PROCEEDINGS OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 00, Number 0, Pages 000–000
S 0002-9939(XX)0000-0

ON A QUESTION OF SLAMAN AND GROSZEK

ANDREW E.M. LEWIS

(Communicated by Julia Knight)

Abstract. We answer a question of Slaman and Groszek by showing that any non-computable perfect tree computes one of its non-computable paths.

1. Introduction

In [SG] Slaman and Groszek showed that if there is a non-constructible real then every perfect set has a non-constructible element, thus answering a question of Prikry [HF]. They then asked if a similar result would hold in an effective context, replacing relative constructibility with Turing reducibility.

Definition 1.1. We say a non-empty $T \subseteq 2^{<\omega}$ is perfect if for every τ in T there exist at least two incompatible τ' extending τ in T.

Question 1.2 (Slaman and Groszek)**.** Does every non-computable perfect T compute one of its non-computable paths?

This paper provides an affirmative answer to question 1.2. Quite apart from the fact that this is a basic and fundamental question in its own right, further motivation is provided here by connections to an old question of Yates, one of the longstanding questions of degree theory.

Definition 1.3. A Turing degree **b** is a **strong minimal cover** for **a** if the degrees strictly below ***b*** are precisely the degrees below and including ***a***.

Question 1.4. (Yates) Does every minimal degree have a strong minimal cover?

In [AL] it was shown that if $A \subseteq \omega$ satisfies the property that for every perfect $T \leq_T A$ there exists perfect $T' \subseteq T$ computable in A such that every path on T' computes A, then the degree of A has a strong minimal cover. The positive solution to Slaman and Groszek's question suffices to show that if A is of minimal degree and perfect $T \leq_T A$ then A computes some path on T which computes A, although we leave open the question as to whether A necessarily computes a perfect $T' \subseteq T$ which has paths only of this kind.

In what follows all notation and terminology will be standard unless explicitly stated otherwise. We fix an effective bijection from ω to the finite subsets of $2^{<\omega}$,

2000 *Mathematics Subject Classification.* Primary 03D28.
The author was supported by Marie-Curie Fellowship MEIF-CT-2005-023657 and partially supported by the NSFC Grand International Joint Project no. 60310213, New Directions in the Theory and Applications of Models of Computation.

and we will often abuse notation by considering the output of a Turing functional on any given input to be a finite subset of $2^{<\omega}$ rather than an element of ω. We write λ in order to denote the string of length 0. We shall use the variable T to range over subsets of $2^{<\omega}$ which may not be downward closed. Given any $T \subseteq 2^{<\omega}$ and $\tau, \tau' \in T$ we say that τ' is a successor of τ in T if $\tau \subset \tau'$ and there does not exist $\tau'' \in T$ with $\tau \subset \tau'' \subset \tau'$. The strings of level l in T are those strings in T which have precisely l proper initial segments in T. We say that $\tau \in T$ is a leaf of T if there does not exist $\tau' \supset \tau$ in T. We say that finite and non-empty $T \subseteq 2^{<\omega}$ is of level l if T has a single element of level 0 and all leaves of T are of level l. T is said to be 2-branching if T has a single element of level 0 and each $\tau \in T$ has precisely two successors in T. We say that finite T is 2-branching to level l if T is of level l and every string in T which is not a leaf has precisely two successors. We let $[T]$ denote the set of infinite paths through T. We let ϕ_i denote the i^{th} partial computable function and $\phi_i[s]$ denotes the longest string τ such that for all $n < |\tau|$, $\phi_i(n) \downarrow = \tau(n)$ in less than s steps. Recall that A is of hyperimmune degree iff it computes a total function f which is not dominated by any total computable function, i.e. such that for any total computable g there exist an infinite number of n with $f(n) \geq g(n)$.

2. The proof

Clearly it suffices to show that whenever $T \subseteq 2^{<\omega}$ is 2-branching and non-computable, T computes some $A \in [T]$ which is non-computable. Our proof will involve multiple levels of non-uniformity. We consider first the easiest case, that T is of hyperimmune degree.

Definition 2.1. The **compatibility sequence** for $\tau \in 2^{<\omega}$ with respect to the finite sequence of strings $\langle \tau_0, .., \tau_s \rangle$ is the string σ of length $s+1$ such that for all $n \leq s$, $\sigma(n) = 0$ if τ is incompatible with τ_n, and $\sigma(n) = 1$ otherwise. We consider the compatibility sequences to be ordered lexicographically.

Lemma 2.2. *If a 2-branching $T \subseteq 2^{<\omega}$ is of hyperimmune degree then T computes a non-computable $A \in [T]$.*

Proof. Let $f : \omega \rightarrow \omega$ be an increasing function computable in T and which is not dominated by any computable function. We define a sequence of strings $\{\tau_s\}_{s \in \omega}$ such that $A = \bigcup_s \tau_s$.

Stage 0. Define τ_0 to be the string of level 0 in T.

Stage $s+1$. We have defined already τ_s, which is of level s in T. Let m be the length of the longest string extending τ_s in T of level $s+2$. Define τ_{s+1} to be a successor of τ_s in T with minimum possible compatibility sequence with respect to $\langle \phi_0[f(m)], .., \phi_s[f(m)] \rangle$.

Now suppose that A is computable. Let i be the least such that $A = \phi_i$ and let $s > i$ be large enough such that, for all $i' < i$, either $\phi_{i'}$ is incompatible with τ_s or $\phi_{i'}[s']$ is an initial segment of τ_s for all s'. For all n let $g(n)$ be the least t such that $\phi_i[t]$ is of length $\geq n$. Let $n > |\tau_{s+1}|$ be such that $f(n) \geq g(n)$. Let s' be the greatest such that $|\tau_{s'}| < n$. Since $A = \phi_i$ we must have that $\tau_{s'-1}$ is compatible with ϕ_i. Since $f(n) \geq g(n)$, if m is the length of the longest string of level $s'+1$ in T extending $\tau_{s'-1}$, then $f(m) \geq g(n)$. Thus $f(m) \geq g(|\tau_{s'}|)$, so that $|\phi_i[f(m)]| \geq |\tau_{s'}|$. As $\phi_{i'}$, for $i' < i$, cannot be incompatible with an extension of τ_s without being incompatible with τ_s and the two extensions of $\tau_{s'-1}$ on T must disagree on some argument less than $|\tau_{s'}|$, this means we must define $\tau_{s'}$ to be incompatible with ϕ_i, which gives the required contradiction. □

Definition 2.3. We say that $T \subseteq 2^{<\omega}$ is **f-compatible** if:

- for every l, all strings in T of level l are of length $f(l)$, and
- T is either 2-branching, or finite and 2-branching to some level.

Lemma 2.4. *If a 2-branching T is of hyperimmune-free degree then there exists some computable function f and some 2-branching $T' \leq_T T$ such that T' is f-compatible and $[T'] \subseteq [T]$.*

Proof. For every l, let $g(l)$ be the length of the longest string in T of level l. Let h be a computable and increasing function which majorizes g (i.e. such that $h(n) \geq g(n)$ for all n). Define $f(0) = g(0)$ and $f(1) = h(1)$ and for all $l \geq 1$ define $f(l+1) = h(f(l)+1)$. Since for any $A \in [T]$ and any l there exist at least two incompatible strings in T extending $A \upharpoonright f(l)$ and of length at most $f(l+1)$, it is clear that T' exists as required. □

Lemma 2.5. *If a 2-branching T is of non-zero hyperimmune-free degree then T computes one of its non-computable paths.*

Proof. By lemma 2.4, and since any non-computable set computes a non-computable path through any perfect computable T', we may suppose that T is f-compatible for some computable f. For the sake of simplicity, let us assume also that the string of level 0 in T is λ. For all n, let $T[n]$ be the set of strings which are of level $\leq n$ in T. The basic form of the proof is as follows. We shall begin by defining a Turing functional Φ such that whenever $\Phi(\tau; n)\downarrow$, it is equal to some finite f-compatible T' of level n. The functional Φ should not be regarded as computing infinite trees, however, since for $n' > n$ we shall not necessarily have that $\Phi(\tau; n)$ is a subset of $\Phi(\tau; n')$ when these two values are defined.

Having defined Φ we shall then show that either T computes one of its non-computable paths, or there exists $A \leq_T T$ in $[T]$ such that for infinitely many n, $\Phi(A; n) = T[n]$. Lastly we shall show that if A is computable then this actually suffices to ensure that there exists $B \in [T]$ which is of the same degree as T.

So let us begin, then, by defining Φ. The key here is to take advantage of the fact that we only require $\Phi(A)$ to be total when A is computable. Suppose that this condition holds and that we subsequently define some $A \leq_T T$ in $[T]$ and then find that $\Phi(A)$ is partial. Then A must be non-computable and so the statement of the lemma holds.

For any l there exist a finite number of finite T' which are f-compatible and of level l. We let $T(l,k)$ denote the k^{th} such T' (for $k \geq 1$). Initially we define $\Phi(\lambda; 0) = \{\lambda\}$. In order to enumerate axioms for Φ on arguments > 0 we shall run a finite number of *modules* at each stage of the construction. Let us begin by considering the (1)-*module above* λ. The role of this module is to enumerate axioms for Φ on argument 1 and on strings extending λ. The activity of this module simply consists of running a finite number of *submodules* at each stage. At stage s of the construction the (1)-module above λ runs stage s of each $(1,i)$-submodule above λ

for $i \leq s$ in turn.

In order to describe the instructions for the $(1, i)$-submodule above λ we consider first a simplified version of this submodule, which does not actually enumerate any axioms for Φ but only *axiom instructions*. We shall explain what these axiom instructions really mean subsequently. Basically the point is that the axiom instructions we enumerate will look rather like axioms for Φ but may not be consistent if regarded simply as axioms. From the axiom instructions, however, we shall be able to enumerate actual axioms for Φ which *are* consistent and which achieve as much as we need them to. We shall refer to an axiom instruction of the form $\Phi(\tau; n) = T'$ (for some finite T') as an axiom instruction on τ for Φ on argument n.

The instructions for the $(1, i)$-submodule above λ at stage s (simplified version). Let k be the number of finite T' which are f-compatible and of level 1. Let $l = (i + 1)k$.

(1) Check to see whether there exists τ of length $f(l)$ such that $\phi_i[s] \supseteq \tau$ and such that no axiom instruction for Φ on argument 1 has previously been enumerated (by any submodule) on any initial segment of τ. If not then do nothing. Otherwise, proceed to the next step.
(2) We say that the submodule *acts on* τ. For each j with $1 \leq j \leq k$, let τ_j be the initial segment of τ of length $f(l - j + 1)$. Perform the following for each j with $1 \leq j \leq k$. For all τ' of length $f(l)$ extending τ_j, enumerate the axiom instruction $\Phi(\tau'; 1) = T(1, j)$ unless $\tau' \supseteq \tau_{j'}$ for some $j' < j$. Once step 2 is completed the submodule never subsequently acts again.

In order to understand how all the $(1, i)$-submodules above λ interact, let us make some simple observations. If the $(1, i)$-submodule acts on τ, T_0 is f-compatible and 2-branching and $\tau \in T_0$, then there exists some $\tau' \in T_0$ for which we enumerate the axiom instruction $\Phi(\tau'; 1) = T_0[1]$. The $(1, i)$-submodule acting on τ at stage s may prevent a $(1, i')$-submodule acting at any subsequent stage if $i' > i$ (even if $\phi_{i'}$ is incompatible with τ), but this is not the case for $i' < i$. A $(1, i')$-submodule for $i' < i$ may subsequently act, even on $\tau' \subset \tau$, so that the axiom instructions we enumerate may well be inconsistent if regarded as a set of axioms for Φ. The crucial point, however, is just this: the levels at which we enumerate axiom instructions are sufficiently far apart that if a $(1, i')$-submodule does subsequently act on $\tau' \subset \tau$ and enumerates an axiom instruction $\Phi(\tau''; 1) = T'$ for some finite T' and some τ'' which is a string in some 2-branching f-compatible T_1, then there *does* exist an infinite path on T_1 extending τ'' and on which we have not yet enumerated any axiom instructions for Φ on argument 1. We can therefore choose an initial segment of this string of sufficient length and enumerate an actual axiom for Φ on this initial segment. This is all we really require, that there should exist *some* string in T_1 extending τ'' on which we can enumerate the axiom.

When we enumerate any axiom instruction $\Phi(\tau; 1) = T'$, then, we also perform the following. We choose some l' which is larger than any number previously mentioned during the course of the construction and for each $\tau' \supset \tau$ of length $f(l')$ such that we have not previously enumerated any axiom instruction for Φ on argument 1 on any initial segment of τ', we enumerate the actual axiom $\Phi(\tau'; 1) = T'$. Now suppose the $(1, i)$-submodule above λ enumerates an axiom instruction on

τ. If $i' > i$ then there exists some τ' of length $f((i'+1)k - k + 1) = f(i'k + 1)$ such that all axiom instructions enumerated by the $(1, i')$-submodule above λ are on strings extending τ'. Let l' be large. Since τ is of length $f((i+1)k)$, every 2-branching f-compatible T_1 containing τ also contains an extension of τ of length $f(l')$, on no initial segment of which any $(1, i')$-submodule has enumerated an axiom instruction.

We are now ready to give the full construction for defining Φ.

The instructions for the (n, i)-submodule above σ at stage s. Let k be the number of finite T' which are f-compatible and of level n. Let $|\sigma| = f(l')$ and let $l = l' + (i+1)k$.

(1) Check to see whether there exists τ of length $f(l)$ such that $\phi_i[s] \supseteq \tau$ and such that no axiom instruction for Φ on argument n has previously been enumerated (by any submodule) on any initial segment of τ. If not then do nothing. Otherwise, proceed to the next step.
(2) For each j with $1 \leq j \leq k$, let τ_j be the initial segment of τ of length $f(l - j + 1)$. Perform the following for each j with $1 \leq j \leq k$. For all τ' of length $f(l)$ extending τ_j, enumerate the axiom instruction $\Phi(\tau'; n) = T(n, j)$ unless $\tau' \supseteq \tau_{j'}$ for some $j' < j$. Proceed to the next step.
(3) For each axiom instruction $\Phi(\tau'; n) = T'$ which was enumerated during step 2, choose some l'' which is larger than any number previously mentioned during the course of the construction and for each $\tau'' \supset \tau'$ of length $f(l'')$ such that we have not previously enumerated any axiom instruction for Φ on argument n on any initial segment of τ'', enumerate the (actual) axiom $\Phi(\tau''; n) = T'$.

The instructions for the (n)-module above σ at stage s. Run stage s of the (n, i)-submodule above σ for each $i \leq s$ in turn.

The construction.
Stage 0. Enumerate the axiom $\Phi(\lambda; 0) = \{\lambda\}$.
Stage $s > 0$. For each σ such that we have enumerated an axiom $\Phi(\sigma; n) = T'$ at a previous stage, run the $(n+1)$-module above σ.

The next thing we do is to attempt to define $A \in [T]$ such that $A \leq_T T$ and such that for infinitely many n, $\Phi(A; n) = T[n]$. In order to do so we define a sequence of finite strings $\{\sigma_m\}$. If this sequence turns out to be finite then we shall be able to show that T computes some $B \in [T]$ which is non-computable. Initially we define $\sigma_0 = \lambda$. Suppose we have defined σ_m. Run the construction for defining Φ until an axiom is enumerated $\Phi(\tau; n) = T[n]$ for any n and for some $\tau \in T$ properly extending σ_m. When and if such an axiom is enumerated, define $\sigma_{m+1} = \tau$.

Let us consider first the possibility that there exists m such that σ_m is defined but σ_{m+1} is not. In this case B which is the leftmost path on T extending σ_m must be non-computable. Towards a contradiction suppose otherwise. Let i be the least such that $B = \phi_i$. Let σ be an initial segment of B extending σ_m, such that an axiom for Φ is enumerated on σ and which is long enough such that no (n, i')-submodule with $i' < i$ (and any n) enumerates an axiom or an axiom instruction on any extension of σ. Let n' be the least such that $\Phi(\sigma; n')$ is undefined. The

(n', i)-submodule above σ will act on an initial segment of B and will enumerate an axiom instruction $\Phi(\tau; n') = T[n']$ for some $\tau \in T$ extending σ. This same submodule will then enumerate an axiom $\Phi(\tau'; n') = T[n']$ for some $\tau' \in T$ extending τ and so σ_{m+1} will be defined.

Next suppose that σ_m is defined for every m and define $A = \bigcup_m \sigma_m$. If A is non-computable then the statement of the lemma holds, so suppose otherwise. Then there exists a computable function g such that:

- for every n, $g(n)$ is some finite f-compatible T_n (say) of level n;
- for infinitely many n we have $T_n = T[n]$.

We define two computable functionals Ψ_0 and Ψ_1 such that $\Psi_0(T) = C$, $\Psi_1(C) = T$ and C is a path on T. We define these two functionals by enumerating axioms in stages. At each stage n we shall enumerate a single axiom $\Psi_0(T_n) = \tau$ for some τ of level n in T_n and also the axiom $\Psi_1(\tau) = T_n$. For any 2-branching f-compatible T', $\Psi_0(T')$ is defined to be the union of all τ such that we enumerate an axiom of the form $\Psi_0(T'[n]) = \tau$.

Stage 0. Define $\Psi_0(\{\lambda\}) = \lambda$ and $\Psi_1(\lambda) = \{\lambda\}$.
Stage $n > 0$. Let $T' \subset T_n$ be the largest such that we have enumerated some axiom $\Psi_0(T') = \tau_0$. Choose some string τ_1 of level n in T_n extending τ_0 such that we have not enumerated any axiom for Ψ_1 on any string τ_2 such that $\tau_0 \subset \tau_2 \subseteq \tau_1$. This must be possible since at each stage n' of the construction we only enumerate one axiom for Ψ_1 and this is on a string of length $f(n')$. Define $\Psi_0(T_n) = \tau_1$ and $\Psi_1(\tau_1) = T_n$.

This completes the proof of lemma 2.5.

□

2.1. **Thanks.** The author would like to thank Ted Slaman for his interest, encouragement and for many helpful conversations, and also the anonymous referee for many helpful suggestions.

References

BC. S.B. Cooper, *Computability theory*, Chapman & Hall, CRC Press, Boca Raton, FL, New York, London (2004).

HF. H. Friedman, 102 problems in mathematical logic, *Journal of Symbolic Logic*, volume 40, (1975), pp 113-129.

AL. A.E.M. Lewis, Π^0_1 classes, strong minimal covers and hyperimmune-free degrees, to appear in the *Bulletin of the London Mathematical Society.*

AL2. ______, A random degree with strong minimal cover, to appear in the *Bulletin of the London Mathematical Society.*

AL3. ______, Strong minimal covers and a question of Yates: the story so far, to appear in the proceedings of the ASL meeting 2006.

SG. T. Slaman and M. Groszek, A basis theorem for perfect sets, *Bulletin of Symbolic Logic*, volume 4, number 2 (1998).

RS. R.I. Soare, *Recursively enumerable sets and degrees*, Springer, New York, (1987).

Dipartimento di Scienze Matematiche ed Informatiche, Pian dei Mantellini 44, 53100 Siena, Italy

E-mail address: andy@aemlewis.co.uk