Math. Struct. in Comp. Science (2009), *vol.* 19, *pp.* 207–215. © 2009 Cambridge University Press
doi:10.1017/S0960129508007275 Printed in the United Kingdom

The method of the Yu–Ding Theorem and its application

YUN FAN[†]

Department of Mathematics, Nanjing University, China
and
Department of Mathematics, Southeast University, China
Email: fanyun1234@hotmail.com

Received 20 February 2008

We begin by reviewing the major results dealing with the structure of the *cl*-degrees, and then focus on the method of the Yu–Ding Theorem, which is an important result in this area. By strengthening the Yu–Ding procedure, we construct a *cl*-cuppable *cl*-degree of c.e. sets.

1. The structure of the *cl*-degrees

We assume some background on computability theory and some knowledge of standard conventions, most of which can be found in Soare (1987); a knowledge of algorithmic randomness will also be helpful.

In the study of relative computations it is natural to consider computations of a real α from another real β in which to compute $\alpha \upharpoonright n$ we are only allowed to use the information of $\alpha \upharpoonright (n + c)$ for some fixed constant c.

The study of oracle computations of this kind and of the reducibility they induce on 2^ω was initiated by Downey, Hirschfeldt and LaForte, who gave it the name *strong weak truth table reducibility* or *sw*-reducibility for short (Downey *et al.* 2004). The motivation for introducing *sw*-reducibility was that it might serve as a measure of relative randomness. However, when taking a computational point of view, instead of *sw*-reducibility, Downey and Hirschfeldt (Downey and Hirschfeldt, to appear) adopt the *computably Lipschitz* reducibility or *cl*-reducibility, which is due to Barmpalias and Lewis (Barmpalias and Lewis 2006) – we will also adopt the terminology of *computably Lipschitz reducibility* here.

Definition 1.1 (Downey and Hirschfeldt, to appear). $\alpha \leqslant_{cl} \beta$ if there is a Turing functional Γ and a constant c such that $\alpha = \Gamma^\beta$ and the use of Γ on any argument n is bounded by $n + c$. The Turing functionals that have their use restricted in such a way are called cl functionals.

cl-reducibility is quite natural since it occurs in many proofs in classical computability theory: it arises when we apply simple permitting for the construction of a set 'below' a given one. *cl*-reducibility (in particular, the case where $c = 0$) has been used in recent

[†] This work is supported by DFG (446 CHV 113/240/0-1), NSFC (10420130638) and NSFC(10701041).

https://doi.org/10.1017/S0960129508007275 Published online by Cambridge University Press

work by Soare, Nabutovsky and Weinberger on applications of computability theory to differential geometry (Soare, to appear).

The main justification for using $\leqslant_{cl}$ as a measure of relative randomness is the following proposition.

Proposition 1.1 (Downey *et al.* 2004). If $\alpha \leqslant_{cl} \beta$ are c.e. reals, then for all n, and $E = K$ or $E = C$, we have $E(\alpha \upharpoonright n) \leqslant E(\beta \upharpoonright n) + O(1)$.

On the other hand, we also have the following theorem.

Theorem 1.1 (Downey and Hirschfeldt, to appear). Let α and β be c.e. reals such that $liminf_n K(\beta \upharpoonright n) - K(\alpha \upharpoonright n) = \infty$. Then $\alpha \leqslant_{cl} \beta$.

Having defined *cl*-reducibility, we can go on to consider the structure of the induced degrees. The *cl*-degrees are the equivalence classes under *cl*-reducibility and the ordering that we consider on these degrees is that induced by *cl*-reducibility on the reals. The *cl*-degrees possess the nice property of having either *only random* reals or *non-random* reals. But we will see that the *cl*-reducibility also has a number of very undesirable features.

Downey, Hischfeldt and Laforte showed that the *cl*-degrees coincide with the *Solovay* degrees on c.e. sets (Downey *et al.* 2004). So the structure of the *cl*-degrees of c.e. reals is not a lower semi-lattice, which can be derived from the structure of the *Solovay*-degrees of c.e. sets. However, it is also not an upper semi-lattice, in that there is no join operation. This property was first proved directly by Downey, LaForte and Hirschfeldt (Downey *et al.* 2004), but it also follows from the next theorem: the Yu–Ding Theorem.

Theorem 1.2 (Yu and Ding 2004). There is no *cl*-complete c.e. real.

In fact, Yu and Ding proved the following stronger result.

Corollary 1.1. There are two c.e. reals α and β such that there is no c.e. real γ with $\alpha \leqslant_{cl} \gamma$ and $\beta \leqslant_{cl} \gamma$.

Although the *cl*-degrees present such difficulties, the structure of the *cl*-degrees is very interesting from a wider perspective.

Yu and Ding (Yu and Ding 2004) asked whether there are maximal *cl*-degrees of c.e. reals, and conjectured that there are such, and that they are exactly the ones that contain random reals. Using a doubly non-uniform argument, Barmpalias and Lewis gave a negative answer for the *cl*-degrees of c.e. sets (Barmpalias 2005), that is, there are no *cl*-maximal degrees of c.e. sets. Barmpalias and Lewis construct a c.e. real so that no c.e. random real can *cl*-compute it (Barmpalias and Lewis 2006). To clarify the situation regarding the *cl*-degrees of random c.e. reals, Frank Stephan proved that there are two random c.e. reals that cannot be *cl*-equivalent. Very little else is known about the *cl*-degrees of c.e. reals. Whether the *cl*-degree of Ω is a maximal *cl*-degree of c.e. reals is still an open problem. Although the ibT degrees of c.e. sets are not dense (Barmpalias and Lewis 2006), we can still ask whether the *cl*-degrees of c.e. reals are dense.

When considering *cl*-reducibility as a measure of randomness, it is meaningful to ask whether there exists some property of *cl*-degrees that can characterise randomness. Lewis and Barmpalias (Lewis and Barmpalias 2006) attempted to do this by defining

https://doi.org/10.1017/S0960129508007275 Published online by Cambridge University Press

quasi-maximal cl-degrees: α is *quasi-maximal* if $\alpha \leqslant_{cl} \beta$ implies that $\beta \leqslant_T \alpha$. They went on to prove that every random real satisfies this *quasi-maximality* property (Lewis and Barmpalias 2007). As a corollary of this result, they get the equivalent of the Yu–Ding theorem for Δ^0_2 reals. However, the bad news is that there also exists a real of quasi-maximal *cl*-degree that is not random. Having proved that quasi-maximality does not characterise the random reals, it is natural to ask whether maximality might provide such characterisation. Such hopes, however, are in vain since Lewis and Barmpalias (2007) showed that no real is of maximal *cl*-degree. These are the main results concerning the global structure of the *cl*-degrees of reals.

2. The method of the Yu–Ding Theorem

In this section we review the method of proving the Yu–Ding Theorem due to Barmpalias and Lewis (Lewis and Barmpalias 2006). The procedure they use is identical to the Yu–Ding procedure, but the verification is a much smoother induction and the proof significantly cleaner. We begin with a description of the position (or digit) of its binary expansion.

Definition 2.1. The positions on the right of the decimal point in the binary expansion are numbered $1, 2, 3, \ldots$ from left to right and the positions on the left of the decimal point are numbered $0, -1, -2, \ldots$ We write 'the positions at b or higher (less) than' as '$\leqslant (\geqslant) b$', and 'the positions between h and l' as an interval $[h, l]$ for $h < l$.

Proof of Theorem 1.2. It suffices to construct two c.e. reals α, β in stages such that for all *cl* functionals Φ, Ψ and c.e. reals γ:

$$R_{\Phi,\Psi,\gamma} \ : \ \alpha \neq \Phi^{\gamma} \vee \beta \neq \Psi^{\gamma} .$$

Consider the following *cl*-game between α and γ. During the stages of the game, α and γ increase, and they have initial value 0. If α changes and b is the leftmost position where an α-digit change occurred, then γ has to increase in such a way that some γ-digit at a position $\leqslant b$ changes. In this game, γ is not less than β. This game describes the *cl*-reduction between α and γ with the identity use.

If γ is used to code two reals α β (during the stages all of them increase and at each stage only one of α and β can change), we get a similar game. We give the Yu–Ding procedure as follows, which shows the idea that if there were a largest *cl*-degree of c.e. real γ, we could select two reals α and β and change them alternately to drive γ to be very large. In the Yu–Ding procedure, we say that γ follows the *least effort strategy* if at each stage γ increases by the least amount that can rectify the functionals holding its computations of α, β.

Lemma 2.1. Let $n > 0$, $h \in \mathbb{Z}$. We are going to define the Yu–Ding procedure amongst α, β, γ with attack interval $(h, h+n]$. We assume that α, β, γ have initial value 0. Repeat the following instructions (at expansionary stages) until $\beta(i) = \gamma(i) = 1$ for all $i \in (h, h+n]$:

1 s odd: Let $\alpha = \alpha + 2^{-h-n}$.
  s even: Let $\beta = \beta + 2^{-h-n}$.

https://doi.org/10.1017/S0960129508007275 Published online by Cambridge University Press

2 Suppose that b is the highest (that is, leftmost) position where a digit change occurs in α or β at the corresponding stage, let $\gamma = \gamma + 2^{-b}$.

Then γ ends up with $n2^{-h}$.

The following observation is, in some sense, at the heart of the Yu–Ding strategy.

Lemma 2.2. In the game where γ has to follow instructions of the type 'change a digit at position $\leqslant b$ (for example, a *cl*-game between γ and α, β), the least effort strategy is the best strategy for γ. In other words, if a different strategy produces γ', then at each stage s of the game, $\gamma_s \leqslant \gamma'_s$.

Lemma 2.3 (Passing through lemma). Suppose that in some game (for example, one like the above) γ has to follow instructions of the type 'change a digit at a position $\leqslant b$'. Although $\gamma_0 = 0$, some γ' plays the same game while starting with $\gamma'_0 = \sigma$ for a finite binary expansion σ. If the strategies of γ, γ' are the same (that is, the 'least effort' strategy described above) and the sequence of instructions only ever demand a change at positions $> |\sigma|$, then at every stage s,

$$\gamma'_s = \gamma_s + \sigma\,.$$

Assume an effective list of all R-requirements and successively assign attack intervals to them. If the attack interval for R_i is $(h, n]$, we take the Yu–Ding strategy with attack interval $(h, n]$, which means the enumerations of α, β as in the Yu–Ding procedure at the stage s, where the length function for R_i $l(\Phi, \Psi, \gamma, s) = max\{x | \forall(y < x)\alpha_s(y) = \Phi^{\gamma}(y)[s] \wedge \beta_s(y) = \Psi^{\gamma}(y)[s]\}$ is larger than n.

Corollary 2.1. If α, β follow the Yu–Ding strategy in attacking R_i with attack interval $(h, h + 2^{h+c}]$ (where the functionals have use bounded by $x + c$), R_i is satisfied or $\gamma \geqslant 1$.

When attack interval $(h, n]$ for R_i is given, we define the one for R_{i+1} to be $(n, n + 2^{n+c'}]$, where $x + c'$ is the use function corresponding to R_{i+1}. Apply the Yu–Ding strategy for each of the requirements on the relevant intervals in a global construction. This causes no problems with the strategies, and the satisfaction of all the requirements follows from Corollary 2.1. □

3. Some results

The proof of the Yu–Ding Theorem gives the stronger result of Corollary 1.1. Imitating the Yu–Ding Theorem's method, it is easy to get the equivalent of Corollary 1.1 for c.e. sets, which can be stated as follows.

Proposition 3.1. The structure of the *cl*-degrees of c.e. sets is not an upper semi-lattice.

We recall the following theorem given in Barmpalias and Lewis (2006). The structure of its proof is similar to the Yu–Ding Theorem's proof, but Barmpalias and Lewis introduce their different procedure for a different purpose.

Theorem 3.1 (Barmpalias and Lewis 2006). There is a c.e. real that cannot be *cl*-computed by any c.e. random real.

https://doi.org/10.1017/S0960129508007275 Published online by Cambridge University Press

Proof. It suffices to construct a c.e. real β and a Martin-Löf test $\{E_j\}_{j\in\omega}$ satisfying $\mu(E_j) < 2^{-j}$ such that for any c.e. real γ

$$R_{\gamma,j} \ : \ \beta = \Phi^{\gamma} \Rightarrow \gamma \in E_j^{\gamma}.$$

Considering a single requirement, γ has to not only *cl*-code β's change, but also follow instructions of the type 'change a digit below n', which come from E_j^{γ}, and is called an E_j^{γ}-test instruction.

For any $j, n \in \mathbb{N}$, they effectively construct the procedure to change β and give out E_j^{γ}-test instructions such that γ, which follows *the least effort strategy*, could be larger than n and $\mu(E_j^{\gamma})$ is less than 2^{-j}. Such a procedure is uniformly computed by j, n.

Assume an effective list of all R-requirements and successively assign attack intervals to them. If the attack interval for R_i is $(h, l]$, they take the strategy such that the enumerations of α, β that follows the procedure for the case $j, n = 2^{h+c}$ mentioned above at the stage s when the length function of R_i is larger than n and $\gamma \notin E_{j,s}^{\gamma}$. Apply the relevent strategy for each of the requirements on the relevant intervals in a global construction. This causes no problems with the strategies and the satisfaction of all the requirements follows. More details can be found in Barmpalias and Lewis (2006). □

4. The $P_{n,k}$ procedure

In this section we will strengthen the Yu–Ding procedure.

Definition 4.1. Assume we are given $n \in N^+$, $k \in \mathbb{N}$. Suppose that α, β and γ have initial value 0, and that $P_{n,k}$ is a procedure to change α, β in stages to drive $\gamma = n$, and during the stages we keep $\alpha \upharpoonright k = 0$. Let $l_{n,k}$ be the largest number to have changed in α, β in $P_{n,k}$. We say $(0, l_{n,k}]$ is the attack interval of $P_{n,k}$.

The Yu–Ding procedure with attack interval $(0, n]$ is a $P_{n,0}$ procedure. Note that α and β in the Yu–Ding procedure keep increasing. It is natural to ask whether there is a $P_{n,k}$ procedure satisfying the condition that during the stages β keeps increasing but α-digit change occurs at most once.

From now on, the $P_{n,k}$ procedure points to the procedure that only allows α-digit change at most once. We design the $P_{n,k}$-procedure inductively on n, k. Before going on to the general case, we will give some examples.

Example 4.1.

1 $P_{1,k}$ can be defined as follows: change $\beta(i) = 1$ for $i \in (0, k]$ from left to right, and then change $\alpha(k) = 1$. It is easy to check that $\gamma = 1$ and $\alpha \upharpoonright k = 0$.

2 To help explain the idea of defining a $P_{n+1,0}$ procedure based on the P_n procedures, we define a $P_{10,0}$ procedure based on the P_1 procedures as an example.

α	000	000	001	0.001	0.101	0.101	00.111
β	010	011	011	0.100	0.100	0.110	00.110
γ	010	011	100	1.000	1.100	1.110	10.000

https://doi.org/10.1017/S0960129508007275 Published online by Cambridge University Press

The action of 'let $\beta = 0.1$ or let $\alpha(1) = 1$' drives γ by 0.1 in either case. Since $\gamma = 10$ can be written as $0.1 + 0.1 + 0.1 + 0.1$ and 'let $\beta = 0.1$' clears all β-digits > 1, it suffices to drive γ by 0.1 both before 'let $\beta = 0.1$' and after it. The $P_{1,1}$ procedure with attack interval $[2, 3]$ could provide 0.1 and at the same time always keep $\alpha(2) = 0$. The fact that $\alpha(2) = 0$ and β-digit > 1 is empty when 'let $\beta = 0.1$' provides enough room to perform the $P_{1,0}$ procedure with digit 2, which can also drive γ by 0.1.
The key points are that

(i) β-digit > 2 can often be used twice; and

(ii) the first sub-procedure P_{1,k_0} applied before 'let $\beta = 0.1$' provides enough room for the next $P_{1,0}$ sub-procedure. Such a $k_0 = l_{1,0}$ is what we want.

3 To help explain the idea of defining a $P_{n+1,k+1}$ procedure based on the $P_{n+1,k}$ procedure and the P_n procedures, we will define $P_{10,1}$ based on the $P_{10,0}$ procedure and the P_1-procedure as an example.

α	00000		00001	0.00001	0.00001		00.01111
β	01000	$\cdots$	01111	0.10000	0.11000	$\cdots$	00.11100
γ	01000	$P_{1,3}$	10000	1.00000	1.01000	$P_{10,0}$	10.00000

We should obey the principle of keeping $\alpha(1) = 0$ during the stages. Note that we can separate 10 into $1 + 0.1 + 0.1$, and 'let $\beta = 0.1$' can play the same role as in defining $P_{10,0}$. It suffices to drive γ by 0.1 before 'let $\beta = 0.1$' and by 1 after it. If we perform the $P_{1,l_{10,0}}$ procedure with digit $\geqslant 2$ before 'let $\beta = 0.1$', we get enough room in α and β to perform the $P_{10,0}$ procedure with digit $\geqslant 2$ after 'let $\beta = 0.1$'. Notice that $\alpha \upharpoonright 2 = 0$.

Lemma 4.1. Assume we are given $k \in N, n > 1 \in \mathbb{N}^+$ and suppose that α, β have initial value 0.

1 Assuming that $P_{n,k}$ have been defined for $k \in \mathbb{N}$, we define a $P_{n+1,0}$ procedure as follows:

— Perform $P_{n,l_{n,0}}$ with attack interval $[2, l_{n,l_{n,0}} + 1]$.

— Let $\alpha(1) = 1$.

— Let $\beta = 0.1$.

— Perform $P_{n,0}$ in attack interval $[2, l_{n,0} + 1]$.

2 Assuming that $P_{n,k}$ for $k \in N$ and $P_{n+1,k'}$ have been defined, we define a $P_{n+1,k'+1}$ procedure as follows:

— Perform $P_{n,l_{n+1,k'}}$ with attack interval $[2, l_{n,l_{n+1,k'}} + 1]$.

— Let $\beta = 0.1$.

— Perform $P_{n,k'}$ with attack interval $[2, l_{n,k'} + 1]$.

Claim: There exists a $P_{n,k}$ procedure with attack interval $(0, l_{n,k}]$ for any $k \in N, n \in \mathbb{N}^+$.

https://doi.org/10.1017/S0960129508007275 Published online by Cambridge University Press

Proof. Example 3.2(1) shows the $P_{1,k}$ procedure for any $k \in N$, which is the foundation for cases $n > 1$.

1 The construction of $P_{n+1,1}$ shows the idea of how to perform the induction on n. During the induction steps:

— The action of 'perform a $P_{n,l_{n,0}}$ procedure' could drive γ by $n/2$.

— 'let $\alpha(1) = 1$' and '$\beta = 0.1$' both drive γ by 0.1.

— The action of 'perform a $P_{n,0}$ procedure' could drive γ by $n/2$.

The room for doing $P_{n,0}$ with $[2, l_{n,0} + 1]$ comes from the fact that:

(i) after performing $P_{n,l_{n,0}}$, it still has $\alpha \upharpoonright (l_{n,0} + 1) = 0$; and

(ii) 'let $\beta = 0.1$' clears all β-digit $\geqslant 2$ used in the $P_{n,l_{n,0}}$ procedure.

Since the whole process obeys the *Passing through lemma*, γ ends up with $n/2 + 0.1 + 0.1 + n/2 = n + 1$.

2 The construction of $P_{n+1,k'+1}$ shows the idea of how to perform the induction on k. During the induction steps:

— The action of 'perform a $P_{n,l_{n+1,k'}}$ procedure' could drive γ by $n/2$.

— 'let $\beta = 0.1$' drives γ by 0.1.

— The action of 'perform a $P_{n,k'}$ procedure' could drive γ by $(n + 1)/2$.

The room for doing $P_{n,k'}$ with $[2, l_{n,k'} + 1]$ comes from the fact that:

(i) after performing $P_{n,l_{n,k'}}$, it still has $\alpha \upharpoonright (l_{n,k'} + 1) = 0$; and

(ii) 'let $\beta = 0.1$' clears all β-digit $\geqslant 2$ used in the $P_{n,l_{n,k'}}$ procedure.

For the case of $\alpha(1) = 0$, we are assured that $\alpha \upharpoonright (k' + 1) = 0$ during each step.
Since the whole process obeys the *Passing through lemma*, γ ends up with $n/2 + 0.1 + (n + 1)/2 = n + 1$. □

5. The *cl*-cuppable *cl*-degrees

In this section we introduce the definition of *cl-cuppable* for the *cl*-degrees. For the case of the no join operator of *cl*-degrees of c.e. reals, it is different from the notion of *cuppable* under *Turing*-reducibility.

Definition 5.1. We say the *cl*-degree **a** is *cl*-cuppable if there exists a *cl*-degree **b** of c.e. reals such that no *cl*-degrees of c.e. reals can *cl*-compute both of them.

Corollary 1.1 can now be restated.

Theorem 5.1. There exist *cl*-cuppable *cl*-degrees of c.e. reals.

It is natural to ask whether there is a *cl*-cuppable *cl*-degree of c.e. sets. By mimicking the method of the Yu–Ding Theorem and taking the $P_{2^n,0}$ strategy for α and β, we can give a positive answer.

Theorem 5.2. There is a *cl*-cuppable *cl*-degree of c.e. sets.

https://doi.org/10.1017/S0960129508007275 Published online by Cambridge University Press

Proof. It suffices to construct a real $\alpha = 0.\chi_A$ for a c.e. set A and a c.e. real β in stages such that for all cl functionals Φ, Ψ and c.e. reals γ,

$$R_{\Phi,\Psi,\gamma} : \alpha \neq \Phi^{\gamma} \vee \beta \neq \Psi^{\gamma}.$$

Lemma 2.2 is still the heart of the proof. Assume an effective list of all R-requirements and successively assign attack intervals to them. We choose the attack interval for R_i to be $(h, h + l_{2^{h+c},0}]$, and take the $P_{2^{h+c},0}$ strategy with attack interval $(h, h + l_{2^{h+c},0}]$, which means the enumerations of α, β, as in the $P_{2^{h+c},0}$ procedure in Section 4 at the stage s when its length function is larger than $h + l_{2^{h+c},0}$. For the case of the definition of $P_{2^{h+c},0}$, we can deduce the following lemma.

Lemma 5.1. If α, β follow the $P_{2^{h+c},0}$ strategy in attacking R_i with attack interval $(h, h + l_{2^{h+c},0}]$ (where the functionals have use bounded by $x + c$), then R_i is satisfied or $\gamma \geqslant 1$.

If the attack interval $(h, n]$ for R_i is given, we define the attack interval for R_{i+1} to be $(n, n + l_{2^{n+c'},0}]$, where $x + c'$ is the use function corresponding to R_{i+1}. Now we apply the corresponding P-procedure strategy for each of the requirements on the relevant intervals in a global construction. There is again no problem caused for the strategies, and the satisfaction of all the requirements follows from Lemma 5.1. □

Corollary 5.1 (Theorem 3.1). There is a (non-random) c.e. real that cannot be cl-computed by any c.e. random real.

Proof. Every random c.e. real is cl-above all c.e. sets by Theorem 1.1, which assures β in Theorem 5.2 is non-random. Furthermore, no c.e. random real can cl-compute β by the same reason as above. □

6. Further research

Two further questions we are interested in are:

1 Can we design the P_n procedure just by controlling the change of β for a given real α?
2 Is there a non-cl-cuppable degree of non-computable c.e. reals?

References

Barmpalias, G. (2005) Computably enumerable sets in the Solovay and the Strong Weak Truth Table Degrees. In: Proceedings of the CiE conference 2005 conference in Amsterdam. *Springer-Verlag Lecture Notes in Computer Science* **3526** 8–17.

Barmpalias, G. and Lewis, A. (2006) A c.e. real that cannot be SW-computed by any Ω number. *Notre Dame J. Formal Logic* **47** (2) 197–209.

Barmpalias, G. and Lewis, A. (2006) The ibT degrees of c.e. sets are not dense. *Annals of Pure and Applied Logic Volume* **16** (1-2).

Downey, R. G. and Hirschfeldt, D. (to appear) Randomness and reducibility. *Springer-Verlag Monographs in Computer Science* (in preparation).

Downey, R. G., Hirschfeldt, D. and LaForte, G. (2004) Randomness and reducibility (extended abstract). In: Sgall, J., Pultr, A. and Kolman, P. (eds.) Mathematical Foundations of Computer

https://doi.org/10.1017/S0960129508007275 Published online by Cambridge University Press

Science 2001. *Springer-Verlag Lecture Notes in Computer Science* **2136** 316–327. (Final version in *Journal of Computing and System Sciences* (2004) **68** 96–114.)

Lewis, A. and Barmpalias, G. (2006) Random reals and Lipschitz continunity. *Mathematical Structures in Computer Science* **16** (5) 737–749.

Lewis, A. and Barmpalias, G. (2007) Randomness and the linear degrees of computability. *Annals of Pure and Applied Logic* **145** (3) 252–257.

Soare, R. I. (1987) *Recursively Enumerable Sets and Degrees*, Springer-Verlag.

Soare, R. I. (to appear) Computability Thoery and Differential Geometry (to appear in *The Bulletin of Symbolic Logic*).

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

https://doi.org/10.1017/S0960129508007275 Published online by Cambridge University Press