A C.E. REAL THAT CANNOT BE SW-COMPUTED BY ANY Ω NUMBER

GEORGE BARMPALIAS AND ANDREW E.M. LEWIS

ABSTRACT. The strong weak truth table (sw) reducibility was suggested by Downey, Hirschfeldt, and LaForte as a measure of relative randomness, alternative to the Solovay reducibility. It also occurs naturally in proofs in classical computability theory as well as in the recent work of Soare, Nabutovsky and Weinberger on applications of computability to differential geometry. We study the sw-degrees of c.e. reals and construct a c.e. real which has no random c.e. real (i.e. Ω number) sw-above it.

1. INTRODUCTION

The strong weak truth table reducibility was suggested by Downey, Hirschfeldt, and LaForte as a measure of relative randomness. Versions of this reducibility are present in computability theory; for instance, these are automatically produced by the basic technique of 'simple permitting' and one of them was used in the recent work of Soare, Nabutovsky and Weinberger on applications of computability theory to differential geometry. The strong weak truth table reducibility naturally induces a degree structure, the sw degrees. Yu and Ding showed that the sw degrees restricted to the c.e. reals have no greatest element, and asked for maximal elements.

In Barmpalias[1] this question was solved for the case of c.e. sets by showing that there are no maximal elements in the sw degrees of the c.e. sets. The strong weak truth table reducibility was originally suggested as an alternative for the Solovay (or *domination*) reducibility which has been a very successful tool for the study of the complexity of c.e. reals but presents various shortcomings outside this class. Of course, the sw degrees present other difficulties (such as the lack of join operator, see below) but they are nevertheless very interesting to study from a wider perspective. Moreover, Downey, Hirschfeldt and LaForte [7] noticed that as far as the computably enumerable sets are concerned, the sw degrees *coincide* with the Solovay

---

Date: February 2005.
Key words and phrases. sw reducibility, c.e. reals, randomness.
Barmpalias was supported by E.P.S.R.C. Research Grant No. EP/C001389/1 "Extensions of Embeddings in the Local Turing Universe". Lewis was supported by E.P.S.R.C. grant No. GR /S28730/01. Both authors were partially supported by the N.S.F.C. Grand International Joint Project, No. 60310213, New Directions in the Theory and Applications of Models of Computation.

degrees. So by [1] we also know that the Solovay degrees of c.e. sets have no maximal element. In the following we assume basic computability theory background (e.g. see Odifreddi[10] and Soare[12]); knowledge of algorithmic randomness is also useful. For definitions, motivation and history of related notions such as randomness, prefix-free complexity and Solovay degrees we refer mainly to [6] and secondly to [4]. The forthcoming monograph [5] by Downey and Hirschfeldt contains all this background and more.

Studying relative randomness, Downey, Hirschfeldt and LaForte [7] found Solovay reducibility insufficient, especially as far as non-c.e. reals are concerned. One of the two new measures for relative randomness they suggested is a strengthening of the weak truth table reducibility, which they called *strong weak truth table reducibility* or sw for short. This reducibility is quite natural since it occurs in many proofs in classical computability theory: it follows when we apply simple permitting for the construction of a set 'below' a given one.

Definition 1. *(Downey, Hirschfeldt and LaForte [7]) We say $A \leq_{sw} B$ if there is a Turing functional Γ and a constant c such that $\Gamma^B = A$ and the use of this computation on any argument n is bounded by $n + c$.*

The special case when $c = 0$ gives the *ibT* reducibility which is closely related with a 'domination' reducibility used by Soare, Nabutovsky and Weinberger (see Soare[11], Nabutovsky and Weinberger [9]) in applying computability theory to differential geometry. In [2] we showed the following.

Theorem 1. *(Barmpalias and Lewis [2]) The ibT degrees of computably enumerable sets are not dense.*

However in this paper we will be concerned with computably enumerable reals and not c.e. sets. We recall the definition of a c.e. real.

Definition 2. *A real number is* computably enumerable (c.e.) *if it is the limit of a computable increasing sequence of rationals.*

C.e. sets are c.e. reals but the converse does not necessarily hold. But of course, c.e. reals are Δ_2. Note that although we often consider reals as sets (with characteristic sequence their binary expansion) and conversely, we do distinguish the meaning of 'c.e.' for the two cases. The main justification for $\leq_{\text{sw}}$ as a measure of relative randomness was the following

Proposition 1. *(Downey, Hirschfeldt, LaForte [7]) If $\alpha \leq_{sw} \beta$ are c.e. reals then for all n, the prefix-free complexity of $\alpha \upharpoonright n$ is less than or equal to that of $\beta \upharpoonright n$ (plus a constant).*

So $\leq_{\text{sw}}$ arguably qualifies as a *measure of relative randomness* for the c.e. reals (and in particular, it preserves randomness, i.e. if α is random and $\alpha \leq_{\text{sw}} \beta$ then β is random). Downey, Hirschfeldt and LaForte [7] have showed that Solovay reducibility (also known as *domination*) and strong weak truth table reducibility *coincide* on the c.e. sets. But, as we see below, this is not true for the c.e. reals. Yu and Ding proved the following

Theorem 2. *(Yu and Ding [13]) There is no sw-complete c.e. real.*

By a 'uniformization' of their proof they got two c.e. reals which have no c.e. real sw-above them. Hence

Corollary 1. *(Downey, Hirschfeldt, LaForte [7])*
The structure of the sw-degrees c.e. reals is not an upper semi-lattice.

They also asked whether there are maximal sw-degrees of c.e. reals. They conjectured that there are such, and they are exactly the ones that contain random c.e. reals. The main idea of their proof of theorem 2 can be applied for the case of c.e. sets in order to get an analogous result. Using different ideas Barmpalias[1] proved the following stronger result.

Theorem 3. *(Barmpalias[1]) There are no sw-maximal c.e. sets. That is, for every c.e. set A, there exists a c.e. set W such that $A <_{sw} W$.*

Note that since the Solovay degrees and sw-degrees coincide on the c.e. sets (see [6]) the following also holds.

Corollary 2. *(Barmpalias[1]) The substructure of the Solovay degrees consisting of the ones with c.e. members (i.e. containing c.e. sets) has no maximal elements.*

In this paper we show the following

Theorem 4. *There are c.e. reals α that cannot be sw-computed by any random c.e. real. That is, for any c.e. $\beta \geq_{sw} \alpha$, the number β is not random.*

To justify the title of this paper we mention that random c.e. reals are known to be exactly the Ω-numbers, i.e. the halting probabilities of universal prefix-free machines. For background and other characterisations of random c.e. reals we refer to Calude[3]. Theorem 4 can be seen as the local version of the following unpublished result of Hirschfeldt which was announced after we submitted this paper.

Theorem 5. *(Hirschfeldt) There are reals α which cannot be sw-computed by any random real. That is, for any $\beta \geq_{sw} \alpha$, the number β is not random.*

Note that Hirschfeldt's theorem concerns the global structure of the sw degrees and does not imply theorem 4.

2. About the structure

We state some easy results about the c.e. sets and reals in the structure of sw degrees. We recall the following definition:

Definition 3. *A Martin-Löf test $\mathcal{M}$ is a uniform sequence (E_i) of c.e. sets of binary strings such that $\mu(E_i) \leq 2^{-i}$. A real α avoids $\mathcal{M}$ if some for i, $\alpha \notin E_i\Sigma^\omega$. A real number is called random if it avoids all Martin-Löf tests.*

Here we identify reals in $(0,1)$ with infinite binary sequences (a real in $(0,1)$ corresponds to its binary expansion) and finite binary strings with intervals of $(0,1)$ (a string σ corresponds to the interval of reals which have the prefix σ in their binary expansion). We note the Lebesque measure of a set of reals by μ. Thus if E is a set of finite strings, $\mu(E)$ is the Lebesque measure of the union of the intervals in E. The set of all infinite binary strings is noted by Σ^ω and $E\Sigma^\omega$ is the set of all infinite binary strings with prefix one of the strings in E.

After the discussion in the previous section, it is natural to ask: are there c.e. reals above all c.e. sets? It is not hard to show the following

Proposition 2. *Every random c.e. real is sw-above every set in the finite levels of the difference hierarchy.*

A proof of this proposition for the first level of the difference hierarchy (the c.e. sets) appears in the forthcoming monograph [5] (in the section about the sw reducibility). The general case follows in a similar way. But are there non-random c.e. reals with this property?

Proposition 3. *There are non-random c.e. reals sw-above every set in the finite levels of the difference hierarchy.*

E.g. $\alpha = \sum_{e\in\mathbb{N}} \sum_{n\in W_e} 2^{-(e+n+2)}$ is non-random (the digit changes below certain levels depend on a finite number of c.e. sets and so a Martin-Löf test capturing α can be easily constructed) and sw-above all c.e. sets. The general case is handled similarly by considering the constants which bound the number of 'mind changes' in the various levels of the difference hierarchy.

3. Proof of theorem 4

From now on all reals will be c.e. and w.l.o.g. members of the unit interval $(0,1)$. We want to construct α such that any β that tries to sw-compute α either fails to do that or fails to escape a Martin-Löf test (which is constructed by us especially for β). In other words, β is unable to cope with both the tight α-coding needed and the digit changes required for the escape from the intervals of our Martin-Löf test. The requirements are

$$\mathcal{Q}_{\Phi,\beta} : \alpha = \Phi^\beta \Rightarrow \beta \in \cap\mathcal{M} \text{ for a Martin-Löf test } \mathcal{M}$$

where Φ runs over all partial computable sw-functionals and β over all c.e. reals in the unit interval. Looking at a single requirement we picture β (i.e. the opponent) having to cope with two kinds of instructions. One comes from the α-changes that have to be sw-coded into β. These occur after a change on a digit n of α and say 'change a digit in β below n'. Of course, if the use of Φ is the identity plus a constant c, the instruction will be 'change a digit in β below $n+c$' but at the moment we may assume $c=0$ for simplicity.

On the other hand β has to follow instructions of the type 'change a digit below n' where the sum $\sum 2^{-n}$ for each n occurring in the sequence of instructions (including repetitions) is bounded. These *test instructions* come from the desire of β to escape our Martin-Löf test and in particular a single member of it. A sequence (n_i) of test instructions can be identified with the sequence of intervals $(\sigma_i\Sigma^\omega)$ where σ_i is the string consisting of the first n_i digits of the current approximation to β (at the time when that test instruction is issued); the latter is of course a member of a Martin-Löf test $\mathcal{M}$ and failure to follow a test instruction means failure to escape that member of $\mathcal{M}$. Since the measure of the members of a Martin-Löf test goes to 0, we have to accept that the above sum $\sum 2^{-n}$ will be as small as our opponent wants.

The above setting is like a game between players A, B where A controls α and the sequence of *test instructions*, and B controls β. Once we sort out this atomic case, i.e. find a winning strategy for A, we can use the same ideas in a global construction which deals with the general case. A winning strategy for A means an enumeration of α and a sequence of test instructions such that any β which manages to code α is unable to follow all test instructions. Further, we want the strategy for A to have arbitrarily small cost, i.e.

(a) α is smaller than any threshold $\epsilon > 0$ set by the opponent. In other words we can implement the strategy by changing α on an arbitrarily remote (from the decimal point) segment.
(b) as mentioned before, the 'measure' $\sum_i 2^{-n_i}$ of the sequence of test instructions (n_i) is smaller than any threshold $\epsilon > 0$ set by the opponent.

The purpose behind (b) was mentioned above while (a) is needed in order to put all strategies together in a global construction (given that we only have one α). Once we have such a winning strategy for A in the above game we can first iterate the strategy with sufficiently low cost each time thus getting that *any β which manages to code α is unable to escape any of the members of the Martin-Löf test induced by the sequences of test instructions.* And by iterating again the previous module we will be able to deal with all requirements $\mathcal{Q}_{\Phi,\beta}$. Note that the cost-effectiveness conditions (a), (b) are vital in implementing these iterations.

The following lemma will simplify things in designing a *winning strategy for A*.

Lemma 1. *In the game described above between A, B a best strategy for B is to increase β by the least amount needed to satisfy A's request (i.e. a change of a digit in β below a certain level), each time a request is put forward by A. In other words if a different strategy for B produces β' then at each stage s of the game $\beta_s \le \beta'_s$.*

Proof. By induction on the stages s. At stage 0, $\beta_0 \le \beta'_0$. If $\beta_s < \beta'_s$ (the case $\beta_s = \beta'_s$ is trivial) then there will be a position n such that $0 = \beta_s(n) < \beta'_s(n) = 1$ and $\beta_s \restriction n = \beta'_s \restriction n$. When A issues a request for a β, β' change on position t or higher in stage $s+1$, if $t < n$ it is clear that $\beta_{s+1} \le \beta'_{s+1}$.

$$\alpha = 0\ .\ \overbrace{0\ \dots\ 0}^{\ell}\ \overbrace{0\ 0\ \dots\ 0}^{n}\ 1\ \longmapsto$$
$$\beta = 0\ .\ \underbrace{0\ \dots\ 0}_{\ell}\ \underbrace{1\ 0\ \dots\ 0}_{n}\ \longmapsto$$

FIGURE 1. An n-ahead (between α, β) at position $n + \ell + 1$.

Otherwise the highest change β will be forced to do is on n and so again $\beta_{s+1} \leq \beta'_{s+1}$. □

This observation allows us to assume a particular strategy for B without losing any generality. The idea is that A forces β to be larger and larger until it exits the unit interval. But since our reals are in $(0, 1)$, B will have to either abandon the α-coding or stop trying to escape the intervals issued by A towards a Martin-Löf test. By lemma 1 if this happens with our standard 'best strategy' β, it will also happen with any other β' which follows a different strategy.

Assuming the best B-strategy of lemma 1, every A-strategy corresponds to a single B-strategy and so a single β.

Definition 4. *The positions on the right of the decimal point in a binary expansion are numbered as* $1, 2, 3, \dots$ *from left to right. The first position on the left of the decimal point is position* 0.

Definition 5. *Suppose* $0 < n \leq t$. *We say that there is an* n*-ahead between* α, β *at position* t *when*

- $\beta(t - n) = 1$ *and* $\beta(i) = 0$ *for* i *strictly between* $t - n, t$
- $\alpha(t) = 1$ *and* $\alpha(i) = 0$ *for* $i = t - n$ *or strictly between* $t - n, t$
- *For all positions* k *strictly between the decimal point and* $t-n$, $\alpha(k) = \beta(k) = 0$.

An illustration of a typical case of this definition is shown in figure 1. Where arrows appear the digits can be anything. Note that this definition allows the possibility $\beta \geq 1$. Since the reals considered in the general construction are in $(0, 1)$ this case will only be used for the sake of deriving a contradiction.

Given arbitrary $n, t > 0$ and a rational $\epsilon > 0$ we wish to design a finite (i.e. consisting of finitely many steps) A-strategy with cost of the occurring test instructions less than ϵ, which leaves α, β with an n-ahead at position t. As mentioned before, the cost of the test instructions $n_1, \dots, n_k$ is $\sum_{i=1}^{k} 2^{-n_k}$.

3.1. **Case** $n = 1$**.** This case is easy: issue the test instruction 'change β on position k' for some (e.g. the least) $k > t$ with $2^{-k} < \epsilon$ and then carry on the consecutive α-changes in positions $k, k-1, \dots, t$ again using the least effort (i.e. adding the least amount to α) required for the changes to take place.

α	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0
β	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0

FIGURE 2. A-strategy for the production of a 1-ahead at position 2 with cost 2^{-4}.

Figure 2 illustrates the A strategy for $n = 1$, $t = 2$ and $\epsilon > 2^{-4}$ (so we chose $k = 4$). The first 4 digits of α, β are shown in consecutive stages of the game and one can see the action of A along with the response of B. When α does not change and β does, a test-instruction is issued.

3.2. **Case $n > 1$.** In this case the strategy for A will be more sophisticated and we are going to build it by recursion. We note that *given n, t, ϵ the A-strategy will only change α at positions $\geq t$ and issue test instructions for β-change at positions $> t$.* The following lemma will be very useful; the idea behind its name will be explained later. When we say 'before position i' we mean 'at i or higher', i.e. $\leq i$.

Lemma 2. (Passing through lemma) *Consider two games (like the one described above), one between A, B and the other between A', B', running in parallel (the first controls α, β and the second one α', β'). Although the strategies of B, B' are the same (i.e. the 'least effort' strategy described above) and the A, A' strategies (i.e. α, α' increments and test instructions) are identical, the first starts with $\alpha[0] = \beta[0] = 0$ (as usual) and the second with $\alpha[0] = \sigma_1$, $\beta'[0] = \sigma_2$ for finite binary expansions σ_1, σ_2. If A, A' only ever demand changes (by either changing α, α' or issuing test instructions) before positions $n > \max\{|\sigma_1|, |\sigma_2|\}$ then at every stage s,*

$$\alpha'[s] = \alpha[s] + \sigma_1 \tag{1}$$

$$\beta'[s] = \beta[s] + \sigma_2. \tag{2}$$

Proof. Equation (1) is an obvious consequence of the hypotheses. By induction on s we show that (2) holds and $\beta'[s], \beta[s]$ have the same expansions after position $|\sigma_2|$. For $s = 0$ it is obvious. Suppose that this double hypothesis holds at stage s. At $s+1$, A demands a change before position $n > |\sigma_2|$ and since β, β' look the same on these positions, $\beta'[s]$ will need to increase by the same amount that $\beta[s]$ needs to increase. So $\beta'[s+1] = \beta[s+1] + \sigma_2$ and one can also see that β, β' will continue to look the same at positions $> |\sigma_2|$ (consider cases whether the change occurred at positions $> |\sigma_2|$ or not). □

The idea for the recursion which will give an efficient strategy for A is as follows. Fix n, t as before. Suppose that for $i = t+1, \ldots, t+n-1$ we have procedures $P(n, i)$ which (working in isolation) produce an n-ahead between α, β at position i and at cost (i.e. measure of test instructions) $C(n, i)$. Moreover they only demand β-changes (either by changing α or by test instructions) on positions $\geq i$. Then we can define $P(n, t)$ which only demands β-changes before position t and produces an n-ahead between α, β at position t as follows:

3.2.1. *$P(n,t)$ modulo $P(n,i)$, $i = t+1, \dots, t+n-1$.*

(p_1) Produce an n-ahead at position $t+1$ by running $P(n, t+1)$.
(a_1) Change α at position t.
(p_2) Produce an n-ahead at position $t+2$ by running $P(n, t+2)$.
(a_2) Change α at position $t+1$.
⋮ ⋮ ⋮
(p_{n-1}) Produce an n-ahead at position $t+n-1$ by running $P(n, t+n-1)$
(a_{n-1}) Change α at position $t+n-2$.
(f) Produce an 1-ahead at position $t+n-1$ by running the usual procedure of 'case $n = 1$'.

Notice the slight abuse of the term n-ahead in the procedure above. Since we begin applying some $P(n,i)$ when $\alpha = \sigma_1$, $\beta = \sigma_2$ (possibly non-zero) a precise statement would be 'after $P(n,i)$ has finished there is an n-ahead between $\alpha - \sigma_1$ and $\beta - \sigma_2$ in position i'. The latter is true because of the *passing through lemma 2* which can be applied because of the nature of $P(n,t)$ and the hypotheses on $P(n,i)$. One can now justify the name of this lemma by looking at the illustration of procedure $P(n,t)$ (figure 3): it allows us to create 'n-aheads' within already developed α, β. Observe that by the hypothesis on $P(n,i)$, *procedure $P(n,t)$ only demands β-changes (either by changing α or by test instructions) on positions $\geq t$.* Also note that the final step (f) is not fully determined as section 'case $n = 1$' describes a bunch of procedures with variable cost (which can be arbitrarily small). So, modulo $P(n,i)$ we can think of $P(n,t)$ as a family of very similar procedures which differ only on the version of their final step (and the cost of their final step is arbitrarily small).

Figure 3 shows a final segment of α (first line) and β (second line) during the stages when strategy $P(n,t)$ is played by A. It is assumed that β responds to all instructions issued by A so far (otherwise it would lose) and that it uses the 'least effort strategy' explained above. The first position shown is position $t-n$ and the first after the first ellipsis mark is position $t-3$. The first position after the second ellipsis mark is position $n+t-3$. The symbol $*$ indicates that a 0 or 1 occupies the position, but it does not matter which. The arrow $\mapsto$ indicates that from that position on it does not matter what follows and the ellipsis mark '...' that the following digits will be as the last one before the mark. Note that the positions higher than $t-n$ are not affected during the whole process and $P(n,t)$ finishes with an n-ahead at position t and a block of 1s of length n starting from position t in α. So after the final step (f) the positions in α from $\mapsto$ on will have 0s.

$$
\begin{array}{cl}
(p_1) & \left| \begin{array}{cccccccccc} 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 & * & \mapsto \\ 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & * & * & \mapsto \end{array} \right. \\[2ex]
(a_1) & \left| \begin{array}{cccccccccc} 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & \dots & 0 & 0 & 1 & 0 & 0 & \dots \end{array} \right. \\[2ex]
(p_2) & \left| \begin{array}{cccccccccc} 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 0 & 1 & \mapsto \\ 0 & 1 & 1 & 0 & \dots & 0 & 0 & 1 & 0 & * & \mapsto \end{array} \right. \\[2ex]
(a_2) & \left| \begin{array}{cccccccccc} 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 1 & 0 & \dots \\ 0 & 1 & 1 & 0 & \dots & 0 & 0 & 1 & 1 & 0 & \dots \end{array} \right. \\[2ex]
 & \qquad\qquad \vdots \qquad\qquad\qquad \vdots \\[2ex]
(p_{n-1}) & \left| \begin{array}{cccccccccccccc} 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 1 & 1 & \dots & 1 & 0 & 1 & \mapsto \\ 0 & 1 & 1 & 1 & \dots & 1 & 1 & 1 & 1 & 1 & \dots & 1 & 0 & * & \mapsto \end{array} \right. \\[2ex]
(a_{n-1}) & \left| \begin{array}{cccccccccccccc} 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 1 & 1 & \dots & 1 & 1 & 0 & \dots \\ 0 & 1 & 1 & 1 & \dots & 1 & 1 & 1 & 1 & 1 & \dots & 1 & 1 & 0 & \dots \end{array} \right. \\[2ex]
(f) & \left| \begin{array}{cccccccccccccc} 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 & 1 & 1 & \dots & 1 & 1 & 1 & \mapsto \\ 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots \end{array} \right.
\end{array}
$$

FIGURE 3. A final segment of α, β during the stages of $P(n,t)$ modulo $P(n,i)$, $i = t+1, \dots, t+n-1$.

Let $C(n,t)$ be the infimum of the costs (measure of test instructions) of procedures $P(n,t)$; as noted before, this is a family of similar strategies depending on the version of the procedure used to produce the 1-ahead on step (f). And let $C(n,t+i)$, $i = 1, \dots, n-1$ be the infimum of the costs of the procedures we have available for step p_i. Since step (f) can have arbitrarily small cost we have

$$C(n,t) \leq \sum_{i=1}^{n-1} C(n,t+i). \tag{3}$$

Note that the more times we apply the recursion we described (procedure $P(n,t)$ modulo $P(n,i)$, $i = t+1, \dots, t+n-1$) in order to get a strategy, the longer the segment of α, β we need to work on is. It is clear that in order to have a definite strategy we need to start from somewhere , i.e. some fixed strategy producing an n-ahead on some large positions. Then we can start applying the recursion in order to get n-aheads on higher positions. We call this fixed strategy *crude* because it has relatively large cost.

α	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
β	0	0	0	0	0	1	0	0	0	1	1	0	0	1	1	1	1	0	0	0

FIGURE 4. Crude strategy for the production of a 3-ahead.

3.2.2. *Crude strategy.* Suppose we want to produce an n-ahead at position t. The idea is the following:

(1) Ask for a β-change on position $t-n+1$ or higher via a test instruction.
(2) Ask for a β-change on position $t-n+2$ or higher via a test instruction.
$\vdots$ $\vdots$ $\vdots$
(n) Ask for a β-change on position t or higher via a test instruction.
(f) Change α at position t.

The problem is that this procedure does not fulfill our requirement that the test instructions are issued only for positions $\geq t$ so that the passing through lemma does not apply and the recursion cannot be done properly. However a simple modification of it suffices: for $i = 1, \ldots, n$ we replace (i) with a $2^{t-(t-n+i)} = 2^{n-i}$ times iteration of the command:

Ask for a β-change on position t or higher via a test instruction.

It is easy to see that this modification does not alter the result (nor the cost) of the previous formulation of the crude strategy and so we are going to adopt it; now (i) means the relevant batch of commands. Figure 4 shows α (first line) and β (second line) during the end of batches (1), (2), (3), (4) and the final step (f) for the case when we want to produce a 3-ahead at some position t (so the first digit shown is $t-3$). The cost has been $2^{-(t-2)} + 2^{-(t-1)} + 2^{-t} < 2^{-(t-3)}$. In general the cost of the crude strategy for the production of an n-ahead at position t is

$$< 2^{-(t-n)}.$$

3.2.3. *A definite strategy for A.* Given $n \leq t$ we want to define a strategy for A which produces an n-ahead at position t and which depends on the length of the initial segment of α, β that we are allowed to use. Suppose that the length of this segment is a given $n_0 \geq n+t-1$. During the production of 1-aheads in final steps f of procedures P we allow the usage of longer segments of α, β (we could avoid this exception but then the argument would be slightly different, see below).

For $n_0 - n + 2 \leq i \leq n_0$ we define $P(n, i)$ to be the crude strategy for an n-ahead at position i. Note that these strategies are exactly what we need in order to define $P(n, n_0-n+1)$ (recall that $n_0-n+1 \geq t$) by the recursion explained above. For $t \leq i < n_0-n+2$ define $P(n, i)$ successively from large

i to small, using the recursion of section 3.2.1. Recall that from $i = n_0 - n + 1$ and less, $P(n, i)$ is a family of similar strategies with members derived from a choice of members from $P(n, i+1), \ldots, P(n, i+n-1)$, application of the recursion and a choice of the version of the final step (f) of the recursion. Note that the procedures P defined with the crude strategy are considered as families with one element.

Now we wish to find the infimum $C(n, t)$ of the costs of procedure $P(n, t)$. We do this by going through the 'backward' recursion of the definition of $P(n, t)$. First of all we know by section 3.2.2 that

$$C(n, n_0) < 2^{-(n_0-n)}, C(n, n_0 - 1) < 2^{-(n_0-1-n)}, \ldots$$
$$\ldots, C(n, n_0 - n + 2) < 2^{-(n_0+1-2n)}$$

Now recall (3) and consider the sequence

$$C(n, n_0), \ldots, C(n, n_0 - n + 2), C(n, n_0 - n + 1), \ldots, C(n, t).$$

After $C(n, n_0 - n + 2)$ each term is less than or equal to the sum of the previous $n - 1$ terms (according to (3)). So if we define

$$D(n, i) := \begin{cases} 2^{i-1}, & \text{if } 1 \le i < n; \\ \sum_{j=1}^{n-1} D(n, i-j), & \text{if } i \ge n. \end{cases}$$

(i.e. D is the sequence with the i-th term for $i \ge n$ equal to the sum of the previous $n - 1$ terms and the first $n - 1$ terms equal to $2^0, \ldots, 2^{n-2}$ respectively) then by induction we have

$$C(n, t) \le D(n, n_0 - t + 1)\ 2^{-(n_0-n)} \tag{4}$$

Since the family of procedures $P(n, t)$ defined depends on the choice of n_0 we write $P(n, t)[n_0]$ to indicate it. If we show that

$$\lim_{n_0 \to \infty} \frac{D(n, n_0)}{2^{n_0}} = 0 \tag{5}$$

then for any given $\epsilon > 0$ we can effectively find a suitable n_0 such that the corresponding family $P(n, t)[n_0]$ contains members which produce an n-ahead at position t at cost less than ϵ. Since the members of $P(n, t)[n_0]$ can be enumerated effectively along with their associated costs, a particular procedure can be found which produces an n-ahead at position t at cost less than ϵ.

3.2.4. *Proof of (5).* First observe that for all k

$$D(n, k+1) \le 2D(n, k). \tag{6}$$

Indeed, for $k < n$ this is clear and for $k \ge n$ we have that

$$D(n, k) = \sum_{j=1}^{n-1} D(n, k-j).$$

But in order to form $D(n,k+1)$ we take $D(n,k)$ and add only

$$\sum_{j=1}^{n-2} D(n,k-j)$$

so that (6) holds and the sequence $(\frac{D(n,k)}{2^k})$ is decreasing. Now if $k \geq n$ this means that

$$2^{n-1} D(n,k-n+1) \geq D(n,k).$$

But then

$$D(n,k+1) = 2D(n,k) - D(n,k-n+1) \leq 2D(n,k) - \frac{D(n,k)}{2^{n-1}} = \\ = 2D(n,k)(1-\frac{1}{2^n}).$$

Thus

$$\frac{D(n,k+1)}{2^{k+1}} \leq (1-\frac{1}{2^n})\frac{D(n,k)}{2^k}$$

and (5) follows.

3.3. **Variation of strategies and many strategies working together.** Recall that so far we assumed that the use of the sw-functional which computes α from β is the identity. In general it will be $x+c$ on argument x and so we need to modify the procedures P which produce n-aheads. For arbitrary c we consider the modification $P_c(n,t)[n_0]$ of $P(n,t)[n_0]$ which produces an n-ahead on position $t+c$. Looking at $P(n,t)[n_0]$ working in isolation, in the presence of a constant c the α-changes cause β-reaction as if they occur c places ahead (i.e. lower). So in order to apply our reasoning as before, we need to modify the test instructions: these will now ask for changes c places ahead than they did before. If we call this modified strategy $P_c(n,t+c)[n_0]$ it is clear (recalling that β follows the 'least effort' strategy) that the usual reasoning applies giving us the production of an n-ahead on position $t+c$ at cost same as in $P(n,t+c)[n_0]$ and by issuing instructions and α changes on positions $\geq t+c$. Assume an effective list of all requirements

$$\mathcal{Q}_1, \mathcal{Q}_2, \ldots$$

based on an effective list (Φ_i, β_i) of all pairs of partial computable sw-functionals and c.e. reals in $(0,1)$. We break each $\mathcal{Q}_i$ into $\mathcal{Q}_{i,1}, \mathcal{Q}_{i,2}, \ldots$ where

$$\mathcal{Q}_{i,j} : \alpha = \Phi_i^{\beta_i} \Rightarrow \beta \in E_j^i$$

where E_j^i is the jth member of the Martin-Löf test we are constructing; so $\mu(E_j^i) < 2^{-j}$. Now we adopt a priority list of all $\mathcal{Q}_{i,j}$ based on the following priority relation

$$\mathcal{Q}_{i_1,j_1} > \mathcal{Q}_{i_2,j_2} \iff \langle i_1, j_1 \rangle < \langle i_2, j_2 \rangle$$

where $\langle\ ,\ \rangle$ is a standard pairing function.

To each $\mathcal{Q}_{i,j}$ we assign an interval I^i_j in the characteristic sequences of α, β_i where it can apply the instructions of its strategy. Here is how we define I^i_j: suppose that the intervals for higher priority requirements have been defined and that n is the least number larger than all numbers in these intervals. The strategy for $\mathcal{Q}_{i,j}$ will be one of the $P_c(n+c, n+c)[n_0]$ for some big enough n_0, where c is the constant in the use of the sw-functional Φ_i. In particular we effectively search for a n_0 such that there are strategies in $P_c(n+c, n+c)[n_0]$ which produce a $(n+c)$-ahead on position $n+c$ at cost less than 2^{-j} (this search will halt due to (5)). Then we fix one of these $P_c(n+c, n+c)[n_0]$-strategies and find the largest position k where a P_c-instruction for β_i-change is issued (either in the form of an α-change or as a test instruction). Define $I^i_j = [n, k]$. Recall that test instructions are translated into members of E^i_j as follows: whenever such an instruction is issued requiring the change of a β-digit $> n$, the sequence $\beta \upharpoonright n$ (where β has current value) is enumerated into E^i_j.

Note that here we also assigned a strategy for $\mathcal{Q}_{i,j}$. We explain why this works, i.e. why $\mathcal{Q}_{i,j}$ is satisfied in this way. Suppose that inside I^i_j β_i only changes when an instruction is issued by $\mathcal{Q}_{i,j}$ and when this happens it increases by the least amount needed to follow the relevant instruction. Then according to the above, P_c produces an $n+c$ ahead at position $n+c$ and so, if β_i follows all instructions, it will end up outside $(0, 1)$, a contradiction. According to lemma 1 this will happen even if β_i diverts from the 'least effort strategy related to $\mathcal{Q}_{i,j}$' described above. So in any case, β_i will fail to follow either an α change or a test instruction. In other words, $\Phi^\beta_i \neq \alpha$ or $\beta \in E^i_j$. If we translate each α change on the nth position into the instruction 'change $\beta \upharpoonright (n + c + 1)$' then the last argument is summarised as:

Lemma 3. *Any real β which follows the instructions of $P_c(n+c, n+c)$ (any procedure in this class) has to be ≥ 1.*

Now we are going to put all strategies together and we will see that there is little interaction between them. In particular, the arguments above work in the case of a global construction because of lemma 3. One thing to note is that when a change happens in α inside I^i_j, all lower (larger) positions *inside* I^i_j become 0 (according to the 'least effort strategy' on the part of α operated by $\mathcal{Q}_{i,j}$). However α remains the same in positions outside I^i_j.

3.4. **Construction.** The construction takes place in stages. At stage s $\mathcal{Q}_{i,j}$ requires attention if $\Phi^{\beta_i}_i = \alpha$ currently holds on all arguments up to the largest number in I^i_j and $\beta_i[s] \notin E^i_j[s]$. At stage s consider $\mathcal{Q}_{i,j}$ for $\langle i, j\rangle < s$ and access the one of highest priority which requires attention. Then perform the next step of its strategy and end stage s.

3.5. **Verification.** Fixing i, j we will show that $\mathcal{Q}_{i,j}$ is satisfied. Since the strategy we assign to $\mathcal{Q}_{i,j}$ produces E^i_j with $\mu(E^i_j) < 2^{-j}$ the satisfaction of all $(\mathcal{Q}_{i,j})_{j\in\omega}$ implies the satisfaction of $\mathcal{Q}_i$ and the satisfaction of all $\mathcal{Q}_{i,j}$ implies the theorem. Suppose that the reduction $\Phi_i^{\beta_i} = \alpha$ is total (otherwise the satisfaction is trivial). If $\beta_i \not\in E^i_j$ then it follows all test instructions issued by $\mathcal{Q}_{i,j}$. And since α is successfully being coded into β_i, it follows all instructions of some procedure in $P_c(n+c, n+c)$. According to lemma 3 $\beta_i \geq 1$, a contradiction.

The lack of induction in the verification (if $\mathcal{Q} > \mathcal{Q}'$ then the satisfaction of $\mathcal{Q}'$ does not depend on the satisfaction of $\mathcal{Q}$) indicates a lack of interaction amongst the strategies.

3.6. **Further comments.** Recall that in procedure $P(n,t)[n_0]$ the final steps (f) in the recursions '$P(n,t)$ modulo $P(n,i)$, $i = t+1,\dots,t+n-1$' were left subject to choice from a pool of 1-ahead procedures with arbitrarily small cost. Instead we could choose for (f) a particular one, e.g. the one which starts with a test instruction 'change β below n_0' and moves the 1-ahead up to the position where we want it. Then the cost of each (f) in $P(n,t)[n_0]$ would be 2^{-n_0} and (4) would hold for

$$D(n,i) := \begin{cases} 2^{i-1}, & \text{if } 1 \leq i < n; \\ \sum_{j=1}^{n-1} D(n, i-j) + 2^{-n}, & \text{if } i \geq n. \end{cases}$$

Then the argument of section 3.2.4 is still valid, giving us (5). So the cost of $P(n,t)[n_0]$ (which is now a single strategy) tends to 0 as $n_0 \to \infty$ and in the construction we only need to search for a big enough n_0.

References

[1] G. Barmpalias, Computably enumerable sets in the Solovay and the Strong Weak Truth Table Degrees, *Lecture Notes in Computer Science*, Volume **3526**, Springer-Verlag, Proceedings of the CiE 2005 Conference 2005.

[2] G. Barmpalias and A.E.M. Lewis, The *ibT* Degrees of Computably Enumerable Sets are Not Dense, to appear in the *Annals of Pure and Applied Logic*.

[3] C.S. Calude, A characterisation of c.e. random reals. *Theoret. Comput. Sci.* **271** no. 1-2 (2002) 3–14.

[4] R. Downey, Some Recent Progress in Algorithmic Randomness, *Lecture notes in Computer Science* **3153** (2004) 42–81, Springer-Verlag, Proceedings of the 29^{th} Annual Conference on Mathematical Foundations of Computer Science, Prague, August 2004.

[5] R. Downey, D. Hirschfeldt, Algorithmic Randomness and Complexity, Monograph in preparation.

[6] R. Downey; D. Hirschfeldt; G. LaForte, Randomness and reducibility. *Journal of Computer and System Sciences* **68** (2004) 96–114

[7] R. Downey; D. Hirschfeldt; G. LaForte, Randomness and reducibility. Mathematical foundations of computer science, 2001, 316–327, *Lecture Notes in Comput. Sci.*, 2136, Springer, Berlin, 2001.

[8] R. Downey, D. Hirschfeldt and A. Nies, Randomness, computability, and density *SIAM Journal on Computing*, Vol. **31** No. 4, pp. 1169–1183 (2002)

[9] A. Nabutovsky and S. Weinberger, The Fractal Nature of Riem/Diff I, *Geometrica Dedicata* **101** (2003) 1–54

[10] P. Odifreddi, 'Classical recursion theory' Amsterdam Oxford: North-Holland, 1989

[11] R. Soare, Computability Theory and Differential Geometry, *Bulletin of Symbolic Logic* **10** (2004) no.4, 457–486.

[12] R. Soare, *Recursively enumerable sets and degrees*, Berlin London: Springer-Verlag, 1987.

[13] L. Yu and D. Ding, There is no SW-complete c.e. real, *J. Symb. Logic*, **69** (2004) no.4, 1163–1170.

E-mail address: georgeb@maths.leeds.ac.uk

E-mail address: thelewisboy@hotmail.com

School of Mathematics, University of Leeds, Leeds LS2 9JT, U.K.