LOWNESS FOR DEMUTH RANDOM

§1. No set of hyperimmune degree can be low for Demuth random. We investigate the notion of lowness for Demuth random. In this section, we show that no set of hyperimmune degree can be low for Demuth random. In particular, no Δ_2 set is low for Demuth random. We let W_x be the x^{th} c.e. set, and we identify finite strings with their code numbers. We treat W_x as a c.e. open set, consisting of basic clopen sets. We say that $[\sigma] \in W_x$ to mean that the code number of σ is in W_x, and we say that a (finite) string $\tau \in W_x$ if $\tau \supseteq \sigma$ for some $[\sigma] \in W_x$. Equivalently we say that τ is *captured by* W_x. The same definition holds if we replace τ by an infinite binary string.

THEOREM 1.1. *No set of hyperimmune degree can be low for Demuth random.*

PROOF. Suppose A is of hyperimmune degree. Let h^A be total computable in A and non-decreasing, which escapes domination by all total computable functions. That is, for all total computable g, $\exists^\infty x(g(x) < h^A(x))$. We build a $Z \leq_T A'$ which is Demuth random, but not Demuth random relative to A. To do this, we give an A-computable approximation $\{Z_s\}$ to Z. The construction will try to achieve two goals. The first is to make Z Demuth random by making Z avoid all Demuth tests. The second goal is to ensure that for infinitely many x, there are at most $h^A(x)$ many mind changes of $Z_s\restriction_x$. Hence Z looks like it is ω-c.e. in A, and cannot be Demuth random relative to A.

1.1. The motivation. Before we describe the strategy used to prove theorem 1.1, let us see why an attempted construction of a c.e. set A which is low for Demuth random fails. Let us consider a single (relativized) Demuth test $\{V_x^A\}$, played by the opponent, where the index for V_x^A can change $h^A(x)$ times. Now we have to cover $V_x^A \subseteq U_x$ with a plain Demuth test $\{U_x\}$. If $h^A(x) = 0$ for all x, then we could just follow the construction of a c.e. set which is low for random. We would enumerate y into A (to make A non-computable) if the associated cost of doing so, is small. Even when h^A is computable, we can always arrange the enumerations so that $V_x^A \subseteq U_x$ eventually, because we could use $h^A(x)$ as the bound for the index change of U_x.

The problem is that an enumeration into A not only increases the amount we have to put into U_x, but also gives the opponent a chance to redefine $h^A(x)$. Suppose he has defined $h^A(x)$ with use b_x. At some stage we will have to commit ourselves to a number $g(x)$, and promise never to change the index for U_x more than $g(x)$ times. We would of course declare that $g(x) > h^A(x)$, but once we do that, the opponent could challenge us to change $A\restriction_{b_x}$ to ensure the non-computability of A. We have to eventually change $A\restriction_{b_x}$ at some x, and allow the opponent to make $h^A(x) > g(x)$, and then we are stuck.

Note that the opponent will be likely to have a winning strategy, if h^A escapes domination by all computable functions. He could then carry out the above, patiently waiting for an x such that $h^A(x) > \varphi_e(x)$ for each e, and then defeat the e^{th} Demuth test. This is the basic idea used in the following proof, where we will play the opponent's winning strategy.

1.2. Listing all Demuth tests. In order to achieve the first goal, we need to specify an effective listing of all Demuth tests. It is enough to consider all

Received by the editors November 2, 2007.

Demuth tests $\{U_x\}$ where $\mu(U_x) < 2^{-3(x+1)}$. Let $\{g_e\}_{e\in\mathbb{N}}$ be an effective listing of all partial computable functions of a single variable. For every g in the list, we will assume that in order to output $g(x)$, we will have to first run the procedures to compute $g(0), \cdots, g(x-1)$, and wait for all of them to return, before attempting to compute $g(x)$. This minor but important restriction on g ensures that:

(i) $dom(g)$ is either $\mathbb{N}$, or an initial segment of $\mathbb{N}$,
(ii) for every x, $g(x+1)$ converges strictly after $g(x)$, if ever,
(iii) g is non-decreasing if it is total (we can arrange for this).

By doing this, we will not miss any total non-decreasing computable function. It is easy to see that there is a total function $k \leq_T \emptyset'$ that is universal in the following sense:

1. if $f(x)$ is ω-c.e. then for some e, $f(x) = k(e,x)$ for all x,
2. for all e, the function $\lambda x k(e,x)$ is ω-c.e.,
3. there is a uniform approximation for k such that for all e and x, the number of mind changes for $k(e,x)$ is bounded by

$$\begin{cases} g_e(x) & \text{if } g_e(x)\downarrow, \\ 0 & \text{otherwise.} \end{cases}$$

Let $k(e,x)[s]$ denote the approximation for $k(e,x)$ at stage s. Denote $U^e_x = W_{k(e,x)}$, where we stop enumeration if $\mu(W_{k(e,x)}[s])$ threatens to exceed $2^{-3(x+1)}$. Then for each e, $\{U^e_x\}$ is a Demuth test, and every Demuth test is one of these. To make things clear, we remark that there are two possible ways in which $U^e_x[s] \neq U^e_x[s+1]$. The first is when $k(e,x)[s] = k(e,x)[s+1]$ but a new element is enumerated into $W_{k(e,x)}$. The second is when $k(e,x)[s] \neq k(e,x)[s+1]$ altogether; if this case applies we say that U^e_x has a *change of index at stage* $s+1$.

1.3. The strategy. Now that we have listed all Demuth tests, how are we going to make use of the function h^A? Note that there is no single universal Demuth test; this complicates matters slightly. The e^{th} requirement will ensure that Z passes the first e many (plain) Demuth tests. That is,

$$\mathcal{R}_e: \ Z \text{ is captured by } U^0_x, U^1_x, \cdots, U^e_x \text{ for only finitely many } x.$$

$\mathcal{R}_e$ would do the following. It starts by picking a number r_e, and decide on $Z\restriction_{r_e}$. This string can only be captured by U^k_x for $x \leq r_e$ and $k \leq e$, so there are only finitely many pairs $\langle k, x\rangle$ to be considered; let S_e denote the collection of these. If any $U^k_x \in S_e$ captures $Z\restriction_{r_e}$, we would change our mind on $Z\restriction_{r_e}$. If at any point in time, $Z\restriction_{r_e}$ has to change more than $h^A(0)$ times, we would pick a new follower for r_e, and repeat, comparing with $h^A(1), h^A(2), \cdots$ each time. The fact that we will eventually settle on a final follower for r_e, will follow from the hyperimmunity of A; all that remains is to argue that we can define an appropriate computable function *at each* $\mathcal{R}_e$.

Suppose that $r^0_e, r^1_e, \cdots$ are the followers picked by $\mathcal{R}_e$. The required computable function P would be something like $P(n) = \sum_{k\leq e}\sum_{x\leq r^n_e} g_k(x)$, for if $P(N) < h^A(N)$ for some N, then we would be able to change $Z\restriction_{r^N_e}$ enough times on the N^{th} attempt. There are two considerations. Firstly, we do not know which of $g_0, \cdots, g_e$ are total, so we cannot afford to wait on non converging computations when computing P. However, as we have said before, we can have a different P at each requirement, and the choice of P can be non-uniform. Thus, P could just sum over all the total functions amongst $g_0, \cdots, g_e$.

The second consideration is that we might not be able to compute $r^0_e, r^1_e, \cdots$, if we have to recover r^n_e from the construction (which is performed with oracle A). We have to somehow figure out what r^n_e is, external to the construction. Observe that however, if we restrict ourselves to non-decreasing $g_0, g_1, \cdots$, it would be sufficient to compute an upperbound for r^n_e. We have to synchronize this with the construction: instead of picking r^n_e when we run out of room to change $Z\restriction_{r_e^{n-1}}$, we could instead pick r_e^n the moment enough of $g_k(x)$ converges and demonstrates that their sum exceeds $h^A(r_e^{n-1})$. To recover a bound for say, r_e^1 externally, we compute the first stage t such that *all of* $g_k(x)[t]$ has converged for $x \leq r_e^0$ and g_k total.

1.4. Notations used for the formal construction. The construction uses oracle A. At stage s we give an approximation $\{Z_s\}$ of Z, and at the end we argue that $Z \leq_T A'$. The construction involves finite injury of the requirements. $\mathcal{R}_1$ for instance, would be injured by $\mathcal{R}_0$ finitely often while $\mathcal{R}_0$ is waiting for hyperimmune permission from h^A. We intend to satisfy $\mathcal{R}_e$, by making $\mu(U_x^e \cap [Z\restriction_r])$ small for appropriate x, r. At stage s, we let $r_e[s]$ denote the follower used by $\mathcal{R}_e$. At stage s of the construction we define Z_s up till length s. We do this by specifying the strings $Z_s\restriction_{r_0[s]}, \cdots, Z_s\restriction_{r_k[s]}$ for an appropriate number k (such that $r_k[s] = s-1$). We adopt the convention of $r_{-1} = -1$ and $\alpha\restriction_{-1} = \alpha\restriction_0 = \langle\rangle$ for any string α. We let $S_e[s]$ denote all the pairs $\langle k, x\rangle$ for which $\mathcal{R}_e$ wants to make Z avoid U_x^k at stage s. The set $S_e[s]$ is specified by

$$S_e[s] = \{\langle k, x\rangle \mid k \leq e \ \wedge \ r_{k-1}[s] + 1 \leq x \leq r_e[s]\}.$$

Define the sequence of numbers

$$M_n = \sum_{j=n}^{2n} 2^{-(1+j)},$$

these will be used to approximate Z_s. Roughly speaking, the intuition is that $Z_s(n)$ will be chosen to be either 0 or 1 depending on which of $Z_s\restriction_n^\frown 0$ or $Z_s\restriction_n^\frown 1$ has a measure of $\leq M_n$ when restricted to a certain collection of U_x^e.

If P is an expression we append $[s]$ to P, to refer to the value of the expression as evaluated at stage s. When the context is clear we drop the stage number from the notation.

1.5. Formal construction of Z. At stage $s = 0$, we set $r_0 = 0$ and $r_e \uparrow$ for all $e > 0$, and do nothing else. Suppose $s > 0$. We define $Z_s\restriction_{r_k[s]}$ inductively; assume that has been defined for some k. There are two cases to consider for $\mathcal{R}_{k+1}$:

1. $r_{k+1}[s] \uparrow$: set $r_{k+1} = r_k[s] + 1$, end the definition of Z_s and go to the next stage.
2. $r_{k+1}[s] \downarrow$: check if $\sum_{\langle e,x\rangle \in S_{k+1}[s]} 2^{r_{k+1}} g_e(x)[s] \leq h^A(r_{k+1}[s])$. The sum is computed using converged values, and if $g_e(x)[s] \uparrow$ for any e, x we count it as 0. There are two possibilities:
   (a) *sum* $> h^A(r_{k+1})$: set $r_{k+1} = s$, and set $r_{k'} \uparrow$ for all $k' > k+1$. End the definition of Z_s and go to the next stage.
   (b) *sum* $\leq h^A(r_{k+1})$: pick the leftmost node $\sigma \supseteq Z_s\restriction_{r_k[s]}$ of length $|\sigma| = r_{k+1}[s]$, such that $\sum_{\langle e,x\rangle \in S_{k+1}[s]} \mu(U_x^e[s] \cap [\sigma]) \leq M_{r_{k+1}[s]}$. We will later verify that σ exists by a counting of measure. Let $Z_s\restriction_{r_{k+1}[s]} = \sigma$.

We say that $\mathcal{R}_{k+1}$ has *acted.* If 2(a) is taken, then we say that $\mathcal{R}_{k+1}$ has *failed the sum check.* This completes the description of Z_s.

1.6. Verification: Clearly, the value of the markers $r_0, r_1, \cdots$ are kept in increasing order. That is, at all stages s, if $r_k[s] \downarrow$, then $r_0[s] < r_1[s] < \cdots < r_k[s]$ are all defined. From now on when we talk about Z_s, we are referring to the fully constructed string at the end of stage s. It is also clear that the construction keeps $|Z_s| < s$ at each stage s.

LEMMA 1.2. *Whenever step 2(b) is taken, we can always define* $Z_s\restriction_{r_{k+1}[s]}$ *for the relevant* k *and* s.

PROOF. We drop s from notations, and proceed by induction on k. Let Υ be the collection of all possible candidates for $Z_s\restriction_{r_{k+1}}$, that is, $\Upsilon = \{\sigma : \sigma \supseteq$

$Z\upharpoonright_{r_k} \wedge |\sigma| = r_{k+1}\}$. Suppose that $k \geq 0$:

$$\sum_{\sigma\in\Upsilon}\sum_{\langle e,x\rangle\in S_{k+1}}\mu(U_x^e\cap[\sigma]) = \sum_{\langle e,x\rangle\in S_{k+1}}\sum_{\sigma\in\Upsilon}\mu(U_x^e\cap[\sigma])$$

$$\leq \sum_{\langle e,x\rangle\in S_{k+1}}\mu(U_x^e\cap[Z\upharpoonright_{r_k}]) \leq \sum_{\langle e,x\rangle\in S_k}\mu(U_x^e\cap[Z\upharpoonright_{r_k}]) + \sum_{x=r_k+1}^{r_{k+1}}\sum_{e\leq k+1}\mu(U_x^e)$$

$$\leq M_{r_k} + \sum_{x=r_k+1}^{r_{k+1}} 2^{-2x} \text{ (since } k\leq r_k) \leq M_{r_k} + \sum_{x=2r_k+1}^{r_k+r_{k+1}} 2^{-(1+x)}$$

$$= \sum_{x=r_{k+1}}^{2r_{k+1}} 2^{-(1+x)}2^{r_{k+1}-r_k} \text{ (adjusting the index } x) = M_{r_{k+1}}|\Upsilon|.$$

Hence, there must be some σ in Υ which passes the measure check in 2(b) for $Z\upharpoonright_{r_{k+1}}$. A similar, but simpler counting argument follows for the base case $k=-1$, using the fact that the search now takes place above $Z\upharpoonright_{r_k} = \langle\rangle$. ⊣

LEMMA 1.3. *For each e, the follower $r_e[s]$ eventually settles.*

PROOF. We proceed by induction on e. Note that once $x_{e'}$ has settled for every $e' < e$, then $\mathcal{R}_e$ will get to act at every stage after that. Hence there is a stage s_0 such that

(i) $r_{e'}$ has settled for all $e' < e$, and
(ii) r_e receives a new value at stage s_0.

Note also that $\mathcal{R}_e$ will get a chance to act at every stage $t > s_0$, and the only reason why r_e receives a new value after stage s_0, must be because $\mathcal{R}_e$ fails the sum check. Suppose for a contradiction, that $\mathcal{R}_e$ fails the sum check infinitely often after s_0.

Let $q(n-1)$ be the stage where $\mathcal{R}_e$ fails the sum check for the n^{th} time after s_0. In other words, $q(0), q(1), \cdots$ are precisely the different values assigned to r_e after s_0. Let $\mathcal{C}$ be the collection of all $k \leq e$ such that g_k is total, and d be a stage where $g_k(x)[d]$ has converged for all $k \leq e$, $k \notin \mathcal{C}$ and $x \in dom(g_k)$. We now define an appropriate computable function to contradict the hyperimmunity of A. Define the total computable function p by: $p(0) = 1 + \max\{s_0, d,$ the least stage t where $g_k(r_e[s_0])[t]\downarrow$ for all $k \in \mathcal{C}\}$. Inductively define $p(n+1) = 1+$ the least t where $g_k(p(n))[t]\downarrow$ for all $k \in \mathcal{C}$. Let $P(n) = \sum_{k\leq e}\sum_{x\leq p(n)} 2^{p(n)} g_k(x)[p(n+1)]$, which is the required computable function.

One can show by a simple induction, that $p(n) \geq q(n)$ for every n, using the fact that $\mathcal{R}_e$ is given a chance to act at every stage after s_0, as well as the restrictions we had placed on the functions $\{g_k\}$. Let N be such that $P(N) \leq h^A(N)$. At stage $q(N+1)$ we have $\mathcal{R}_e$ failing the sum check, so that $h^A(N) < h^A(q(N)) < \sum_{\langle k,x\rangle\in S_e} 2^{q(N)} g_k(x)$, where everything in the last sum is evaluated at stage $q(N+1)$. That last sum is clearly $< P(N) \leq h^A(N)$, giving a contradiction. ⊣

Let $\hat{r}_e$ denote the final value of the follower r_e. Let $Z = \lim_s Z_s$. We now show that $Z \leq_T A'$, and is not Demuth random relative to A. For each e and s, $Z_{s+1+\hat{r}_e}\upharpoonright_{\hat{r}_e}$ is defined, by lemma 1.2, and the fact that any value assiged to r_e at stage t has to be t itself.

LEMMA 1.4. *For each e, $|t \geq 1+\hat{r}_e : Z_t\upharpoonright_{\hat{r}_e} \neq Z_{t+1}\upharpoonright_{\hat{r}_e}| \leq h^A(\hat{r}_e)$.*

PROOF. Suppose that $Z_{t_1}\upharpoonright_{\hat{r}_e} \neq Z_{t_2}\upharpoonright_{\hat{r}_e}$ for some $1+\hat{r}_e \leq t_1 < t_2$. We must have $r_{e'}$ already settled at stage t_1, for all $e' \leq e$. Suppose that $Z_{t_2}\upharpoonright_{\hat{r}_e}$ is to the left of $Z_{t_1}\upharpoonright_{\hat{r}_e}$, then let e' be the least such that $Z_{t_2}\upharpoonright_{\hat{r}_{e'}}$ is to the left of $Z_{t_1}\upharpoonright_{\hat{r}_{e'}}$. The fact that $\mathcal{R}_{e'}$ didn't pick $Z_{t_2}\upharpoonright_{\hat{r}_{e'}}$ at stage t_1, shows that we must have a change of index for U_b^a between t_1 and t_2, for some $\langle a,b\rangle \in S_{e'} \subseteq S_e$. Hence, the total number of mind changes is at most $2^{\hat{r}_e} \sum_{\langle a,b\rangle \in S_e} g_a(b)$, where divergent values count as 0. $2^{\hat{r}_e}$ represents the number of times we can change our mind from left to right consecutively without moving back to the left, while $\sum_{\langle a,b\rangle \in S_e} g_a(b)$ represents the number of times we can move from right to left. Since $\mathcal{R}_e$ never fails a sum check after $\hat{r}_e$ is picked, it follows that the number of mind changes has to be bounded by $h^A(\hat{r}_e)$. $\dashv$

By asking appropriate 1-quantifier questions of A', we can recover $Z = \lim_s Z_s$, because of lemma 1.4, and hence Z is well-defined. To see that Z is not Demuth random in A, define the Demuth test $\{V_x\}$ by the following: run the construction and enumerate $[Z_s\restriction_x]$ into V_x when it is first defined. Subsequently each time we get a new $Z_t\restriction_x$, we change the index for V_x, and enumerate the new $[Z_t\restriction_x]$ in. If we ever need to change the index $> h^A(x)$ times, we stop and do nothing. By lemma 1.4, Z will be captured by $V_{\hat{r}_e}$ for every e.

Lastly, we need to see that Z passes all $\{U_x^e\}$. Suppose for a contradiction, that $Z \in U_x^e$ for some e and $x > \hat{r}_e$. Let δ be such that $Z \in [\delta] \in U_x^e$, and let $e' \geq e$ such that $\hat{r}_{e'} > |\delta|$. Go to a stage in the construction where δ appears in U_x^e and never leaves, and $r_{e'} = \hat{r}_{e'}$ has settled. At every stage t after that, observe that $\langle e, x\rangle \in S_{e'}$, and that $\mathcal{R}_{e'}$ will get to act, in which it will discover that $\mu(U_x^e \cap [Z\restriction_{\hat{r}_{e'}}]) = 2^{-\hat{r}_{e'}} > M_{\hat{r}_{e'}}$. Thus, $\mathcal{R}_{e'}$ never pick $Z\restriction_{\hat{r}_{e'}}$ as an initial segment for Z_t, giving us a contradiction. $\dashv$

SCHOOL OF MATHEMATICS, STATISTICS AND COMPUTER SCIENCE
VICTORIA UNIVERSITY OF WELLINGTON
PO BOX 600, WELLINGTON, NEW ZEALAND