Trivial Reals*

Rod G. Downey[†]
School of Mathematical and Computing Sciences
Victoria University of Wellington
New Zealand

Denis R. Hirschfeldt[‡]
Department of Mathematics
University of Chicago
U.S.A.

André Nies
Department of Computer Science
Auckland University
New Zealand

Frank Stephan[§]
Mathematical Institute
University of Heidelberg
Germany

Abstract

Solovay showed that there are noncomputable reals α such that $H(\alpha \upharpoonright n) \leqslant H(1^n) + O(1)$, where H is prefix-free Kolmogorov complexity. Such H*-trivial* reals are interesting due to the connection between algorithmic complexity and effective randomness. We give a new, easier construction of an H-trivial real. We also analyze various computability-theoretic properties of the H-trivial reals, showing for example that no H-trivial real can compute the halting problem. Therefore, our construction of an H-trivial computably enumerable set is an easy, injury-free construction of an incomplete computably enumerable set. Finally, we relate the H-trivials to other classes of "highly nonrandom" reals that have been previously studied.

*Some of the material in this paper was presented by Downey in his talk *Algorithmic Randomness and Computability* at the 8th Asian Logic Meeting in Chongqing, China. A preliminary version of this paper appeared as an extended abstract in Brattka, Schröder, and Weihrauch (eds.), *Computability and Complexity in Analysis*, Malaga, Spain, July 12–13, 2002, *Electronic Notes in Theoretical Computer Science* 66, vol. 1, 37–55.

[†]Supported by the Marsden fund of New Zealand.

[‡]Partially supported by NSF Grant DMS-02-00465.

[§]Supported by the Heisenberg program of the Deutsche Forschungsgemeinschaft (DFG), grant no. Ste 967/1–2.

1 Introduction

Our concern is the relationship between the intrinsic computational complexity of a real and the intrinsic randomness of the real. Downey, Hirschfeldt, LaForte and Nies [8, 9] 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 $K(\alpha \upharpoonright n) \leqslant K(\beta \upharpoonright n) + O(1)$, 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 [8, 9] 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 $\Omega = \sum_{\sigma \in \mathrm{dom}(M)} 2^{-|\sigma|}$. 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 $\alpha \leqslant_R \beta$ (for $R \in \{H, K\}$), 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 (= recursively enumerable) reals, that is to the 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 $\alpha \leqslant_R \beta$ then $\alpha \leqslant_T \beta$. However, this is not true in general.

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 [20], Chaitin [3] proved that if α is a real with $K(\alpha \upharpoonright n) \leqslant K(1^n) + O(1)$ 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 $H(\alpha \upharpoonright n) \leqslant H(1^n) + O(1)$ 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 [27] proved that there are Δ^0_2 noncomputable reals β such that $H(\beta \upharpoonright n) \leqslant H(1^n) + O(1)$. 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. (Such a proof also appears in Vereshchagin [30].) To state an extension of this result, we need another triviality notion.

Answering a question of Kučera and of van Lambalgen, Kučera and Terwijn [15] constructed a set X which is low for random. Here we say that X is low for random (also known as Martin-Löf-low) if 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. (Recently, Nies [23] has shown that in fact a real is H-trivial if and only if it is 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 real can be Turing complete, or even high. (Nies [23] has extended this result by showing that every H-trivial real is low.) An immediate application of this result is that the construction of a noncomputable H-trivial real provides a very simple injury-free solution to Post's problem. Indeed, in Section 3 we give an alternate construction of a noncomputable H-trivial real that is not only injury-free but priority-free, in a sense that will be discussed in that section.

We also prove that there is an effective listing of the H-trivial reals along with constants witnessing their H-triviality. This is in contrast to the strongly H-trivial reals, which have no such listing, as shown by Nies [24]. (A real is strongly H-trivial if H-complexity relativized to A is the same as H-complexity, up to an additive constant.) Recently, Nies and Hirschfeldt (see Nies [23]) have shown that a real is H-trivial if and only if it is strongly H-trivial. This was conjectured by Hirschfeldt and obtained as a direct modification of Nies' result that the H-trivial reals are downward closed under Turing reducibility. (See the introduction to [23] for more details on the history of this result.) Our result shows that there is no computable way of passing from a constant witnessing H-triviality to one witnessing strong H-triviality.

In an unpublished report, Zambella [31] proved that there is a computable function f such that for each c there are at most $f(c)$ many reals α with

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

We will give a unified proof of this result and Chaitin's result that every H-trivial real is Δ^0_2.

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

The study of relative randomness seems intimately related to weak truth table reducibility. Recall that $A \leqslant_{wtt} B$ if 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 form an ideal in the wtt-degrees.

Related to the topic of H-triviality is work of Kummer [16]. Kummer investigated "Kummer trivial" computably enumerable sets. In terms of classical (non-prefix-free) Kolmogorov complexity, we know that if A is a computably enumerable set then $K(A \upharpoonright n) \leqslant 2 \log n + O(1)$ 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 computably enumerable $A \in \mathbf{a}$ are "Kummer trivial" in the sense that $K(A \upharpoonright n) \leqslant (1+\epsilon) \log n + O(1)$ for all $\epsilon > 0$. (By Chaitin's work [3], if $K(A \upharpoonright n) \leqslant \log n + O(1)$ then A is computable, so this result is sharp.) Kummer proved that the degrees containing such complicated sets are exactly the array noncomputable (= array nonrecursive) degrees (see Section 7 for a definition). We prove that (I) no 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 Kolmogorov complexity and K for non-prefix-free complexity. Following a recent proposal to change terminology, we call the recursively enumerable sets computably enumerable and the array nonrecursive sets array noncomputable. The remaining computability-theoretic notation follows Soare's textbook [26].

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 computably enumerable if its left cut is computably enumerable 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 $M(\tau)\downarrow \Rightarrow M(\tau')\uparrow$ for all finite binary strings $\tau \subsetneq \tau'$. It is *universal* if for each prefix-free machine N there is a constant c such that, for all binary strings τ, if $N(\tau)\downarrow$ then $M(\mu)\downarrow = N(\tau)$ for some μ with $|\mu| \leqslant |\tau| + c$. We call c the *coding constant* of N.

For a prefix-free machine M and a binary string τ, let $H_M(\tau)$ be the length of the shortest binary string σ such that $M(\sigma)\downarrow = \tau$, if such a string exists, and let $H_M(\tau)$ be undefined otherwise. We fix a universal prefix-free machine U and let $H(\tau) = H_U(\tau)$. The number $H(\tau)$ is the *prefix-free complexity* of τ. (The choice of U 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 $H(\alpha \upharpoonright n) \geqslant n - O(1)$. There are several equivalent definitions of 1-randomness, the best-known of which is due to Martin-Löf [21]. References on algorithmic complexity and effective randomness include Ambos-Spies and Kučera [1], Calude [2], Chaitin [4], Downey and Hirschfeldt [7], Fortnow [12], Kautz [14], Kurtz [17], Li and Vitanyi [19], and van Lambalgen [29].

The above definitions can be relativized to any set A in the obvious way. The prefix-free complexity of σ relative to A is denoted by $H^A(\sigma)$.

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

2.1 Theorem (Kraft-Chaitin). *From a computably enumerable 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*.*

A sequence satisfying the hypothesis of the Kraft-Chaitin Theorem is called a *Kraft-Chaitin sequence.*

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 [27]. The proof there is complicated and only constructs a Δ^0_2 real.

3.1 Theorem (after Solovay [27]). *There is a noncomputable computably enumerable set* A *such that* $H(A \upharpoonright n) \leqslant H(n) + O(1)$*.*

3.2 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 the Kraft-Chaitin Theorem it would be enough to build M *implicitly*, enumerating the length axiom $\langle |\sigma|, 0^n\rangle$. 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 the Kraft-Chaitin Theorem 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 ensure that $\sum_{\tau\in\mathrm{dom}(M)} 2^{-|\tau|} < 1/2$.

Proof of Theorem 3.1. The idea is the following. We will build a noncomputable computably enumerable 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 an axiom $\langle |\sigma| + 1, A_s \upharpoonright n\rangle$ 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 e-th computably enumerable set and $H(j)[s]$ is the stage s approximation to the H-complexity of j.

Clearly A is computably enumerable. 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 the result, which appears to be due to Muchnik, that there exists a noncomputable computably enumerable set A that is strongly H-trivial, in the sense that H-complexity relativized to A is the same as H-complexity, up to an additive constant, i.e., $\forall \sigma \, (H(\sigma) \leqslant H^A(\sigma) + c)$. Such an A is both H-trivial and low for random. (As mentioned above, Nies and Hirschfeldt (see Nies [23]) improved the result by showing that a real is H-trivial if and only if it is strongly H-trivial.)

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 nonhigh (and in fact, as shown by Nies [23], low).

As we see in the next section, the construction above automatically yields a Turing incomplete computably enumerable set. It is thus an *injury-free* solution to Post's problem. It is not, however, *priority-free*, in that the construction depends on an ordering of the simplicity requirements, with stronger requirements allowed to use up more of the domain of the machine M. We can do methodologically better by giving a priority-free solution to Post's problem, in the sense that no explicit diagonalization (such as that of W_e above) occurs in the construction of the incomplete computably enumerable set, and therefore the construction of this set (as opposed to the verification that it is H-trivial) does not depend on an ordering of requirements. We now sketch this method, which is rather more like that of Solovay's original proof of the existence of a Δ_2^0 H-trivial real.

Let us reconsider the key idea in the proof of Theorem 3.1. At certain stages we wish to change an initial segment of A for the sake of diagonalization. Our method is to make sure that the total measure added to the domain of our machine M (which proves the H-triviality of A) due to such changes is bounded by 1. Suppose, on the other hand, we were *fortunate* in the sense that the universal machine itself "covered" the measure needed for these changes. That is, suppose we were lucky enough to be at a stage s where we desire to put n into $A_{s+1} - A_s$ *and* at that very stage $H_s(j)$ changes for all $j \in \{n, \ldots, s\}$. That would mean that *in any case* we would need to enumerate new axioms describing $A_{s+1} \upharpoonright j$ for all $j \in \{n, \ldots, s\}$, whether or not these initial segments change. Thus at that very stage, we could also change $A_s \upharpoonright j$ for all $j \in \{n, \ldots, s\}$ at no extra cost.

Notice that we would not need to copy the universal machine U at every stage. We could also enumerate a collection of stages $t_0, t_1 \ldots$ and only update M at stages t_i. Thus for the lucky situation outlined above, we would only need the approximation to $H(j)$ to change for all $j \in \{n, \ldots, t_s\}$ at some stage u with $t_s \leqslant u \leqslant t_{s+1}$. This observation would seem to allow a greater possibility for the lucky situation to occur, since many more stages can occur between t_s and t_{s+1}.

The key point in all of this is the following. Let $t_0, t_1, \ldots$ be a computable collection of stages. *Suppose that we construct a set* $A = \bigcup_s A_{t_s}$ *so that for* $n \leqslant t_s$, *if* $A_{t_{s+1}} \upharpoonright n \neq A_{t_s} \upharpoonright n$ *then* $H_{t_s}(j) < H_{t_{s+1}}(j)$ *for all* j *with* $n \leqslant j \leqslant t_s$. *Then* A *is* H*-trivial.*

We are now ready to define A in a priority-free way.

Let $t_0, t_1, \ldots$ be a collection of stages such that t_i as a function of i dominates all primitive recursive functions. (Actually, as we will see, dominating the overhead in the Recursion Theorem is enough.) At each stage u, let $\{a_{i,u} : i \in \omega\}$ list $\overline{A}_u$. Define

$$A_{t_{s+1}} = A_{t_s} \cup \{a_{n,t_s}, \ldots, t_s\},$$

where n is the least number $\leqslant t_s$ such that $H_{t_{s+1}}(j) < H_{t_s}(j)$ for all $j \in \{n, \ldots t_s\}$. (Naturally, if no such n exists, $A_{t_{s+1}} = A_{t_s}$.) Requiring the complexity change for all $j \in \{n, \ldots, t_s\}$, rather than just $j \in \{a_{n,t_s}, \ldots, t_s\}$, ensures that A is coinfinite, since for each n there are only finitely many s such that $H_{t_{s+1}}(n) < H_{t_s}(n)$.

Note that there is no priority used in the definition of A. It is like the Dekker deficiency set or the so-called "dump set" (see [26, Theorem V.2.5]).

It remains to prove that A is noncomputable. By the Recursion Theorem, we can build a prefix-free Turing machine M and know the coding constant c of M in U. That is, if we declare $M(\sigma) = j$ then we will have $U(\tau) = j$ for some τ such that $|\tau| \leqslant |\sigma| + c$. Note further that if we put σ into the domain of M at stage t_s, then τ will be in the domain of U by stage $t_{s+1}-1$. (This is why we chose the stages to dominate the primitive recursive functions. This was the key insight in Solovay's original construction.)

Now the proof looks like that of Theorem 3.1. We will devote 2^{-e} of the domain of our machine M to making sure that A satisfies the e-th simplicity requirement. When we see a_{n,t_s} occur in W_{e,t_s}, where $\sum_{n\leqslant j\leqslant t_s} 2^{-H_{t_s}(j)} < 2^{-(e+c+1)}$, we change the M_{t_s} descriptions of all j with $n \leqslant j \leqslant t_s$ so that $H_{t_{s+1}}(j) < H_{t_s}(j)$ for all such j. The cost of this change is bounded by 2^{-e}, and a_{n,t_s} will enter $A_{t_{s+1}}$, as required.

4 Turing degrees of H-trivials

In this section we give a proof that every H-trivial real α is Turing incomplete, and in fact not high (i.e., $\alpha' <_T \emptyset''$).

4.1 Theorem. *If the real A is H-trivial then A is not high.*

Proof. By Corollary 6.7 part (II), we can choose a Δ^0_2 approximation for A. We enumerate a Kraft-Chaitin sequence L of axioms of the form $\langle r_n, n\rangle$. If $E \subseteq \mathbb{N}$, the *weight* of E is $\sum_{n\in\mathbb{N}} 2^{-r_n}$.

Our enumeration of L depends on the behavior of a total Turing reduction Γ^A. At first we will be quite general about what Γ is, but we adopt the usual conventions on the use γ of Γ.

The basic idea is as follows. We enumerate an axiom $\langle r, n\rangle$ into L for some small r, thus ensuring that $H(n)$ is small. Each time $A \upharpoonright n$ changes, the "opponent" has to provide a corresponding short description of $A \upharpoonright n$ via the fixed universal prefix-free machine U. Thus we make the opponent load up many descriptions of approximations to $A \upharpoonright n$, while we enumerate only one short description of n. We will be able to argue that there are enough $A \upharpoonright n$-changes for certain n, which will be picked so that $n > \gamma(m)$ for certain numbers m.

We first consider the simpler case where we only want to show that A is Turing incomplete. Fix a Turing reduction $K = \Psi^A$. We build a computably enumerable set B. By the Recursion Theorem, we can assume that we know in advance a computably enumerable index i for B. Since B equals the i-th row of K, this means we also know a total reduction Γ such that $\Gamma^A = B$. (For the full construction, we will use the fact that if $\emptyset'' \leqslant_T A'$ then there is a reduction Γ such that $f = \Gamma^A$ eventually dominates each computable function.)

Let b be a number such that $\forall n\,(H(A \upharpoonright n) \leqslant H(n) + b)$. By the Recursion Theorem we can assume that the coding constant d for the prefix-free machine M corresponding to L via the Kraft-Chaitin theorem is known ahead of time. Let $c = b + d$, so that $\forall n\,(H(A \upharpoonright n) \leqslant H_M(n) + c)$.

All values appended by $[s]$ will be taken at the end of the stage. We carry out our construction at stages $s_0 < s_1 < \cdots$ so that $\forall i \, \forall n \leqslant s_i \, (H(A \upharpoonright n)[s_{i+1}] \leqslant H_M(n)[s_{i+1}] + c)$ and $\forall i \, \forall n \leqslant s_i \, (\Gamma^A(n)[s_{i+1}] \downarrow)$.

Let $k \in \mathbb{N}$ be a sufficiently large number (the precise value will be determined below). We describe procedures $P_i(g)$, $0 \leqslant i < k$, where the parameter g is a rational of the form 2^{-x}, $x \in \mathbb{N}$. Each procedure P_i, $0 \leqslant i < k-1$, calls P_{i+1} many times.

The basic idea is that, for each n we work with, each level of procedures is responsible for creating a change in $A \upharpoonright n$, thus forcing the opponent to provide a short description of a new string of length n. For certain n, however, changes will come too early, thus creating a certain amount of "trash", that is, axioms enumerated into L that do not cause the appropriate number of short descriptions to appear in U. We will need to show that this trash is small enough that it will not cause us problems.

The bottom procedure $P_{k-1}(g)$.

The output of this procedure (if the procedure returns at all) is a set $C = C_{k-1}$ of numbers n such that

(I) first we put an axiom $\langle r_n, n \rangle$ into L (and therefore have $M(\sigma) = n$ for some σ of length r_n), and

(II) then we see strings σ_0, σ_1 of length n with U-descriptions of length $\leqslant r_n + c$. (The strings are approximations to $A \upharpoonright n$ at certain stages.)

Moreover, the weight $\sum_{n \in C} 2^{-r_n}$ of C equals g, and the "trash" put into L by this procedure, namely $\sum_{n \notin C} 2^{-r_n}$, is at most $2^{-2(k-1)}g$.

Here is the procedure:

1. Choose a fresh number $m = m_{k-1}$.
2. Let $q_{k-1} = 0$ and $C = \emptyset$.

   WHILE $q_{k-1} < g$:

   (a) Choose a fresh number n (in particular, $n > \gamma(m)$) and put an axiom $\langle r, n \rangle$ into L, where r is given by $2^{-r} = 2^{-2(k-1)} 2^{-(u+1)} g$, and u is the number of times the expression $A \upharpoonright \gamma(m)$ has changed so far.

   (b) At the next stage (in the sense described above):

   IF $A \upharpoonright \gamma(m)$ changed, add 2^{-r} to the real *trash* (which is global for the construction, being 0 initially), and continue at 2(a).

   ELSE put n into C and add 2^{-r} to q_{k-1}.

   END WHILE

3. Wait for an $A \upharpoonright \gamma(m)$ change. If this happens, RETURN the set C.

   (In the case where we only want to show that A is not Turing complete, we are enumerating B, and we know $B = \Gamma^A$. Thus we may simply put m into B. For the proof of the full theorem, it remains to show that this A-change happens often enough.)

This completes the description of the procedure.

Note that the procedure can only get stuck at step 3, because we assume $A \upharpoonright \gamma(m)$ settles.

We verify the required property (II) of C. If n gets into C at a stage s, then $\gamma(m) < n$ (and we see a short description of $A[s] \upharpoonright n$). Since we assume the procedure returns at a stage $t > s$, $A \upharpoonright \gamma(m)$ has changed from its value at s, so at t there is a short description of a second string of length n.

Clearly $\sum_{n \in C} 2^{-r_n} = g$, since g has the form 2^{-x} and we stop when q_{k-1} reaches this value. Moreover, *trash* $\leqslant 2^{-2(k-1)} g \sum_{u \geqslant 0} 2^{-(u+1)} \leqslant 2^{-2(k-1)} g$.

The procedure $P_i(g)$ for $0 \leqslant i < k-1$.

The output of this procedure (if the procedure returns at all) is a set $C = C_i$ of numbers n such that

(I) first we put an axiom $\langle r_n, n \rangle$ into L, and

(II) then at later stages we see strings σ_j $(0 \leqslant j \leqslant k-i+2)$ of length n with U-descriptions of length $\leqslant r_n + c$. (Again, these strings are approximations to $A \upharpoonright n$ at those stages.)

Moreover, the weight $\sum_{n \in C} 2^{-r_n}$ of C is g, and the trash put into L by this procedure, namely $\sum_{n \notin C_i} 2^{-r_n}$, is at most $2^{-2i} g$.

Here is the procedure:

1. Choose a fresh number $m = m_i$.

2. Let $q_i = 0$ and $C = C_i = \emptyset$.

   WHILE $q_i < g$:

   (a) Call $P_{i+1}(g_{i+1})$, where $g_{i+1} = 2^{-2(i+1)} 2^{-(u+1)} g$, and u is the number of times the expression $A \upharpoonright \gamma(m)$ has changed so far. If $A \upharpoonright \gamma(m)$ changes during the execution of P_{i+1}, put the current $\sum_{j>i} q_j$ into *trash* and end the subprocedures called by P_i.

   (b) If P_{i+1} returns a set C_{i+1} then put this set into C_i, and add g_{i+1} to q_i.

   END WHILE

3. Wait for an $A \upharpoonright \gamma(m)$ change. If this happens RETURN the set C.

This completes the description of the procedure.

Again, we verify the required property (II) of C_i. When P_{i+1} returns a set C_{i+1} at stage s, by induction, for each $n \in C_{i+1}$ we have already seen short descriptions of distinct strings σ_j $(0 \leqslant j < k-i+1)$ of length n. Since this run of P_{i+1} was not stopped, $A \upharpoonright \gamma(m_i)$ did not change during this run, and in particular $\gamma(m_i) < n$. If $A \upharpoonright \gamma(m_i)$ changes before P_i returns, this gives a new string of length n with a short description. Otherwise $A \upharpoonright \gamma(m_i)$ changes when P_i leaves step 3, again giving a new string of length n with a short description.

4.2 Lemma. *If $P_i(g)$ is called and stopped during the loop performed at step 2, then the amount added to trash by $P_i(g)$ and the subprocedures it calls is at most $2^{-2i}g$.*

Proof. We use induction on descending values of i. Above, we verified the lemma for $i = k-1$. Now suppose the statement is true for $i+1$.

(a) If $P_i(g)$ calls P_{i+1} at stage s with goal $g_{i+1,s}$, then by the induction hypothesis, at most $2^{-2(i+1)}g_{i+1,s}$ is put into *trash* by this subprocedure. Since $\sum_s g_{i+1,s} \leqslant g_i$, the total contribution to *trash* of the subprocedures called by $P_i(g)$ is $\leqslant 2^{-2(i+1)}g_i$.

(b) When P_i stops its subprocedures during execution of step 2, it puts $\sum_{j>i} q_j$ into *trash*. But always $q_{j+1} \leqslant q_j/2$, and letting u be the number of times the expression $A \upharpoonright \gamma(m)$ has changed so far, $q_{i+1} \leqslant 2^{-2i-2}2^{-(u+1)}g$. So $\sum_{j>i} q_j \leqslant 2^{-2i-1}2^{-(u+1)}g$, and the total sum over all u is $\leqslant 2^{-2i-1}g$.

The trash contributed by (a) and (b) together is at most $2^{-2i}g$. ☐

The proof of Turing incompleteness of A runs as follows. Let $k = 2^{c+3}$. We start the construction by calling $P_0(g)$ with $g = 1/4$. Then $\sum_{n \in C_0} 2^{-r_n} = g$ and, by Lemma 4.2, *trash* $\leqslant g$. Thus the total weight put into L (i.e., the weight of $\mathbb{N}$) is $\leqslant 2g$, and hence L is a Kraft-Chaitin sequence.

Now, by induction on descending $i < k$, we can show that *each run of a procedure $P_i(g)$ returns unless stopped.* Suppose $i = k-1$, or $i < k-1$ and the claim is true for $i+1$. Since Γ^A is total, eventually the counter u in step 2 is constant. So if $i < k-1$ then eventually we call $P_{i+1}(g')$ for a fixed g' often enough to reach the required weight (for $i = k-1$ the argument is similar). We can enforce the $A \upharpoonright \gamma(m_i)$ change needed at step 3 by enumerating m_i into B. Thus $P_{i(g)}$ returns.

When the initial procedure $P_0(g)$ returns, by the property (II) of C_0, the opponent has to provide at least measure $kg2^{-c}$ in descriptions of length n strings. So if $g = 1/4$, we reach the contradiction $\mu(\mathrm{dom}(U)) > 1$.

We now complete the Proof of Theorem 4.1. Assume $\emptyset'' \leqslant_T A'$. Then there is a total function $f = \Gamma^A$ that eventually dominates each computable function.

Again let $k = 2^{c+3}$.

Modify the construction as follows. Run procedures $P_0(2^{-p})$, $p \geqslant 3$, in parallel. At stage s, look for the least $p \geqslant 3$ such that no procedure $P_0(2^{-p})$ is running, and start such a procedure. Stop when the sum of the q_0 values for the various procedures P_0 reaches 1/4.

This completes the construction. We now verify its correctness.

We first show that L is a Kraft-Chaitin sequence. The trash produced by the stopping subprocedures of a run $P_0(g)$ is bounded by g, by Lemma 4.2. Thus the overall trash of this type is at most 1/4. However, there is a new type of trash, produced by procedures $P_0(2^{-p})$ that never receive an $A \upharpoonright \gamma(m_0)$ change at step 3. For each $p \geqslant 3$, at most one procedure $P_0(2^{-p})$ can get stuck in this way, in which case the sum of wasted descriptions is $\leqslant 2^{-p}$. So this trash adds up to at most 1/4. Altogether, the weight of L is bounded by 3/4.

4.3 Lemma. *Almost all runs of procedure P_0 terminate.*

Proof. We prove by induction on descending $i < k$ that almost all runs of procedure P_i stop. We number the successive runs of a P_i procedure $R_0, R_1, \ldots$. If $i < k-1$, then by the induction hypothesis, there is a v_0 such that, for $v \geqslant v_0$, no run R_v gets stuck while performing a subroutine P_j, $k > j > i$. We want to prove that only finitely many runs R_v get stuck at step 3. Let the total computable function $f_i(x)$ be defined as follows. If x is not the parameter m_i of a run R_v with $v \geqslant v_0$ by stage x, then $f(x) = 0$. Otherwise, $f_i(x)$ is one greater than the first stage where m_i has been canceled or $\gamma(x)$ is defined and R_v reaches step 3. (Such a stage exists by the induction hypothesis and the fact that Γ is total.)

Now for almost all x, the initial segment $A \upharpoonright \gamma(x)$ changes after the stage when $f(x)$ has been defined, so the corresponding run R_v does not get stuck. □

Thus, for a sufficiently large p, we complete procedures $P_0(2^{-p})$ as many times as needed to reach a set C_0 of weight 1/4. This gives a contradiction as before. □

One final limitation is the following.

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

Proof. Nies (unpublished) has shown that every H-trivial real is Turing reducible to an H-trivial computably enumerable set, so it is enough to prove the theorem for the degrees of H-trivial computably enumerable sets. Notice that the statement "$H(W_i \upharpoonright n) \leqslant H(n) + c$ for all n" is Π^0_2 in the parameters i, c. Thus the collection of indices of H-trivial computably enumerable sets is Σ^0_3. We can enumerate a piecewise computably enumerable 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. By Theorem 7.3, the H-trivials are closed under join, so such a set has the property that $\bigoplus_{m \leqslant n} A^{(m)}$ is Turing incomplete for all m. Hence, the result follows from the strong form of the Thickness Lemma (see Soare [26], Ch. VIII, Theorems 2.3 and 2.6). □

In unpublished work, Nies has shown that the degrees of H-trivial computably enumerable sets are bounded below $\mathbf{0}'$ by a low$_2$ computably enumerable set. This is much more difficult to prove.

5 Listing the H-trivials

We next prove a result about the presentation of the class $\mathcal{H}$ of H-trivial reals. First consider the computably enumerable case. As is true for every class that contains the finite sets and has a Σ^0_3 index set, there is a uniformly computably enumerable listing (A_e) of the computably enumerable sets in $\mathcal{H}$. Here we show there is a listing that includes the witnesses of the Σ^0_3 statement, namely the constants via which the A_e are H-trivial. This is true even in the Δ^0_2 case.

We say that α is H-trivial via the constant c if $H(\alpha \upharpoonright n) \leqslant H(n) + c$ for all n. A Δ^0_2-approximation is a computable $\{0,1\}$-valued function $\lambda x, s\ B_s(x)$ such that $B_s(x) = 0$ for $x \geqslant s$ and $B(x) = \lim_s B_s(x)$ exists for each x.

5.1 Theorem. *There is an effective list $((B_{e,s}(x))_{s\in\mathbb{N}}, d_e)$ of Δ^0_2-approximations and constants such that each H-trivial real occurs as a real $B_e = \lim_s B_{e,s}$, and each B_e is H-trivial via the constant d_e. Moreover, $B_{e,s}(x)$ changes at most $O(x^2)$ times as s increases, with effectively given constant.*

Proof. We define, uniformly in e, Δ^0_2-approximations $B_{e,s}$ and Kraft-Chaitin sequences V_e such that, for effectively given constants g_e and for each stage u,

$$\forall w \leqslant u\, \exists r \leqslant H_s(n) + g_e + 3\, (\langle r, B_{e,s} \upharpoonright n\rangle \in V_{e,s}). \tag{5.1}$$

Then we obtain d_e by adding to $g_e + 2$ the coding constant of a prefix-free machine uniformly obtained from V_e.

We need a lemma whose proof will be obtained by analyzing the proof of Theorem 5.8 in [23]. For those familiar with that paper, we include a proof of this lemma below. The lemma says that there is a uniformly computable set Q_e of "good stages" such that B_e changes only at a good stage, and the cost of these changes, namely the weight of short descriptions of the new initial segments $B_{e,s} \upharpoonright m$, is bounded by an effectively given constant 2^{g_e}.

5.2 Lemma. *There is an effective list $((B_{e,s}(x))_{s\in\mathbb{N}}, g_e, Q_e)$ of Δ^0_2-approximations, constants, and (indices for) computable sets of stages, with the following properties.*

1. $B_{e,u}(x) \neq B_{e,u-1}(x) \Rightarrow u \in Q_e$.

2. *Let* $Q_e = \{q_e(0) < q_e(1) < \cdots\}$ *(*Q_e *may be finite). If* $q_e(r+1)$ *is defined, then let* $\hat{c}(z,r) = \sum_{z<y\leqslant q_e(r)} 2^{-H_{q_e(r+1)}(y)}$ *and let*

$$\hat{S}_e = \sum \{\hat{c}(x,r) : u = q_e(r+2) \text{ defined } \wedge \\ x \text{ is minimal such that } B_{e,u}(x) \neq B_{e,u-1}(x)\}.$$

Then $\hat{S}_e < 2^{g_e}$.

Moreover, $B_{e,s}(x)$ *changes at most* $O(x^2)$ *times as* s *increases, with effectively given constant.*

We first complete the proof of the theorem assuming the lemma. We obtain V_e by emulating the construction of an H-trivial real in Theorem 3.1 (see also [23, Proposition 3.3]). At stage u, for each $w \leqslant u$, put $\langle H_u(w) + g_e + 3, B_u \upharpoonright w\rangle$ into V_u in case

(a) $u = w$, or

(b) $u > w \ \wedge \ H_u(w) < H_{u-1}(w)$, or

(c) $B_{u-1} \upharpoonright w \neq B_u \upharpoonright w$.

Clearly, each V_e satisfies (5.1). It remains to show that V_e is a Kraft-Chaitin sequence. We drop the subscript e in what follows. The weight contributed by axioms added for reasons (a) and (b) is at most $2^{-g-2} \leqslant 1/4$. Now consider the axioms added for reason (c). Since B only changes at stages in Q, for each w there are at most two enumerations at a stage $u = q(r+2)$ such that $w > q(r)$. The weight contributed by all w at such stages is at most $\Omega/4$. Now assume $w \leqslant q(r)$, and let $u = q(r+2)$.

Case 1. $H_{q(r+1)}(w) > H_u(w)$. This happens at most once for each value $H_u(w)$, $u \in Q$. Since each value corresponds to a new description of w, the overall contribution is at most $\Omega/8$.

Case 2. $H_{q(r+1)}(w) = H_u(w)$. Since $B(x)$ changes for some minimal $x < w$ at u, the term $2^{-H_u(w)}$ occurs in the sum $\hat{c}(x,r)$. Since $\hat{S} \leqslant 2^g$, the overall contribution is at most 1/8. □

Proof of Lemma 5.2. By [23, Theorem 6.2], let $(\Gamma_m)_{m\in\mathbb{N}}$ be a list of (total) tt-reductions such that the class of H-trivial reals equals $\{\Gamma_m(\emptyset') : m \in \mathbb{N}\}$. Let $A_m = \Gamma_m(\emptyset')$, with the Δ^0_2-approximation $A_{m,s} = \Gamma_m(\emptyset'_s)$. We refer to the proof of [23, Theorem 5.8] and adopt its notation.

Let e be a computable code for a tuple consisting of the following: m, a constant b (we hope A_m is H-trivial via b), numbers i (a level in the tree of runs of procedures) and n (we consider the n-th run of a procedure $P_i(p,\alpha)$, hoping it will be golden), and a constant g_e which we hope will be such that $2^{g_e} = p/\alpha$. (We assume that g_e is at least the constant via which the empty set is H-trivial.) Given e, we define a set Q_e. If e meets our expectations then Q_e will be equal to A_m and will be H-trivial via g_e. Otherwise, Q_e will be finite, but g_e will still be a correct constant via which Q_e is H-trivial.

As in the main construction, we obtain a coding constant d for a prefix-free machine by applying the Recursion Theorem with parameters to m, b, let $k = 2^{b+d}$, and only consider those $i \leqslant k$.

Given e, we run the construction as in [23, Theorem 5.8] in order to define Q_e. For each u, we can effectively determine if u is a *stage* in the sense of that construction. Moreover we can determine if by stage u we started the n-th run $P_i(p, \alpha)$ of a procedure P_i. We leave Q_e empty unless $g_e = p/\alpha$. In that case we check if $u = q(r)$ in the sense of [23, Theorem. 5.8]. If so we declare $u \in Q_e$.

Finally we let $B_{e,u}(x) = A_{m,\max(Q_e \cap \{0,\dots,u\})}$. Thus if Q_e is finite we are stuck with $A_{m,\max Q_e}$. The property $\hat{S} \leqslant 2^{g_e}$ is verified in the proof of [23, Theorem 5.8]. The $O(x^2)$ bound on the number of changes follows as in [23, Fact 3.6]. ☐

Note that we can replace the list (Γ_m) in the above proof by a listing of a subclass of the H-trivials containing the finite sets. Thus there are also effective listings with constants for the H-trivial computably enumerable sets and for the H-trivial computably enumerable reals.

Let C be a set of computably enumerable indices closed under equality of computably enumerable sets. We say that C is *uniformly* Σ^0_3 if there is a Π^0_2 relation P such that $e \in C \leftrightarrow \exists n\,(P(e,n))$ and there is an effective sequence (e_n, b_n) such that $P(e_n, b_n)$ and $\forall e \in C\, \exists n\,(W_e = W_{e_n})$. We have proved that $\mathcal{H}$ is uniformly Σ^0_3. It would be interesting to see which other properly Σ^0_3 index sets have that property, for instance the class of computable sets.

Recall that A is strongly H-trivial via a constant c if $\forall \sigma\,(H(\sigma) \leqslant H^A(\sigma) + c)$, where H^A is H-complexity relativized to A. In [23] it is proved that each H-trivial real is strongly H-trivial. However, in the proof the constant of strong H-triviality is not obtained in a uniform way. The following corollary shows that this non-uniformity is necessary.

5.3 Corollary. *There is no effective way to obtain from a pair (A, b), where A is a computably enumerable set that is H-trivial via b, a constant c such that A is strongly H-trivial via c.*

Proof. Otherwise, by Theorem 5.1 above we would obtain a listing (A_e, c_e) of all the strongly H-trivials with appropriate constants. Nies [24, Theorem 5.9] showed that such a listing does not exist. ☐

6 Theorems of Chaitin and of Zambella

In this section we give a unified proof of some unpublished material of Zambella and of Chaitin's result that all H-trivials are Δ^0_2, while establishing some intermediate results of independent interest.

6.1 Definition. Given a prefix-free machine D, let $Z_D(\sigma) = \mu(D^{-1}(\sigma))$.

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 $Z_D(\sigma)$.

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

We make a few comments before proving this theorem. A measure of complexity is any function $F : 2^{<\omega} \to \omega$ such that $\sum_\sigma 2^{-F(\sigma)} < 1$ and $\{\langle \sigma, k\rangle : F(\sigma) \leqslant k\}$ is computably enumerable. Chaitin [3] introduced this concept and showed that H-complexity is a minimal measure of complexity in the sense that, for any measure of complexity F, we have $H(\sigma) \leqslant F(\sigma)+O(1)$. Notice that $-\log_2 Z(\sigma)$ 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 6.2, we know that for some constant d,

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

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

6.3 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 $H(\alpha \upharpoonright n) \leqslant H(\beta \upharpoonright n) + d$ for all n. 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 there is a c such that $Z(\alpha \upharpoonright n) \geqslant cZ(\beta \upharpoonright n)$ for all n. The other direction is similar. □

6.4 Remark. For any σ, the real $Z(\sigma)$ 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 [9] 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.

Proof of Theorem 6.2. 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,
- $n > 2r \geqslant 2$,
- $H(\sigma)[s] = n$, 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

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

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 result allows us to get an analog of the result of Chaitin [3] on the number of descriptions of a string.

6.5 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 6.3. 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.

6.6 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. □

6.7 Corollary. (I) **(Zambella [31])** *For a fixed d, there are at most $O(2^d)$ many reals α with*

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

for all n.

(II) **(Chaitin [3])** *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. □

7 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 the earlier papers mentioned in the introduction, wtt-reducibility seems to have a lot to do with randomness considerations. Triviality is no exception.

7.1 Theorem. *Suppose that $\alpha \leqslant_{wtt} \beta$ 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 computable use φ and that β is H-trivial. We have

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

by the above. □

Nies [23] has extended this result to Turing reducibility, but with a much more difficult proof.

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

7.2 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 6.6 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. □

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

Proof. By Theorem 7.2, 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. $\square$

Theorem 7.2 suggests the question of whether addition is a join on the H-degrees. In general, it is not, as the example $\Omega \not\leqslant_H \Omega + (1 - \Omega)$ shows. But for computably enumerable reals it is. This fact considerably simplifies the analysis of the H-degrees of computably enumerable reals (compare for instance the difficulties in studying the sw-degrees considered in [8], many of which arise from the lack of a join operation).

7.4 Theorem. *If α, β are computably enumerable 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 representing 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 $\max\{H(\alpha \upharpoonright n), H(\beta \upharpoonright n)\} + 2$.

The machine 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 $\tilde\alpha, \tilde\beta$ 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 $\tilde\alpha + \tilde\beta + b \cdot 2^{-n}$.

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 $\tilde\alpha$ and $\tilde\beta$ (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 $\tilde{\gamma} \upharpoonright n = \sigma$. As $\tilde{\alpha} + \tilde{\beta} = \tilde{\gamma}$, 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 $(\tilde{\alpha} \upharpoonright n, \tilde{\beta} \upharpoonright n)$, 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 [19, Exercise 2.58].) Recall also that, by [16], a computably enumerable degree $\mathbf{d}$ either has complex computably enumerable sets or every computably enumerable 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 computably enumerable set A is called array noncomputable relative to such a very strong array if for all computably enumerable 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.

7.5 Theorem (Downey, Jockusch and Stob [10, 11]). *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 array noncomputable relative to some very strong array, then $\mathbf{d}$ contains one that is array noncomputable relative to $\{F_x : x \in \mathbb{N}\}$.*

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

7.6 Theorem. *If $\mathbf{d}$ is an array noncomputable and computably enumerable 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 7.5, $\mathbf{d}$ contains a set A array noncomputable 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 computably enumerable 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 $z > \max\{x : x \in F_{e+c}\}$. The universal machine must respond at some stage s by converging to $A_s \upharpoonright z$ on some input σ of length $\leqslant d+e+1+c$. We then enumerate into V_s, our "kill" computably enumerable 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. □

One might be tempted to think that the Kummer trivial computably enumerable sets and the H-trivial sets correspond. The next result shows that at least one inclusion fails. Since the proof is rather technical, and the result also follows from the recent results of Nies [23] mentioned above, we restrict ourselves to a brief proof sketch.

7.7 Theorem. *There is a computably enumerable Turing degree* **a** *that consists only of Kummer trivials but contains no H-trivials.*

Proof sketch. We construct a contiguous computably enumerable degree **a** containing no H-trivials. A *contiguous* degree is a Turing degree that consists of a single wtt-degree. Such degrees were first constructed by Ladner and Sasso [18]. By Downey [5], every contiguous computably enumerable degree is array computable. Hence, by Kummer's theorem, all of its members are Kummer trivial. The argument is a Π^0_2 priority argument using a tree of strategies.

We must meet the requirements below:

$$\mathcal{R}_{e,i} : \Phi_e^A = B \wedge \Phi_i^B = A \text{ implies } A \equiv_{wtt} B.$$

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 $\mathcal{R}_{e,i}$ having outcomes $\infty <_L f$. The outcome ∞ is meant to say that $\Phi_e^A = B \wedge \Phi_i^B = A$. The other outcome is meant to say that the limit of the length of the agreement function

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

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

The usual H-nontriviality requirements are that

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

These are met, as one would expect, by changing A sufficiently often when the universal prefix-free machine U threatens to demonstrate that A has the same H-complexity as 1^ω 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. If a version of $\mathcal{P}$ guessing that the outcome of R_α is ∞ (and so associated with some node $\beta \supseteq \alpha^\frown\infty$) generates a follower x, then that will only happen at an $\alpha^\frown\infty$ (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 $\alpha^\frown\infty$ 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 $\alpha^\frown\infty \subseteq \beta$.) 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 $\alpha^\frown\infty$-stages, $A \upharpoonright x$ can be computed from the least $\alpha^\frown\infty$-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 $\alpha^\frown\infty$-stage t with $\ell(e,i,s) > z$, then $B \upharpoonright z = B_s \upharpoonright z$, where $s > t$ is the least $\alpha^\frown\infty$-stage with $A_s \upharpoonright x = A \upharpoonright x$. More details can be found in, for example, Downey [5].

Returning to the $\mathcal{P}$ requirements, we build a prefix-free machine M via the Kraft-Chaitin Theorem. By the Recursion Theorem we know the coding constant d of M. We split the domain of M up into chunks for the various requirements. That is, each $\mathcal{P}_\alpha$ on the priority tree will be allowed to add at most $2^{-k(\alpha)}$ to the measure of the domain of M, where the sum of $2^{-k(\alpha)}$ over all strategies $\mathcal{P}_\alpha$ is one.

Suppose that $\mathcal{P}_\alpha$ is the version of $\mathcal{P}_e$ on the true path, and we are at stage where this version has priority (i.e., the construction never again moves to the left of $\mathcal{P}_\alpha$). Let 2^{-k} be the amount of the measure of the domain of M devoted to $\mathcal{P}_\alpha$.

We wait until there are a large number of α-confirmed followers for $\mathcal{P}_\alpha$ (the exact number necessary is not hard to compute from k and d). Specifically, we pick a big x_1 at an α-stage, then when the total length of agreement is bigger than x_1, we initialize and α-confirm, as usual. Then we pick x_2, and so on, until the whole entourage of followers is stable.

Let x_n be the largest of our followers. This is where we will satisfy $\mathcal{P}_e$. 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 $k+d$. If we see $U(\tau)\downarrow = A \upharpoonright x_n$, with $|\tau| \leqslant k+d+e$, then we change A. This is done using the followers in reverse order, first x_n, then later x_{n-1} if necessary, and so on. The reverse order guarantees the contiguity as usual.

Note that U has the option of later choosing something shorter than $k+d$ to compute

1^{x_n}, but this can only happen $k+d$ times, and we have enough x_i's to cope with this.

The remaining details consist of implementing this strategy on a priority tree. □

Array noncomputable sets 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 [15] were the first to construct such sets. They used a theorem of Sacks [25] 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 [22], 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 Ishmukhametov [13] we call a set A *traceable* or *weakly computable* if 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

1. $|W_{h(x)}| \leqslant f(x)$ for almost all x and
2. $g(x) \in W_{h(x)}$ for all x.

Ishmukhametov [13] observed that if a degree is weakly computable then it is array computable, and the notions coincide for computably enumerable sets. Ishmukhametov proved the remarkable theorem that the computably enumerable degrees with strong minimal covers are exactly the weakly computable degrees. Furthermore, any weakly computable degree (in general) has a strong minimal cover.

7.8 Theorem. *Suppose that A is low for random. Then A is low* **(Nies [22])** *Furthermore, A is weakly computable.*

Proof sketch. If one mimics the proof by Terwijn and Zambella [28] that Schnorr low sets are computably traceable, but using Martin-Löf lowness in place of Schnorr lowness, then the "if" direction proves the theorem. □

References

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

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

[3] G. Chaitin, *A theory of program size formally identical to information theory*, Journal of the Association for Computing Machinery 22 (1975), 329–340, reprinted in [4].

[4] G. Chaitin, *Information, Randomness & Incompleteness*, 2nd edition, Series in Computer Science 8, World Scientific, River Edge, NJ, 1990.

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

[6] R. Downey 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.

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

[8] R. Downey, D. Hirschfeldt, and G. Laforte, *Randomness and reducibility*, in *Mathematical Foundations of Computer Science 2001* (Sgall, Pultr, P. Kolman, eds.), Lecture Notes in Computer Science 2136, Springer, 2001, 316–327.

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

[10] R. Downey, 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 1432, Springer-Verlag, Heidelberg, 1990, 141–174.

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

[12] L. Fortnow, *Kolmogorov complexity,* in [6], 73–86.

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

[14] S. Kautz, *Degrees of Random Sets*, Ph.D. Diss., Cornell University, 1991.

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

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

[17] S. Kurtz, *Randomness and Genericity in the Degrees of Unsolvability*, Ph.D. Thesis, University of Illinois at Urbana-Champaign, 1981.

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

[19] M. Li and P. Vitanyi, *An Introduction to Kolmogorov Complexity and its Applications*, 2nd edition, Springer-Verlag, New York, 1997.

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

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

[22] A. Nies, *Low for random sets are* Δ^0_2, Technical Report, University of Chicago, 2002.

[23] A. Nies, *Lowness properties of reals and randomness*, to appear.

[24] A. Nies, *Reals which compute little*, to appear.

[25] G. Sacks, *Degrees of Unsolvability,* Princeton University Press, 1963.

[26] R. Soare, *Recursively enumerable sets and degrees*, Springer, Berlin, 1987.

[27] R. Solovay, *Draft of a paper* (*or series of papers*) *on Chaitin's work*, unpublished manuscript, May, 1975, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 215 pages.

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

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

[30] N. Vereshchagin, *A computably enumerable undecidable set with low prefix complexity: a simplified proof*, Electronic Colloquium on Computational Complexity, Revision 01 of Report TR01-083.

[31] D. Zambella, *On Languages with simple initial segments*, Technical Report, University of Amsterdam, 1990.