy for *Pe* will be the usual strategy for the construction of a simple set: if *We* ∩ *g* = ∅ wait until some *x >* 2*e* appears in *We* such that it does not violate any *g*-restraint of higher priority and enumerated into *g*. The outcomes of *Pe* are diag *<*∗ wait

(i.e. *diagonalized* and *wait*). The domination and *Q* strategies can restrain *f* and so they need backing strategies as explained above. Let *Me* be the *e*th backing strategy and *De* be the *e*th domination strategy.

The construction will be a tree argument where the level 3*e* belongs to the backing strategy *Me*, the levels 3*n* + 1 belong alternately to domination and minimal pair strategies (e.g. 3(2*e*) + 1 belongs to *De* and 3(2*e* + 1) + 1 belongs to *Qe*) and level 3*e* +2 belongs to *Pe*.

Backing strategies.

Before giving the details about how the domination and *Q* strategies work, we need to describe the backing strategies *Me* which provide a good approxi- mation of *μ*(domΦ) and which both of them need. An *M* strategy comes with a *resolution n* and the *error allowance δ* = 3−*n*; its job is to find a rational *q* which approximates *μ*(domΦ) from below to within *δ*. An *M* strategy first divides [0*,* 1) into 3*n* subintervals

[*q*1*, q*0)*,* [*q*2*, q*1)*,... ,* [*qk, qk*−1)

(where *q*0 = 1*, qk* = 0 and *qi > qi*+1) of length *δ*. Its outcomes will be the following specially ordered set

*q*1 *<*∗ ··· *<*∗ *qk.*

When *M* is visited at stage *s* it looks for the least *t* such that *μDt* has moved into a different interval of the partition above since the last stage *s*0 it was visited. If such does not exist (or *s*0 does not exist) it outputs *qk*. Otherwise it outputs the small endpoint of the interval of the partition in which *μDt* currently belongs. That is, if *μDt* ∈ [*qi, qi*−1) then the current outcome is *qi*. To get a picture of the approximation of the outcomes note that for every *t* the measure *μDt* is non-decreasing during the stages of the construction (more precisely the stages of the enumeration of Φ). Moreover

*μDt* ≥ *μDt*+1

for all *t* at all stages. So we can picture *μDt*, *t* ∈ N as a collection of markers on the unit interval which move monotonically towards 1 while preserving their order. Now if *μ*(domΦ) ∈ [*qi, qi*−1) then there will only be finitely many *t* such that *qi*−1 *< μDt* and so after some stage no marker will go beyond [*qi, qi*−1). Also, for almost all *t* the marker *μDt* will settle in [*qi, qi*−1). Thus *qi* will be the leftmost infinitely often visited outcome (and so, the *ﬁnal outcome* and it can be seen as an approximation of *μ*(domΦ) from below within *δ*).

Domination strategies.

Since the domination strategy also has to make sure that lim*s fs* exists it will have infinitely many versions (let *De* be the *e*th domination strategy); each of them will try to fix *f* on a certain initial segment, while ensuring that it is a.e. dominating. Each (version of the) domination strategy inherits a restraint *N* which prohibits it from changing *f* T *N* , and is based on a backing strategy which provides an approximation *q* of *μ*(domΦ) as above. It operates as follows:

1. Wait for a stage *s* + 1 such there exists *σ* ∈ *s<s* with |*σ*| *> N* and the following properties:
   * *σ* ⊃ *fs* T *N*
   * (∀*n <* |*σ*|) *fs*(*n*) ≤ *σ*(*n*)
   * *μ*(*D*[*N,*|*σ*|)[*σ*] *> q* − *δ*
2. Let *fs*+1 ⊃ *σ* and restrain *f* T |*σ*|.

Then, if the guess *q* is correct, the reals *β* such that Φ*β* is not majorized by *f* in [*N,* |*σ*|) will have measure less than 2*δ*. This strategy also contributes to the convergence of (*fs*) since it restrains it on a certain segment. Note that in [[3](#_bookmark16)] the domination strategies are incorporated into other strategies rather than distinguished as in this construction. If this strategy sits on a node *α* the restraint it imposes will be *rα* and its state/current outcome will be active or inactive according to whether *rα* /=0 or *rα* = 0 respectively.

Minimal pair strategy.

Suppose that *α* works for *Qe*. If it works in isolation, it just has to follow the usual minimal pair strategy: it enumerates a partial computable function *hα* as follows. At expansionary stages (i.e. stages where the length of agreement

*le* = max{*x* | Φ*f* T *x* = Ψ*g* T *x*}

*e e*

is larger than ever before) it defines *hα* = Φ*f* T *l* = Ψ*g* T *l* and defines restraint

*e e*

*rα* equal to the (maximum) use of these computations. Also it requires that

when one of *f* T *rα*, *g* T *rα* changes, then the other one does not change until the next expansionary stage occurs. By the usual inductive argument on the stages one can verify that this strategy succeeds (see [[6](#_bookmark18)]).

The strategies for all *Qe* can work together with non-computability re- quirements for *f, g* on a tree, thus constructing a minimal pair of c.e. degrees.

|  |  |  |
| --- | --- | --- |
| off | *f* -switched | *g*-switched |

Table 1

The states of Q*e*.

However this construction has the feature that unpredictable number of re- peated restraints may be applied to *f* for the sake of a single requirement *Qe* in such a way that when a new restraint is imposed, the existing restraints must remain in force. To visualize this situation in the context of a tree argu- ment, these are the restraints *on the left of the true path* which can easily be accumulated and should be respected. As we pointed out earlier this feature prevents us from making *f* a.e. dominating.

To overcome this obstacle we do the following: whenever we wish to impose a restraint on *f* for the sake of *Qe* but there is already such a restraint in force for *Qe* we try to *push the existing restraint to g*. The reason that we can do this is that *a restraint on g is as good as a restraint on f* given that the purpose of issuing restraints is to preserve one side of the double computation

Φ*f* T *x* = Ψ*g* T *x.*

*e e*

Of course this means that the restraint of a minimal pair strategy may increase due to the request of lower priority strategies (i.e. strategies on the same level of the tree and to its right) and in general the restraint on *g* will be much more than usual. But it will be sufficiently well behaved so that it allows the *Pe* requirements to succeed in making *g* non-computable. However it will be clear by the end of the proof that this method does not allow us to ask *g* to be a.e. dominating as well. In other words, we cannot apply this trick to both *f* and *g* in order to construct a minimal pair of a.e. dominating degrees.

Adapted minimal pair strategy on the tree.

Suppose that *α* on the tree works for *Qe*. As usual, *α* will have its own enumerations Φ*α,* Ψ*α* of the functionals Φ*e,* Ψ*e*. In particular Φ*α*[*s*] will be Φ*α*[*t*] where *t* is the largest *α*-stage ≤ *s* (and similarly for Ψ*α*). The length of agreement *lα* is now defined according to Φ*α,* Ψ*α*.

We look at the minimal pair argument in some detail in order to be able to make the modifications we need. Every *Qe* strategy *α* has a restraint *rα* which may tend to infinity (we use one restraint for both *f, g* and direct it to one of these functions if needed). The restraint *rα* (or the strategy *α*) can be in one of the following states: off, *f* -switched, *g*-switched (see the table [1](#_bookmark5)). The states of *Qe* are shown in the following table.

|  |  |  |
| --- | --- | --- |
| off |  | *f* -switched |
| off |  | *g*-switched |
| *f* -switched | → | *g*-switched |

Table 2

Possible transitions between the states of Q*e*.

Switching of the restraint to different states will be performed explicitly during the construction. The possible transitions are shown in table [2](#_bookmark6).

The idea is that when we have an expansionary stage we can switch to off since it does not matter if one side of the double computation is injured. However if the *f* side is injured we need to switch the restraint to *g* (and similarly if the *g* side is injured). A *g*-switched or *f* -switched restraint can return to off state if another expansionary stage comes, in which case both computations will be restored and the restraint *rα* will possibly increase.

*The extra transition* which is not a feature of the usual minimal pair ar- gument is the third one in table [2](#_bookmark6). If a *Q* strategy wants to start defining *hα* and setting up a positive *rα* it is actually exposing itself to possibly becoming *f* -switched since the *g* side of its computations may later be injured. If there is already an *f* -switched strategy *β* on the left of *α* and on the same level then *α* cannot take this risk since it would allow the possibility of two *f* restraints on the same level—something we want to avoid. So in that case *α* will try to *g*-switch the restraint on the left before it starts. It does that by waiting for a stage where the length of agreement *lα* is larger than *lβ*. Then it can increase *rβ* up to the use of Ψ*g* T |*hβ*| (where *Qe* is the requirement that *α, β* are trying to fulfill), set *β* to *g*-switched state and proceed.

*e*

Comments.

Before we state formally the set of instructions that a *Q* node *α* follows we make some remarks. First, when a *Q* strategy switches from a restraint on *f* to no restraint on *f* (i.e. to off or *g*-switched) then all domination strategies below *α-*fin must be initialized since they counted a restraint that is not permanent. Second, the fact that we can only pass from *f* -switched to *g*- switched and not vice-versa is crucial. This transition causes the restraint *rβ* to increase for nodes *β* on the left of the true path when they are not accessible. The fact that this transition is one-way guarantees that the state of the *Q* nodes on the left of the true path will settle down and so their restraint will reach a limit as well. This is something we need since otherwise the simplicity requirements *P* for *g* would not get the chance to be satisfied.

Finally we point out the different attitude of the positive strategies for *g* and *f* , i.e. the simplicity strategies *P* and the domination strategies *D*. A *P* strategy will respect *both* the usual restraint from the nodes on the left (it will enumerate no number which is less than the last stage where a node to the left of it was accessible) *and* the *g*-switched restraints of *Q* nodes to the left of it or above it. The standard restraint from the nodes on the left prevents the situation where there is an *f* -switched node on the true path, a strategy *β* in off state at the same level to the left of it and a node on the right of the true path switches *β* to an *f* restraint thus creating a second restraint on *f* at the same level. Domination strategies are more demanding and will respect less restraint: they only listen to *f* -switched restraints of higher priority (i.e. above or to the left of them) and not the standard restraint.

* 1. *Formal strategies*

We give the sets of instructions for each strategy which constitute the programs actually run in the construction.

*Me*-node *α*.

These are the backing strategies and they run exactly as described above. In particular, if *α* is an *Me*-node then it works with resolution *e*. This means that it has *k* = 3*e* outcomes

*q*1 *<*∗ ··· *<*∗ *qk.*

which are rationals such that *q*0 = 1*, qk* =0 and *qi > qi*+1.

*Qe*-node *α*.

Let *Nα* be the largest restraint imposed on *f* by a higher priority require- ment. That is, the largest *rβ* such that *β <L α* or *β* ⊂ *α* and one of the following holds:

* *β* is an *f* -switched *Q* node.
* *β* is an active *D* node.

Also, let *qn* be the outcome of the predecessor of *α* (i.e. the outcome of its backing strategy) and *δα* = 3−*e* (the distance between the outcomes of the backing strategy). Follow the instructions:

1. If *lα* is larger than ever before and larger than all values that the pa- rameters *lβ* with *β <L α* and |*β*| = |*α*| have ever taken then do the following.
   * if there is an *f* -switched strategy *β* such that *β <L α* and |*β*| = |*α*| then set *rβ* equal to the use of Ψ*g* T |*hβ*|; also *g*-switch *β*. Initialize all *D* strategies of lower priority than *β*.

*e*

'

* + If *μ*(*D*[*Nα,N* ')[*f* ]) *> qn* − *δα* where *N* = max{*Nα, u*} +1 and *u* is the

*α*

use of the computations Φ*f*

*α*

T (|*hα*| + 1), Ψ*g*

T (|*hα*| + 1), then go to

the next step. Otherwise access outcome fin and pass control to the

construction.

* + Define

*hα* T (|*hα*| + 1) = Φ*f*

*α*

T (|*hα*| + 1) = Ψ*g*

T (|*hα*| + 1)

set *α* to off state, define *rα* equal to the use of the above computations, access outcome inf and pass control to the construction.

*α*

1. Otherwise access outcome fin and pass control to the construction.

*(apart from the states of a Q node that we discussed, such a strategy may be active or inactive according to whether rα* /=0 *or rα* = 0*. This qualiﬁcation refers to the potential of the strategy to restrain a segment of f.)*

*Pe*-node *α*.

1. If *We* ∩ *g* /= 0 access outcome diag and pass control to the construction.
2. Otherwise check if currently there exists *x* (less than the current stage) such that
   * *x >* 2*e*
   * *x* ∈ *We*
   * *x > r* where *r* is the least number greater than all stages where a node to the left of *α* was accessible and all *g*-switched restraints *rβ* where *β* is any node of lower priority (i.e. to the left or below *α*).

If yes, put the least one into *g* and pass control to the construction along with a request to terminate the current stage. Otherwise access outcome wait and pass control to the construction.

*De*-node *α*.

Let *Nα* be the largest restraint imposed on *f* by a higher priority require- ment (as described in the *Q*-node strategy) and *s* the current stage. Also,

let *qn* be the outcome of the predecessor of *α* (i.e. the outcome of its backing strategy) and *δα* = 3−*e* (the distance between the outcomes of its backing strategy).

1. If *α* is in active state access outcome active and pass control to the construction.
2. If *α* is in inactive state and there exists *σ* ∈ *s<s* with the following properties
   * |*σ*| *> Nα*
   * *σ* ⊃ *fs* T *Nα*
   * (∀*n <* |*σ*|) *fs*(*n*) ≤ *σ*(*n*)
   * *μ*(*D*[*Nα,*|*σ*|)[*σ*] *> qn* − *δα*

then define *f* T |*σ*| = *σ*, set *rα* = |*σ*| and access the active outcome (thus setting *α* in active state). Pass control to the construction along with the request to terminate the current stage.

1. Otherwise access outcome inactive and pass control to the construction.
   1. *Construction*

Start with *f*0(*n*) = 0 for all *n* and *g* = ∅. Initialize all strategies *α* (set *rα* = 0,

*hα* = ∅). At stage *s* do the following:

1. Initialize every active *D* or *Q* strategy which became active at some stage

*s*0 *< s* and is based on outcome *qi* of its backing strategy such that

*μDt*[*s*] ≥ *qi*−1

where *t* is the largest number such that *μDt*[*s*0] *< qi*−1. Also if *α* was initialized, do the same for the strategies of lower priority than *α* (the ones below it or to the right of it).

*(This cancellation is allowed since the outcomes of the backing strategies considered have been proved wrong. Also, it is necessary since such strategies may impose ‘too much’ restraint on f, thus preventing it from being a.e. dominating. Note that this initialization does not happen automatically since some of these strategies may lie on the left of the true path.)*

1. Start accessing the nodes of the tree starting from the top node , running the corresponding strategies and determining the successor according to the outcome suggested by the previous strategy. Continue until either stage *s* is terminated by a request of a strategy we run or we reach a strategy *α* with |*α*| = *s*. If *α* is the last node we access, initialize all nodes to the left of it. If a *Q* strategy *α* was in off-state at the beginning of the stage and *g* T *rα* or *f* T *rα* changed, *α* changes to *f* or *g*-state respectively.
   1. *Veriﬁcation*

The following lemma is evident from the construction.

Lemma 2.3 *The state of a Q node α can only change provided that it is*

*active (i.e. rα* /= 0*) and one of the following reasons holds.*

* + 1. *Initialization of α sets its state to* off*.*
    2. *A node to the right of α on the same level changes the state of α from*

*f-switched to g-switched.*

* + 1. *A domination node of lower priority (i.e. below or to the right of it) changes its state from* off *to f-switched.*
    2. *A simplicity strategy for g below α switches its state from* off *to g- switched.*

Lemma 2.4 *The following basic facts are true.*

* *There is an inﬁnite leftmost inﬁnitely often visited path TP through the tree.*
* *The states and restraints of the Q and D nodes to the left of TP reach a limit.*
* *If a node lies on TP then it is initialized ﬁnitely many times.*

Proof. We prove these claims by simultaneous induction on *TP* . Note that every node has finitely many branches and so we can immediately conclude that the outcomes of a node on *TP* will have limit infimum. For the top node all claims are trivial since it is a backing strategy. Suppose that *α* = *TP* T *n* in an infinitely often visited node such that after some stage *s*0 no node to the left of it is visited. Since *α* has finitely many branches there will be an outcome *o* such that the node *α*+ = *α-o* is accessed infinitely often and any node to the right of it finitely often. So *α*+ belongs to *TP* and after some stage *s*1 no node to the left of it is visited. At *s*1 there are only finitely many active *Q* or *D* nodes to the left of *α*+ and so, after some stage *s*2 *> s*1 there will be no initializations of *α*+. According to lemma [2.3](#_bookmark7) the only state transitions that can happen to nodes on the left of *α*+ after *s*2 concern active *Q* nodes (of which there are finitely many) which turn from off to *f* -switched or from *f* -switched to *g*-switched. This can only happen finitely often and thus the states of all nodes to the left of *α*+ will stabilize at some stage *s*3 *> s*2. At stage *s*3 the restraints *rβ* of the nodes *β <L α*+ have reached their final values.

Lemma 2.5 *If α is a Q node and α-fin is on TP then rα reaches a limit.*

Proof. After some stage *α-*fin will stop being initialized and so, it will stop having expansionary stages. Then *rα* can not be increased anymore.

Lemma 2.6 *If α is a P or D strategy on TP then the restraint imposed on*

*α reaches a limit. Hence all P are satisﬁed.*

Proof. Suppose that *α* is a *P* node. By lemma [2.4](#_bookmark8) the *g* restraint coming from nodes to the left of *α* will reach a limit. The only other restraint that *α* has to consider come from *g*-switched *Q* nodes *β* such that *β-*fin ⊆ *α*. But according to lemma [2.5](#_bookmark9) this reaches a limit as well. The same argument applies to the case where *α* is a *Q* node, only that now we do not have to consider the restraint “last stage where a node to the left was accessible”. The satisfaction of *P* is evident.

By construction at any stage there is at most one strategy at each level which restrains *f* (i.e. which is an *f* -switched *Q* strategy or an active *D* strat- egy). So at each level of the tree there is at most one strategy with a permanent restraint on *f* (i.e. a permanently active *D* node or a permanently *f* -switched *Q* node with a final restraint). Let (*σi*) be inductively defined as follows. Set *σ*0 = ∅ and let *σi*+1 be the node *β* of least length which holds a permanent restraint on *f* and has |*β*| *>* |*σi*|. Then |*σi*| *<* |*σi*+1| and all these nodes lie to the left or on *TP* . Also, if *Ni* is the final value of *rσi* (and *N*0 = 0) we have *Ni < Ni*+1.

Lemma 2.7 *The sequence* (*σi*) *is inﬁnite.*

Proof. Suppose for a contradiction that *σi, Ni* is the last term of the sequence. Let *σ* ⊃ *σi* be a *D* node on the true path. If *q*0*,... , qk* are the outcomes of its backing strategy and *μ*(domΦ) ∈ [*qi, qi*−1) then *σ* will be under *qi* = *qσ*. Also all markers *μDm* will eventually pass the threshold *qσ* − *δσ*. After all states and restraints on the left of *σ* have stabilized no node on the same level as *σ* and to the left of it will be active (according to hypothesis). After such a stage and when *μDrσ > qσ* − *δσ* (where *rσ* = *Ni*) *σ* will be visited and it will define *f* such that *μD*[*Ni,rα*)[*f* ] *> qσ* − *δσ* thus becoming active. By hypotheses it will remain active and this is a contradiction. So (*σi*) is infinite.

Lemma 2.8 *f* = lim*s fs exists.*

Proof. Once *σi* becomes active for the last time thus imposing restraint *Ni*, the segment *f* T *Ni* will not change.

Lemma 2.9 *For every i* ≥ 1 *the measure of the reals β in the domain of* Φ *such that* Φ*β is not dominated by f in* [*Ni*−1*, Ni*) *is less than* 2 · 3−*ni, where* 3*ni* +1 *is the level of σi. Formally,*

*μ*(*dom*Φ − *D*[*N ,N* )[*f* ]) *<* 2 · 3−*ni.*

*i*−1 *i*

*Also, ni* ≥ *i.*

Proof. If 3*ni* +1 or 3*ni* + 2 is the level of *σi* then its resolution is *ni* and *i* ≤ *ni*. Also the error allowance is *δi* = 3−*ni* . Let *qj* be the outcome of its backing strategy. Then if *σi* became permanently active at stage *s*0 and *m* is least such that *μDm*[*s*0] ∈ [*qj, qj*−1), by step *A* of the construction we have *μDm*[*s*] ∈ [*qj, qj*−1) for all *s* ≥ *s*0 , so *μ*(domΦ) ∈ [*qj, qj*−1); we also have that *μ*(*D*[*Ni*−1*,Ni*)[*f* ]) *> qj* − *δi* (otherwise *σi* would not be active). So the reals *β* for which Φ*β* is not majorized by *f* in [*Ni*−1*, Ni*) can have measure at most 2*δi* which is 2 · 3−*ni* .

Lemma 2.10 *The degrees of f, g form a minimal pair.*

Proof. Suppose that *h* ≤*T f* and *h* ≤*T g* for some function *h*. We show that

*h* is computable. Let Φ*e*, Ψ*e* be programs such that

Φ*f* = Ψ*g* = *h.* (4)

*e e*

Then there is *α* ⊂ *TP* such that Φ*α* = Φ*e* and Ψ*α* = Ψ*e*. We show that *h* = *hα*. By ([4](#_bookmark11)) we have that *α-*inf ⊂ *TP* . The whole argument takes place in a final segment of stages where *α* is not initialized and all nodes to the left of *α* have finalized their states and restraints. It follows by [4](#_bookmark11) that there will be no *f* -switched strategy to the left and on the same level as *α*.

First we wish to show that outcome inf is accessed infinitely often and thus *hα* is total. Since we get infinitely many expansionary *α*-stages (stages where *α* is accessible) it is enough to show that

*μD*[*Ni,N* )[*f* ] *> qα* − *δα*

where *N > Ni* is a fixed number, *qα*, *δα* are the outcome and resolution respectively of the backing strategy of *α* and *i* is the maximum such that

|*σi*| *<* |*α*|. If |*α*| = 3*e* +1 then *δα* = 3−*e*. The level 3*ni*+1 +1 of *σi*+1 will be larger than the level 3*e* +1 of *α*. So by lemma [2.9](#_bookmark10) we have

*μ*(domΦ − *D*[*N ,*∞)[*f* ]) ≤ Σ *μ*(domΦ − *D*[*N ,N* )[*f* ]) ≤ Σ 2 · 3−*j* ≤ 3−*e.*

*i j*−1 *j*

*j>i j>e*

So

*μ*(*D*[*N ,N*)[*f* ]) ≥ *μ*(*D*[*N ,*∞)[*f* ]) *> μ*(domΦ) − 3−*e* ≥ *qα* − 3−*e.*

*i* *i*

So *α* will be allowed to access outcome inf infinitely many times thus increas- ing |*hα*| infinitely often. Now we only need to show *hα* = *h*. For a contradiction suppose that for a least *x*, *hα*(*x*) /= *h*(*x*). When *hα*(*x*) was defined at some expansionary *α*-stage *s*0 we had

*hα*(*x*)= Φ*f* (*x*)= Ψ*g* (*x*)*.* (5)

*α α*

If *s*1 is the largest expansionary stage for *lα* at which ([5](#_bookmark12)) holds, *rα* is larger than the corresponding uses of the computations. Let *s*2 be the least stage after *s*1 at which one of *f* T *rα,g* T *rα* changes (from the value they had at *s*1), and *s*3 the next expansionary stage after *s*1. By construction at each stage at most one of *f* , *g* can change. So we have the following cases:

1. *f changes*: When the change happens *α* becomes *g*-switched. Note that before *s*3 the outcome *α-*inf will not be accessible and so *α* remains *g*-switched until *s*3. So no node to the right of *α* or below it will change *g* T *rα* before stage *s*3. This means that at *s*3 ([5](#_bookmark12)) will hold again, a contradiction.
2. *g changes*: When the change happens *α* becomes *f* -switched. If it remains in this state until *s*3 by the same argument we get that ([5](#_bookmark12)) will hold at stage *s*3, a contradiction. Otherwise a node to the right of *α* turns this strategy to *g*-switched state at a stage *s*∗ between *s*2 and *s*3. But at that stage ([5](#_bookmark12)) must hold (possibly with new use *u*) and *rα* will increase to cover the use of these (possibly new) computations. From stage *s*∗ to stage *s*3 the node *α* will remain in *g*-state and as before it will preserve the *g*-side of the computation ([5](#_bookmark12)), thus guaranteeing that this equality survives until stage *s*3; this is a contradiction.

This shows that ([5](#_bookmark12)) holds for all *x*.

Lemma 2.11 *The function f is almost everywhere dominating.*

Proof. If a real *β* is such that the function Φ*β* is total and is not dominated by *f* , then it must belong to infinitely many of the sets domΦ − *D*[*Ni*−1 *,Ni*)[*f* ] for *i* ≥ 1. So according to lemma [2.9](#_bookmark10) these reals belong to a set of measure

0.

# References

1. Binns, Stephen, Bjørn Kjos-Hanssen, Joseph Miller and Reed Solomon, *Lowness notions,* *measure and domination*, in preparation.
2. Binns, Stephen, Bjørn Kjos-Hanssen, Manuel Lerman and Reed Solomon, *On a conjecture of* *Dobrinen and Simpson concerning almost everywhere domination*, J. Symbolic Logic 71 (2006), no. 1, 119–136.
3. Cholak, Peter, Noam Greenberg and Joseph Miller, *Uniform almost everywhere domination*, to appear.
4. Dobrinen, Natasha, and Stephen Simpson, *Almost everywhere domination*, J. Symbolic Logic

69 (2004), no. 3, 914–922.

1. Kurtz, Stuart, ”Randomness and Genericity in the Degrees of Unsolvability”, Ph.D. thesis, University of Illinois at Urbana-Champaign, 1981. VII + 131 pages.
2. Soare, Robert I., “Recursively enumerable sets and degrees”, Berlin London: Springer-Verlag, 1987.