Computably Enumerable Reals and Uniformly Presentable Ideals*

Rod Downey and Sebastiaan A. Terwijn
School of Mathematical and Computing Sciences
Victoria University
PO Box 600, Wellington
New Zealand

January 5, 2011

Abstract

We study the relationship between a computably enumerable real and its presentations. A set A presents a computably enumerable real α if A is a computably enumerable prefix-free set of strings such that $\alpha = \sum_{\sigma\in A} 2^{-|\sigma|}$. Note that $\sum_{\sigma\in A} 2^{-|\sigma|}$ is precisely the measure of the set of reals that have a string in A as an initial segment. So we will simply abbreviate $\sum_{\sigma\in A} 2^{-|\sigma|}$ by $\mu(A)$. It is known that whenever A so presents α then $A \leq_{wtt} \alpha$, where $\leq_{wtt}$ denotes weak truth table reducibility, and that the wtt degrees of presentations form an ideal $\mathcal{I}(\alpha)$ in the computably enumerable wtt degrees. We prove that any such ideal is Σ^0_3, and conversely that if $\mathcal{I}$ is any Σ^0_3 ideal in the computably enumerable wtt degrees then there is a computable enumerable real α such that $\mathcal{I} = \mathcal{I}(\alpha)$.

1 Introduction

Although the larger part of computability theory has been concerned with the computational complexity of sets of numbers and strings, from the beginnings of the subject (see e.g. Turing [16]) there has been an interest also in *reals*. Of particular importance to computable analysis (see e.g. Weihrauch [17], Pour-El and Richards [13], Ko [9]) and algorithmic information theory (see e.g. Chaitin [6], Calude [2], Martin-Löf [11], Li-Vitanyi [10]) is the collection of *computably enumerable reals.*

Following Soare [14], a real α is called computably enumerable (c.e.) if we can effectively generate it from below. That is if there is a computable sequence of rationals $\{q_i : i \in \mathbb{N}\}$ with $q_{i+1} \geq q_i$ converging to α. If we can in addition effectively compute the radius of convergence, then α is said to be computable, in which case we can compute effectively the n-th bit of its dyadic expansion.

*The research in this paper was supported by the Marsden Fund of New Zealand.

A driving force in the analysis of computably enumerable reals is algorithmic information theory. A classic example of a computably enumerable real that is *not* computable is Chaitin's halting probability:

$$\Omega = \sum_{U(\sigma)\downarrow} 2^{-|\sigma|},$$

where U denotes a universal prefix-free Turing machine[1].

Given a computably enumerable real, it is natural to ask how it can be generated. That is, what kinds of effective sequences can be used to "present" the real. For simplicity we consider only reals between 0 and 1. Two classical representations of reals are Cauchy sequences and Dedekind cuts. Let α be a real, and $L(\alpha) = \{q \in \mathbb{Q} : q \leq \alpha\}$ the Dedekind cut associated with α. Soare [14] investigated the relation between $L(\alpha)$ and sets A with $\alpha = \sum_{n \in A} 2^{-n}$. That there are c.e. α for which such A cannot be c.e. had already been noted by C. G. Jockusch. It is clear that α is a c.e. real if and only if $L(\alpha)$ is a computably enumerable set of rationals. Some basic equivalences were proven by Calude et al. [4]:

Theorem 1 (Calude et al. [4]) *The following are equivalent for a real α.*

(i) *α is computably enumerable.*

(ii) *The lower Dedekind cut of α is computably enumerable.*

(iii) *There is an infinite computably enumerable prefix-free set $A \subseteq 2^{<\omega}$ such that $\alpha = \sum_{x \in A} 2^{-|x|}$.*

(iv) *There is an infinite computable prefix-free set $A \subseteq 2^{<\omega}$ such that $\alpha = \sum_{x \in A} 2^{-|x|}$.*

(v) *There is a computable function $f(x, y)$ of two variables such that*

  (a) *for all k, s, if $f(k, s) = 1$ and $f(k, s+1) = 0$, then there exists $k' < k$ such that $f(k', s) = 0$ and $f(k', s+1) = 1$.*

  (b) *$\alpha = a_1 a_2 \ldots$, where $a_i = \lim_s f(i, s)$.*

(vi) *There is a computable increasing sequence of rationals with limit α.*

Although the apparently-stronger (iv) is not explicitly stated in [4], it follows from (iii), since we can always make the enumeration of strings in A nondecreasing in length.

[1]Recall that a machine U is called prefix-free if, whenever $U(\sigma) \downarrow$, then for all $\sigma \sqsubset \tau$, $U(\tau) \uparrow$. The prefix-free Kolmogorov complexity of a string ν relative to a machine N is defined as $H_N(\nu) = (\mu n)(\exists \sigma)[|\sigma| = n \wedge N(\sigma) = \nu]$. Prefix-free machine U is universal iff for all prefix-free M there is a constant c such that for all ν $K_U(\nu) \leq K_M(\nu) + c$. A real α is called random iff there is a constant c such that, for all n $H_U(\alpha \upharpoonright n) \geq n - c$. It seems that computable enumerable reals occupy the same central place in algorithmic information theory that computable enumerable sets occupy in classical computability theory. We refer the reader to Li-Vitanyi [10] for further details and motivation.

Prefix-free sets correspond to open (hence measurable) sets of reals: if A is prefix-free then the set $\{X \in 2^\omega : \exists \sigma \in A(\sigma \sqsubset X)\}$ is open in the usual topology on 2^ω, and its Lebesgue measure $\mu(A)$ is $\sum_{\sigma \in A} 2^{-|\sigma|}$. Hence we can rewrite (iii) (and (iv)) as $\alpha = \mu(A)$ for some computable enumerable (computable) prefix-free A.

Definition 2 *For any $A \subseteq 2^{<\omega}$, we say A is a* presentation *of a c.e. real α if A is a prefix-free c.e. set such that $\alpha = \mu(A)$.*

Downey and LaForte asked the following question: given a computably enumerable real α, what can be said about the possible presentations α might have? In particular they asked: suppose that α is not computable. Does α necessarily have noncomputable presentations? Surprisingly the answer is "no."

Theorem 3 (Downey and LaForte [7]) *There is a c.e. real α which is not computable, but such that if A presents α, then A is computable.*

The proof of Theorem 3 was a surprisingly difficult $\mathbf{0''}$ or "infinite injury" priority argument, an argument of a type hitherto not found in computable analysis. The remainder of of Downey and LaForte [7] was devoted to trying to understand what sorts of reals are "nearly computable" in the sense that they only have computable presentations; and what can be said about the types of presentations that a real might have. For instance, Downey and LaForte proved that such reals can have high c.e. degree. They also showed that some classes of degrees do not contain such nearly computable reals, namely they showed that such degrees could not be "promptly simple." Finally, Wu [18] used a $\mathbf{0'''}$ priority argument to construct a nonzero computably enumerable degree $\mathbf{a}$ such that no nonzero lesser degree contains a nearly computable real.

In the present paper, we turn to the question: *What types of presentations can a computably enumerable real have? In particular, if a computably enumerable real has one noncomputable presentation, what others does it have?*

As with several other questions arising from computable mathematics, the answer seems to lie in strong reducibilities, specifically, *weak truth table* reducibility. Recall that $A \leq_{wtt} B$ iff there exists a Turing reduction Γ and a computable function γ such that $\Gamma^B = A$, and for all x the maximum element queried in the computation $\Gamma^B(x)$ is $\leq \gamma(x)$. Wtt reducibility has proven useful in other parts of computable mathematics, notably Calude and Nies [5] proved that Chaitin's number Ω is wtt-complete but not tt-complete, and Downey and Remmel [8], showed that the degrees of c.e. bases of a c.e. vector space V are precisely the wtt-degrees below $\deg_{wtt}(V)$.

Theorem 4 (Downey and LaForte [7])

(i) *Let α be a computably enumerable real, with $\alpha = .\chi_A$ for some set A. Suppose that B is any presentation of α. Then $B \leq_{wtt} A$ with use function the identity.*

(ii) *If A is a presentation of a c.e. real α and $C \leq_{wtt} A$ is computably enumerable, then there is a presentation B of α with $B \equiv_{wtt} C$.*

It is easy to see that if α is strongly c.e., in the sense that $\alpha = .\chi_A$ for some c.e. set A, then there is a presentation B of α with $A \leq_m B$.[2] It follows from Theorem 4 that

[2]Namely, choose a prefix-free domain containing exactly one string of every length, and enumerate this string into B iff its length is enumerated into A.

such α have presentations in every c.e. wtt-degree below A. So we have two extremes: A c.e. real can be only computably presentable, and at the other extreme a c.e. real can have presentations of each c.e. wtt-degree.

We remark that it is easy to construct a presentation B of a c.e. real α such that $B \not\leq_{tt} \alpha$ (Proposition 6), and hence weak truth table reducibility is the natural reducibility in this context.

As noted in [7],

$$\mathcal{I}(\alpha) = \{W_e : \exists A \text{ presentation of } \alpha.\ W_e \equiv_{wtt} A\}$$

is an ideal[3] in the c.e. wtt-degrees. It is not difficult to prove (Section 3) that any such ideal is Σ^0_3. Σ^0_3-ideals in the c.e. wtt-degrees were studied by Ambos-Spies et al. [1]. They proved that these are exactly the ideals having an exact pair. The main goal of the present paper is to completely classify the possible ideals that can be realized as ideals of degrees of presentations. The answer is "anything that is not explicitly ruled out can be realized."

Theorem 5 *Suppose that I is Σ^0_3 and that $\mathcal{I} = \{W_e : e \in I\}$ forms an ideal in the c.e. wtt-degrees. Then there is a computably enumerable real α such that $\mathcal{I}(\alpha) = \mathcal{I}$.*

One consequence of this result is that there are c.e. reals that have no greatest degree of presentation etc. The proof of Theorem 5 is an infinite injury priority argument, which combines several ingredients. In particular, it combines an approximation argument (for the Σ^0_3 representation of $\mathcal{I}$), a coding argument (for the members of $\mathcal{I}$), and infinitary negative requirements (like those used in [7]), and is of some technical depth.

2 Some notation

In the following sections, we generally use standard notation from computability theory. In particular, when we construct c.e. sets to be presentations of reals with various computational properties, we generally follow the terminology of Odifreddi [12] and Soare [15]. An important abbreviation that deserves special notice is the following: We fix in advance an enumeration of all c.e. sets W_e as the output of some suitable universal Turing machine such that exactly one pair $\langle e, x\rangle$ with $x \in W_e$ is listed at each stage s. We can then use "$[s]$" to relativize entire expressions involving computable dynamic processes with the meaning that the state of each such process is evaluated at stage s. By convention, all computations etc. at stage s are bounded by s, and the word "fresh" refers to a number or string bigger than any previously seen in the construction, and, in particular, will exceed all uses.

3 Presentations and ideals

First we show that Theorem 4 (i) does not hold for tt-reducibility:

[3]That is, it is closed downwards (by Theorem 4) and closed under joins: If B and C present α then also $\{0\sigma : \sigma \in B\} \cup \{1\sigma : \sigma \in C\}$ of degree $B \oplus C$ presents α.

Proposition 6 *There exist a c.e. real α and a presentation B of α such that $B \not\leq_{tt} \alpha$.*

Proof. We construct a c.e. real α and a prefix-free c.e. domain B presenting α such that for every e, if φ_e is a tt-reduction, then there is a string σ such that

R_e : $\quad \sigma \in B \iff \alpha \not\models \varphi_e(\sigma)$.

Let $D = \{0^n 1 : n \in \omega\}$, so that D is a recursive prefix-free domain.

Stage 0. Let $\alpha[0] = 0$, $B[0] = \emptyset$, $\sigma_{e,0} = 0^e 1$.

Stage $s > 0$. Look at the smallest e for which R_e has not yet been satisfied and for which $\varphi_e(\sigma_e)\downarrow [s]$. If $\alpha[s] + 2^{-|\sigma_e|} \models \varphi_e(\sigma_e)$ then instead of putting σ_e into B we put the extensions $\sigma_e 0$ and $\sigma_e 1$ into B. If $\alpha[s] + 2^{-|\sigma_e|} \not\models \varphi_e(\sigma_e)$ then put σ_e into B. In both cases add $2^{-|\sigma_e|}$ to α, and initialize all R_i with $i > e$ by redefining σ_i to be fresh strings from D.

Clearly B is c.e., and B is prefix-free because D is. Furthermore, $\mu(B) = \alpha$ because every time we add measure to B we add the same amount numerically to α. Finally, if φ_e is a tt-reduction, then $\sigma_e \in B \Leftrightarrow \alpha \not\models \varphi_e(\sigma_e)$ because σ_e is kept out of B precisely when $\alpha \models \varphi_e(\sigma_e)$. Because after every diagonalization the lower priority σ_i, $i > e$, are picked fresh, they do not interfere with the action taken for R_e. Hence the construction is finite injury. □

As mentioned in the introduction, for a c.e. real α the family

$$\mathcal{I}(\alpha) = \{W_e : \exists A \text{ presentation of } \alpha.\ W_e \equiv_{wtt} A\}$$

forms an ideal. Let us determine the complexity of $\mathcal{I}(\alpha)$. The statement "$\mu(W_e) = \alpha$" is Π^0_2 ("for all diameters ε there is a stage s such that $\mu(W_e)[s]$ and $\alpha[s]$ are closer than ε"). Saying that W_e is prefix-free is Π^0_1 ($\forall \sigma, \tau \in W_e.\ \sigma \not\sqsubset \tau$). For a given c.e. set A the set $\{W_e : W_e \equiv_{wtt} A\}$ is Σ^0_3 (see Odifreddi [12, p627]; roughly, we have to say "there exists a wtt-reduction such that $\forall x \forall y \leq x \exists s > x$ such that at stage s the reduction gives the right answers on y"). All in all, $W_e \in \mathcal{I}(\alpha)$ if and only if there exists d such that a Σ^0_3 statement holds true of W_d. So we see that $\mathcal{I}(\alpha)$ is a Σ^0_3-ideal. To see that this is optimal, note that for α computable we have by Theorem 4 that $\mathcal{I}(\alpha) = \{W_e : W_e \text{ computable}\}$, and this set is Σ^0_3-complete. $\mathcal{I}(\alpha)$ is not always Σ^0_3-complete: For $\alpha = \chi_K$ we already saw that $\mathcal{I}(\alpha) = \{W_e : e \in \omega\}$ is trivial (as an index set). We now prove a result in the spirit of Rice's Theorem, saying that this is in fact the *only* case where $\mathcal{I}(\alpha)$ is not Σ^0_3-complete.

Theorem 7 *$\mathcal{I}(\alpha)$ is either $\{W_e : e \in \omega\}$ or Σ^0_3-complete.*

Proof. Let $\alpha = \chi_A$ be a c.e. real. It is easy to see that $\mathcal{I}(\alpha) = \omega$ iff A is wtt-complete. Suppose that A is not wtt-complete. We prove that $\mathcal{I}(\alpha)$ is Σ^0_3-complete. This can be proved using the methods of Rogers and Kallibekov, see Odifreddi [12, p625-627]. We sketch the proof and leave the details to the reader. Let $\text{Inf} = \{e : W_e \text{ is infinite}\}$. We use that the weak jump $\{x : W_x \cap \text{Inf} \neq \emptyset\}$ of Inf is Σ^0_3-complete. It suffices to build sets B_x uniformly in x such that

$$W_x \cap \text{Inf} \neq \emptyset \implies B_x \text{ computable}$$
$$W_x \cap \text{Inf} = \emptyset \implies B_x \not\leq_{wtt} A.$$

In the first case clearly the wtt-degree of B_x contains a presentation of α, while in the second case it follows from Theorem 4 that this is not the case.

We have requirements

P_e : $\quad e \in W_x \wedge W_e$ infinite $\Longrightarrow (\forall i \geq e)\ [\omega^{[i]} \subseteq B_x]$

that try to make B_x computable, and

R_e : $\quad (\Gamma_e, \gamma_e)$ is total wtt-reduction $\Longrightarrow \exists z[B_x(z) \neq \Gamma_e^A(z)]$.

for making $B_x \not\leq_{wtt} A$, and give them the priority ordering $P_0 < R_0 < P_1 < R_1 < \ldots$

R_e is handled by Sacks's coding strategy ([12, p512]): We maintain a length of agreement function $l(e,s)$ monitoring agreement between B_x and Γ_e^A. We code $K \upharpoonright l(e,s)$ into $\omega^{[2e+1]}$. Then, provided that the higher priority requirements are finitary, R_e is also finitary (and hence satisfied), since otherwise the whole of K would be coded into B_x and still we would have $B_x \leq_{wtt} A$, contradicting the incompleteness of A.

P_e is handled directly by filling the rows above $\omega^{[2e]}$ up to $\max W_e[s]$ at every stage s whenever e is found to be in W_x.

If W_x contains no code of an infinite c.e. set then all P_e are finitary, hence every R_e succeeds and $B_x \not\leq_{wtt} A$. If on the other hand $e \in W_x$ is a minimal code of an infinite c.e. set then $(\forall i \geq e)[\omega^{[i]} \subseteq B_x]$. Since all higher priority requirements are finitary, $B_x \cap \omega^{[i]}$ is finite for every $i < e$. Hence B_x is computable. □

We have seen that $\mathcal{I}(\alpha)$ is a Σ_3^0-ideal. Theorem 5 says that conversely every Σ_3^0-ideal in the c.e. wtt-degrees is of the form $\mathcal{I}(\alpha)$ for some c.e. real α. The proof will make use of the following lemma, that implies that every Σ_3^0-ideal is generated by a uniform collection of c.e. sets.

Lemma 8 (Yates [12, II.5.25]) *Let $\{W_e : e \in I\}$, $I \in \Sigma_3^0$, be any collection of c.e. sets containing all the finite sets. Then there is a uniformly c.e. collection $\{V_f : f \in \omega\}$ such that $\{W_e : e \in I\} = \{V_f : f \in \omega\}$.*

Proof. Suppose $W_e \in \mathcal{C} \Leftrightarrow \exists f \forall n \exists m R(e,f,n,m)$ for some recursive predicate R. For every f construct a c.e. set V_f as follows. For every successive n, V_f looks for an m confirming $R(e,f,n,m)$, and if it finds such m it copies W_e by setting $V_{f,n} = W_{e,n}$. If f is true then for all n the appropriate confirmation m will be found, and V_f equals W_e. If f is false then for some n, V_f will search forever, so it is finite. It is now clear that the collection of all V_f generates $\mathcal{C}$. □

4 Proof of Theorem 5

Outline of the proof. By Lemma 8 we may suppose that the Σ_3^0-ideal is given to us by a uniform collection of c.e. sets $U_0, U_1, U_2, \ldots$ We want to construct α such that for all e:

C_e : $\quad$ code U_e into $\mathcal{I}(\alpha)$ by constructing $A_e \equiv_{wtt} U_e$ with $\alpha = \mu(A_e)$,

N_e : $\quad W_e$ presents $\alpha \Longrightarrow W_e \leq_{wtt} \bigoplus_{i \leq e} A_i$.

First we describe the strategies for meeting these requirements in isolation, and then we describe how we will combine the strategies (using a tree of strategies).

We will try to satisfy $U_e \leq_{wtt} A_e$ by permitting: Along with A_e we define a use function ψ_e such that whenever a number x enters U_e we put (or at least try to put) a string $\psi_e(x)$ into A_e.

We will try to ensure $A_e \leq_{wtt} U_e$ by allowing a small string to enter A_e only for the sake of coding U_e. So, assuming that $\psi_e(x) \geq x$ we will have $A_e \leq_{wtt} U_e$ with the identity as use function.

Along with the construction we will define α by enumerating rational values in it (see item (vi) of Theorem 1). $\alpha[s]$ will be the approximation of α determined by the numbers put into it by stage s. The second part of C_e will be satisfied by ensuring that there are infinitely many stages s with $(\alpha = \mu(A_e))[s]$, so that indeed all the A_e will present the same α.

For N_e, if $\alpha[s]$ and $\mu(W_e)[s]$ grow close we will try to make W_e computable by restraining $\alpha[s]$. We will monitor how close the two get by defining a monotone unbounded sequence of numbers $m(e)[s]$, and every time we see that $|\alpha - \mu(W_e)| < 2^{-m(e)}[s]$ we will try to keep $\alpha[s]$ from changing on short strings, thus allowing only minor changes. Were we to completely succeed in this, then W_e would be computable as follows: When asked if $\gamma \in W_e$, run the construction until $|\alpha - \mu(W_e)| < 2^{-m(e)}[s]$, with $m(e)[s] >> |\gamma|$. Then $\gamma \in W_e$ if and only if $\gamma \in W_e[s]$.

A coding strategy C_e can easily live with the action of a higher priority coding strategy C_i simply by picking different coding locations. We describe how the other strategies can be combined.

First we look at how N_e can deal with the outcome of a higher priority C_i. As described above, when at stage s it holds that $|\alpha - \mu(W_e)| < 2^{-m(e)}[s]$, N_e tries to restrain $\alpha[s]$ by trying to keep it from changing more than $2^{-m(e)}$. (It will allow minor changes in α to give lower priority requirements a chance of succeeding.) However, the coding action of C_i may spoil this. Suppose that, despite the injuries of C_i, at the end of the construction W_e presents α. Although we cannot argue anymore that W_e is computable, we can still argue that it is computable in A_i, which is good enough for us. To compute whether $\gamma \in W_e$, A_i compute s so large that $\mu(A_i)$ changes no more than $2^{-|\gamma|+1}$ after s by the coding of U_i. Then, using that the construction is recursive, compute a stage s such that $|\alpha - \mu(W_e)| < 2^{-m(e)}[s]$, where $2^{-m(e)}[s] < 2^{-|\gamma|+2}$. Then $\alpha[s]$ is not changed more than $2^{-m(e)}[s]$ by N_e, and $\alpha[s]$ is not changed more than $2^{-|\gamma|+1}$ because of the coding of C_i, so $\mu(W_e)[s]$ cannot change more than $2 \cdot 2^{-m(e)}[s] + 2^{-|\gamma|+1} < 2^{-|\gamma|}$. So $\gamma \in W_e$ if and only if $\gamma \in W_e[s]$. Hence W_e is computable in A_i.

Second we look at how C_i can deal with the outcome of a higher priority N_e. There are two relevant outcomes of N_e: The infinitary outcome is when at infinitely many stages (which we will call e-expansionary stages) $\mu(W_e)$ grows closer to α. The finitary outcome is when from a certain stage onwards, $\mu(W_e)[s]$ is bounded away from $\alpha[s]$. Suppose that x enters U_e at stage s. Then A_e wants to code this event by enumerating a string δ. Suppose further that $|\alpha - \mu(W_e)|[s] < 2^{-m(e)}[s] < 2^{-|\delta|}$. Then A_e is not allowed to enumerate a string as short as δ, since this would cause $\alpha[s]$ to change $2^{-|\delta|}$, which is more than N_e allows. To get around this we use the trick of Downey and LaForte [7, Theorem 8]. Namely, in the above situation we let A_e announce that it wishes to enumerate δ, without actually doing so. Furthermore,

we make α slightly bigger, so little that the computability of W_e as described above is not affected, namely that if $\mu(W_e)$ is to stay close to α then W_e cannot enumerate a short string. Then there are two possibilities for W_e: Either it does not respond, remaining forever more than $2^{-m(e)}[s]$ apart from α, in which case it does not present α and the outcome of N_e will be finitary. Or it responds by growing closer than $2^{-m(e)}[s]$ to α again, in which case we repeat the procedure. If W_e keeps responding to the small changes we make in α, by repeating enough times we will be able to create enough space between A_e and α for δ to enter A_e. Note that it is important that we do not allow N_e to let its value $m(e)$ grow during this procedure. We will refer to this strategy as the "drip feed strategy", since we think of C_i succeeding by feeding α changes small enough to be allowed by N_e, and doing this often enough to be able to finally make its move.

The strategy for C_e becomes a little more complicated when it has to deal with the outcome of more than one N-strategy. Suppose that C_e is below N_i, which in its turn is below N_j. Suppose that we try to put δ into A_e using the drip feed strategy described above. Then C_e will try to change α by an amount of 2^{-n} in $2^{-|\delta|+n}$ steps, where $n = m(i)$, the maximum change in α that N_i allows for. Now while waiting for N_i to respond to the first change, N_j may let its value $m(j)$ grow, since it does not know (or care) whether N_i is going to respond. When N_i finally does respond, $m(j)$ may have become so big that N_j does not allow a change of 2^{-n} in α, thus frustrating the drip feed strategy of C_e. The solution is to let N_i in turn use a drip feed strategy to let N_j allow for a change of 2^{-n}. If both N_i and N_j are infinitary, in the end all the changes in α requested by C_e will be allowed for. After every successfully completed drip feeding strategy, the N-strategies are allowed to let their m-value grow. This works in general for any finite number of N-strategies above C_i, by recursively nesting the drip feed strategies.

The tree of strategies. In general, of course, a requirement has to deal with the outcome of *all* the higher priority requirements, not just one. We handle the combinatorics of this using the usual infinite injury framework of putting all the strategies on a tree. We use $2^{<\omega}$ as a tree of strategies, assigning both C_e and N_e to every string of length e. Define

$$g(e) = \begin{cases} 0 & \text{if } W_e \text{ presents } \alpha \\ 1 & \text{otherwise.} \end{cases}$$

The path defined by g is called the *true path* of the construction. At every stage s we will have a finite approximation g_s of g of length at most s such that $g = \liminf_{s\to\infty} g_s$. For any string σ, a stage s is a *σ-stage* if $s = 0$ or $s > 0$ and $\sigma \sqsubseteq g_s$. A σ-strategy is *initialized* if all its parameters are set to being undefined. For any two strings σ and τ, $\sigma <_L \tau$ if and only if there is a string ρ such that $\rho 0 \sqsubseteq \sigma$ and $\rho 1 \sqsubseteq \tau$. A σ-strategy has *higher priority* than a τ-strategy if $\sigma \sqsubset \tau$ or $\sigma <_L \tau$. For the two σ-strategies, C_σ has priority over N_σ.

To coordinate the drip feed strategies, we equip every node σ with a *counter* $c(\sigma)$ and let σ only act at stages where $c(\sigma) = 0$. The counter $c(\sigma)$ will indicate how many steps a drip feed strategy needs to be successfully completed. So, at the start of a drip feed strategy initiated by some low priority coding requirement that wants to put a short string δ into α, the counter $c(\sigma)$ is set to $2^{-|\delta|+n}$ for some number n determined

by the restraint of the first infinitary N-requirement above it. Every time a package of size 2^{-n} passes $N_{|\sigma|}$ the counter $c(\sigma)$ is decreased by 1. Every time the counter reaches 0 we allow σ to act.

In order to enable the infinitary coding actions of low priority requirements to interleave with the coding of high priority requirements, we equip the strategies with *lists* Λ for bookkeeping which strings wish to enter α using a drip feed strategy. After a drip feed strategy is successfully completed, the top element of Λ is removed. Every string on Λ has to wait until it is on the top of the list before it can start a drip feed strategy. (Such lists were not needed in the proof of Theorem 3 since there the positive requirements were finitary.)

Construction. Every σ will build its own copy A_σ and try to satisfy $C_{|\sigma|}$ by building a wtt-reduction ψ_σ from $U_{|\sigma|}$ to A_σ. The construction will feature several auxiliary parameters and functions: For every σ we have lists $\Lambda(C_\sigma)$ and $\Lambda(N_\sigma)$, a counter $c(\sigma)$, functions $l(\sigma)$ and $m(\sigma)$ monitoring the length of agreement, and a restraint function $r(\sigma)$. The construction proceeds in stages.

Stage $s = 0$. Set $\alpha[0] = 0$, $g[0] = \lambda$. For all σ, let $A_\sigma[0] = \emptyset$, and initialize all σ-strategies, i.e. set all parameters to be $\uparrow$ (undefined).

Stage $s > 0$. The finite approximation $g[s]$ (of length at most s) to the constructions true path will be defined by the σ that are active at stage s. These σ are defined by recursion, in increasing order. Given an active σ, the next (if any) active node is determined by N_σ.

Action for the positive requirement C_σ. Let $e = |\sigma|$. First we pick suitable coding locations for U_e in A_σ: For every $x \leq s$, if $\psi_\sigma(x)[s-1]\downarrow$ then let $\psi_\sigma(x)[s] = \psi_\sigma(x)[s-1]$. If $\psi_\sigma(x)[s-1]\uparrow$ then pick for $\psi_\sigma(x)[s]$ a fresh string of length bigger than x and not extending any string previously enumerated into A_σ. For every x that enters U_e at s do the following: Let ρ be the longest initial segment of σ such that $\rho 0 \sqsubseteq \sigma$ and $r(\rho) > \psi_\sigma(x)[s]$. (This means that ρ is the largest initial such that $\psi_\sigma(x)$ has to use the drip feed strategy to get into A_σ.) Add $\psi_\sigma(x)[s]$ to the list $\Lambda(C_\sigma)[s]$ and add $|\psi_\sigma(x)[s]|$ to $\Lambda(N_\rho)[s]$. ($\Lambda(C_\sigma)$ is the list C_σ uses to keep track of the strings it wants to put into A_δ using a drip feed strategy. $\Lambda(N_\sigma)$ is a list of numbers that N_σ uses to bookkeep for which (sizes of) strings a request has been made by a lower priority requirement to enter α.) If ρ does not exist, put $\psi_\sigma(x)[s]$ into A_σ straightaway.

Let δ be the string on top of the list $\Lambda(C_\sigma)[s]$. Let ρ be the longest initial segment of σ such that $\rho 0 \sqsubseteq \sigma$ and $r(\rho)[s] > |\delta|$. (The existence of such ρ is guaranteed by the fact that δ is on $\Lambda(C_\sigma)$.) See if $|\delta|$ is on the list $\Lambda(N_\rho)[s]$. If so, do nothing. If not, this means that δ has been successfully processed by N_ρ, and hence, by recursion, by all relevant N-strategies above σ. So in this case we put δ into A_σ and remove it from $\Lambda(C_\sigma)$, and we initialize all *negative* strategies $\tau \geq \sigma$ (meaning that the restraint value of these strategies becomes undefined).

We make $\alpha = \mu(A_\sigma)[s]$ by putting, if necessary, either fresh strings (not extending previous ones) into A_σ, or numbers into α.

Action for the negative requirement N_σ. Let $e = |\sigma|$. Define the following length

of agreement functions,

$$l(\sigma)[s] \;=\; \begin{cases} s & \text{if } \mu(W_e) = \alpha[s] \\ \min\{n : |\alpha - \mu(W_e)|[s] > 2^{-n}\} - 1 & \text{otherwise.} \end{cases}$$

$$m(\sigma)[s] \;=\; \max\{l(\sigma)[t] : t < s\}.$$

so that we always have $|\alpha - \mu(W_e)| \leq 2^{-l(\sigma)}[s]$. A σ-stage s is *σ-expansionary* if $l(\sigma) > m(\sigma)[s]$.

If s is not σ-expansionary we let $\sigma 1$ act at s, and we initialize all τ-strategies with $\sigma <_L \tau$.

If s is σ-expansionary we initialize all τ-strategies, $\tau \geq \sigma 1$. Whether $\sigma 0$ is allowed to act depends on the value of σ's counter $c(\sigma)$ (see two cases below).

If $\Lambda(N_\sigma)[s]$ is empty we set $r(\sigma) = l(\sigma)[s]$ and let $\sigma 0$ act. Otherwise, let n be the number on top of the list $\Lambda(N_\sigma)[s]$. If $c(\sigma) \uparrow [s-1]$ then set $c(\sigma) = 2^{-n+r(\sigma)}[s]$ and let $\sigma 0$ act. If $c(\sigma) \downarrow [s-1]$ there are two cases:

I. $c(\sigma)[s-1] = 0$. This means that σ's drip feed strategy for n has been successfully completed. Remove n from $\Lambda(N_\sigma)[s]$, set $r(\sigma) = l(\sigma)[s]$, and let $\sigma 0$ act.

II. $c(\sigma)[s-1] > 0$. This means that at a previous stage t this counter was set to some number $2^{-n+r(\sigma)}[t]$, and $r(\sigma)$ has not changed since. Let ρ be maximal with $\rho 0 \sqsubseteq \sigma$ and $r(\rho) > r(\sigma)[s]$. Since ρ acts its counter $c(\rho)$ must be 0 at s. See if $r(\sigma)[s]$ is on the list $\Lambda(N_\rho)[s]$. If so, do nothing. (In this case $r(\sigma)[s]$ is still waiting for its turn to start a drip feed strategy.) If not, $r(\sigma)[s]$ was removed from $\Lambda(N_\rho)$ when its counter $c(\rho)$ became 0 at some stage $\leq s$. So we let $c(\sigma)[s] = c(\sigma)[s-1] - 1$, and we put the next copy of $r(\sigma)[s]$ on the list $\Lambda(N_\rho)[s]$. If there is no such ρ we add $2^{-r(\sigma)}[s]$ to α.

This completes the construction.

Verification. Because the construction is recursive, α is a c.e. real, and every A_σ is a c.e. prefix-free domain.

We prove by induction along the true path g that for σ on g we have $A_\sigma \equiv_{wtt} U_e$, $\mu(A_\sigma) = \alpha$, and if W_e presents α then $W_e \leq_{wtt} \bigoplus_{\tau \sqsubseteq \sigma} A_\tau$, where $e = |\sigma|$.

First note that always $A_\sigma \leq_{wtt} U_e$: A string δ can only enter A_σ after $\psi_e(x)$ has been picked when $|\delta| > |\psi_e(x)|$ or $\delta = \psi_e(y)$ for some y. Since $|\psi_e(x)| > x$, if U_e does not change below $|\delta|$ then also A_σ doesn't. So $A_\sigma \leq_{wtt} U_e$ with use the identity function.

Claim For $\rho 0 \sqsubset g$, every number put on $\Lambda(N_\rho)$ is eventually removed from $\Lambda(N_\rho)$ again.
Namely, since there are infinitely many stages at which ρ acts, and ρ can only act when $c(\rho) = 0$ or when $\Lambda(N_\rho)$ is empty, infinitely often $\Lambda(N_\rho)$ is empty or its top element is removed. Since any element on the list has only finitely many predecessors on the list, every element is eventually removed.

Suppose that C_σ is never initialized after stage s, i.e. $g[s]$ is never to the left of σ. Note that $\tau \sqsubset \sigma$ can then only initialize σ's negative strategy. We prove that

$U_e \leq_m A_\sigma$ via ψ_σ. Now suppose x enters U_e at σ-stage $t > s$. Let $\rho \sqsubseteq \sigma$ be maximal with $\rho 0 \sqsubseteq \sigma$ and $r(\rho) > r(\sigma)[t]$. If no such ρ exists $\psi_\sigma(x)$ enters A_σ immediately. Otherwise, $\psi_\sigma(x)$ is added to the list $\Lambda(C_\sigma)$ and $|\psi_\sigma(x)|$ is added to $\Lambda(N_\rho)$. By the above claim it is eventually removed from $\Lambda(N_\rho)$. But then it is also eventually removed from $\Lambda(C_\sigma)$, and put into A_σ at the same stage. Since no other strategy can put $\psi_\sigma(x)$ into A_σ we have $x \in U_e \Leftrightarrow \psi_\sigma(x) \in A_\sigma$.

Now we also have that $\mu(A_\sigma) = \alpha$ because $\alpha = \mu(A_\sigma)[s]$ at the end of C_σ's action at every σ-stage s.

We verify that N_σ is satisfied. Suppose that $W_e \subseteq 2^{<\omega}$ is prefix-free and $\mu(W_e) = \alpha$. Suppose that s is such that the σ-strategies are never initialized after stage s. Let $\gamma \in 2^{<\omega}$. We compute whether $\gamma \in W_e$ as follows. Determine a σ-stage $t \geq s, |\gamma| + 1$ such that $A_\tau[s] \upharpoonright |\gamma|+1 = A_\tau \upharpoonright |\gamma|+1$ for all $\tau \sqsubseteq \sigma$, and such that $l(\sigma)[t] > |\gamma|+1$ and $r(\sigma) = l(\sigma)[t]$. We can compute the latter because the construction is recursive and $\lim_s l(\sigma)[s] = \infty$ by the assumption that $\mu(W_e) = \alpha$. Furthermore, we may choose t such that $r(\sigma) = l(\sigma)[t]$, because $\sigma 0$ will act infinitely often. By the definition of l we have $|\alpha - \mu(W_e)| \leq 2^{-l(\sigma)}[t]$. Which things can change $\alpha[t]$, and what is the effect on $\mu(W_e)$?

- The coding strategies C_τ with $\tau \sqsubseteq \sigma$ are done below $|\gamma| + 1$ by choice of t.
- Since all $\tau \geq \sigma 1$ are initialized at every σ-expansionary stage, the coding strategies C_τ with $\sigma <_L \tau$ cannot change α more than $2^{-t} \leq 2^{-|\gamma|-1}$ in total.
- The coding strategies C_τ with $\tau \sqsupseteq \sigma 0$ may wish to change $\alpha[t]$ below $r(\sigma)$. Note that $r(\sigma) = l(\sigma)[t]$. But they have to use the drip feed strategy in order to do this, meaning that they cannot disrupt the inequality $|\alpha - \mu(W_e)| < 2^{-r(\sigma)}[t]$ by more than $2^{-r(\sigma)}[t]$ ever, after which they have to wait until $\mu(W_e)$ grows closer to α than $2^{-r(\sigma)}[t]$ again. This means that $\mu(W_e)$ cannot change on any string of length smaller than $r(\sigma)[t]$, and in particular not on γ.

Summing up, $\mu(W_e)[t]$ cannot change more than

$$2^{-|\gamma|-1} + 2^{-r(\sigma)}[t] < 2 \cdot 2^{-|\gamma|-1} = 2^{-|\gamma|}.$$

So $\gamma \in W_e \Leftrightarrow \gamma \in W_e[t]$.

This completes the verification, and the proof of Theorem 5.

References

[1] Ambos-Spies, K., Nies, A., Shore, R. A., *The theory of the recursively enumerable weak truth-table degrees is undecidable*, Journal of Symbolic Logic 57 (1992) 864–874.

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

[3] Calude, C., Coles, R., Hertling, P., Khoussainov, B., *Degree-theoretic aspects of computably enumerable reals*, in *Models and Computability,* (ed. Cooper and Truss) Cambridge University Press, 1999.

[4] 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 1373 (1998) 596–606.

[5] Calude, C., and Nies, A., *Chaitin's Ω numbers and strong reducibilities*, Journal of Universal Computer Science 11(3) (1997) 1162–1166.

[6] Chaitin, G., *A theory of program size formally identical to information theory*, Journal of the ACM 22 (1975) 329–340.

[7] Downey, R., and LaForte, G., *On presentations of computably enumerable reals*, to appear in Theor. Comput. Sci.

[8] Downey, R. G., and Remmel, J. B., *Classification of degree classes associated with r.e. subspaces*, Ann. Pure and Appl. Logic, **42** (1989) 105–125

[9] Ko, Ker-I, *On the continued fraction representation of computable real numbers*, Theor. Comput. Sci, A, 47 (1986) 299–313,

[10] Li, M., and Vitányi, P., *Kolmogorov Complexity and its Applications*, 2nd edition, Springer-Verlag, 1997.

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

[12] Odifreddi, P. G., *Classical Recursion Theory Vol. II*, North-Holland, 1999.

[13] Pour-El, M., and Richards, I., *Computability in Analysis and Physics*, Springer-Verlag, Berlin, 1989.

[14] Soare, R., *Recursion theory and Dedekind cuts*, Trans. Amer. Math. Soc. 140 (1969) 271–294.

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

[16] Turing, A., *On computable numbers with an application to the Entscheidungsproblem*, Proc. Amer. Math. Soc. 43 (1937) 544–546.

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

[18] Wu, G., *Prefix-free languages and initial segments of computably enumerable degrees,* in: J. Wang (ed.), Computing and Combinatorics, Springer Lecture Notes in Computer Science 2108 (2001) 576–585.