

## 1. BIOGRAPHICAL BACKGROUND

My research interests are driven by much the same considerations that inspired me to work in computability theory in the first place: the challenge of building a solution to an interesting problem or pinpointing the reason it can't be done. Thus, just like computability theory itself, my interest is not in any particular class of objects, but in a certain kind of construction oriented approach to mathematical problems. Aesthetic sensibilities, mathematical or otherwise, are nearly impossible to capture in writing so the best I can do is observe that I'm often attracted to the interplay between computational properties and some non-computational notion (set containment, function domination, paths through trees, etc..). So when it came time to start my serious graduate research, I was attracted to a problem posed by Groszek and Slaman [7] about the relation between computational properties of perfect trees and those of their paths. In particular the problem asked whether there was a non-computable perfect tree  $T \subset 2^{<\omega}$  so that every path  $f \in 2^\omega$  through  $T$  computable by  $T$  was actually computable.

Of course the downside to attacking problems with large intricate constructions is they provide few partial results. A lesson that I learned when, after getting repeatedly stuck trying to construct such a tree, Professor Lewis gave a clever proof showing such an object couldn't be built. But while the components of a failed construction aren't very interesting on their own in a roundabout fashion my work did lead to some interesting mathematics. Issues I raised in a presentation of my partial results suggested a question to Slaman and Groszek that lead them to define the notion of a modulus (of computation). This notion in turn inspired my current research and freed from my prior problem by Lewis's result I was able to extend Groszek and Slaman's results and explore interesting new directions.

## 2. BACKGROUND

**2.1. Computation and Rate of Growth.** The notion of relative (Turing) computability provides a way to compare the 'uncomputability' of two sets (of integers). If there is an algorithm to decide membership in  $A$  by asking questions about membership in  $B$  we say that  $A$  is computable in  $B$  (written  $A \leq_T B$ ). The equivalence classes of  $T$  are called Turing degrees and the equivalence class of the empty set (denoted  $\mathbb{0}$ ) is the class of computable sets. This structure is supplemented with a jump operator taking a set  $A$  to the set  $A'$ , a maximally complex set definable via existential quantification over  $A$ . The jump operator can be thought of as a natural 'step up' in computational power and the number of applications of the jump operator to  $A$  required to compute  $B$  can be thought of as a measure of the 'complexity' of  $B$  relative to  $A$ . For any computable ordinal (an ordinal whose ordertype can be computably represented)  $\alpha$  there is a natural definition of what it means to iterate the jump operation  $\alpha$  many times starting from  $A$  which we write  $A^{(\alpha)}$ .

One traditional area of interest in computability theory is the relation between the rate of growth of a (total) function (identified with its graph) and its Turing degree [3, 10, 11, 13]. Originally motivated by Post's program [14] these explorations have long since become a topic of interest in their own right, but most of the work still retains the same basic form: classify the degrees of functions which are 'large' relative to some fixed class of functions. Jockusch and Soare, however, introduced a fascinating alternative approach in [9] by introducing the notion of a recursively encodable set. Formally, Jockusch and Soare defined a set  $A$  to be recursively encodable if for every infinite  $X$  there is a  $Y \subset X$  with  $Y \geq_T A$ . Intuitively, a recursively encodable set is a set that is computable from any sufficiently fast growing function making them an interesting alternative way to approach the connection between rate of growth and Turing degree. However, it wasn't long before Solovay [12] identified the recursively encodable sets with the  $\Delta_1^1$  sets (sets with a particularly simple second order syntactic definition) seemingly exhausting this research direction. However, Groszek and Slaman's notion of a modulus (of computation) refines the notion of a recursively encodable set opening up a whole new set of questions.

## 2.2. Self-Moduli.

**Notation 2.1.**  $f \succ g \iff (\forall x)(f(x) \geq g(x))$

**Definition 2.2** (Groszek-Slaman [7]). A function  $f$  is a (self-)modulus for  $X$  if ( $f \equiv_T X$  and)  $(\forall g \in \omega^\omega)(g \succ f \implies g \geq_T X)$

**Definition 2.3** (Groszek-Slaman [7]). A function  $f$  is a uniform (self-)modulus for  $X$  if ( $f \equiv_T X$  and) there is a recursive functional  $\Phi$  so that  $g \succ f \implies \Phi(g) = X$ .

We will occasionally abuse notation and say a degree has a (self-)modulus if there is a set of that degree with a (self-)modulus.

Since every degree in the REA hierarchy (including degrees of the form  $0^{(\alpha)}$ ) has a uniform self-modulus, we can identify those sets with a modulus with the  $\Delta_1^1$  sets and thus the recursively encodable sets. Indeed a modulus (or uniform modulus) for a set  $X$  can be viewed as a witness to the recursive encodability of  $X$  and thus its membership in  $\Delta_1^1$ . Thus it is interesting to compare the natural witnesses of a set's membership in  $\Delta_1^1$  defined 'from above' (a modulus/uniform modulus) with those defined 'from below' (Kleene H-sets/REA sets computing the set in question).

**Theorem 2.4** (Groszek-Slaman [7]). *If  $X$  has a modulus then  $X$  has a uniform modulus and is  $\Delta_1^1$*

Those functions which are their own moduli are of particular interest for several reasons. In [7] Slaman points out that these functions are useful for running permitting arguments as well as providing an example of a set that isn't 2-random relative to any continuous measure. Another example of the sort of constructions that self-moduli support is the following theorem.

**Theorem 2.5** (Groszek-Slaman [7]). *If  $f$  is a self-modulus then either  $f$  is  $\Delta_2^0$  or  $f$  can compute a 1-generic.*

Moreover, this isn't satisfied in a trivial fashion (say by every self-modulus being  $\Delta_2^0$ ).

**Theorem 2.6** (Groszek-Slaman [7]). *There is a non-computable (uniform) self-modulus  $f$  such that  $X \in \Delta_2^0 \implies X \not\leq_T f$ .*

### 3. RESULTS

**3.1. Extending To The Transfinite.** The jumping off point for my research was the question of whether Theorem 2.6 could be extended past 2. Not only was it possible to replace  $\Delta_2^0$  with  $\Delta_3^0$  it was actually possible to go up the entire hyperarithmetical hierarchy.

**Theorem 3.1.** *For every computable ordinal  $\alpha$  there is a non-computable uniform self-modulus  $f$  such that  $X \in \Delta_\alpha^0 \wedge X \leq_T f \implies X \equiv_T \emptyset$ .*

Noting that every  $\alpha$ -REA set is either computable or bounds a non-computable  $\Delta_3^0$  set we can immediately infer the following interesting corollary. Professor Shore has asked whether Theorem 2.6 would be sufficient to establish this result but to my knowledge this question is still open.

**Corollary 3.2.** *There are uniform self-moduli that compute no non-trivial  $\alpha$ -REA sets. In particular there are uniform self-moduli not of  $\alpha$ -REA degree.*

The basic machinery used in my proof is the same as that in the Groszek-Slaman result. I build a computable functional  $\Phi$  along with approximations to  $f$  so that  $\Phi$  witnesses  $f$  is a uniform self-modulus. The construction proceeds by making irrevocable commitments of the form  $g \supset \sigma \implies \Phi(g) \supset \tau$  based on the current guess at  $f$  in such a manner that  $\Phi(g) = f$  for any  $g$  above the final  $f$ . However, extending this machinery to higher ordinals significantly complicates the argument. The  $\Delta_2^0$  case only required a finite injury argument to avoid conflict between different  $\Delta_2^0$  sets we wished to diagonalize against. Once we move up to  $\Delta_3^0$  the machinery to avoid a single set may use an infinite number of subrequirements each of which might act infinitely many times. This means the machinery we need to use to meet a higher priority requirement could potentially injure a lower priority argument infinitely often forcing us to adopt a more complicated type of finite injury argument. Moving up past  $\omega$  introduces more difficulties and to meet them we must allow some subrequirements to 'protect' other requirements (taking care never our protection never extends infinitely far) and make use of wellfoundedness to ensure all requirements are met.

The generalization of Theorem 2.5 is much more straightforward.

**Theorem 3.3.** *If  $f$  is a self-modulus then either  $f$  is  $\Delta_{\alpha+1}^0$  or  $f$  can compute an  $\alpha$ -generic.*

Since self-moduli are a relatively unexplored area in computability theory there are a host of other simple lemmas that help establish some basic properties but must be skipped over in the interests of succinctness.

**3.2. Witnessing Membership In  $\Delta_1^1$ .** In Theorem 2.4 Groszek and Slaman demonstrated that every set with a modulus must also have a uniform modulus, but the argument produces a very complex uniform modulus. Since those sets with a modulus must themselves be  $\Delta_1^1$  they must have a  $\Delta_1^1$  modulus but is this the only restriction on the ‘distance’ between a modulus for  $X$  and a uniform modulus for  $X$ ? I hypothesize the answer is yes based on the following partial result.

**Theorem 3.4.** *For every  $n \in \omega$  there is a self-modulus  $f$  such that no  $h \leq_T f^{(n)}$  is a uniform modulus for  $f$ .*

This construction is fairly complex, but the basic idea is as follows. We build  $f$  to be computable in  $\mathcal{Q}^{(n+2)}$  as well as  $n+2$ -generic on some perfect tree  $T$ . The genericity guarantees that  $f$  has no uniform moduli within  $n$  jumps while we keep  $f$  computable in  $\mathcal{Q}^{(n+2)}$  so that any function growing fast enough to compute  $\mathcal{Q}^{(n+2)}$  can compute  $f$ . It therefore only remains to ensure that every ‘small’  $g \succ f$  can compute  $f$  without interfering with the demand that  $f$  be sufficiently generic on some perfect tree.

The trick here is to code  $f$  into the locations where  $f$  is ‘small’ so any function able to enumerate infinitely many places  $f$  is ‘small’ can compute  $f$ . Thus if  $g \succ f$  then either  $g$  will be ‘large’ and therefore have enough computational power to calculate  $f$  or is infinitely often ‘small’ and can use the locations where it is ‘small’ to decode  $f$ . Of course this is a bit of a simplification since we want any  $g \succ f$  to be able to enumerate locations it is ‘small’ so we actually need to define a notion of smallness for every  $k \leq n+2$  which we do by identifying the notion of being  $k+1$ -small with being below a canonical self-modulus for  $\mathcal{Q}^{(k+1)}$  for an interval that is large compared to what happens at level  $k$  in our construction.

Now given any  $g \succ f$  we select the smallest  $k$  so that  $g \not\leq_T \mathcal{Q}^{(k+1)}$ . If  $k \geq n+2$  then  $g$  can directly compute  $f$ . Otherwise infinitely often  $g$  is  $k+1$ -small and as  $g \geq_T \mathcal{Q}^{(k)}$   $g$  can enumerate these locations and recover  $f$ . Of course, a considerable amount of work is required to implement this strategy while still remaining  $n+2$  generic on some perfect tree. In particular, we code  $f$  into the locations that it is  $k+1$ -small by constructing  $f$  to be a path through some tree  $T$  with the property that at any given location  $x$  there is at most one string  $\sigma$  on  $T$  of length  $x$  that is  $k+1$  small. If  $T$  was computable in  $\mathcal{Q}^{(k)}$  this would be unproblematic but to be sufficiently generic we can only guarantee  $T \leq_T \mathcal{Q}^{(k+1)}$ . However, by the limit lemma  $T = \lim_{s \rightarrow \infty} T[s]$  where  $T[s]$  is computable in  $\mathcal{Q}^{(k)}$  and we can arrange the construction so that if  $g \succ f$  and  $g$  is  $k+1$  small at  $x$  then at the stage  $s$  at which  $\mathcal{Q}^{(k)}$  (and hence  $g$ ) realizes this  $f \upharpoonright_x$  is the unique string of length  $x$  satisfying  $g \succ f \upharpoonright_x$  and  $f \upharpoonright_x \in T[s]$ .

**3.3. Syntactic Definitions.** Interestingly the degrees containing a uniform self-modulus have a particularly simple syntactic definition.

**Theorem 3.5.** *A degree  $\underline{d}$  contains a uniform self-modulus if and only if it contains a  $\Pi_2^0$  singleton.*

Not only does this provide a natural semantic analog of the syntactic class of  $\Pi_2^0$  singletons, it also provides an alternative proof that there are self-moduli not of  $\alpha$ -REA degree as a result of Jockusch and Shore [8] (relying on an unpublished result of Harrington) tells us that not every  $\Pi_2^0$  singleton is of  $\alpha$ -REA degree. In fact the particular  $\Pi_2^0$  singleton that Harrington constructed turns out to give an alternate proof of Theorem 3.1 for the case  $\alpha = \omega$ .

Those degrees with a uniform self-modulus are an interesting subclass of the  $\Delta_1^1$  sets with a particularly simple syntactic definition in second order arithmetic. Thus one measure of the ‘complexity’ of a set  $X$  relative to this sort of second order definition is the number of jumps of  $X$  required to compute a uniform modulus of  $X$ . An alternative measure is the number of jumps of  $X$  required to compute a uniform self-modulus  $f \geq_T X$ . The following result shows these always give the same result.

**Theorem 3.6.** *The following are equivalent for  $\alpha < \omega_1^{ck}$*

- *There is a uniform modulus of  $X$  computable in  $X^{(\alpha)}$*
- *There is a uniform self-modulus  $f$  with  $X \leq_T f \leq_T X^{(\alpha)}$*
- *$X$  is a  $\Pi_{2+\alpha}^0$  singleton (note the order of addition).*

Note that one can fairly easily demonstrate that  $X$  has a modulus computable in  $X^{(\alpha)}$  if and only if there is a self-modulus  $f$  with  $X \leq_T f \leq_T X^{(\alpha)}$ . These results illustrate that studying self-moduli, particularly non-uniform self-moduli, is a good way to get a grip on the difference between moduli and uniform moduli in general and suggests that the notion of a self-modulus provides a useful tool to refine our understanding of the  $\Delta_1^1$  sets.

## 4. FUTURE DIRECTIONS

**4.1. Extending Current Work.** The obvious next step in my research is to extend Theorem 3.4 to ordinals past  $\omega$ . However, the current approach I’m taking looks promising and I expect to have the problem solved before I graduate but these leaves no shortage of interesting questions about moduli to be solved. I list a few examples below phrased in terms of self-moduli but with the understanding that explicating self-moduli usually answers the related questions about moduli in general.

- (1) Can the self-moduli be identified with any interesting syntactic class as was done with the uniform self-moduli?

- (2) What sets are low for self-moduli, i.e., for what  $X$  is being a self-modulus relative to  $X$  the same as being a self-modulus relative to  $\mathbb{Q}$ .
- (3) Do the self-moduli relative to  $X$  always ‘resemble’ (isomorphic? Same syntactic definitions?) the self-moduli relative to  $\mathbb{Q}$ .
- (4) Is the self-modulus Slaman and Groszek constructed in Theorem 2.6 of the same degree as an  $\alpha$ -REA set (suggested by a question of Shore).
- (5) Let  $\bigsqcup_i f_i = f$  where  $f(x) = \max_{i \leq x} f_i(x)$ . If  $f_i$  is a sequence of self-moduli and  $g$  a self-moduli with  $g \geq_T f_i$  for all  $i$  must  $g$  compute a function  $g^* \succ f$ ? More generally what is the relation between the join operation on Turing degrees and  $\bigsqcup$ .
- (6) When can we build another self-moduli above/below/between other self-moduli?

On a more long range view there are several potential directions I can see taking this research. In Solovay’s paper [12] identifying the recursively encodable sets with the  $\Delta_1^1$  sets he also classified the hyperarithmetically encodable sets (those sets hyperarithmetical in some subset of every infinite set). An interesting research direction would be to define an analogous notion of a modulus of hypercomputation and pursue similar investigations in the hyperdegrees. I’ve long been curious about higher computability theory and this would be a good opportunity to extend my work in that direction.

Another potential direction would be to pursue connections with randomness and reverse mathematics. Groszek and Slaman have already observed that self-moduli are highly non-random (not 2-random relative to any continuous measure [7]) and there is a growing body of work using measure theoretic notions to specify categories of fast growing functions (e.g., almost everywhere dominating) [3, 4, 1]. Alternatively it might be interesting to go the other way and try to adapt the measure theoretic notions of domination like almost everywhere domination used in the above work to weaken the concept of a modulus. Of course there is no guarantee that the notion of a modulus has any important consequences in these areas but it certainly seems worth examining and even if moduli aren’t specifically useful it would still be interesting to continue work on the broader issue of relating rate of growth to computational properties.

What I’ve mentioned above is where I could take my current research but I also have some more speculative ideas about other directions that might be interesting. One area that I’ve long thought warranted study is the application of randomness notions to the philosophical problem of inductive inference. In particular extending the early work of people like Blum and Gold [2, 6] by defining a formal analog to the notion of a probabilistic scientific explanation (like quantum mechanics) and then investigating the class of reals for which a computable machine could settle on an acceptable theory. In light of the Kuera-Gács theorem [5] this class may be much larger than just the computable reals. Of course this is just one speculative interest of mine and other problems may draw me in different directions.

## REFERENCES

- [1] S. Binns, B. Kjos-Hanssen, M. Lerman, R. Solomon, and PDF File. On a conjecture of dobrinen and simpson concerning almost everywhere domination. *J. Symbolic Logic*, 71(1):119–136, 2006.
- [2] L. Blum and M. Blum. Inductive inference: A recursion theoretic approach. *Recursive Function Theory: Newsletter*, pages 15–29, 1973.
- [3] Peter A Cholak, Noam Greenberg, and Joseph S. Miller. Uniform almost everywhere domination. *Journal of Symbolic Logic*, 71:1057–1072, 2006.
- [4] Natasha L Dobrinen and Stephen G Simpson. Almost everywhere domination. *The Journal of Symbolic Logic*, 69:914–922, 2004.
- [5] P. Gács. Every set is reducible to a random one. *Information and Control*, 70:186–192, 1986.
- [6] E. Mark Gold. Limiting recursion. *The Journal of Symbolic Logic*, 30(1):28–48, mar 1965.
- [7] Marcia J. Groszek and Theodore A. Slaman. Moduli of computation. Lecture delivered at "Logic, Computability and Randomness", January 2007.
- [8] Carl G. Jockusch and Richard A Shore. Pseudo-jump operators. ii: Transfinite iterations, hierarchies and minimal covers. *The Journal of Symbolic Logic*, 49:1205–1236, December 1984. Shows that there are  $\Pi_2^0$  singletons that are not  $RE$  degrees.
- [9] Carl G. Jockusch and Robert I Soare. Encodability of kleene's  $\mathcal{O}$ . *The Journal of Symbolic Logic*, 38:437–440, December 1973.
- [10] Donald A Martin. Classes of recursively enumerable sets and degrees of unsolvability. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 12:295–310, 1966.
- [11] T. G McLaughlin. On the relations between some rate-of-growth conditions. *Proceedings of the American Mathematical Society*, 69:151–155, April 1978.
- [12] Robert M Solovay. Hyperarithmetically encodable sets. *Transactions of the American Mathematical Society*, 239:99–122, May 1978.
- [13] S. Tennenbaum. *Degree of unsolvability and the rate of growth of functions*, pages 71–73. Polytechnic Press of Polytechnic Inst. of Brooklyn, Brooklyn, N.Y., 1963.
- [14] CEM Yates. Three theorems on the degrees of recursively enumerable sets. *Duke Math. J*, 32(3):461–468, 1965.