Trivial Reals*

(Extended Abstract)

Rod G. Downey
School of Mathematical and Computing Sciences
Victoria University of Wellington

Denis R. Hirschfeldt
Department of Mathematics
University of Chicago

André Nies
Department of Computer Science
Auckland University

Frank Stephan
Mathematical Institute
University of Heidelberg

May 7, 2002

1 Introduction

Our concern is the relationship between the intrinsic computational complexity of a real and the intrinsic randomness of the real. In [11, 12], together with LaForte, we looked at ways of understanding the intrinsic randomness of reals by measuring their relative initial segment complexity. (In this paper, "random" will always mean "1-random"; see Section 2 for basic definitions.) Thus, for instance, if α and β are reals (in $(0,1)$), given as binary sequences, then we can compare the complexities of α and β by studying notions of reducibility based on relative initial segment complexity. For example, we define $\alpha \leqslant_K \beta$ if there is a constant c such that, for all n, $K(\alpha \upharpoonright n) \leqslant K(\beta \upharpoonright n) + c$, where we will be denoting classical Kolmogorov complexity by K. For prefix-free Kolmogorov complexity H, we define $\alpha \leqslant_H \beta$ analogously.

The goal of the papers [11, 12] was to look at the structure of reducibilities like the above, and interrelationships among them, as a way of addressing questions such as: How random is a real? Given two reals, which is more random? If we partition reals into equivalence classes of reals of the "same degrees of randomness", what does the resulting structure look like?

The classic example of a random real is the halting probability of a universal prefix-free machine M, Chaitin's $\Omega = \sum_{M(\sigma)\downarrow} 2^{-|\sigma|}$. It is well-known that Ω has the property that $\alpha \leqslant_H \Omega$ for all reals α.

A natural question to ask is the following: Given reals $\alpha \leqslant_Q \beta$ (for $Q \in \{H, K\}$), what can be said about the computational complexity of α and β measured relative to, say, Turing reducibility?

For example, if we restrict our attention to computably enumerable (c.e.) reals (ones whose left cuts are computably enumerable), then being H-complete like Ω implies that the real is Turing complete. A natural guess would be that for all reals, if $\alpha \leqslant_Q \beta$ then $\alpha \leqslant_T \beta$. However, this is not true in general.

*The first author is supported by the Marsden fund of New Zealand. The fourth author is supported by the Heisenberg program of the Deutsche Forschungsgemeinschaft (DFG), grant no. Ste 967/1–1.

The present paper is concerned with "trivial reals". These are reals whose complexity is "low" or trivial from the point of view of randomness, in the sense that such reals resemble computable reals like 1^ω.

Building on work of Loveland [29], Chaitin proved that if α if a real with $K(\alpha \upharpoonright n) \leqslant K(1^n)+O(1)$ for all n, then α is computable. That is, if in terms of its Kolmogorov complexity, α looks like 1^ω, then it must be trivial computationally. What about the prefix-free version? Chaitin also proved that if a real α has the property that $H(\alpha \upharpoonright n) \leqslant H(1^n) + O(1)$ for all n then α is Δ^0_2. He asked if this could be improved to say that α must be computable.

It is a remarkable result that one *cannot* so improve this: Solovay [35] proved that there are Δ^0_2 noncomputable reals β such that for all n, $H(\beta \upharpoonright n) \leqslant H(1^n) + O(1)$. Solovay's proof is in an unpublished manuscript, and is long and difficult. All known proofs of Solovay's theorem use variations of his technique.

In Section 3 we will give a new, short and easy proof of a strengthening of Solovay's result that such noncomputable "H-trivial reals" exist. To state an extension of this result, we need another triviality notion.

Answering a question of Kučera and of van Lambalgen, Kučera and Terwijn [22] constructed a set X which is low for random. Here we say that X is low for random (also known as Martin-Löf-low) iff the collection of sets random relative to X is exactly the collection of random sets. It is possible to modify the construction given in Section 3 to show that there exist noncomputable computably enumerable sets that are both H-trivial and low for random.

H-triviality is surely a remarkable phenomenon. The remainder of the present paper is devoted to exploring this concept.

We prove that no H-trivial set can be Turing complete, and indeed every H-trivial set has array non-computable (anc) degree, where a degree is anc if it contains one of the array non-computable[1] sets of Downey, Jockusch and Stob [13, 14]. This means that the construction of a noncomputable H-trivial real provides a very simple (one line, essentially) injury-free solution to Post's problem.

In an unpublished report, Zambella [40] proved that there is a computable function f such that for each c there are at most $f(c)$ many reals α with

$$H(\alpha \upharpoonright n) \leqslant H(1^n) + c.$$

No proof of Zambella's theorem has appeared in print. We will give a proof of this result.

The reducibility $\leqslant_H$ is a preordering and hence we can form a degree structure from it, the H-degrees. The resulting degree structure has as its join operation ordinary addition. That is, $[\alpha] \vee [\beta] = [\alpha + \beta]$, where $[\alpha]$ is the H-degree of α. As a consequence, we show that the H-trivial reals form an ideal in the H-degrees.

The notion of triviality seems intimately related to weak truth table reducibility. Recall that $A \leqslant_{wtt} B$ iff there is a Turing procedure Φ and a computable function φ such that $\Phi^B = A$ and for all x the maximum number queried of B on input x is bounded by $\varphi(x)$. We prove that the H-trivial reals *also* form an ideal in the wtt-degrees.

Related to the topic of H-triviality is work of Kummer [24]. Kummer investigated "Kummer trivial" computably enumerable sets. In terms of classical (non-prefix free) Kolmogorov complexity, we know that if A is computably enumerable then $K(A \upharpoonright n) \leqslant n + 2\log n$ for all n. Kummer constructed computably enumerable sets A and constants c such that, infinitely often, $K(A \upharpoonright n) \geqslant 2\log n - c$. He called such sets *complex.* Kummer also showed that the computably enumerable degrees exhibit a gap phenomenon. Namely, either a degree $\mathbf{a}$ contains a complex set A, or all $A \in \mathbf{a}$ are "Kummer trivial" in the sense that for all $\epsilon > 0$, there

[1] A very strong array $\{F_x : x \in \mathbb{N}\}$ is a collection of disjoint finite sets given by canonical indices, such that $|F_x| < |F_{x+1}|$ for all x. A set A is called array noncomputable relative to such an array iff for all computable enumerable sets W, there exists an x with $A \cap F_x = W \cap F_x$. A computably enumerable degree contains a set that is anc relative to some very strong array iff it contains one that is anc relative to any other very strong array.

is a c such that $K(A \upharpoonright n) \leqslant (1+\epsilon)\log n$ for all n. (By Chaitin's work, if $K(A \upharpoonright n) \leqslant \log n + c$ for all n, then A is computable, so this result is sharp.) Kummer proved that the degrees containing such complicated sets are exactly the anc degrees. We prove that (i) not every array-noncomputable computably enumerable set is H-trivial, and (ii) there exist Turing degrees containing only Kummer trivial sets which contain no H-trivial sets. The result (ii) implies that being Kummer trivial does not make a set H-trivial.

2 Basic Definitions

Our notation is standard, except that we follow the tradition of using H for prefix-free complexity and K for non-prefix-free complexity. Our computability-theoretic notation follows Soare [33].

We work with reals between 0 and 1, identifying a real with its binary expansion, and hence with the set of natural numbers whose characteristic function is the same as that expansion. A real α is c.e. if its left cut is c.e. as a set, or equivalently, if α is the limit of a computable increasing sequence of rationals.

We work with machines with input and output alphabets $\{0, 1\}$. A machine M is *prefix-free* (or *self-delimiting*) if, for all finite binary strings σ and $\tau \subsetneq \tau'$, we have $M^\sigma(\tau)\downarrow \Rightarrow M^\sigma(\tau')\uparrow$, where M^σ means that M uses σ as an oracle. It is *universal* if for each prefix-free machine N there is a constant c such that, for all binary strings σ and τ, if $N^\sigma(\tau)\downarrow$ then $M^\sigma(\mu)\downarrow = N^\sigma(\tau)$ for some μ with $|\mu| \leqslant |\tau| + c$. We call c the *coding constant* of N.

The prefix-free complexity $H(\tau)$ of a binary string τ is the length of the shortest binary string σ such that $M(\sigma)\downarrow = \tau$ for a fixed universal prefix-free machine M. (The choice of M does not affect the prefix-free complexity, up to a constant additive factor.) For a natural number n, we write $H(n)$ for $H(1^n)$. A real α is *random*, or more precisely, 1-random, if there is a constant c such that $\forall n(H(\alpha \upharpoonright n) \geqslant n - c)$.

An important tool in building prefix-free machines is the Kraft-Chaitin Theorem.

2.1 Theorem (Kraft-Chaitin). *From a c.e. sequence of pairs $(\langle n_i, \sigma_i\rangle)_{i\in\omega}$ (known as* axioms*) such that $\sum_{i\in\omega} 2^{-n_i} \leqslant 1$, we can effectively obtain a prefix-free machine M such that for each i there is a τ_i of length n_i with $M(\tau_i)\downarrow = \sigma_i$, and $M(\mu)\uparrow$ unless $\mu = \tau_i$ for some i.*

3 A short proof of Solovay's theorem

We now give our simple proof of Solovay's theorem that H-trivial reals exist. This was proved by Solovay in his 1974 manuscript [35]. The proof there is very complicated and only constructs a Δ^0_2 real.

3.1 Theorem (after Solovay [35]). *There is a noncomputable c.e. set A such that $H(A \upharpoonright n) \leqslant H(n) + O(1)$.*

Remark. While the proof below is easy, it is slightly hard to see why it works. So, by way of motivation, suppose that we were to asked to "prove" that the set $B = \{0^n : n \in \omega\}$ has the same complexity as $\omega = \{1^n : n \in \omega\}$. A complicated way to do this would be for us to build our own prefix-free machine M whose only job is to compute initial segments of B. The idea would be that if the universal machine U converges to 1^n on input σ then $M(\sigma)\downarrow = 0^n$. Notice that, in fact, using Kraft-Chaitin it would be enough to build M *implicitly*, enumerating the length axiom (or "requirement") $\langle|\sigma|, 0^n\rangle$. We are guaranteed that $\sum_{\tau\in\mathrm{dom}(M)} 2^{-|\tau|} \leqslant \sum_{\sigma\in\mathrm{dom}(U)} 2^{-|\sigma|} \leqslant 1$, and hence Kraft-Chaitin applies.

Note also that we could, for convenience and as we do in the main construction, use a string of length $|\sigma| + 1$, in which case we would force $\sum_{\tau\in\mathrm{dom}(M)} 2^{-|\tau|} < 1/2$.

Proof. The idea is the following. We will build a noncomputable c.e. set A in place of the B described in the remark and, as above, we will slavishly follow U on n in the sense that whenever U enumerates, at stage s, a shorter σ with $U(\sigma) = n$, then we will enumerate a requirement $\langle |\sigma| + 1, A_s \upharpoonright n\rangle$ for our machine M. To make A noncomputable, we will also sometimes make $A_s \upharpoonright n \neq A_{s+1} \upharpoonright n$. Then for each j with $n \leqslant j \leqslant s$, for the currently shortest string σ_j computing j, we will also need to enumerate an axiom $\langle |\sigma_j|, A_{s+1} \upharpoonright j\rangle$ for M. This construction works by making this extra measure added to the domain of M small.

We are ready to define A:

$$A = \{\langle e, n\rangle : \exists s\,(W_{e,s} \cap A_s = \emptyset \wedge \langle e, n\rangle \in W_{e,s} \wedge \sum_{\langle e,n\rangle \leqslant j \leqslant s} 2^{-H(j)[s]} < 2^{-(e+2)})\},$$

where $W_{e,s}$ is the stage s approximation to the eth c.e. set and $H(j)[s]$ is the stage s approximation to the H-complexity of j.

Clearly A is c.e.. Since $\sum_{j \geqslant m} 2^{-H(j)}$ goes to zero as m increases, if W_e is infinite then $A^{[e]} \cap W_e^{[e]} \neq \emptyset$. It is easy to see that this implies that A is noncomputable. Finally, the extra measure put into the domain of M, beyond one half of that which enters the domain of U, is bounded by $\sum_e 2^{-(e+2)}$ (corresponding to at most one initial segment change for each e), whence

$$\sum_{\sigma \in \mathrm{dom}(M)} 2^{-|\sigma|} < \sum_{\sigma \in \mathrm{dom}(U)} 2^{-(|\sigma|+1)} + \sum_e 2^{-(e+2)} \leqslant \frac{1}{2} + \frac{1}{2} = 1.$$

Thus M is a prefix-free machine, and hence $H(A \upharpoonright n) \leqslant H(n) + O(1)$. □

We remark that the above proof can be modified to prove a stronger result, which appears to be due to Muchnik, namely that there exists a noncomputable c.e. set A that is super-H-trivial, in the sense that H-complexity relativized to A is the same as H-complexity, up to an additive constant. Such an A is both H-trivial and low for random.

Clearly the proof also admits many variations. For instance, we can make A promptly simple, or below any nonzero computably enumerable degree. We cannot control the jump or make A Turing complete, since, as we will see, all H-trivials are Turing incomplete, and in fact, low_2. We do not know if the H-trivials coincide with the low for random sets. An interesting test case is whether all H-trivials are of low Turing degree, since all low for random sets are of low Turing degree (by combining results of Kučera and Terwijn [22] and Nies [31]).

4 Turing degrees of H-trivials

In this section we give a proof that every H-trivial c.e. real is Turing incomplete. This proof can be modified to show that every H-trivial real α is Turing incomplete, and in fact not high (i.e., $\alpha' <_T \emptyset'$). A more complicated modification shows that α must be array computable, and hence low_2 (i.e., $\alpha'' \equiv_T \emptyset''$). A sketch of this modified proof is contained in the appendix.

4.1 Theorem. *Every H-trivial c.e. real is Turing incomplete.*

Proof. Since for every c.e. real there is a wtt-equivalent c.e. set, it is enough to prove the theorem for c.e. sets.

Suppose that A is H-trivial, c.e., and complete. For convenience, assume that $0 \notin A$, so that, thought of as a real, $A \leqslant 1/2$. Let e be such that $\Phi_e^A = K$, where K is the halting problem, and let d be such that $\forall n\ (H(A \upharpoonright n) \leqslant H(n) + d)$.

We build a c.e. set B and a prefix-free machine M (via the Kraft-Chaitin Theorem). By the Recursion Theorem, we can assume that we know i such that $\Phi_i^K = B$ and the coding constant b of M. Let $\Gamma = \Phi_i \circ \Phi_e$, so that $\Gamma^A = B$, and let $c = b + d$, so that $\forall n\ (H(A \upharpoonright n) \leqslant H_M(n) + c)$. Let $k = 2^{c+2}$.

We adopt the usual conventions on the use γ of Γ. All values appended by $[s]$ will be taken at the end of the stage. We also assume that we have slowed down our construction enough so that $\forall s\ \forall n \leqslant s\ (H(A \upharpoonright n)[s] \leqslant H_M(n)[s] + c)$ and $\forall s\ \forall n \leqslant s\ (\Gamma^A(n)[s]\downarrow)$.

We will have the following global variables: natural numbers $m_0, \dots, m_{k-1}$, which are always kept in increasing order; reals *Used* and *Trash*; and sets of natural numbers $\mathit{Current}_j$ for $j \leqslant k$ and *Successful*. In each procedure below, q will always be a local variable.

Let us first define the procedure INCREASE:

1. Let $s+1$ be the current stage.

2. Choose a fresh large n and enumerate $\langle n, 2^{-\gamma^A(m_{k-1})[s]}\rangle$ as an axiom for M. Add $2^{-\gamma^A(m_{k-1})[s]}$ to *Used*.

3. If during stage $s+1$ some number enters A below $\gamma^A(m_{k-1})[s]$ then remove $2^{-\gamma^A(m_{k-1})[s]}$ from *Used* and add it to *Trash*. In this case, return 0.

4. Enumerate n into $\mathit{Current}_k$, conclude stage $s+1$, and return $2^{-\gamma^A(m_{k-1})[s]}$.

Let us now define the procedure LOAD(j):

1. If $j = k$ then return INCREASE.

2. Let $q = 0$ and let $s+1$ be the current stage.

3. While $q < 2^{-\gamma^A(m_{j-1})}[s]$ (or $q < 1/2$ if $j = 0$):

   (a) Call LOAD$(j+1)$ and add the returned value to q.

   (b) End the current stage t.

   (c) If $\gamma^A(m_{j-1})[t] \neq \gamma^A(m_{j-1})[s]$ then remove q from *Used* and add it to *Trash*. In this case, empty $\mathit{Current}_{j+1}$ and return 0.

4. Enumerate m_j into B and choose a fresh large value for each m_i, $i \geqslant j$.

5. If, during the current stage $t+1$, some number enters A below $\gamma^A(m_{j-1})[t]$ then remove q from *Used* and add it to *Trash*. In this case, empty $\mathit{Current}_{j+1}$ and return 0.

6. Enumerate every number in $\mathit{Current}_{j+1}$ into $\mathit{Current}_j$, empty $\mathit{Current}_{j+1}$, conclude stage $t+1$, and return q.

Our main procedure is as follows:

1. Let *Used* $=$ *Trash* $= 0$ and let each $\mathit{Current}_j$ be empty. Choose values for $m_0, \dots, m_{k-1}$.

2. Call LOAD(0)

3. Let *Successful* $=$ $\mathit{Current}_0$.

It is easy to check that the total amount of the axioms for M is equal to *Used* $+$ *Trash*, that *Used* $\leqslant 1/2$, and that *Trash* $\leqslant A \leqslant 1/2$. (To see that *Trash* $\leqslant A$, notice that, for each $n \in A$, the amount that is put into *Trash* as a consequence of n entering A at stage s is at most the amount corresponding to the elements of $\mathit{Current}_j$ for the least j such that $n < \gamma^A(m_j)[s]$, and that this amount is bounded by $2^{-\gamma^A(m_j)[s]}$.)

Furthermore, the main procedure must terminate. This can be argued by reverse induction. First of all, LOAD(k) always returns. Now assume that LOAD$(j+1)$ always returns. Then the only way LOAD(j) can get stuck is if the values returned by successive calls to LOAD$(j+1)$ go to 0. But that would mean that $\lim_s \gamma^A(m_j)[s] = \infty$, contradicting the choice of Γ. Thus, by induction, LOAD(0) always returns, and by the same argument as above, the values returned by successive calls to this routine do not go to 0. This means that the main procedure eventually terminates.

It is now easy to check that $Used = \sum_{n \in Successful} 2^{-H_M(n)} = 1/2$. We claim that for each $n \in$ *Successful*, if s is least such that $H_M(n)[s+1] = H_M(n)$ then $A \upharpoonright n$ assumes at least k many values after stage s. If this is true then we have a contradiction, since $\sum_{n \in Successful} k 2^{-(H_M(n)+c)} = k2^{-c} \sum_{n \in Successful} 2^{-H_M(n)} = 2^{c+2}2^{-c}(1/2) > 1$.

To verify the claim, let $n \in$ *Successful*. At some stage $s+1$ we enumerate $\langle n, 2^{-\gamma^A(m_{k-1})[s]} \rangle$ as an axiom for M. Notice that $H_M(n)[s+1] = H_M(n)$, since we never enumerate another axiom involving n.

For each $j < k$, let $t_j + 1$ be the stage at which n enters $Current_j$ and let $l_j = \gamma^A(m_j)[s]$. Since m_j enters A stage $t_j + 1$, we have $A[t_j+1] \upharpoonright l_j \neq A[s] \upharpoonright l_j$. On the other hand, $A[t_j+1] \upharpoonright l_{j-1} = A[s] \upharpoonright l_{j-1}$, since otherwise n could not enter $Current_j$.

So we have $A[t_{k-1}] \upharpoonright n \neq \cdots \neq A[t_0] \upharpoonright n$, as desired. □

One final limitation is the following.

4.2 Theorem. *The Turing degrees of H-trivial c.e. sets are bounded by a c.e. degree strictly below $\mathbf{0}'$.*

Proof. Notice that the statement "$H(W_i \upharpoonright n) \leqslant H(1^n) + c$ for all n" is Π^0_2 in the parameters i, c. Thus the collection of indices of H-trivial c.e. sets is Σ^0_3. We can enumerate a piecewise c.e. set A where the $\langle i, c\rangle$-th column is equal to W_i iff W_i is H-trivial with constant c, and finite otherwise. Notice that such a set has the property that $\oplus_{m \leqslant n} A^{(m)}$ is Turing incomplete for all m. Hence, the result follows from the strong form of the thickness lemma (e.g., Soare [33], Ch VIII, Theorems 2.3 and 2.6). □

In unpublished work, Nies has shown that the degrees of H-trivial c.e. sets are bounded below $\mathbf{0}'$ by a low$_2$ c.e. set. This is much more difficult to prove. At the end of the next section, we give a proof that the degrees of all H-trivials are bounded by a degree strictly below $\mathbf{0}'$ (Theorem 5.7).

5 Zambella's Theorem

In this section we will give proofs of some unpublished material of Zambella. Our proofs are different and we establish some intermediate results of independent interest.

5.1 Definition. Given a prefix free machine D, we define the *Zambella measure* with respect to D as

$$Z_D(\sigma) = \mu(D^{-1}(\sigma)).$$

That is, $Z_D(\sigma)$ is the probability that D outputs σ. If D is the fixed universal machine we will write $Z(\sigma)$ for $Z_D(\sigma)$.

5.2 Theorem. $Z_D(\sigma) = O(2^{-H(\sigma)})$.

We make a few comments before proving this theorem. Notice that $Z(\sigma)$ is a measure of complexity, and hence, by the minimality of H among measures of complexity, we know that $2^{H(\sigma)} \leqslant Z(\sigma)$. Therefore, by Theorem 5.2, we know that for some constant d,

$$2^{-H(\sigma)} \leqslant Z(\sigma) \leqslant d2^{-H(\sigma)}.$$

Therefore, we can often replace usage of H by Z. As an illustration, for reals α and β, we have the following result.

5.3 Theorem. $\alpha \leqslant_H \beta$ *iff there is a constant c such that for all n,*

$$cZ(\beta \upharpoonright n) \geqslant Z(\alpha \upharpoonright n).$$

Proof. Suppose that $\alpha \leqslant_H \beta$. Then there is a constant d such that for all n, $H(\alpha \upharpoonright n) \leqslant H(\beta \upharpoonright n) + d$. This happens iff there is a constant d' such that for all n,

$$2^{-H(\alpha\upharpoonright n)} \geqslant d' 2^{H(\beta\upharpoonright n)}.$$

This happens iff for some c and all n, $Z(\alpha \upharpoonright n) \geqslant cZ(\beta \upharpoonright n)$. The other direction is similar. $\square$

Remark. For any σ, $Z(\sigma)$ is random.

To see that the remark is true we use the Kraft-Chaitin Theorem to build a machine M and show that $\omega \leqslant_S Z(\sigma)$, where $\leqslant_S$ is Solovay reducibility (see [11] for a definition and discussion of Solovay reducibility). At stage s, if we see $U(\nu)\downarrow$, where U is the universal machine, we declare that $M(\nu) = \sigma$. Then for some $c = c_M$, there is a ν' with $U(\nu') = \sigma$ and furthermore $|\nu| \leqslant |\nu'| + c$. Thus whenever we add $2^{-|\nu|}$ to Ω, we add $2^{-(|\nu|+c)}$ to $Z(\sigma)$, and hence $\Omega \leqslant_S Z(\sigma)$, which implies $Z(\sigma)$ is random.

We now turn to the proof of Theorem 5.2. The idea of the proof is the following: We will use the Kraft-Chaitin Theorem to define a prefix-free machine M as follows. Whenever we see $Z_D(\sigma) \geqslant 2^{2r-n}$, where n is the current H-complexity of σ, we will enumerate an axiom $\langle n-r+1, \sigma\rangle$ (saying that some string of length $n-r+1$ is mapped to σ by M). For large enough r we will get to contradict the minimality of H. In detail, at stage s, we do the following.

For each $\sigma, n, r < s$, if

- σ, n, s is not yet attended to,
- $n > 2r \geqslant 2$,
- $H(\sigma) = n$, and
- $Z_D(\sigma) \geqslant 2^{2r-n}$, then attend to σ, r, n by enumerating an axiom $\langle n-r+1, \sigma\rangle$.

Notice that for any fixed σ, r, we put in axioms $\langle n-r+1, \sigma\rangle$ for descending values of n. Let $h_{\sigma,r}$ be the last value put in. We add at most

$$\sum_{n=0}^{\infty} 2^{-(h_{\sigma,r}-r+1+n)} = 2^{-h_{\sigma,r}+r}$$

to the measure of the domain of M.

When we put in the last axiom $\langle h_{\sigma,r} - r + 1, \sigma\rangle$, we see that $Z_D(\sigma) \geqslant 2^{2r-h_{\sigma,r}}$. Since D is prefix-free, for this fixed r we can conclude that

$$\sum_{\sigma} 2^{2r-h_{\sigma,r}} \leqslant 1.$$

Therefore,

$$2^r \sum_{\sigma} 2^{-h_{\sigma,r}+r} \leqslant 1.$$

Hence, for r we can add at most 2^{-r} to the measure of the domain of M. Thus, as $r \geqslant 1$, we can apply the Kraft-Chaitin Theorem to conclude that M exists.

Let c be such that

$$H(\sigma) \leqslant H_M(\sigma) + c.$$

Let $d = 2^{2(c+2)}$. Then we claim that

$$Z_D(\sigma) \leqslant d 2^{-H(\sigma)}.$$

To see this, let $r = c+2$. If $Z_D(\sigma) > 2^{2r-H(\sigma)}$, then eventually we put in an axiom $\langle H(\sigma) - r + 1, \sigma\rangle$, and hence $H_M(\sigma) \leqslant H(\sigma) - (c+1)$, a contradiction. This concludes the proof of Theorem 5.2.

This result allows us to get a analog of the result of Chaitin for the number of descriptions of a string.

5.4 Corollary. *There is a constant d such that for all c and all σ,*

$$|\{\nu : D(\nu) = \sigma \wedge |\nu| \leqslant H(\sigma) + c\}| \leqslant d2^c.$$

Proof. Trivially, $\mu(\{\nu : D(\nu) = \sigma \wedge |\nu| \leqslant H(\sigma) + c\}) \geqslant 2^{-(H(\sigma)+c)} \cdot |\{\nu : D(\nu) = \sigma \wedge |\nu| \leqslant H(\sigma) + c\}|$. But also, $\mu(\{\nu : D(\nu) = \sigma \wedge |\nu| \leqslant H(\sigma) + c\}) \leqslant d \cdot 2^{-H(\sigma)}$, by Theorem 5.3. Thus,

$$d2^{-H(\sigma)} \geqslant 2^{-c}2^{-H(\sigma)}|\{\nu : D(\nu) = \sigma \wedge |\nu| \leqslant H(\sigma) + c\}|.$$

Hence, $d2^c \geqslant |\{\nu : D(\nu) = \sigma \wedge |\nu| \leqslant H(\sigma) + c\}|$. □

We can now conclude that there are few H-trivials.

5.5 Theorem. *The set $S_d = \{\sigma : H(\sigma) < H(|\sigma|) + d\}$ has at most $O(2^d)$ many strings of length n.*

Proof. Given a universal prefix-free machine U, there is another machine V with the following property: V has for each n a program of length m (on which it converges) whenever the sum of all $2^{-|p|}$ such that $U(p)$ is defined and has length n is at least 2^{1-m}; furthermore V has for every n and every length m at most one program of length m. As U is universal, it follows that there is a constant c such that the following holds: If the sum of all $2^{-|p|}$ such that $U(p)$ is defined and has length n is at least 2^{c-m}, then there is a program q of length m with $U(q) = n$.

Let $m = H(n)$ and n be any length. There are less than 2^{d+c+1} many programs p of length $m + d$ or less such that $U(p)$ has length n, as otherwise the sum $2^{-|p|}$ over these programs would be at least 2^{c+1-m}, which would cause the existence of a program of length $m - 1$ for n, a contradiction to $H(n) = m$. So the set $S_d = \{\sigma : H(\sigma) < H(|\sigma|) + d\}$ has at most 2^{d+c+1} many strings of length n, where c is independent of n and d. □

5.6 Corollary. *(i)* [**Zambella [40]**] *For a fixed d, there are at most $O(2^d)$ many reals α with*

$$H(\alpha \upharpoonright n) \leqslant H(1^n) + d$$

for all n.

(ii) [**Chaitin [6]**] *If a real is H-trivial, then it is Δ^0_2.*

Proof. Consider the Δ^0_2 tree $T_d = \{\sigma : \forall \nu \subseteq \sigma(\nu \in S_d)\}$. This tree has width $O(2^d)$, and hence it has at most $O(2^d)$ many infinite paths. For each such path X, we can choose $\sigma \in T_d$ such that X is the only path above σ. Hence such X is Δ^0_2. □

The theorem above also has consequences for the possible degrees below $\mathbf{0}'$ of H-trivial sets. This generalizes a weaker form of Theorem 4.2 to all sets.

5.7 Theorem. *The degrees of H-trivial reals are bounded by a degree strictly below $\mathbf{0}'$.*

Proof. The infinite paths of the trees T_d above are uniformly computable from $\mathbf{0}'$, since $\mathbf{0}'$ can compute the real H-complexity of each 1^n, $n \in \mathbb{N}$. Thus, letting d vary, we can produce an enumeration $\{B_i : i \in \mathbb{N}\}$ of uniformly $\mathbf{0}'$-computable sets which consists of all the H-trivial reals. Moreover, by Theorem 6.3, the H-trivials are closed under join, so no finite join of the B_i can be Turing complete. Now we can use an oracle version of the thickness lemma, as in Theorem 4.2, to get a degree strictly below $\mathbf{0}'$ and above every member of the sequence of degrees of the B_i. □

6 Triviality and wtt-reducibility

Recall that $A \leqslant_{wtt} B$ iff there is a procedure Φ with computable use φ such that $\Phi^B = A$. As we have seen in earlier papers, *wtt*-reducibility seems to have a lot to do with randomness considerations. Triviality is no exception.

6.1 Theorem. *Suppose that $\alpha \leq_{wtt} \beta$ and β is H-trivial. Then α is H-trivial.*

Proof. For each computable $\varphi : \mathbb{N} \mapsto \mathbb{N}$,

$$H(\varphi(n)) \leqslant H(n) + O(1).$$

(To see this consider the prefix-free machine M such that for all σ, if $U(\sigma) = n$ then $M(\sigma) = \varphi(U(\sigma))$, where U is a universal prefix-free machine.)

Now suppose that $\alpha = \Phi^\beta$ with use φ and β is H-trivial. We have

$$H(\alpha \upharpoonright n) \leqslant H(\beta \upharpoonright \varphi(n)) + O(1), \text{ and hence,}$$

$$H(\alpha \upharpoonright n) \leqslant H(\varphi(n)) + O(1) \leqslant H(n) + O(1),$$

by the above and since β is H-trivial. □

We now show that the H-trivials are closed under $\oplus$, and hence form an ideal in the *wtt*-degrees. We begin by showing that the H-trivials are closed under addition.

6.2 Theorem. *If α and β are H-trivial then so is $\alpha + \beta$.*

Proof. Assume that α, β are two H-trivial reals. Then there is a constant c such that $H(\alpha \upharpoonright n)$ and $H(\beta \upharpoonright n)$ are both below $H(n) + c$ for every n. By Theorem 5.5 there is a constant d such that for each n there are at most d strings $\tau \in \{0,1\}^n$ satisfying $H(\tau) \leqslant H(n) + c$. Let e be the shortest program for n. One can assign to $\alpha \upharpoonright n$ and $\beta \upharpoonright n$ numbers $i, j \leqslant d$ such that they are the i-th and the j-th string of length n enumerated by a program of length up to $|e| + c$.

Let U be a universal prefix-free machine. We build a prefix-free machine V witnessing the H-triviality of $\alpha + \beta$. Representing i, j by strings of the fixed length d and taking $b \in \{0,1\}$, $V(eijb)$ is defined by first simulating $U(e)$ until an output n is produced and then continuing the simulation in order to find the i-th and j-th string α and β of length n such that both are generated by a program of size up to $n + c$. Then one can compute $2^{-n}(\alpha + \beta + b)$ and derive from this string the first n binary digits of the real $\alpha + \beta$. These digits are correct provided that e, i, j are correct and b is the carry bit from bit $n+1$ to bit n when adding α and β – this bit is well-defined unless $\alpha + \beta = z \cdot 2^{-m}$ for some integers m, z, but in that case $\alpha + \beta$ is computable and one can get the first n bits of $\alpha + \beta$ directly without having to do the more involved construction given here. □

6.3 Corollary. *The wtt-degrees containing H-trivials form an ideal in the wtt-degrees.*

Proof. By Theorem 6.2, we know that if α and β are H-trivial, then so is $\alpha + \beta$, where $+$ is normal addition. Now let $\alpha' = \alpha(0)0\alpha(1)0\ldots$, where $\alpha(n)$ is the nth bit of α, and let $\beta' = 0\beta(0)0\beta(1)\ldots$. Both α' and β' are H-trivial, since they have the same *wtt*-degrees as α and β, respectively. It follows that $\alpha' + \beta' = \alpha \oplus \beta$ is H-trivial. □

Theorem 6.2 suggests the question of whether addition is a join on the H-degrees. In general, it is not (for example, $\Omega \nleqslant_H \Omega + (1 - \Omega)$), but for c.e. reals it is. This fact considerably simplifies the analysis of the H-degrees of c.e. reals (compare for instance the difficulties in studying the sw-degrees considered in [12], many of which arise from the lack of a join operation).

6.4 Theorem. *If α, β are c.e. reals then the H-complexity of $\alpha + \beta$ is – up to an additive constant – the maximum of the H-complexities of α and β. In particular, $\alpha + \beta$ represents the join of α and β with respect to H-reducibility.*

Proof. Let $\gamma = \alpha + \beta$. Without loss of generality, the reals represented by α, β are in $(0, 1/2)$, so that we do not to have to care about the problem of represent digits before the decimal point. Furthermore, we have programs i, j, k which approximate α, β, γ, respectively, from below, such that at every stage and also for the limit the equation $\alpha + \beta = \gamma$ holds.

First we show that $H(\gamma \upharpoonright n) \leqslant \max\{H(\alpha \upharpoonright n), H(\beta \upharpoonright n)\} + c$ for some constant c. Fix a universal prefix-free machine U. It is sufficient to produce a prefix-free machine V that for each n computes $(\alpha + \beta) \upharpoonright n$ from some input of length up to $\max\{H(\alpha \upharpoonright n), H(\beta \upharpoonright n)\} + 2$.

V receives as input eab where $a, b \in \{0, 1\}$ and $e \in \{0, 1\}^*$. The length of the input is $|e| + 2$. First V simulates $U(e)$. In the case that this simulation terminates with some output σ, let $n = |\sigma|$. Now V simulates the approximation of α and β from below until it happens that either

- $a = 0$ and $\sigma = \alpha \upharpoonright n$ or
- $a = 1$ and $\sigma = \beta \upharpoonright n$.

Let $\widetilde{\alpha}, \widetilde{\beta}$ be the current values of the approximations of α and β, respectively, when the above simulation is stopped. Now V outputs the first n bits of the real $\widetilde{\alpha} + \widetilde{\beta} + b \cdot 2^{-n}$.

In order to verify that this works, given n, let a be 0 if the approximation of β is correct on its first n bits before the one of α and let a be 1 otherwise. Let e be the shortest program for $\alpha \upharpoonright n$ in case $a = 0$ and for $\beta \upharpoonright n$ in case $a = 1$. Then $U(e)$ terminates and $|e| \leqslant \max\{H(\alpha \upharpoonright n), H(\beta \upharpoonright n)\}$. In addition, we know both values $\alpha \upharpoonright n$ and $\beta \upharpoonright n$ once $U(e)$ terminates. So $\widetilde{\alpha}$ and $\widetilde{\beta}$ (defined as above) are correct on their first n bits, but it might be that bits beyond the first n cause a carry to exist which is not yet known. But we can choose b to be that carry bit and have then that $V(eab) = \gamma \upharpoonright n$.

For the other direction, we construct a machine W that computes $(\alpha \upharpoonright n, \beta \upharpoonright n)$ from any input e with $U(e) = \gamma \upharpoonright n$. The way to do this is to simulate $U(e)$ and, whenever it gives an output σ, to simulate the enumerations of α, β, γ until the current approximation $\widetilde{\gamma} \upharpoonright n = \sigma$. As $\widetilde{\alpha} + \widetilde{\beta} = \widetilde{\gamma}$, it is impossible that the approximations of α, β will later change on their first n bits if $\gamma \upharpoonright n = \sigma$. So the machine W then just outputs $(\widetilde{\alpha} \upharpoonright n, \widetilde{\beta} \upharpoonright n)$, which is correct under the assumption that e, and therefore also σ, are correct. □

Recall that a computably enumerable set X is (Kummer) complex iff $K(X \upharpoonright n) \geqslant 2 \log n - c$ infinitely often. (No computably enumerable set can have $K(X \upharpoonright n) \geqslant 2 \log n - c$ for all n (see [28], Exercise 2.58).) Recall also that, by [24], a computably enumerable degree $\mathbf{d}$ either has complex c.e. sets or every c.e. set $D \in \mathbf{d}$ is Kummer trivial in the sense that for all $\epsilon > 0$ there is a constant c such that for all n,

$$K(D \upharpoonright n) \leqslant (1 + \epsilon) \log n + d.$$

The relevant degrees containing the complex sets are the array noncomputable degrees of Downey, Jockusch and Stob. Recall that a very strong array $\{F_x : x \in \mathbb{N}\}$ is a strong array such that $|F_x| < |F_{x+1}|$ for all x. A c.e. set A is called anc relative to such a very strong array iff for all c.e. sets W there are infinitely many x such that $W \cap F_x = A \cap F_x$. A relevant fact for our purposes is the following.

6.5 Theorem (Downey, Jockusch and Stob [13, 14]). *For all wtt degrees $\mathbf{d}$, and all very strong arrays $\{F_x : x \in \mathbb{N}\}$, if $\mathbf{d}$ contains a set that is anc relative to some very strong array, then $\mathbf{d}$ contains one which is anc relative to $\{F_x : x \in \mathbb{N}\}$.*

We first show that array noncomputable *wtt*-degrees (i.e., ones containing anc c.e. sets) cannot be H-trivial.

6.6 Theorem. *Suppose that* $\mathbf{d}$ *is an array noncomputable c.e. wtt-degree. Then no set in* $\mathbf{d}$ *is* H*-trivial.*

Proof. We will build a prefix-free machine M. The range of M will consist of initial segments of 1^ω. By the recursion theorem, we can assume we know the coding constant d of our machine in the universal prefix-free machine U. Choose a very strong array such that $|F_e| = 2^{d+e+1}$. By Theorem 6.5, $\mathbf{d}$ contains a set A anc relative to this array. We claim that A is not H-trivial, and hence the wtt-degree $\mathbf{d}$ contains no H-trivials.

Suppose that A is H-trivial and that $A \leqslant_H 1^\omega$ with constant c. We will build a c.e. set V with $V \cap F_g \neq A \cap F_g$ for all $g > c$, contradicting the array noncomputability of A. For each e, we do the following. First we "load" 2^{d+e+1} beyond $\max\{x : x \in F_{e+c}\}$, by enumerating into our machine an axiom $\langle 2^{d+e+1}, 1^z\rangle$ for some (fresh) $z > \max\{x : x \in F_{e+c}\}$. When the universal machine responds it must respond by enumerating $\langle \sigma, A_s \upharpoonright z\rangle$, for some σ of length $\leqslant d+e+1+c$. At such a stage s, we then enumerate into V_s, our "kill" c.e. set, the least $p \in F_{e+c}$ not yet in V_s, making $F_{e+c} \cap A[s] \neq V_s \cap F_{e+c}[s]$. Notice that we only trigger enumeration into V at stages *after* a quantum of $2^{e+1+c+d}$ has been added to the measure of the domain of U. Now the possible number of changes we can put into V for the sake of $e+c$ is $|F_{e+c}|$, which is bigger than $2^{e+c+1+d}$. Hence A cannot respond each time, since if it did then the domain of U would have measure bigger than 1. $\square$

One might be tempted to think that the Kummer trivial c.e. sets and the H-trivial sets correspond. The next result shows that at least one inclusion fails. The proof is technical, and we leave a sketch of it to the appendix.

6.7 Theorem. *There is a c.e. Turing degree* $\mathbf{a}$ *which consists only of Kummer trivials but contains no* H*-trivials.*

Array noncomputable degrees have one further connection with our investigations. Recall that a set A is low for random iff every random set is still random relative to A. Kučera and Terwijn [22] were the first to construct such sets. They used a theorem of Sacks [32] to prove that any low for random set A must be of GL_1 Turing degree. That is, $A \oplus \emptyset' \equiv_T A'$. This was improved by Nies [31], who also showed that there are only countably many low for random sets, and that they are all Δ^0_2 and hence low (i.e., $A' \equiv_T \emptyset'$). The following result seems to be a theorem of Zambella. Following Ismukhametov [16] we call a set A *traceable* or *weakly computable* iff there is a computable function f such that for all $g \leqslant_T A$, there is a weak array $\{W_{h(x)} : x \in \mathbb{N}\}$ such that

(i) $|W_{h(x)}| \leqslant f(x)$ for almost all x and

(ii) $g(x) \in W_{h(x)}$ for all x.

Ismukhametov [16] observed that if a degree is weakly computable then it is array computable, and the notions coincide for computably enumerable sets. Ismukhametov proved the remarkable theorem that the c.e. degrees with strong minimal covers are exactly the weakly computable degrees. Furthermore, any weakly computable degree (in general) has a strong minimal cover.

6.8 Theorem. *Suppose that* A *is low for random. Then* A *is low* **(Nies)** *and additionally* A *must be weakly computable.*

Proof. (sketch) If one mimics the Terwijn-Zambella proof that Schnorr low sets are computably traceable ([37]), but using Martin-Löf lowness in place of Schnorr lowness, then the "if" direction proves the theorem. $\square$

7 Open Questions

The following are some interesting open questions about H-trivial reals.

1. Are the H-trivials downward closed under Turing reducibility?
2. Are all H-trivials low?
3. Do the classes of H-trivials and low for random sets coincide?
4. Is there an H-trivial that is not ω-c.e.? (A set is ω-c.e. if it has a computable approximation whose number of mind changes is computably bounded.)
5. Is there a maximal H-trivial c.e. real?
6. What can be said about resource bounded variations of the ideas presented here?
7. Given $A <_H \Omega$ and c.e., is it true that there exists a $B \equiv_H A$ with $A \not\equiv_T B$?
8. What can be said about the K-complexity of H-trivial reals?

References

[1] Ambos-Spies K., and A. Kučera, *Randomness in computability theory*, in *Computability Theory and its Applications*, (Cholak, Lempp, Lerman, Shore, eds) Contemporary Mathematics Vol. 257 (2000), 1-14.

[2] Calude, C., *Information Theory and Randomness, an Algorithmic Perspective,* Springer-Verlag, Berlin, 1994.

[3] Calude, C., Hertling, P., Khoussainov, B., Wang, Y., *Recursively enumerable reals and Chaitin's Ω number*, in STACS '98, Springer Lecture Notes in Computer Science, Vol 1373, 1998, 596-606.

[4] Calude, C., and A. Nies, *Chaitin's Ω numbers and strong reducibilities*, Journal of Universal Computer Science, 1997.

[5] Chaitin, G. *Information-theoretical characterizations of recursive infinite strings,* Theoretical Computer Science, vol. 2 (1976), 45-48.

[6] Chaitin, G., *A theory of program size formally identical to information theory*, Journal of the Association for Computing Machinery 22 (1975), pp. 329-340.

[7] Downey, R., Δ_2^0 degrees and transfer theorems, *Illinois J. Math*, **31** (1987) 419-427.

[8] Downey, R. and E. Griffiths, *On Schnorr randomness*, in preparation.

[9] Downey, R. and D. Hirschfeldt, *Aspects of Complexity,* (Short courses in complexity from the New Zealand Mathematical Research Institute summer 2000 meeting, Kaikoura) Walter De Gruyter, Berlin and New York, 2001, vi+172 pp.

[10] Downey, R. and D. Hirschfeldt, *Algorithmic Randomness and Complexity*, Springer-Verlag, in preparation.

[11] Downey, R., D. Hirschfeldt, and A. Nies, *Randomness, computability and density*, *SIAM Journal on Computing* **31** (2002) 1169–1183 . (extended abstract in proceedings of STACS 2001).

[12] Downey, R, D. Hirschfeldt, and G. Laforte, *Randomness and reducibility,* in proceedings of MFCS 2001.

[13] Downey, R., C. Jockusch and M. Stob, *Array nonrecursive sets and multiple permitting arguments,* in *Recursion Theory Week,* (Ambos-Spies, Muller, Sacks, eds.) Lecture Notes in Mathematics, vol. 1432 (1990), 141-174.

[14] Downey, R., C. Jockusch and M. Stob, *Array nonrecursive degrees and genericity,* (with C. Jockusch and M. Stob) in *Computability, Enumerability, Unsolvability,* (ed. Cooper, Slaman, and Wainer), London Mathematical Society Lecture Notes Series Vol 224, Cambridge University Press (1996), 93-105.

[15] Fortnow, L., *Kolmogorov complexity,* in [9], 73-86.

[16] Ishmukhametov, S., *Weak recursive degrees and a problem of Spector,* in *Recursion Theory and Complexity*, (ed. M. Arslanov and S. Lempp), de Gruyter, (Berlin, 1999), 81-88.

[17] Kautz, S., *Degrees of random sets,* Ph.D. Diss. Cornell, 1991.

[18] Ko, Ker-I, *Complexity of Real Functions*, Birkhäuser, Berlin, 1991.

[19] Ko, Ker-I, and H. Friedman, *On the computational complexity of real functions,* Theoretical Computer Science, 20 (1982), 323-352.

[20] Kučera, A., *On relative randomness,* Annals of Pure and Applied Logic, vol. 63, (1993), 61-67.

[21] Kučera, A., *On the use of diagonally nonrecursive functions,* in *Logic Colloquium, '87*, North-Holland, Amsterdam, 1989, 219-239.

[22] Kučera, A., and S. Terwijn, *Lowness for the class of random sets,* Journal of Symbolic Logic, vol. 64 (1999), 1396-1402.

[23] Kučera, A. and T. Slaman, *Randomness and recursive enumerability,* SIAM Journal of Computing, to appear.

[24] Kummer, M., *Kolmogorov complexity and instance complexity of recursively enumerable sets,* SIAM Journal of Computing, Vol. 25 (1996), 1123-1143.

[25] Kurtz, S., *Randomness and Genericity in the Degrees of Unsolvability*, Ph. D. Thesis, University of Illinois at Urbana,

[26] Ladner R. E. and L. P. Sasso, Jr., The weak truth table degrees of recursively enumerable sets, Ann. Math. Logic **8** (1975), 429-448.

[27] Levin, L., *Measures of complexity of finite objects (axiomatic description),* Soviet Mathematics Dolkady, vol. 17 (1976), 552-526.

[28] Li, Ming and Vitanyi, P., *Kolmogorov Complexity and its Applications*, Springer-Verlag, 1993.

[29] Loveland, D. *A variant of the Kolmogorov concept of complexity*, Information and Control, vol. 15 (1969), 510-526.

[30] Martin-Löf, P., *The definition of random sequences*, Information and Control, 9 (1966), 602-619.

[31] Nies, A., *Low for random sets are* Δ_2^0, Technical Report, University of Chicago, 2002.

[32] Sacks, G., *Degrees of Unsolvability,* Princeton Universit Press, 1963.

[33] Soare, R., *Recursively enumerable sets and degrees* (Springer, Berlin, 1987).

[34] Solomonoff, R., *A fromal theory of inductive inference, part 1 and part2,* Information and Control, 7 (1964), 224-254.

[35] Solovay, R. *Draft of paper (or series of papers) on Chaitin's work*, unpublished notes, May, 1975, 215 pages.

[36] Turing, A., *On computable numbers with an application to the Entscheidungsproblem*, Proceedings of the London Mathematical Society, 42 (1936), 230-265, correction in Proceedings of the London Mathematical Society, 43 (1937), 544-546.

[37] Terwijn, S. and D. Zambella, *Algorithmic randomness and lowness,* Journal of Symbolic Logic, vol. 66 (2001), 1199-1205.

[38] van Lambalgen, M, *Random Sequences,* Ph. D. Diss. University of Amsterdam, 1987.

[39] Weihrauch, K., *Computability*, Springer-Verlag, Berlin, 1987.

[40] Zambella, D, *On Languages with simple intial segments*, Technical Report, University of Amsterdam,

A Appendix

In this section we present a strengthening of Theorem 4.1 and a proof sketch of Theorem 6.7. We begin with a definition.

A.1 Definition. A c.e. set $T \subseteq \mathbb{N}$ is a *trace* if for some computable h, $\forall x\,(|T^{[x]}| \leqslant h(x))$.

A set A is traceable if there is a c.e. trace T such that $\forall \Gamma\,(\Gamma^A$ total $\Rightarrow \Gamma^A(m) \in T^{[m]}$ for almost every m).

The function $g(x) = \max T^{[x]}$ is wtt below K. Thus a traceable set A is array computable in the sense that some $g \leq_{wtt} K$ bounds all A-computable functions. This is a general definition of array computability given in [14], and implies low$_2$ness.

A.2 Theorem. *Each H-trivial real is traceable, which implies that it is array computable, and hence low$_2$.*

Proof. Suppose A is H-trivial via a constant b. Let (A_s) be a Δ_2 approximation of A. We will enumerate a trace for A. We will also enumerate a set L of axioms $\langle r, n\rangle$ such that $\sum_{\langle r,n\rangle \in L} 2^{-r} \leqslant 1$ (such a set is called a Kraft-Chaitin set, or KC set). For each n at most one axiom $\langle r_n, n\rangle$ will be enumerated. For $E \subseteq \mathbb{N}$, let the weight of E be defined by $wt(E) = \sum_{n\in E} 2^{-r_n}$.

The following technicality reduces the complexity of the construction: by the Recursion Theorem, we can assume that a c.e. index for L is given. Then, in fact, we can assume that an index d for a machine M_d corresponding to L by the Kraft-Chaitin Theorem is given, because, for any c.e. set $B \subseteq \mathbb{N} \times 2^{<\omega}$, we can effectively obtain an index for a KC set $\widetilde{B}$ such that $\widetilde{B} = B$ in case B already is a KC set. Let M_d be the machine effectively obtained from $\widetilde{B}$ via the Kraft-Chaitin Theorem. Our construction effectively produces a KC set L from d. Thus, if $B = L$, then B is a KC set and M_d is a machine for L.

Let $c = b + d$. Let (Γ_e) be a list of all Turing functionals which output natural numbers. For each e we run a construction which ensures that

$$\Gamma_e^A \text{ total } \Rightarrow \Gamma_e^A(m) \in T^{[m]} \text{ for almost every } m. \tag{1}$$

Let $k(e) = 2^{e+c+3}$. All the parameters below have a (suppressed) superscript e (we also write Γ for Γ_e). We say that a finite set $E \subseteq \mathbb{N}$ is a j-set at stage t if, for all $n \in E$, at some stage $u < t$ an axiom $\langle r_n, n\rangle$ went into L and now there are j distinct strings σ of length n such that $H_t(\sigma) \leqslant r_n + c$. A c.e. set with an enumeration $E = \bigcup E_t$ is a j-set if each E_t is a j-set at stage t.

Intuitively, the strings σ have the form $A_s \upharpoonright n$ at certain stages s, $u \leqslant s < t$. We will use the fact that A is H-trivial via b to obtain $k(e)$-sets of large weight if (1) fails. More precisely, we attempt to enumerate a $k(e)$-set $C_{k(e)}$ of weight $2^{-(e+2)}$, which is a contradiction since it would imply that the measure of the domain of the universal machine is $\geqslant 2$.

Let $k = k(e)$. To obtain C_k, we go through j-sets C_j $(1 \leq j \leqslant k)$ and j-sets D_j $(1 \leqslant j \leqslant k-1)$, $C_1 \supseteq D_1 \supseteq \cdots \supseteq D_{k-1} \supseteq C_k$. We describe procedures P_i $(1 < i \leqslant k)$ moving numbers n from D_{i-1} to C_i (thereby adding a further string σ as above), and procedures Q_j $(1 \leqslant j < k)$ enumerating C_1 (if $j = 1$) and moving numbers from C_j to D_j. Roughly the P procedures try to obtain A changes enumerating T, while the Q procedures deal with Γ changes.

In more detail, the P_i procedure works with numbers m and subsets D of D_{i-1} such that $\gamma(m) \leqslant \min(D)$, where γ is the use function of Γ. It enumerates $\Gamma^A(m)$ into $T^{[m]}$. If (1) fails then there are infinitely many m where this always leads to an $A \upharpoonright \gamma(m)$ change. To find an m, P_i must try all m above some parameter $v \in \mathbb{N}$. It calls, for all $m \geqslant v$ in parallel, a procedure Q_{i-1} with a parameter m, whose job it is to obtain an $i-1$-set D of sufficient weight and $\gamma(m) \leqslant \min(D)$ (calling procedures P_{i-1} unless already $i = 2$).

We write $P_{i,v}$ ($Q_{i,m}$) for the procedure with parameter v (m). $P_{i,v}$ has further parameters p (the goal) and α (the amount it is allowed to waste by leaving it in $D_{i-1} - C_i$). Similarly, $Q_{j,m}$ has goal q and trash quota β, the amount it may leave in $C_j - D_j$. All those are numbers of the form 2^{-l}, $l \in \mathbb{N}$. To ensure that L is a KC-set, we must limit the amount wasted. At any stage s, let $\alpha_i = 2^{-(e+2i+3+n_{P,i})}$, where $n_{P,i}$ is the number of P_i-procedures started so far at a stage $\leqslant s$ (while we may call several procedures at stage s, we think of this as being done in some order so that the above makes sense). Let $\beta_j = 2^{-(e+2j+2+n_{Q,j})}$, where $n_{Q,j}$ is the number of Q_j-procedures started so far at a stage $\leqslant s$. When P_i is called at stage s, its parameter α will be at most α_i, and similarly for Q_j. This ensures $wt(C_1 - C_k) \leq 2^{-(e+2)}$. Since we will be able to show that $wt(C_k) \leq 2^{-(e+2)}$ as well, L will be a KC set.

We now give the procedures in detail.

Procedure $P_{i,v}(p, \alpha)$, where $1 < i \leqslant k$ and $\alpha \leqslant p$.
Let $C = \emptyset$. Declare each m, $v \leqslant m < s$, available.
WHILE $wt(C) < p$ DO

P1 Declare $m = s$ available. For each m that is currently available, call the procedure $Q_{i-1,m}(2^{-m}\alpha, \beta)$, where $\beta = \min(2^{-m}\alpha, \beta_i)$. Declare m unavailable.

P2 When some procedure $Q_{i-1,m}$ returns a set D, put $\Gamma^A(m)[s]$ into $T^{[m]}$ (this will be defined). Declare m available.

P3 Suppose we carried out P2 at a previous stage and s is the first stage where $A \upharpoonright \gamma(m)$ has changed and $H_s(A \upharpoonright n) \leq r_n + c$ for each $n \in D$, the output of $Q_{i-1,m}$ at that stage. Put D into C and C_i, unless $wt(C \cup D) > p$ (P_i overfulfills its goal). In that case, choose a subset $\widetilde{D}$ of D such that $wt(C \cup \widetilde{D}) = p$, and put $\widetilde{D}$ into C and C_i (it is easy to see that $\widetilde{D}$ exists, since $p = 2^{-l}$ for some l, and $r_n > l$ for each $n \in D$ – now order the r_n in a nondecreasing way).

END WHILE. Return the set C.

Procedure $Q_{j,m}(q, \beta)$, where $0 < j < k$ and $\beta \leqslant q$.
WHILE $wt(D) < q$ DO

Q1 Wait for a stage such that $\Gamma^A(m)[s]$ converges.

Q2 Let $p = 2^{-(g_m+1)}\beta$, where g_m is the number of times the expression $A \upharpoonright \gamma(m)$ has changed so far.

Case (a). $j = 1$. Pick a fresh number $n \in \mathbb{N}^e$. Put $\langle r_n, n \rangle$ into L, where $2^{-(r_n)} = p$.

Case (b). $j > 1$. Call $P_{j,s}(p, \alpha)$, where $\alpha = \min(p, \alpha_j)$.

Wait for the first stage where one of the following applies, choose the first case which applies and take the indicated action.

Q3.1 $A \upharpoonright \gamma(m)$ changes. Terminate all subprocedures and goto Q1.

Q3.2 $j = 1$ and $H(A \upharpoonright n) \leqslant r_n + c$. Put n into D and D_i.

Q3.3 $j > 1$ and the procedure P_j returns C. Put C into D and D_i.

END WHILE. Return the set D.
Clearly $\Gamma^A(m)$ converges when $Q_{j,m}$ returns.

To enumerate T and L, at each stage s, start the procedure $P^e_{k(e),e}(1, \alpha^e_{k(e)})$. We call this run construction e.

Since we have a tree of procedures, each $n \in C_1$ corresponds to a unique procedure at each level (we say n belongs to that procedure). We now verify that L is a KC set and T is a trace.

A.3 Lemma. *a) Let $1 \leqslant j < k$. The weight of all numbers in $C_j - D_j$ which belong to $Q_{j,m}(q, \beta)$ is at most β.*

b) Let $1 < i \leqslant k$. The weight of all numbers in $D_{i-1} - C_i$ which belong to $P_{i,v}(p, \alpha)$ is at most α.

Proof. a) Suppose n belongs to $Q_{j,m}(q, \beta)$ and is in C_j at stage t. If n does not reach D_j, this is because $Q_{j,m}$ was interrupted after t, or $A \upharpoonright \gamma(m)$ changed (i.e., we are in Q3.1), or, for $j > 1$, the run of $P_{j,v}$ which put n into C_j does not return. At a stage s where $A \upharpoonright \gamma(m)$ has changed g times so far, the total weight of such n is at most $p = 2^{-(g+1)}\beta$, since for $j > 1$ the run of $P_{j,v}$ has not yet reached its goal. Together this makes at most β.

b) Suppose n belongs to $P_{i,v}(p, \alpha)$ and is in D_{i-1} at stage t. It was put there during a run of a $Q_{i-1,m}$ procedure called by $P_{i,v}$. If n does not reach C_i, this is for one of the following reasons:

(i) $P_{i,v}$ does not return

(ii) After we put $\Gamma^A(m)[s]$ into $T^{[m]}$ at P2, we wait forever for a stage s as in P3.

(iii) $P_{i,v}$ did not need n to reach its goal.

(iv) $P_{i,v}$ is terminated (by a Q procedure at a higher level).

Any of these events can happen at most once for each $m \geqslant v$. The total weight of those n for each m is at most $2^{-m}\alpha$. □

A.4 Lemma. $wt(C_1 - C_{k(e)}) \leqslant 2^{-(e+2)}$.

Proof. By the previous lemma and the choice of the values α_i, β_j. □

A.5 Lemma. $wt(C_{k(e)}) \leqslant 2^{-(e+2)}$.

Proof. Let $k = k(e)$. C_k is a k set, so for all $n \in C_k$, there are k distinct strings σ of length n such that $H(\sigma) \leq r_n + c$. Then $1 \geqslant \mathrm{dom}(U) \geqslant k \sum_{n \in C_k} 2^{-(r_n+c)}$. Since $k = 2^{(e+2+c)}$, this implies $wt(C_k) \leq 2^{-(e+2)}$. □

We conclude that L is a KC-set.

A.6 Lemma. *T is a c.e. trace.*

Proof. It is sufficient to give a computable bound on the part of $|T^{[m]}|$ enumerated by the e-th construction, uniformly in e, because we start it by calling $P^e_{k(e),e}$, so only a construction $e \leqslant m$ can enumerate $|T^{[m]}|$.

A procedure $P_{i,v}$ initiated at s only calls procedures $Q_{i-1,m}$ for $m \geqslant s$. Thus it suffices to consider those finitely many $P_{i,v}(g, \alpha)$ running by stage m. Each one needs at most $2^m g/\alpha$ enumerations into $T^{[m]}$ to reach its goal via calls to $Q_{i-1,m}$. Take the maximum of those values over all relevant procedures to obtain the computable bound. □

A.7 Lemma. *A is traceable via T.*

Proof. Suppose Γ^A_e is total but there are infinitely many m such that $\Gamma^A_e(m) \notin T^{[m]}$. We claim that each procedure returns unless terminated. This will yield a contradiction, since we call $P^e_{k(e),e}$ with goal 1, and this procedure is not terminated, so it enumerates weight 1 into $C_{k(e)}$.

Again let $k = k(e)$. To prove the claim we induct on levels of procedures Q_1, P_2, Q_2, …, Q_{k-1}, P_k.

$Q_{j,m}(q, \beta)$: Since $\Gamma^A_e(m)$ converges, eventually $A \upharpoonright \gamma(m)$ reaches a final value. From then on, each time we add a fixed quantity p to the weight of D in Q3.2 or Q3.3, using that for $j > 1$, inductively the procedure $P_{j,v}$ returns as it is not terminated. Thus $Q_{j,m}$ reaches its goal.

$P_{i,v}(p, \alpha)$: Choose $m \geqslant v$ such that $\Gamma_e^A(m) \notin T^{[m]}$. Each time we call $Q_{i-1,m}$, by induction it returns, as it is not terminated, and we get the desired stage s in P3 so we can call $Q_{i-1,m}$ again. Thus $P_{i,v}$ reaches its goal. □

This completes the proof of Theorem A.2. □

6.7. Theorem. *There is a c.e. Turing degree* **a** *which consists only of Kummer trivials but contains no H-trivials.*

Proof. We merely sketch the proof since it is quite technical, and certainly not the central result of the paper. We will construct a contiguous c.e. degree **a** containing no H-trivials. A *contiguous* degree is a Turing degree that contains a single *wtt*-degree. Such degrees were first constructed by Ladner and Sasso [26]. By Downey [7], every contiguous c.e. degree is array computable. Hence, by Kummer's theorem, all of its members are Kummer trivial. The argument is a Π_2^0 priority argument using a tree of strategies. We shall meet the requirements below:

$$\mathcal{R}_e : \Delta_e^A = B \wedge \Phi_e^B = A \text{ implies } A \equiv_{wtt} B.$$

There is a standard way to do this via *dumping* and *confirming*. Specifically, one has a priority tree $PT = \{\infty, f\}^{<\omega}$ with versions of $\mathcal{R}_e$ having outcomes $\infty <_L f$. The outcome ∞ is meant to say that $\Delta_e^A = B \wedge \Phi_e^B = A$. The other outcome is meant to say that the limit of the length of the agreement function

$$\ell(e, s) = \max\{x : \forall y \leqslant x(\Phi_e^B(y) = A(y) \wedge \forall z \leqslant \varphi(y)(\Delta_e^A(z) = B(z)[s]))\},$$

the so-called A-controllable length of agreement, is finite.

The usual H-nontriviality requirements are that

$$\mathcal{P}_e : A \text{ is not } H\text{-trivial with constant } e.$$

These are met, as one would expect, by changing A sufficiently often when the universal prefix-free machine U threatens to demonstrate A has the same H-complexity as $\mathbb{N}$ up to the additive constant e. We will discuss this further below.

As we see below, versions of $\mathcal{P}$ type requirements generate followers. The $\mathcal{R}$ requirements refine the collections of requirements into a well-behaved stream. Their action is purely negative. For a version of $\mathcal{P}$ guessing ∞ (so associated with some node $\beta \supset \alpha$), if it generates a follower x, then that will only happen at an $\alpha\widehat{\ }\infty$ (i.e., α-expansionary) stage s_0. (Note that any follower with a weaker guess will be canceled at such a stage.) Then at the next $\alpha\widehat{\ }\infty$ stage, we will *confirm* the number x. This means that we cancel all numbers $\geqslant x$ (which will necessarily be weaker than x). Thus x can only enter A after it needs to and at a β stage; at which point it must be β-confirmed. (That is, α-confirmed for all $\alpha\widehat{\ }\infty \subseteq \beta$.) Finally, we *dump* in the sense that if we ever enumerate x into A at stage s, then we promise to also enumerate z for $x \leqslant z \leqslant s$ into A.

It is a standard argument to show that in the limit, for any follower x that survives $\alpha\widehat{\ }\infty$-stages, $A \upharpoonright x$ can be computed from the least $\alpha\widehat{\ }\infty$-stage $s_1 > s_0$ where $B_{s_1} \upharpoonright s_0 = B \upharpoonright s_0$ (assuming the standard convention that uses at stage s are bounded by s). Similarly, it is also a standard argument to prove that if x is the least α-confirmed follower appointed at some $\alpha\widehat{\ }\infty$-stage t with $\ell(e, s) > z$, then $B \upharpoonright z = B_s \upharpoonright z$ where $s > t$ is the least $\alpha\widehat{\ }\infty$-stage with $A_s \upharpoonright x = A \upharpoonright x$. More details can be found in, for example, Downey [7].

Returning to the $\mathcal{P}$ requirements, by the recursion theorem we have at our disposal 2^{-d} we can dump in to the universal machine, and we know d. Split this up into chunks for the relevant requirement. That is, for each $\mathcal{P}_\alpha$ on the priority tree, we will set aside $2^{-k(\alpha)}$ for this requirement. These will all be disjoint sections of 2^{-d}, and will have no interference for differing α.

Suppose the relevant guess for $\mathcal{P}_e$ on the true path is α, and we are at stage n where this guess has priority, so we are never again to the left of $\mathcal{P}_\alpha$, and hence the chunk for $\mathcal{P}_\alpha$ is now stable in the sense that no higher priority cancellations or confirmations will affect numbers from this block. Let's suppose that we have 2^{-k} devoted to this version of $\mathcal{P}_e$.

We will wait till there are a large number of α-confirmed followers for $\mathcal{P}_\alpha$. Something like $2^{2^{2^{k+f}}}$ will be enough, where f is the absolute constant for the number of H-trivials with constant e given by Zambella's theorem. Specifically, pick a big x_1 at an α-stage, then when the total length of agreement is bigger than x_1, initialize and α-confirm, as usual. Then pick x_2, etc., till the whole entourage of followers are stable.

Let x_n be the largest. This is where we will kill $\mathcal{P}_e$. The first action is to enumerate $\langle k, 1^{x_n}\rangle$ as an axiom for M. Thus we are saying that the H-complexity of 1^{x_n} is at most k. If we see $U(\tau)\downarrow = A \upharpoonright x_n$, with $|\tau| \leqslant k + e$, then we will change A. This is done using the followers in reverse order, first x_n, then later, x_{n-1} if necessary, etc. The reverse order guarantees the contiguity as usual.

Note that U has the option of later choosing something shorter than k to compute 1^{x_n}, but this can only happen k times, and we have enough x_i's to cope with this.

The remaining details are to implement this strategy on a priority tree. $\square$