SCHNORR TRIVIALITY AND GENERICITY

JOHANNA N.Y. FRANKLIN
DEPARTMENT OF MATHEMATICS
NATIONAL UNIVERSITY OF SINGAPORE
2, SCIENCE DRIVE 2
SINGAPORE 117543

Abstract. We study the connection between Schnorr triviality and genericity. We show that while no 2-generic is Turing equivalent to a Schnorr trivial and no 1-generic is tt-equivalent to a Schnorr trivial, there is a 1-generic that is Turing equivalent to a Schnorr trivial. However, every such 1-generic must be high. As a corollary, we prove that not all K-trivials are Schnorr trivial. We also use these techniques to extend a previous result and show that the bases of cones of Schnorr trivial Turing degrees are precisely those whose jumps are at least $\mathbf{0}''$.

Contents

1. Introduction

In this paper, we relate algorithmic randomness to more classical recursion theory. We study a particular class of reals that have low initial-segment complexity in the context of Schnorr randomness, the Schnorr trivial reals. Unlike the Turing degrees that contain K-trivial reals, which have low initial-segment complexity in the context of Martin-Löf randomness, the Turing degrees that contain Schnorr trivial reals do not seem to have a neat characterization. For instance, each Turing degree whose jump is at least $\mathbf{0}''$ contains a Schnorr trivial real [6], and they are not closed downward under Turing reductions [2]. This suggests that it would be informative to consider other properties of reals and determine the extent to which they are compatible with Schnorr triviality.

Here, we study the relationship between genericity and Schnorr triviality. We find that the two are largely incompatible. We prove that no real can be both Schnorr trivial and 1-generic. At the level of degrees, we show that no Schnorr trivial real is tt-equivalent to a 1-generic or Turing equivalent to a 2-generic. However, it is possible for a Schnorr trivial real to be Turing equivalent to a 1-generic. To show that a real is K-trivial, we only need to consider prefix-free Turing machines. On the other hand, to show that a real is Schnorr trivial, we need to exhibit a Turing machine that is not only prefix free but computable as well. To ensure

Date: October 25, 2009.
This material comprises part of the author's Ph.D. thesis which was written under the supervision of Theodore A. Slaman at the University of California, Berkeley. The author acknowledges the support of NSF grant DMS-0501167 and thanks her advisor and Leo Harrington for many insightful conversations and the editor and referees for valuable comments.

that we can produce a computable machine to witness Schnorr triviality, we must either have a stronger reduction than Turing reducibility, a more powerful generic, or additional information about the generic in question.

These results have two primary consequences. The technique used in the 2-generic case allows us to show that the primary theorem in [6] is sharp; i.e., that if $\mathbf{a}' \not\geq_T \mathbf{0}''$, $\mathbf{a}$ is not the base of a cone of Schnorr trivial degrees. Secondly, since no Schnorr trivial is 1-generic, we can easily see that there are K-trivial reals that are not Schnorr trivial.

1.1. **Terminology and Definitions.** Most of the notation is standard and follows Soare [13]. We will call the elements of 2^ω reals and consider Turing machines to be partial recursive functions from $2^{<\omega}$ to $2^{<\omega}$. Any Turing machine we consider will be prefix free; that is, its domain must be a prefix-free set. The elements of $2^{<\omega}$ that extend a particular $\tau \in 2^{<\omega}$ will be denoted as $[\tau]$, and we will define $[S]$ similarly for any $S \subseteq 2^{<\omega}$.

We will use μ to denote Lesbesgue measure throughout the paper. We will often consider the measure of the domain of a Turing machine, but never the range. Therefore, we will simply write $\mu(M)$ for $\mu(\text{dom}(M))$. It should be noted that if M is a prefix-free Turing machine and we list the elements of the graph of M as $\langle \tau_i, \sigma_i \rangle$, we can see that $\mu(M) = \sum_i \frac{1}{2^{|\tau_i|}}$.

As in [3], we will use K to denote prefix-free Kolmogorov complexity. In this paper, we will not consider prefix-free Kolmogorov complexity with respect to a universal machine, but Kolmogorov complexity with respect to some other particular Turing machine. This will make the following notation necessary.

Definition 1.1. Let M be a Turing machine, and let $\sigma \in 2^{<\omega}$. The *prefix-free Kolmogorov complexity of σ with respect to M* is $K_M(\sigma) = \min\{|\tau| \mid M(\tau) = \sigma\}$.

It is clear that the measure of each Turing machine's domain is a recursively enumerable real; i.e., effectively approximable from below. However, we will consider only the Turing machines that satisfy the following condition.

Definition 1.2. A Turing machine M is *computable* if the measure of its domain is a recursive real.

Now we present the standard definition for a Schnorr random real, which is measure-theoretic.

Definition 1.3. [12] A *Schnorr test* is a uniformly r.e. sequence $\langle V_i \rangle_{i\in\omega}$ of Σ^0_1 classes such that $\mu(V_i) = \frac{1}{2^i}$ for each i. A real A is *Schnorr random* if for all Schnorr tests $\langle V_i \rangle_{i\in\omega}$, $A \notin \cap_{i\in\omega} V_i$.

However, Downey and Griffiths used the notion of a computable Turing machine to develop an initial-segment complexity definition as well.

Theorem 1.4. [5] *A real A is Schnorr random if and only if*

$$(\forall M \ \textit{comp.})(\exists c \in \omega)(\forall n \in \omega)[K_M(A\restriction n) \geq n - c].$$

Later, Downey, Griffiths, and Laforte developed the following characterization of Schnorr triviality in [2]. They began by defining a notion of relative initial-segment complexity for Schnorr randomness.

Definition 1.5. We say that $A \leq_{Sch} B$ if for every computable Turing machine M, there is a computable Turing machine M' and a constant $c \in \omega$ such that $(\forall n \in \omega)[K_{M'}(A\restriction n) \leq K_M(B\restriction n) + c]$.

A real is said to be *trivial* for a particular randomness notion if its initial-segment complexity is no more than a recursive real's relative to that randomness notion's definition of initial-segment complexity. This enabled Downey, Griffiths, and Laforte to make the following definition in [2].

Definition 1.6. A real A is *Schnorr trivial* ($A \leq_{Sch} 0^\omega$) if $A \leq_{Sch} 0^\omega$; i.e., if the following statement holds.

$$(\forall M \text{ comp.})(\exists M' \text{ comp.})(\exists c \in \omega)(\forall n \in \omega)[K_{M'}(A\restriction n) \leq K_M(0^n) + c]$$

In the course of this paper, we will need to construct computable Turing machines. This will be simplified by the following theorem from [1].

Theorem 1.7 (Kraft-Chaitin Theorem)**.** *Let $\langle d_i, \sigma_i \rangle_{i \in \omega}$ be a recursive sequence with $d_i \in \omega$ and $\sigma_i \in 2^{<\omega}$ for all i such that $\sum_i \frac{1}{2^{d_i}} \leq 1$. (Such a sequence is called a* Kraft-Chaitin set*, and each element of the sequence is called a* Kraft-Chaitin axiom*.) Then there are strings τ_i and a prefix-free machine M such that $dom(M) = \{\tau_i \mid i \in \omega\}$ and for all i and j in ω,*

(1) *if $i \neq j$, then $\tau_i \neq \tau_j$,*
(2) *$|\tau_i| = d_i$,*
(3) *and $M(\tau_i) = \sigma_i$.*

The Kraft-Chaitin Theorem allows us to construct a prefix-free machine by specifying only the lengths of the strings in the domain rather than the strings themselves. We will therefore identify $\langle \tau, \sigma \rangle$ with $\langle d, \sigma \rangle$, where $d = |\tau|$, throughout.

2. The 2-generic case

Theorem 2.1. *Suppose* **a** *contains a 2-generic real. Then* **a** *does not contain a Schnorr trivial real.*

Proof. Let $G \in \mathbf{a}$ be a 2-generic real, and suppose that A is an arbitrary element of **a**. We will show that A cannot be Schnorr trivial.

Since $A \equiv_T G$, there are Turing functionals Φ and Ψ such that $\Phi(A) = G$ and $\Psi(G) = A$. We can write "Φ and Ψ are total, and $\Phi \circ \Psi$ is the identity function" as the following Π^0_2 statement.

$$\varphi := (\forall n \in \omega)\,(\exists s \in \omega)\,[\Phi_s(n)\!\downarrow \wedge \Psi_s(n)\!\downarrow] \wedge (\forall n \in \omega)\,(\exists s \in \omega)\,[(\Phi \circ \Psi)_s(n) = n]$$

Therefore, since G is 2-generic, there is an initial segment of G that forces φ. We call this initial segment p and consider the set of forcing conditions $\mathcal{P} = \{q \in 2^{<\omega} \mid q \supseteq p\}$, which we will write as $\langle q_i \rangle_{i \in \omega}$. We will order these conditions in the standard way by writing $q \succeq r$ if and only if $q \subseteq r$. Now we may define the set $T = \{\Psi(q) \mid q \preceq p\}$.

The set of initial segments of the elements of T forms a tree in $2^{<\omega}$. Without loss of generality, we will identify T with this tree. We note that this tree is perfect. If this were not the case, T would have an isolated branch, which would necessarily be recursive. However, this is not possible, since we have forced every branch in T to be Turing equivalent to a nonrecursive real. We also see that this tree is recursively enumerable, since Ψ is a Turing functional and $\mathcal{P}$ is a recursive set of finite binary strings.

To see that A cannot be Schnorr trivial, we must construct a computable machine M such that the following holds.

$$(\forall M_e \text{ comp.})\,(\forall c \in \omega)\,(\forall q \preceq p)\,(\exists r \preceq q)\,(\exists n \in \omega)\,[K_{M_e}(\Psi(r)\restriction n) > K_M(0^n) + c]$$

If this is the case, if we are given an M_e and c and a forcing condition p, no matter how we extend p, we can find an extension r of that extension such that the Σ^0_1 statement "$(\exists n \in \omega)\,[K_{M_e}(\Psi(r)\restriction n) > K_M(0^n) + c]$" will hold. Since G is 2-generic, we can force this statement to be true, so each branch of T will be non-Schnorr trivial with respect to each M_e and c. Since A is a branch in T, this will be enough to show that A is not Schnorr trivial.

We enumerate the pairs $\langle e, c \rangle$ such that $(\exists s \in \omega)\left[\mu(M_{e,s}) \geq 1 - \frac{1}{2^c}\right]$. If M_e is a computable machine with domain 1, $\langle e, c \rangle$ will be enumerated for all c. Since such M_e exist, this list is infinite, and we can naturally write it as $\langle e_m, c_m \rangle_{m \in \omega}$.

We begin by considering an arbitrary pair $\langle e_m, c_m \rangle$. As we build our machine M, we will allot $\frac{1}{2^{m+1}}$ of its measure to ensure that every possible $\Psi(q_i)$, where $q_i \in \mathcal{P}$, has an extension that is not Schnorr trivial with respect to M_{e_m} and c_m.

Consider a particular element q_i of $\mathcal{P}$. We will dedicate $\frac{1}{2^{i+1}}$ of the measure that we have allotted to $\langle e_m, c_m \rangle$ to ensuring that we can extend $\Psi(q_i)$ in such a way as to remove the possibility of forcing Schnorr triviality with respect to M_{e_m} and the constant c_m, for a total measure of $\frac{1}{2^{(m+1)+(i+1)}}$. Therefore, we will add $\langle (m+1) + (i+1), 0^n \rangle$ to M for some n.

For simplicity, we will now write $\langle e_m, c_m\rangle$ as $\langle e, c\rangle$. We know that $\mu(M'_e) \geq 1 - \frac{1}{2^c}$, so no more than $\frac{1}{2^c}$ additional measure can be assigned to $\mu(M'_e)$ after some stage s. This allows us to define the following values.

$$\begin{aligned} s_{e,c} &= \min\left\{s \mid \mu(M_{e,s}) \geq 1 - \frac{1}{2^c}\right\} \\ n_{e,c} &= \max\{|\sigma| \mid \sigma \in M_{e,s_c}\} \end{aligned}$$

We have defined $s_{e,c}$ to be the least stage s at which $\mu(M'_{e,s}) \geq 1 - \frac{1}{2^c}$ and $n_{e,c}$ to be the length of the longest string in the range of $M'_{e,s_{e,c}}$.

We find a height h such that there are at least $2^{(m+1)+(i+1)} + 1$ distinct branches of length h extending $\Psi(q_i)$ in T. This is possible because T is a perfect tree. Now we observe that since we have more than $2^{(m+1)+(i+1)}$ branches, the least measure that M'_e assigns an extension of $\Psi(q_i)$ of length at least h after stage $s_{e,c}$ is strictly less than $\frac{1}{2^{(m+1)+(i+1)}} \cdot \frac{1}{2^c} = \frac{1}{2^{(m+1)+(i+1)+c}}$. Therefore, M'_e must assign at least one such extension a complexity strictly greater than $(m+1)+(i+1)+c$. We let n be the least integer such that the following three conditions hold.

(1) $n > n_{e,c}$.
(2) $n \geq h$.
(3) At this stage in the construction of M, there is no d such that $\langle d, 0^n\rangle \in M$.

We now add $\langle (m+1)+(i+1), 0^n\rangle$ to M. Since $K_M(0^n) = (m+1)+(i+1)$, we can see that there is some $r_i \preceq q_i$ such that for this n, we have the following.

$$K_{M'_e}(\Psi(r_i)\restriction n) > (m+1)+(i+1)+c = K_M(0^n)+c$$

This means that q_i cannot force $K_{M'_e}(\Psi(r_i)\restriction n)$ to be less than or equal to $K_M(0^n)+c$, so q_i must force this inequality to hold instead.

We construct M by repeating the above procedure for each pair $\langle m, i\rangle$ in turn. For each $\langle m, i\rangle$, a new axiom $\langle (m+1)+(i+1), 0^n\rangle$ will enter M, so we can calculate the measure of M as follows.

$$\mu(M) = \sum_{m\in\omega}\sum_{i\in\omega} \frac{1}{2^{(m+1)+(i+1)}} = \sum_{m\in\omega} \frac{1}{2^{m+1}} = 1,$$

Since this procedure is recursively enumerable and $\mu(M) = 1$, we can see by the Kraft-Chaitin Theorem that M is a computable Turing machine.

Since A is a branch through T, the existence of M shows that we cannot force it to be Schnorr trivial with respect to any computable Turing machine with domain 1 and any constant. Therefore, it must not be Schnorr trivial with respect to any computable Turing machine with domain 1 and any constant. By a result of Downey and Griffiths [5], it is enough to consider Turing machines of domain 1 when determining the Schnorr triviality of a real, so A cannot be Schnorr trivial. Finally, since we chose A to be an arbitrary element in $\mathbf{a}$, we can see that no element of $\mathbf{a}$ is Schnorr trivial.

□

3. The 1-generic case

We now turn our attention to 1-generic reals. In this case, we will consider different reducibilities. We recall that $A \leq_{tt} B$ if there is a Φ such that $\Phi^B = A$ and Φ is total for every oracle.

We may use the same technique as we did in the proof of Theorem 2.1 to show the following.

Theorem 3.1. *Suppose that G is a 1-generic real and that $A \equiv_{tt} G$. Then A is not Schnorr trivial.*

Proof. Let Ψ be a tt-functional witnessing $A \leq_{tt} G$. Since Ψ is a tt-functional rather than simply a Turing functional, we are able to construct a recursively enumerable perfect tree containing $\Psi(G) = A$ as before such that G cannot force all of the extensions of any point in the tree to be Schnorr trivial. Therefore, A must not be Schnorr trivial. □

This gives us the following corollary.

Corollary 3.2. *No 1-generic real is Schnorr trivial.*

However, this proof cannot be extended to show that there is no Schnorr trivial real in the Turing degree of any 1-generic. In the proof of Theorem 2.1, we can use an initial segment of our 2-generic G to force the totality of the functionals Ψ and Φ. However, 1-genericity is not sufficient for a real to force totality. At best, a 1-generic real could force $\Phi \circ \Psi$ to be equal to the identity function at every point at which it converged. This is not sufficient, since if the tree T is not perfect, we may not be able to repeat the procedure in the proof of Theorem 2.1 for all $\langle m, i\rangle$ because there may not be enough branches above some q_i. Therefore, the resulting machine M may not be computable. The following theorem shows that Theorem 3.1 cannot be generalized to Turing reducibility.

Theorem 3.3. *There is a 1-generic real G and a real $X \equiv_T G$ such that X is Schnorr trivial.*

To see this, it is enough to note that there is a high 1-generic real, since by Theorem 9 in [6], there must be a Schnorr trivial real in every high degree. However, we also present a direct construction that illustrates the relationship between 1-genericity and Schnorr triviality clearly.

Proof. We will construct a 1-generic G and two Turing functionals Φ and Φ^{-1} such that Φ^G is Schnorr trivial and Φ^{-1} witnesses that $G \leq_T \Phi^G$. The 1-generic G will be recursive in $\mathbf{0}''$, and Φ and Φ^{-1}, of course, will both be recursively enumerable. We will construct G to be a branch of a subtree of $2^{<\omega}$. Naturally, we will have to construct machines M'_e for each e to witness the Schnorr triviality of $\Phi(G)$.

There are two different types of requirements that we will need to meet. We will need $\Phi(G)$ to be Schnorr trivial with respect to each computable Turing machine M_e with domain 1, and we will need G to meet every dense Σ^0_1 set. We will enumerate the Σ^0_1 sets as $\langle W_e\rangle_{e\in\omega}$ and express these requirements as follows.

Sch_e: $\Phi(G)$ is Schnorr trivial with respect to M_e.

Gen_e: $(\exists \sigma \subset G)\,[\sigma \in W_e \text{ or } (\forall \tau \supset \sigma)\,[\tau \notin W_e]]$.

These requirements will be ordered as $Sch_0 < Gen_0 < Sch_1 < \ldots < Sch_n < Gen_n < Sch_{n+1} < \ldots$.

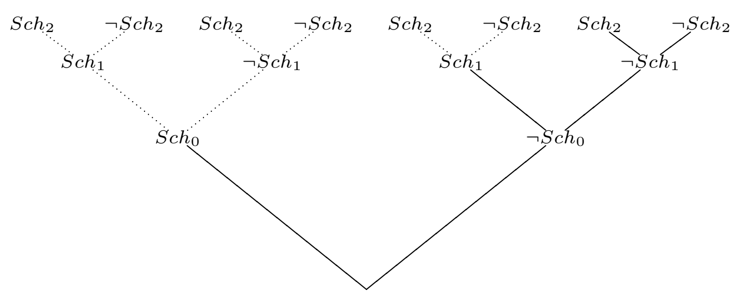
We will construct a tree T and a "companion" tree $\Phi(T)$ in stages so that every infinite branch B of T will be 1-generic and for some such B, the corresponding branch $\Phi(B)$ of $\Phi(T)$ will be Schnorr trivial. We will represent the finite trees at the end of stage s as T_s and $\Phi(T_s)$ as usual. Certain strings in T will be dedicated to satisfying the requirements of the form Sch_e and others to satisfying the Gen_e requirements.

If a string σ is dedicated to satisfying requirement Sch_e, we will ensure that every infinite path in $\Phi(T)$ through $\Phi(\sigma)$ is Schnorr trivial with respect to M_e if M_e is a computable Turing machine with domain 1. To do this, we will begin by ensuring that we will only extend a string in T extending σ (and thus the corresponding element in $\Phi(T)$) if we have new evidence that M_e is computable with domain 1; i.e., if $\mu(M_e) \geq 1 - \frac{1}{2^c}$ for some c greater than one we used the last time we extended σ in T. When we extend the corresponding string $\Phi(\sigma)$ in $\Phi(T)$, we extend it to be long enough that we can be sure that we can build a computable machine M'_e witnessing its Schnorr triviality. This will be enough, since when we build a tree of Schnorr trivial reals, as in [6], the only concern is the minimum acceptable branching heights in the tree. If we can extend this string σ infinitely often, we will have infinite branches in T and $\Phi(T)$ extending σ and $\Phi(\sigma)$, respectively. If not, M_e is not computable with domain 1 and there will be no infinite path in T through σ. Therefore, we will only satisfy Sch_e when M_e is actually computable with domain 1.

However, the requirements Sch_e will often interact. For instance, a string σ dedicated to satisfying Sch_e may be a substring of another string τ dedicated to satisfying Sch_{e+1}. If M_e is not computable with domain 1, then at some point, we will no longer extend any string extending σ. This includes τ, whether M_{e+1} is computable or not. Therefore, even if M_e is a computable machine with domain 1, it is possible that in the course of the construction, not every string dedicated to satisfying Sch_e will succeed. We deal with this problem by constructing our tree T so that not every branch contains a string dedicated to satisfying each Sch_e.

Our method of satisfying the Sch_e requirements is illustrated in Figure 1. We label strings which are dedicated to satisfying requirement Sch_i with Sch_i. Strings on the same level as a string labeled with Sch_i that are not dedicated to satisfying Sch_i are labeled with $\neg Sch_i$ for clarity, although no such label will be attached in the actual construction. If the measure of a machine M_e is large enough, we will branch above strings dedicated to it; otherwise, we will not. This is illustrated by the dotted branches in the figure. For instance, if $\mu(M_0) < 1 - \frac{1}{2^1}$, the tree will never be extended about the string labeled with Sch_0. However, the rightmost branch of the tree will always be extended since it contains no substrings dedicated to satisfying a requirement of the form Sch_i, so it is indicated by a solid line. We can see from this that if $Z \subseteq \omega$ is a set of indices of machines with domain 1, there will be an infinite branch through T with strings dedicated to satisfying Sch_i for $i \in Z$.

FIGURE 1. Satisfying the Sch_e requirements



We will satisfy each genericity requirement along each infinite path in T. Each time we branch in T in an attempt to satisfy some Sch_e, we dedicate a string on each new branch to satisfying Gen_i, where i is the least integer such that no string on that branch has previously been dedicated to satisfying Gen_i. If that string is never seen to be comparable to an element of W_i, then Gen_i is met trivially. Otherwise, when we find such an element of W_i, we will prune the tree to ensure that every extension of that string in T passes through that element and thus meets W_i. This has no effect on $\Phi(T)$ except to prune it slightly. However, it may require us to start over again with lower-priority requirements along that path through T. If every string dedicated to a requirement of the form Sch_e on that branch of the tree is dedicated to a requirement for a computable machine with domain 1, this will always be possible; otherwise, the branch will be finite and irrelevant to our analysis of the final tree anyway.

We will label a string with m_e if we wish every string extending it to satisfy Sch_e, and we will label a string with g_e if we intend to use it to guarantee that requirement Gen_e is met above it. We will call a string labeled with m_e a *machine string*, and a string that has been labeled with g_e will be called a *generic string*.

We will also call a string σ a *leaf* of the finite tree T_s if no extension of it is in T_s. We will name three types of leaves. The first type of leaf will be referred to as a *machine leaf*. A leaf τ of T_s will be called a machine leaf if there is some machine string $\rho \subseteq \tau$ such that ρ extends the highest branching point in T_s below τ. The second type of leaf will be called a *generic leaf*. A generic leaf τ will have some substring extending the highest branching point in T_s below τ that has been given a label at a previous point in the construction, but no such label will be of the form m_e (that is, all such labels will be of the form g_e). Finally, we may, over the course of the construction, describe some leaves as *resting* in order to meet a genericity requirement. Leaves will be described this way if the labels on their substrings extending the highest branching point below them in T_s have been removed to meet a requirement of the form Gen_e. Once a leaf is described as resting, it will not be extended in any T_t for $t \geq s$ unless a higher priority genericity requirement leads us to "awaken" it. It will be clear from the construction that every nonresting leaf is either a machine leaf or a generic leaf, but not both.

We will have, in general, several strings with the label m_e. However, no single computable Turing machine M'_e will witness the Schnorr triviality of the infinite branches extending all of them. Therefore, we will build a different Turing machine $M'_{e,\sigma}$ for each σ with the label m_e.

We say that a string σ labeled with g_e is *active at stage s* if there is some τ comparable to σ in $W_{e,s}$. The condition for a string labeled with m_e to be active is slightly more complicated. We say that a string σ labeled with m_e is active with n at stage s if the following conditions hold.

(1) σ has not yet acted with n.
(2) $\mu(M_e) \geq 1 - \frac{1}{2^n}$.

The construction proceeds as follows.

$s = 0$: We begin by taking T_0 to be the downward closure of the strings $\langle 00\rangle$ and $\langle 1\rangle$. We attach the label m_0 to $\langle 0\rangle$ and the label g_0 to $\langle 00\rangle$ and $\langle 1\rangle$. Now we may define $\Phi(\langle 0\rangle) = \Phi(\langle 00\rangle) = \langle 0\rangle$ and $\Phi(\langle 1\rangle) = \langle 1\rangle$. Φ^{-1} is defined accordingly.

$s > 0$: We consider the requirements that are active for some σ for stage s in increasing order. After handling each active requirement as described below for each string that is active at s for it, we will go on to the next requirement that has not been deactivated by one with higher priority.

Case 1: Sch_e**:** We will allow the tree above a string σ labeled with m_e to branch when $\mu(M_e)$ becomes greater than or equal to $1 - \frac{1}{2^n}$ for some n and this is permitted by the activation of all the other strings $\tau \subset \sigma$ labeled with some m_i, where $i < e$. Then we will extend and branch $\Phi(T_{s-1})$ in such a way as to ensure that the branches extending $\Phi(\sigma)$ are Schnorr trivial with respect to M_e. We will do the following for each string labeled with m_e. It will be clear from the construction that all strings labeled with m_e are incomparable, so our treatments of these requirements will not interact.

Let n be the greatest number such that σ is active with n at stage s. We consider the $\tau \supset \sigma$ that are active at stage s for a requirement of the form $Sch_{e'}$ and, for each such τ, we let n_τ be the greatest n such that τ has not yet acted with n_τ at any stage t. This allows us to determine which strings above σ will permit branching.

If there is $\rho \subseteq \sigma$ such that ρ has a label of the form m_i but is not currently active, we will not be able to extend and branch T_s above it at this point. Therefore, we will not act for σ at stage s. Otherwise, we consider all of the extensions of σ in T_s to see how many branches would be necessary for us to act. We will work downward from the non-resting leaves in T_s above σ through the branching points in T_s to determine how many leaves we will need and which leaves will result in additional branching. We will call the number of leaves needed to satisfy the branching requirements above a branching point τ k_τ. Note that there will be a generic string and a machine string immediately above each branching point (assuming that neither is resting).

Suppose that τ_b is a branching point with no other branching points above it in $[\sigma] \cap T_s$. We add 0 to k_{τ_b} for a resting leaf above τ_b, 2 for a generic leaf, 2 for a machine leaf if its machine string is active, and 1 for a machine leaf with a nonactive machine string. This indicates that the resting leaves are not extended at all, that the generic leaf and the active machine leaves must branch, and that a machine leaf with a nonactive machine string must be considered but not branched. For instance, if there is a generic leaf and an machine leaf with an active machine string above τ_b, we say that $k_{\tau_b} = 4$.

For the next highest branching point $\tau_{b'}$, we calculate $k_{\tau_{b'}}$. Suppose τ_1 is the branching point immediately above $\tau_{b'}$'s machine string and τ_2 is the branching point immediately above $\tau_{b'}$'s generic string. If $\tau_{b'}$'s machine string τ is such that $n_\tau \geq k_{\tau_1}$; we let $k_{\tau_{b'}} = k_{\tau_1} + k_{\tau_2}$ because τ is active for a large enough n to permit all the branching that we require to take place above it. Otherwise, we say that $k_{\tau_{b'}}$ is the number of nonresting leaves in T_s above τ_1 plus k_{τ_2} and say that only the leaves above $\tau_{b'}$'s generic string will branch. We repeat this procedure, working downward, for all branching points above σ. We will let $k = k_{\sigma'}$, where σ' is the branching point in T_s immediately above σ.

If $n \not\geq k$, more branching would be required above σ than σ authorizes. Therefore, we will not let σ act with n and will not adjust anything in $[\sigma] \cup T_s$ before going on to the next requirement. Otherwise, we extend Φ, Φ^{-1}, $M'_{e,\sigma}$, and T_s. We will work with Φ and M_e simultaneously to ensure that every extension of $\Phi(\sigma)$ is Schnorr trivial with respect to M_e. This part of the construction is like that in [6]. We will define $s_{e,j}$ to be the least stage s in the enumeration of M_e such that $\mu(M_e) \geq 1 - \frac{1}{2^j}$ for all $j < n$. We will refer to stages in the enumeration of M_e as M_e-stages and to stages in our construction simply as stages.

Let n' be the greatest number for which σ was previously active and for which we made adjustments to T_s. For each r between n' and n, whenever an axiom $\langle d, \rho \rangle$ enters M_e between M_e-stages $s_{e,r-1}$ and $s_{e,r}$, we add $\langle d{+}1, \Phi(\tau){\upharpoonright}|\rho|\rangle$ to $M'_{e,\sigma}$ if $|\rho| \leq |\Phi(\tau)|$ and $\langle d{+}1, \Phi(\tau)^\frown 1^{|\rho|-|\Phi(\tau)|}\rangle$ to $M'_{e,\sigma}$ if $|\rho| \leq |\Phi(\tau)|$ for all nonresting leaves τ in T_s extending σ. If there are $r' < r$ such τ, we add the same axiom $r - r'$ additional times for the leftmost such τ for a total of $r' + (r - r') = r$ new axioms. If $\Phi(\tau)^\frown 1^u$ has been added to $\Phi(T_s)$ in this process, we define $\Phi^{-1}(\Phi(\tau)^\frown 1^u) = \tau$. Now, for each i and ρ such that there is a string $\rho \supseteq \sigma$ labeled m_i that we can branch above, we will have to adjust $M'_{i,\rho}$. We will proceed as in the previous paragraph.

Now we create new branching points in T_s and $\Phi(T_s)$. For each leaf τ that we determined could branch, we add $\tau^\frown 0$, $\tau^\frown 00$ and $\tau^\frown 11$ to T_s. Let ρ be the longest string extending $\Phi(\tau)$ in $\Phi(T_s)$. Now we define $\Phi(\tau^\frown 0) = \Phi(\tau^\frown 00) = \rho^\frown 0$ and $\Phi(\tau^\frown 1) = \rho^\frown 1$. Again, we define Φ^{-1} accordingly.

Finally, we will label the elements above the branching points we just created. Suppose that the highest branching point below τ_i in T_s is such that the machine string immediately above it is labeled with $m_{i'}$. We will label $\tau_i^\frown 0$ with $m_{i'+1}$ and $\tau_i^\frown 00$ and $\tau_i^\frown 1$ with $g_{i'+1}$. At this point, we will say that σ has acted with n. Note that we will not act for any Sch_i for any $\tau \supset \sigma$ at this stage now.

Case 2: Gen_e**:** For each σ for which Gen_e is active, there is a τ comparable to σ such that $\tau \in W_e$. For each such σ, there are two subcases to consider.

In the first subcase, $\tau \subseteq \sigma$. If this is so, we simply remove the label g_e from σ since all extensions of σ will automatically extend an element of W_e.

In the second subcase, $\tau \supset \sigma$. If $\tau \in T_s$, we choose the leftmost leaf of T_s extending τ and call it τ'; if not, we add τ to T_s as a leaf and refer to it from this point as τ'. This is the only manner in which a resting leaf can be "awakened." We will now remove all labels from the strings extending σ. This means that all the leaves incomparable to τ' that extend σ will be resting from this point onwards unless they are awakened by another such requirement with higher priority. If this has not been done already, we will extend our definition of Φ by setting $\Phi(\tau') = \Phi(\sigma)$.

We finish the second subcase by setting $M_{i,\rho} = 0$ for all i and ρ such that $\rho \subseteq \sigma$ and ρ is labeled with m_i in T_s.

We repeat this procedure for each active requirement that has not been deactivated by a previous action in this stage. At the end of the stage, we extend τ, the rightmost leaf of T_s, as well as $\Phi(\tau)$. Clearly, by construction, τ will be a generic leaf. Suppose that the highest branching point below τ is associated with a machine node labeled with m_i. We will extend τ by $\langle 00 \rangle$ and $\langle 1 \rangle$ and label $\tau^\frown 0$

with m_{i+1}, $\tau^\frown 00$ with g_{i+1}, and $\tau^\frown 1$ with g_{i+1}. Let ρ be the longest string extending $\Phi(\tau)$ in $\Phi(T_s)$. Now we define $\Phi(\tau^\frown 0) = \Phi(\tau^\frown 00) = \rho^\frown 0$ and $\Phi(\tau^\frown 1) = \rho^\frown 1$. Again, we define Φ^{-1} accordingly.

Let $T = \cup_{s\in\omega} T_s$. We say that σ is *labeled with m_e (g_e) in T* if there is a stage s such that for all $t \geq s$, $\sigma \in T_t$ and σ is labeled with m_e (g_e) at stage t.

Now we must verify our construction.

Lemma 3.4. *If $B \in [T]$ and a substring of B is labeled with m_e in T, then M_e must be a computable Turing machine with domain 1.*

Proof. We proceed by contradiction. Suppose that B is an infinite branch of T and let σ be an initial segment of B that is labeled with m_e in T for some Turing machine M_e such that $\mu(M_e) < 1$. Then there is a smallest n such that $\mu(M_e) < 1 - \frac{1}{2^n}$. This means that σ will not act with n at any stage, so there are at most $n-1$ stages at which σ acts for m_e. Let s be the last such stage.

If a string τ extending σ labeled with m_i for some i becomes active at a stage $t > s$, branching will not result above σ since there will be a string below τ that is not active at that stage; i.e., σ itself. Alternately, if an initial segment τ of σ labeled with m_i becomes active at a stage $t > s$, there will be no branching above σ since σ will not be active at stage t. Therefore, no branching will occur above σ after stage s. This has two primary consequences: after stage s, no new labels will be assigned to strings above σ, and $[\sigma] \cap T_s$ will not be extended due to a requirement of the form Sch_i.

However, the elements of $[\sigma] \cap T_s$ may still be extended when we act to satisfy a requirement of the form Gen_i. If τ labeled with g_i is comparable with σ and acts after stage s, we may extend a leaf of T_s to meet the set W_i. However, there are only finitely many such τ, since no labels will be assigned to a string above σ after stage s, and there can be only finitely many such labels below σ. Therefore, elements of $[\sigma] \cap T_s$ can be extended no more than finitely often in this way. Since all such extensions are finite, this will not be enough to generate an element of $[T]$ with σ as an initial segment either.

Since $[\sigma] \cap T_s$ is a finite tree and no path through it can be extended more than finitely often, no extension of σ in T is infinite. Since we assumed that B was in $[T]$, we have a contradiction. □

Lemma 3.5. *Let M_e be a computable Turing machine with domain 1 and let σ be labeled with m_e in T. Then every infinite path in $\Phi([\sigma] \cap T)$ is Schnorr trivial with respect to M_e.*

Proof. Let M_e and σ be as described in the statement of the lemma, and let $B \in [\sigma] \cap T$. We begin by noting that if $\Phi(B) \in [\Phi(T)]$, $B \in [T]$. We must show that

$$(\exists M'_e \text{ comp.})\, (\exists c \in \omega)\, (\forall n \in \omega) \left[K_{M'_e}(\Phi(B)\restriction n) \leq K_{M_e}(0^n) + c\right]$$

We will show that $M'_{e,\sigma}$ is such a computable Turing machine and that $M'_{e,\sigma}$ and the constant 1 witness the Schnorr triviality of $\Phi(B)$ with respect to M_e. First, we must show that $M'_{e,\sigma}$ is a computable Turing machine. From this point on, we will write $M'_{e,\sigma}$ as M'_e for brevity.

Lemma 3.6. *M'_e is a Kraft-Chaitin set.*

Proof. We first note that, since our construction is recursive, the elements of M'_e form an r.e. set.

Now we must show that $\mu(M'_e) \leq 1$. We know that $\mu(M_e) = 1$. Suppose we divide the measure of M_e into a sequence of intervals $\langle I_k \rangle_{k>0}$, where we define $I_k = (1 - \frac{1}{2^{k-1}}, 1 - \frac{1}{2^k}]$ for each k. Note that I_k has measure precisely $\frac{1}{2^k}$ for each k. We would like to show that for each I_k, no more than $\frac{k}{2^{k+1}}$ enters M'_e. Theoretically, precisely the axioms that enter M_e at a stage s such that $s_{s,k-1} < s \leq s_{e,k}$ should contribute to I_k. However, it is possible that this will not be the case, since $s_{e,k-1}$ is defined as the least M_e-stage s such that $\mu(M_{e,s}) \geq 1 - \frac{1}{2^{k-1}}$ and not the least stage s such that $\mu(M_{e,s}) = 1 - \frac{1}{2^{k-1}}$. Therefore, it may be that $\mu(M_{e,s_{k-1}}) > 1 - \frac{1}{2^{k-1}}$, and the most we can say is that the axioms that contribute to I_k enter M_e at or before M_e-stage s_k.

Suppose $n \leq k$, and suppose $\langle d, \sigma \rangle$ enters M_e at an M_e-stage s such that $s_{e,n-1} < s \leq s_{e,n}$. There are now four possibilities: that $\mu(M_{e,s}) < 1 - \frac{1}{2^{k-1}}$ and $\langle d, \sigma \rangle$ contributes nothing to I_k, that $\mu(M_{e,s-1}) <$

$1 - \frac{1}{2^{k-1}} < \mu(M_{e,s})$, that $1 - \frac{1}{2^{k-1}} < \mu(M_{e,s-1}) < \mu(M_{e,s}) < 1 - \frac{1}{2^k}$, or that $1 - \frac{1}{2^{k-1}} < \mu(M_{e,s-1}) < 1 - \frac{1}{2^k}$ and $\mu(M_{e,s}) > 1 - \frac{1}{2^k}$.

In the first case, $\langle d, \sigma \rangle$ contributes nothing to I_k, so we need not consider it. In the third case, all of the measure that $\langle d, \sigma \rangle$ contributes to $\mu(M_e)$ is contained in I_k, so $n = k$ and $\frac{k}{2^{d+1}}$ enters $\mu(M_e')$ when $\frac{1}{2^d}$ enters $\mu(M_e)$.

The second and fourth cases are analogous. In the second case, some of the $\frac{1}{2^d}$ that $\langle d, \sigma \rangle$ contributes to $\mu(M_e)$ is contained in I_{k-1}. Suppose that $0 < q < 1$ is the fraction of this $\frac{1}{2^d}$ that is actually in I_k. Since $\langle d, \sigma \rangle$ is dealt with in our construction at stage n, $\frac{n}{2^{d+1}}$ will enter $\mu(M_e')$. We assign the proportional amount of measure $q \cdot \frac{n}{2^{d+1}}$ to the part of $\mu(M_e')$ corresponding to I_k, so for the $q \cdot \frac{1}{2^d}$ entering I_k, $q \cdot \frac{n}{2^{d+1}} \leq q \cdot \frac{k}{2^{d+1}}$ enters $\mu(M_e')$.

In the fourth case, some of the measure contributed to $\mu(M_e)$ by $\langle d, \sigma \rangle$ is contained in I_k and some is contained in I_m for $m > k$. We treat this precisely as we did in the second case and consider only the portion of this measure that is contained in I_k.

In each of these cases, when $\langle d, \sigma \rangle$ enters M_e and contributes some measure to I_k, no more than $\frac{k}{2}$ of that measure enters $\mu(M_e')$. After summing up the contributions to the measure of M_e' corresponding to each I_k, we can see that for each interval I_k, no more than $\frac{k}{2} \cdot \frac{1}{2^k} = \frac{k}{2^{k+1}}$ is added to $\mu(M_e')$. This allows us to see that $\mu(M_e') \leq \sum_{k \in \omega} \frac{k}{2^{k+1}} = 1$, so M_e' is a Kraft-Chaitin set.

□

Lemma 3.7. *$\mu(M_e')$ is a recursive real.*

Proof. It is enough to show that $\mu(M_e')$ is the limit of a recursive sequence of rationals $\langle q_c \rangle_{c \in \omega}$ such that there is a recursive function f such that for all c, $|\mu(M_e') - q_{f(c)}| < \frac{1}{2^c}$. We let $M_{e,c}'$ be the part of M_e' that has been enumerated at the earliest point in the construction by which Sch_e has acted with c for σ.

We begin by observing that since $\mu(M_{e,c}')$ is a finite sum of rationals for each c, each $\mu(M_{e,c}')$ is a rational. Furthermore, by our construction, $\langle \mu(M_{e,c}') \rangle_{c \in \omega}$ is a recursive sequence, and $\mu(M_e') = \lim_c \mu(M_{e,c}')$, so we may take our recursive sequence of rationals to be $\langle \mu(M_{e,c}') \rangle_{c \in \omega}$.

Finally, after the M_e-stage $s_{e,c}$, no more than $\frac{1}{2^c}$ will enter M_e, so, as demonstrated in the proof of the previous lemma, no more than $\sum_{j > c} \frac{j}{2^{j+1}}$ can be added to $\mathrm{dom}(M_e')$. We can therefore see that

$$|\mu(M_e') - \mu(M_{e,c}')| \leq \sum_{j > c} \frac{j}{2^{j+1}} = \frac{c+2}{2^{c+1}}$$

for each c. This inequality will allow us to find a recursive function f as mentioned in the first paragraph, so we can see that $\mu(M')$ is a recursive real. □

Now, by Lemmas 3.6 and 3.7 and the Kraft-Chaitin Theorem, we may take M_e' to be a computable Turing machine.

We must show that M_e' and the constant 1 witness the Schnorr triviality of $\Phi(B)$. If $K_{M_e}(0^n)$ is infinite, we are done. Otherwise, suppose that $K_{M_e}(0^n) = d$. Then $\langle d, 0^n \rangle \in M_e$. At some point after $\langle d, 0^n \rangle$ enters M_e, we will reach a stage in our construction s where σ will act, since $\mu(M_e) = 1$ and thus σ will act infinitely often. At that stage s, for every $\tau \in \Phi(T)$ of length n that is comparable to $\Phi(\sigma)$, we will add $\langle d+1, \tau \rangle$ to M_e'. $\Phi(B) \upharpoonright n$ will be among these τ, so $K_{M_e'}(\Phi(B) \upharpoonright n) \leq d + 1 = K_{M_e}(0^n) + 1$, and M_e' and 1 will witness the Schnorr triviality of $\Phi(B)$ with respect to M_e. □

Lemma 3.8. *Let $Z \subseteq \omega$, and suppose that $\langle M_i \rangle_{i \in Z}$ is a list of computable machines with domain 1. Let B_Z be the set of infinite paths in T such that if $B \in B_Z$ and $\sigma \subseteq B$ is labeled with m_e in T, then $e \in Z$. Then any element of B_Z is Schnorr trivial with respect to the elements of $\langle M_i \rangle_{i \in Z}$.*

Proof. This is clear from Lemma 3.5. □

Lemma 3.9. *Requirement Gen_e is satisfied for each e on every infinite path of T.*

Proof. Let $B \in [T]$. Since B is infinite, there has been infinitely much branching in T along B. Therefore, for every e, there is some initial segment of B that was labeled g_e at some point.

If a string σ is labeled with g_e in T, then σ has never acted in the construction. This can only happen if no element of W_e is comparable to σ. In this case, Gen_e is satisfied trivially, since for all $\tau \supseteq \sigma$, $\tau \notin W_e$.

Now suppose that no string is labeled with g_e in T. Let σ be the initial segment that had this property at a later stage of the construction than any other. Since σ's label must have been removed when some τ comparable to σ entered W_e at a stage s in the construction, σ must have acted. If $\tau \subseteq \sigma$, T_s was unchanged, and Gen_e was met because there is $\tau \in B$ such that $\tau \in W_e$. If $\tau \supset \sigma$, we altered T_s so that all branches extending σ would also go through τ. Since σ was the initial segment that was last labeled g_e, B will be an extension of τ. Otherwise, the label g_e would have been removed from σ due to the demand of some Gen_i for some $i < e$, and another initial segment of B would have been labeled with g_e at a later stage in the construction. Therefore, we will meet Gen_e once again, since there is $\tau \in B$ such that $\tau \in W_e$. □

Lemma 3.10. *If $B \in [T]$, then $B \equiv_T \Phi(B)$.*

Proof. This can be easily seen from the construction. □

Now we can finish the proof. This time, we let Z be the subset of ω such that $i \in Z$ if and only if M_i is computable with domain 1. We then choose the path G through T such that some initial segment of G will be labeled with m_e in T if and only if $e \in Z$. By Lemma 3.8, $\Phi(G)$ will be Schnorr trivial with respect to every computable Turing machine with domain 1 and therefore, simply Schnorr trivial. Additionally, since Gen_e is satisfied for every e on B by Lemma 3.9, G is 1-generic. We may note that $G \leq_T 0''$ since $Z \leq_T 0''$.

Finally, by Lemma 3.10, $\Phi(G) \equiv_T G$, so we have produced a 1-generic that is Turing equivalent to a Schnorr trivial.

□

We can see that the resulting functionals Φ and Φ^{-1} in the proof of Theorem 3.3 are not total, since by Lemma 3.4, the tree T that we build will be extended only finitely often above a string labeled with m_e if $\mu(M_e) < 1$. This explains why this proof cannot give us *tt*-equivalence.

We may now ask what properties a 1-generic that is Turing equivalent to a Schnorr trivial may have. As previously mentioned, such a 1-generic may be high. We show here that, in fact, such a 1-generic must be high and that a nonhigh 1-generic cannot even compute a nonrecursive Schnorr trivial real.

Theorem 3.11. *Suppose that G is a nonhigh 1-generic real. Then if $A \leq_T G$ and A is not recursive, A cannot be Schnorr trivial.*

The following lemma by Kummer [8] will be necessary for this proof. If T is a recursively enumerable tree, we say that $f : 2^{\leq n} \longrightarrow T$ is an embedding if $\sigma \subseteq \tau$ exactly when $f(\sigma) \subseteq f(\tau)$.

Lemma 3.12. *If T is a recursively enumerable tree such that for some n, there is no embedding of $2^{\leq n}$ into T, then every branch of T is recursive.*

Proof of Theorem 3.11. Suppose that G is a nonhigh 1-generic real and that $A \leq_T G$ as stated. Since $A \leq_T G$, there is a Turing functional Ψ such that $\Psi(G) = A$, and we can assume that for every $\sigma \in 2^{<\omega}$, $\Psi(\sigma)$ converges. Now we can define, for each $\sigma \in 2^{<\omega}$, the tree $T_\sigma = \{\alpha \mid (\exists \beta \supseteq \alpha)[\alpha \subseteq \Psi(\beta)]\}$. This produces a family of uniformly r.e. trees.

We consider two distinct cases: that in which $2^{\leq n}$ cannot be embedded into $T_{G\upharpoonright k}$ for some n and k, and that in which $2^{\leq n}$ can be embedded into $T_{G\upharpoonright k}$ for every n and k.

If there are n and k such that $2^{\leq n}$ cannot be embedded into $T_{G\upharpoonright k}$, by Lemma 3.12, every branch of $T_{G\upharpoonright k}$ must be recursive. Since A is necessarily one of these branches, A is recursive.

If not, the finite tree $2^{\leq n}$ is embeddable into every $T_{G\upharpoonright k}$ for every k and n. For every $\sigma \in 2^{<\omega}$, we can define S_σ to be the first enumerated subset of T_σ that is topologically equivalent to $2^{\leq |\sigma|+1}$ if such a subset exists; otherwise, we let S_σ be undefined. By our assumption, $S_{G\restriction n}$ is defined for all n. We now recall the following theorem by Martin.

Theorem 3.13. [9] *The following are equivalent for a real B.*
(1) $B' \not\geq_T 0''$.
(2) $(\forall f \leq_T B)\,(\exists g : \omega \longrightarrow \omega \text{ rec.})\,(\exists^\infty n \in \omega)\,[f(n) < g(n)]$.

Consider the function that maps each n to the number of steps required to enumerate $S_{G\restriction n}$ and the function that maps each n to the length of the greatest element of $S_{G\restriction n}$. Since both of these functions are recursive in G and G is not high, we can find a recursive function f such that for infinitely many n, no more than $f(n)$ steps are needed to enumerate $S_{G\restriction n}$, and all of the elements of $S_{G\restriction n}$ will have length less than or equal to $f(n)$.

Now, for each σ, we define $g^A(\sigma)$ to be the leaf of S_σ that is lexicographically closest to $A\restriction(f(|\sigma|))$ whenever S_σ is enumerated within $f(|\sigma|)$ steps and all its elements have lengths no greater than $f(|\sigma|)$ and to be the empty string $\langle\rangle$ otherwise. Therefore, f is not only Turing reducible to A, but truth-table reducible to it, and the set $R = \{\sigma \mid g^A(\sigma) \neq \langle\rangle\}$ is recursive.

We now assume for a contradiction that A is Schnorr trivial. By a result in [7], there is a recursive function h that, given a string σ, will produce a list of $2^{|\sigma|}$ possibilities for $g^A(\sigma)$. We observe that for each $\sigma \in R$, the set S_σ has $2^{|\sigma|+1}$ leaves, so there must be a leaf α_σ that is not in the list of possibilities produced by h. The mapping from σ to α_σ is partial recursive, and we can use this mapping to find another partial recursive mapping that takes a string σ to an extension β_σ such that $\alpha_\sigma \subseteq \Psi^{\beta_\sigma}$.

Since G is not high, $G\restriction n \in R$ for infinitely many n, and since G is 1-generic, there is an n such that $G\restriction n \in R$ and $G\restriction n \subseteq \beta_{G\restriction n} \subseteq G$. Therefore, $\alpha_{G\restriction n} \subseteq A$ and $\alpha_{G\restriction n}$ is the leaf of $S_{G\restriction n}$ that is lexicographically closest to $A\restriction f(n)$. However, by our definition of α_σ, $\alpha_{G\restriction n}$ cannot be in the list of possibilities for $g^A(\sigma)$, so A must not be Schnorr trivial. □

We end this section with a corollary to Corollary 3.2.

Corollary 3.14. *There is a K-trivial real which is not Schnorr trivial.*

Proof. We note that every nonzero recursively enumerable degree bounds a 1-generic degree [11]. There are several different proofs that there is a recursively enumerable K-trivial B [14, 4], and Nies has shown that the K-trivial reals are closed downward in the Turing degrees [10]. Therefore, there are K-trivial 1-generics. However, by Corollary 3.2, no 1-generic is Schnorr trivial, so we have a K-trivial real that is not Schnorr trivial. □

We may also note that since the K-trivial reals are closed downward in the Turing degrees, a degree that contains a 1-generic either contains only K-trivials or no K-trivials.

4. Degrees without cones of Schnorr trivials

In [6], we proved the following theorem.

Theorem 4.1. *Let $\mathbf{h}$ be a Turing degree such that $\mathbf{h}' \geq_T \mathbf{0}''$. Then $\mathbf{h}$ contains a Schnorr trivial.*

Here, we show that this theorem is optimal.

Theorem 4.2. *Suppose $\mathbf{a}$ is a Turing degree such that $\mathbf{a}' \not\geq_T \mathbf{0}''$. Then there is a Turing degree above $\mathbf{a}$ that contains no Schnorr trivial reals.*

Proof. Let such a degree $\mathbf{a}$ be given, and let $A \in \mathbf{a}$. We will show that if G is a 2-generic relative to A, then $A \oplus G$ is not Turing equivalent to any Schnorr trivial real.

Let G be 2-generic relative to A, and suppose that $B \equiv_T A \oplus G$. Then there are Ψ^A and Φ^A such that $\Psi^A(G) = B$ and $\Phi^A(B) = G$.

Since the statement in Section 2 relativizes, we can write "Φ^A and Ψ^A are total, and $\Phi^A \circ \Psi^A$ is the identity function" as the following Π^A_2 statement.

$$\varphi := (\forall n \in \omega)\,(\exists s \in \omega)\left[\Phi^A_s(n)\downarrow \wedge \Psi^A_s(n)\downarrow \wedge (\Phi^A \circ \Psi^A)_s(n) = n\right]$$

Therefore, since G is 2-generic relative to A, there is an initial segment of G that forces φ. We call this initial segment p and consider the set of forcing conditions $\mathcal{P} = \{q \in 2^{<\omega} \mid q \supseteq p\}$. We will order these conditions by writing $q \succeq r$ if and only if $q \subseteq r$. Now we may consider the set $T = \{\Psi^A(q) \mid q \preceq p\}$ as before. Since p forces the totality of Ψ^A, we can see that T may be interpreted as a perfect tree that is recursive in A. We note that $B \in [T]$.

Now we use Theorem 3.13 to construct a computable Turing machine witnessing the non-Schnorr triviality of B.

We define $f : \omega \longrightarrow \omega$ such that for all n, $f(n)$ is the least height at which at least 2^n distinct strings $\sigma \oplus \tau$ in T such that τ extends $\Psi^A(q)$ for each $q \in \mathcal{P}$ such that $q \in \{p^\frown\sigma \mid |\sigma| = n\}$. We will refer to the set of such q as $\mathcal{Q}_n$ for each n. Since T is a perfect tree, f is total, and it is clear that $f \leq_T A$. Therefore, we may take a recursive function $g : \omega \longrightarrow \omega$ such that $f(n) < g(n)$ for infinitely many n. We will use this g to build a computable Turing machine M witnessing the fact that B cannot be Schnorr trivial; i.e., we will require that M satisfy the following statement.

$$(\forall M_e \text{ comp.})\,(\forall c \in \omega)\,(\forall q \preceq p)\,(\exists r \preceq q)\,(\exists n \in \omega)\left[K_{M_e}(\Psi^A(r)\restriction n) > K_M(0^n) + c\right]$$

We will not satisfy this statement directly for all $q \in \mathcal{P}$, since $f(n) < g(n)$ only infinitely often and not for all n. However, if we attempt to satisfy it for all elements of each $\mathcal{Q}_n$, we will be correct infinitely often, and if we succeed for the elements of $\mathcal{Q}_n$, we will clearly succeed for all of the elements of $\mathcal{Q}_i$ for $i < n$.

The rest of this proof is very much like that of Theorem 2.1. As before, we begin by listing the pairs $\langle e, c\rangle$ such that $\mu(M_e) \geq 1 - \frac{1}{2^c}$. This list is recursively enumerable, and if M_e is computable with domain 1, $\langle e, c\rangle$ will appear in this list for all $c \in \omega$. Therefore, this list will be infinite, and we may write it as $\langle e_m, c_m\rangle_{m\in\omega}$.

We will consider the pair $\langle e_m, c_m\rangle$. We will allot $\frac{1}{2^{m+1}}$ of the measure of M to ensure that for each $q \in \mathcal{P}$, there is an extension of $\Psi^A(q)$ that is not Schnorr trivial with respect to M_{e_m} and c_m. To do this, we will ensure that we cannot force the statement that there is no such extension. Since G is 2-generic with respect to A and $B \in [T]$, this will be sufficient.

We will take care of this particular pair for the elements of $\mathcal{Q}_i$ with $\frac{1}{2^{(m+1)+(i+1)}}$ of the $\frac{1}{2^{m+1}}$ we originally allotted. We will have 2^i elements in $\Psi(\mathcal{Q}_i) = \{\Psi^A(q) \mid q \in \mathcal{Q}_i\}$, so we will be able to use a maximum of $\frac{1}{2^{(m+1)+(i+1)}} \cdot \frac{1}{2^i} = \frac{1}{2^{(m+1)+2i+1}}$ of the measure of M above each such element. From this point on, we will refer to $\langle e_m, c_m\rangle$ simply as $\langle e, c\rangle$.

Now we may use the values of g to determine a height h such that there are at least $2^{(m+1)+2i+1} + 1$ extensions of $\Psi^A(q)$ of each q in $\mathcal{Q}_i$ at this height. The smallest measure that M_e can assign a string of this length after a stage at which $\mu(M_e) \geq 1 - \frac{1}{2^c}$ is less than $\frac{1}{2^{(m+1)+2i+1}} \cdot \frac{1}{2^c} = \frac{1}{2^{(m+1)+2i+1+c}}$. Therefore, the greatest complexity that M_e can assign an extension of $\Psi^A(q)$ for a q in $\mathcal{Q}_i$ of height at least h after such a stage must be strictly greater than $(m+1) + 2i + 1 + c$.

Let $n_{e,c}$ be the length of the longest string in the range of M_e that appears by the smallest stage s at which $\mu(M_{e,s}) \geq 1 - \frac{1}{2^c}$, and let n be the smallest integer such that the following three conditions hold.

(1) $n > n_{e,c}$.
(2) $n \geq h$.
(3) At this stage in the construction of M, $0^n \notin M$.

We now add $\langle (m+1) + 2i + 1, 0^n\rangle$ to M, so $K_M(0^n) = (m+1) + 2i + 1$.

We repeat this procedure for each m and i, so we can see that M is a computable machine based on the following calculation.

$$\begin{aligned}\mu(M) &= \sum_{m\in\omega}\sum_{i\in\omega}\frac{1}{2^{(m+1)+2i+1}}\\ &= \sum_{m\in\omega}\frac{1}{2^{m+1}}\cdot\frac{1}{6}\\ &= \frac{1}{6}\end{aligned}$$

Now we can see that for infinitely many i, for each $q \in \mathcal{Q}_i$, there is an $r \in \mathcal{P}$ such that $\Psi^A(r)$ extends $\Psi^A(q)$ and the following occurs.

$$K_{M_e}(\Psi^A(r)\restriction n) > (m+1) + 2i + c = K_M(0^n) + c$$

This is enough, since if this statement is true for all $q \in \mathcal{Q}_i$, it is true for all $q \in \mathcal{Q}_n$ for $n < i$ as well. Therefore, for every element of every $\mathcal{Q}_i$, there will be an extension of its image under Ψ^A that will, together with M, witness the non-Schnorr triviality of its branch with respect to M_e and c. Since there are infinitely many such i, this will enable us to force B to be non-Schnorr trivial.

□

This theorem, together with Theorem 9 of [6], gives the following result.

Theorem 4.3. *Let* $\mathbf{a}$ *be a Turing degree. Then* $\mathbf{a}' \geq_T \mathbf{0}''$ *if and only if every degree above* $\mathbf{a}$ *contains a Schnorr trivial real.*

5. A further question

While a 1-generic cannot be tt-equivalent to a Schnorr trivial, it can be Turing equivalent to one. This naturally leads us to consider the question of wtt-reducibility. We recall that $A \leq_{wtt} B$ if there is some Φ such that $\Phi(B) = A$ and there is a recursive function f such that to calculate $A(n)$ using Φ, $B\restriction f(n)$ will be sufficient. This function need not be total, so wtt-reducibility is intermediate between tt-reducibility and Turing reducibility.

We note that neither the proof of Theorem 3.1 nor the proof of Theorem 3.3 can be adapted to the wtt case. We cannot guarantee that the tree T in the proof of Theorem 3.1 is perfect if we have wtt-equivalence instead of tt-equivalence, since a wtt-functional may or may not be total. Therefore, this proof does not work for wtt-equivalence. Similarly, in the construction in the proof of Theorem 3.3, it does not seem that we can bound the length of the initial segment of either X or G necessary to compute the other recursively. To determine the amount of X we need to compute G, we need to know which r.e. sets G must meet, which is a Σ^0_1 question. Similarly, to determine the amount of G we need to compute X, we must know which M_e have measure $\geq 1 - \frac{1}{2^c}$ for certain c and e, which is, again, a Σ^0_1 question. Therefore, the following question remains open.

Question 5.1. Is there a 1-generic real that is wtt-equivalent to a Schnorr trivial real?

References

[1] Gregory J. Chaitin. A theory of program size formally identical to information theory. *J. Assoc. Comput. Mach.*, 22:329–340, 1975.

[2] Rod Downey, Evan Griffiths, and Geoffrey Laforte. On Schnorr and computable randomness, martingales, and machines. *Math. Log. Q.*, 50(6):613–627, 2004.

[3] Rod Downey, Denis R. Hirschfeldt, André Nies, and Sebastiaan A. Terwijn. Calibrating randomness. *Bull. Symbolic Logic*, 12(3):411–491, 2006.

[4] Rod G. Downey, Denis R. Hirschfeldt, André Nies, and Frank Stephan. Trivial reals. In *Proceedings of the 7th and 8th Asian Logic Conferences*, pages 103–131, Singapore, 2003. Singapore Univ. Press.

[5] Rodney G. Downey and Evan J. Griffiths. Schnorr randomness. *J. Symbolic Logic*, 69(2):533–554, 2004.
[6] Johanna N.Y. Franklin. Schnorr trivial reals: A construction. *Arch. Math. Logic*, 46(7–8):665–678, 2008.
[7] Johanna N.Y. Franklin and Frank Stephan. Schnorr trivial sets and truth-table reducibility. Technical Report TRA3/08, School of Computing, National University of Singapore, 2008.
[8] Martin Kummer. A proof of Beigel's cardinality conjecture. *J. Symbolic Logic*, 57(2):677–681, 1992.
[9] Donald A. Martin. Classes of recursively enumerable sets and degrees of unsolvability. *Z. Math. Logik Grundlagen Math.*, 12:295–310, 1966.
[10] André Nies. Lowness properties and randomness. *Adv. Math.*, 197(1):274–305, 2005.
[11] Piergiorgio Odifreddi. *Classical Recursion Theory, Volume II*. Number 143 in Studies in Logic and the Foundations of Mathematics. North-Holland, 1999.
[12] C.-P. Schnorr. *Zufälligkeit und Wahrscheinlichkeit*, volume 218 of *Lecture Notes in Mathematics*. Springer-Verlag, Heidelberg, 1971.
[13] Robert I. Soare. *Recursively Enumerable Sets and Degrees*. Perspectives in Mathematical Logic. Springer-Verlag, 1987.
[14] Domenico Zambella. On sequences with simple initial segments. Technical Report ILLC ML-1990-05, University of Amsterdam, 1990.

E-mail address: franklin@math.nus.edu.sg