Rigidity and biinterpretability in the hyperdegrees

Richard A. Shore*
Department of Mathemcatics
Cornell Univeristy
Ithaca NY 14853

October 31, 2005

Abstract

Slaman and Woodin have developed and used set-theoretic methods to prove some remarkable theorems about automorphisms of, and definability in, the Turing degrees. Their methods apply to other coarser degree structures as well and, as they point out, give even stronger results for some of them. In particular, their methods can be used to show that the hyperarithmetic degrees are rigid and biinterpretable with second order arithmetic. We give a direct proof using only older coding style arguments to prove these results without any appeal to set-theoretic or metamathematical considerations. Our methods also apply to various coarser reducibilities.

1 Introduction

Slaman and Woodin [2006] (see also Slaman [1991]) have developed and used set-theoretic and metamathematical techniques to prove some remarkable theorems about the Turing degrees, $\mathcal{D}_T$. These techniques include forcing over models of ZFC to make the set of reals in the ground model countable in the generic extension as well as absoluteness arguments. One key result is that every relation on $\mathcal{D}_T$ invariant under automorphisms and definable in second order arithmetic is actually definable in $\mathcal{D}_T$. They also prove that the double jump is invariant and hence definable. (This result was then used by Shore and Slaman [2000] to prove that the Turing jump itself is definable in $\mathcal{D}_T$.) As other examples, we mention their results that every degree above $0''$ is fixed under every automorphism;

*Partially supported by NSF Grant DMS-0100035 and the IMS of the National University of Singapore. The material in this paper was presented in a talk at the IMS workshop Computational Prospects of Infinity in July, 2005. We also want to thank Ted Slaman for some conversations and suggestions.

there are at most countably many automorphisms of $\mathcal{D}_T$; and in fact every 5-generic $\mathbf{g}$ is an automorphism base (i.e. if π is an automorphism of $\mathcal{D}_T$ and $\pi(\mathbf{g}) = \mathbf{g}$ then π is the identity map).

Slaman [1991] points out that their methods apply to a wide array of degree structures, often giving stronger results based on specific special properties of the reducibility. For example, in the arithmetic degrees, $\mathcal{D}_a$, every automorphism is the identity on the degrees above $0^{(\omega)}$, the first arithmetic jump of 0, while the hyperdegrees $\mathcal{D}_h$ are rigid and biinterpretable with second order arithmetic. Thus every relation on $\mathcal{D}_h$ is definable if and only if it is definable in second order arithmetic.

(We say that X is arithmetic in Y, $X \leq_a Y$, if $X \leq_T Y^{(n)}$ for some $n \in \omega$ and X is hyperarithmetic in Y, $X \leq_h Y$, if $X \leq_T Y^{(\alpha)}$ for some ordinal α recursive in Y where $Y^{(\alpha)}$ is the α^{th} iterate of the Turing jump applied to Y. Kleene showed (see Sacks [1990, II.1-2]) that $X \leq_h Y$ if and only if X is $\Delta^1_1(Y)$. A degree structure $\mathcal{D}$ is biinterpretable with second order arithmetic if there is a definable standard model of arithmetic (or class of structures all isomorphic to $\mathbb{N}$) with definable schemes for both quantification over subsets of the model and a relation matching degrees with codes for sets in the model which are of the specified degrees. Of course, this immediately gives the desired result on definability of relations on $\mathcal{D}$. See Slaman and Woodin [2006] for more details.)

Our goal here is to prove first that $\mathcal{D}_h$ is rigid by a direct coding argument similar to that used in Abraham and Shore [1986] to prove (under mild set theoretic hypotheses such as $\aleph_1^{L[r]}$ being countable for every real r) that the constructibility degrees of reals are rigid. Both arguments are based on lattice embedding construction from the Turing degrees as in Shore [1982a]. The methods needed do the required embeddings are just Cohen-like forcing in the setting of the hyperarithmetic hierarchy. We also use ideas from Shore [1981] and [1982] to make the codings sufficiently effective to make the recovery of the set coded hyperarithmetic in the top degree of the embedded lattice. We then use the coding methods of Slaman and Woodin [1986] (in the hyperarithmetic setting) to translate the rigidity proof to one of biinterpretability. We exploit the specific proof of rigidity to do this translation and so avoid the need to definably deal with automorphisms as is possible using the set-theoretic methods and absoluteness results of Slaman and Woodin.

In Section 2, we present the proof of rigidity in abstract terms based on the existence of a coding scheme satisfying certain properties. In Section 3, we describe the specific lattices we employ that implement these coding requirements. Then, in Section 4, we use Cohen forcing in the hyperarithmetic setting as introduced by Feferman [1965] and presented in Sacks [1990] to show that all countable lattices (with 0) can be embedded into $\mathcal{D}_h$. Finally, in Section 5, we describe the translation to biinterpretability and comment on the applicability of our methods to other coarser degree structures mentioned in Slaman [1991].

2 Rigidity

Intuitively, our basic idea is to code any given set X of degree $\mathbf{x}$ into $\mathcal{D}_h$ "near $\mathbf{x}$" in such a way as to be able to uniquely pick out the set X in way that is definably tied to the degree $\mathbf{x}$. We want some structure $\mathcal{L}_X$ from which X can be "easily" recovered and that we can embed into $\mathcal{D}_h$ near $\mathbf{x}$.

First we explain what constitutes "easily" recoverable. In the setting of the Turing degrees, the underlying obstacle to improving the results fixing, for example, the cone above $0''$ under all automorphisms is the complexity of the notion of Turing reducibility. The relation $X \leq_T Y$ is $\Sigma_3(X, Y)$ and, in general, a formula of the form $(\exists X \leq_T Y)\Phi(X, Y)$ is $\Sigma_3(Y)$ even when Φ is recursive. Thus the best one can hope for is that a formula about $\mathcal{D}_T$ in the language with $\leq_T$ is that it will be Σ_3 in the degrees mentioned and even this only for positive existential formulas. (As the sets Σ_3 in X determine the degree of X'' and vice versa, this is the source of the ubiquitous nature of the double jump, and of $0''$ in particular, in the results on $\mathcal{D}_T$.) In the setting of the constructibility degrees of reals, $\mathcal{D}_c$, the relation $X \leq_c Y$ is itself constructible and so any quantifier free relation in the language with $\leq_c$ on degrees is itself constructible. It is that advantage that permits the coding proof of rigidity of Abraham and Shore [1986] to work in $\mathcal{D}_c$ but leaves it a couple of jumps short in $\mathcal{D}_T$. The situation for hyperarithmetic reducibility is intermediate but close enough to that for constructibility to allow a slightly modified proof to work. Not only is $X \leq_h Y$ a $\Pi^1_1(X, Y)$ relation but a formula $\exists X \leq_h Y \Phi(X, Y)$ is equivalent to a Π^1_1 formula when Φ is itself Π^1_1. Thus a positive coding of both X and its complement $\bar{X}$ (i.e. one that can be decoded by considering only positive existential formulas in the language with $\leq_h$) would show both X and $\bar{X}$ to be Π^1_1, and so hyperarithmetic, in the coding degrees. In all of these settings, the addition of $\vee$ to the language does not cost any more as we can go effectively between X,Y and $X \oplus Y$.

To be a bit more specific we work with lattices with 0 and 1 and plan to associate with each set X a lattice $\mathcal{M}_X$ which codes membership in both X and $\bar{X}$ by a recursive list of positive elementary formulas (in $\leq$ and $\vee$). (Note that we must avoid using $\wedge$ in our decoding formulas since it costs another quantifier.) Thus we will have a lattice $\mathcal{M}_X$ such that, for any lattice embedding $f : \mathcal{M}_X \to \mathcal{D}_h$, $X \leq_h f(1_{\mathcal{M}_X})$. Actually, we work with partial lattices. A partial lattice is a partial order in which join and infimum are determined by relations which define partial functions satisfying the usual conditions (in terms of order) for $\vee$ and $\wedge$ when defined. Our constructions are no different for partial lattices than for lattices and realizing this means that when describing a structure we do not have to specify all the infima and suprema but only the ones relevant to our concerns. Thus whenever we say lattice below we include partial lattices as well.

Next, the notion of "near $\mathbf{x}$" might suggest that we want $f(1_M) \leq_h \mathbf{x}$. This is done for many of the arguments in $\mathcal{D}_T$ but only once one is above, for example, $0'$ or $0''$ or is considering degrees with some other special property. It was also done in $\mathcal{D}_c$ below Cohen generic reals (Abraham and Shore [1986]) and a similar argument would work in the setting of $\mathcal{D}_h$. Instead, we employ one in which "near" means that $f(0_{\mathcal{M}}) = \mathbf{x}$. This procedure avoids some additional argumentation used for $\mathcal{D}_c$ by building a bit more into our structure. We form a new lattice $\mathcal{L}_X$ from two disjoint copies $\mathcal{M}_X$ and $\hat{\mathcal{M}}_X$ of our original one by letting $1_{\mathcal{L}_X}$ be the join of $1_{\mathcal{M}_X}$ and $1_{\hat{\mathcal{M}}_X}$ and $0_{\mathcal{L}_X}$ be their infimum.

Now, suppose we have an embedding $f : \mathcal{L}_X \to \mathcal{D}_h$ that takes $0_{\mathcal{L}_X}$ to $\mathbf{x}$ and we consider any automorphism π of $\mathcal{D}_h$. As it is an automorphism, π carries the image of $\mathcal{L}_X$ under f to another image $\pi f(\mathcal{L}_X)$ in $\mathcal{D}_h$. The coding scheme assumed above insures that $\mathbf{x} \leq_h \pi f(1_{\mathcal{M}_X}), \pi f(1_{\hat{\mathcal{M}}_X})$. On the other hand, as f is a lattice embedding, $\mathbf{x} \equiv_h f(1_{\mathcal{M}_X}) \wedge f(1_{\hat{\mathcal{M}}_X})$ and so applying the automorphisms gives $\pi(\mathbf{x}) \equiv_h \pi f(1_{\mathcal{M}_X}) \wedge \pi f(1_{\hat{\mathcal{M}}_X})$. Thus $\mathbf{x} \leq_{\mathbf{h}} \pi(\mathbf{x})$. The same argument applied to π^{-1} gives $\mathbf{x} \leq_{\mathbf{h}} \pi^{-1}(\mathbf{x})$ and so $\pi(\mathbf{x}) \leq_h \mathbf{x}$. Thus $\mathbf{x} \equiv_{\mathbf{h}} \pi(\mathbf{x})$ for every automorphism π of $\mathcal{D}_h$. To prove rigidity all we have to do now is describe, for each X, a lattice $\mathcal{L}_X$ with the desired properties and prove that it can be embedded in $\mathcal{D}_h$ with $0_{\mathcal{L}_X}$ going to $\mathbf{x}$.

3 Coding

The essential ingredient in making X “easily” recoverable from the coding lattice $\mathcal{L}_X$ is the “effectively generated” model of arithmetic introduced in Shore [1982],[1981]. We here need only the successor function and the coding of the set X. The elements of our lattice that generate a copy of $\mathbb{N}$ are designated by $d_0, e_0, e_{1,} f_0$ and f_1. The element of the lattice corresponding to $n \in \mathbb{N}$ is d_n. The generating scheme that implements the successor function on $\mathbb{N}$ is determined by the following requirements:

1. $(d_{2n} \vee e_0) \wedge f_1 = d_{2n+1}$ and
2. $(d_{2n+1} \vee e_1) \wedge f_0 = d_{2n+2}$.

These conditions clearly guarantee that we can enumerate the d_n recursively in the lattice structure and write a recursive list of quantifier free formulas in this language which define each of them. We wish to convert this procedure and these formulas into ones that are positive in the language with just $\leq$ and $\vee$ at least to the extent that we can use them to code X and $\bar{X}$ (with the aid of other parameters c and $\bar{c}$). In Shore [1982] and [1981] the lattices were embedded as initial segments with the d_n as minimal elements and so it sufficed to say, for example, that $d_{2n+1} \leq (d_{2n} \vee e_0), f_1$ and $d_{2n+1} \neq 0$. As for being different from 0, we can simply add two other parameters p and q and require in our lattice that $q \neq 0$ and $p \vee d_n \geq q$ for each n. Thus we can say of an x that we view as a candidate for being one of the d_n that $x \vee p \geq q$ in place of saying that $x \neq 0$. In the context of coding X (done by exact pairs outside the basic lattice rather than by internal elements in Shore [1981] for reasons extraneous to our concerns here) we can replace the initial segment features of the structure with additional purely lattice theoretic requirements on the coding parameters. Specifically, we require the lattice to have two additional parameters c_X and $\bar{c}_X$ such that $d_n \leq c_X$ for $n \in X$, $d_n \wedge c_X = 0$ for $n \notin X$, $d_n \leq \bar{c}_X$ for $n \notin X$ and $d_n \wedge \bar{c}_X = 0$ for $n \in X$.

We now show how to recursively generate positive existential formulas $\phi_n(x)$ using just $\leq$ and $\vee$ such that, in any lattice $\mathcal{L}_X$ with elements $d_0, e_0, e_{1,} f_0, f_1, p$ and q as described, $\phi_n(x)$ holds of x if and only $0 < x \leq d_n$. Given such formulas, our requirements on c and $\bar{c}$ allow us to define X by $n \in X \Leftrightarrow \exists x(\phi_n(x) \ \&\ x \leq c_X)$ and $n \notin X \Leftrightarrow \exists x(\phi_n(x) \ \&\ x \leq \bar{c}_X)$. As we have already noted, when interpreted in an isomorphic copy of $\mathcal{L}_X$ in $\mathcal{D}_h$, such formulas are equivalent to ones Π^1_1 in the relevant parameters.

We begin with $x = d_0$ as ϕ_0. Recursively, we let $\phi_{2n+1}(x)$ be $\exists z(\phi_{2n}(z) \ \&\ x \leq z \vee e_0, f_1 \ \&\ q \leq x \vee p)$ and $\phi_{2n+2}(x)$ be $\exists z(\phi_{2n+1}(z) \ \&\ x \leq z \vee e_1, f_0 \ \&\ q \leq x \vee p)$. Consider any x such that $\phi_{2n+1}(x)$ holds. We then have a z as described such that, by induction, $0 < z \leq d_{2n}$. Thus $z \vee e_0 \leq d_{2n} \vee e_0$ and so $x \leq d_{2n} \vee e_0, f_1$. As $d_{2n+1} = (d_{2n} \vee e_0) \wedge f_1$, $x \leq d_{2n+1}$ as required. Of course, $q \leq x \vee p$ guarantees that $x > 0$ as well. The argument for ϕ_{2n+2} is essentially the same.

We now turn to embedding countable lattices in $\mathcal{D}_h$.

4 Embedding lattices

In this section, we describe how the elementary methods of Shore [1982a] in the Turing degrees can be used to embed any countable (partial) lattice with 0 in $\mathcal{D}_h$ preserving 0. Relativization to any degree $\mathbf{x}$ supplies the desired embeddings of $\mathcal{L}_X$ (the partial lattice generated as specified above by elements $d_0, e_0, e_{1,} f_0, f_1, p, q, c$ and $\bar{c}$) as it is recursive in X as a partial lattice, i.e. the partial order, the partial functions $\vee$ and $\wedge$ and their domains are recursive in X.

The standard lattice representation arguments (originally from Jonsson [1953] but translated into the language of Lerman [1971] or [1983] and as presented also in Shore [1982a]) give our desired representation theorem. (A simple proof without the requirement for 0 is in Shore [1982a]. Adding the requirement that the value of each function in the representation is 0 at 0 at the beginning presents no difficulties nor does relativization to X.)

Theorem 4.1. *Let $\{p_i\}$ enumerate a recursively presentable partial lattice $\mathcal{P}$ with p_0 its least element. There is a uniformly recursive array of functions $\alpha_n : \omega \to \omega$ such that for all i, j, k, n, m:*

0. $\alpha_n(0) = 0,$

1. $p_i \leq p_j \Rightarrow \alpha_n(j) = \alpha_m(j) \to \alpha_n(i) = \alpha_m(i)$ *and*
$p_i \nleq p_j \Rightarrow \exists q, r(\alpha_q(j) = \alpha_r(j) \ \&\ \alpha_q(i) \neq \alpha_r(i)),$

2. $p_i \vee p_j = p_k \Rightarrow [\alpha_n(i) = \alpha_m(i) \ \&\ \alpha_n(j) = \alpha_m(j) \ \to\ \alpha_n(k) = \alpha_m(k)],$

3. $p_i \wedge p_j = p_k \ \&\ \alpha_n(k) = \alpha_m(k) \ \Rightarrow\ \exists q_1, q_2, q_3[\alpha_n(i) = \alpha_{q_1}(i) \ \&\ \alpha_{q_1}(j) = \alpha_{q_2}(j)$
$\&\ \alpha_{q_2}(i) = \alpha_{q_3}(i) \ \&\ \alpha_{q_3}(j) = \alpha_m(j)].$

We can define an embedding of $\mathcal{P}$ into $\mathcal{D}_h$ from any sufficiently generic function $g : \omega \to \omega$ by setting the image of p_i to be the degree of the function h_i defined by $h_i(n) = \alpha_{g(n)}(i)$. Intuitively, if one iterates the construction of trees of $(n+1)$-generics inside ones of n-generics into the transfinite taking appropriate diagonal-like intersections of the trees at limits one gets paths P which are generic at each level γ of the hyperarithmetic hierarchy and make $P^{(\gamma)} \equiv_T P \oplus 0^{(\gamma)}$. If one does this uniformly and properly for all recursive γ one gets a generic P (indeed one recursive in $\mathcal{O}$) such that, in addition, $\omega_1^P = \omega_1^{CK}$ and so everything hyperarithmetic in P is recursive in $P \oplus 0^{(\gamma)}$ for some recursive γ. This reduces the arguments about hyperarithmetic reducibility to ones about Turing reducibility and so the correctness of the embedding in the hyperarithmetic setting can be read off from the proof for the Turing degrees.

Formalizing this idea seems to require a hierarchy of languages in which one can talk about formulas (and terms representing the sets constructed) at each level of the hyperarithmetic hierarchy. The needed facts can probably be extracted from, e.g. MacIntyre [1977] who modifies the basic approach to Cohen forcing in the hyperarithmetic setting as introduced by Feferman [1965] (or from the analysis of a more general setting in Jockusch and Shore [1984]). We describe a somewhat coarser but more readily available analysis based on essentially the presentation in Sacks [1990] of Feferman's results. For convenience we make the purely notational change of using a function symbol $\mathcal{G}$ in our forcing language in place of one $\mathcal{T}$ for a set as in Sacks [1990] and so the conditions consist of consistent finite conjunctions of formulas of the form $\mathcal{G}(n) = m$ thought off as nonempty subbasic open subsets of ω^ω (in place of 2^ω). (If one prefers, one can keep the set version and code our desired function into a set in any standard way.). We also introduce standard terms $\mathcal{H}_i$ for the h_i with the specified interpretations and the obvious forcing relations. By generic we now mean generic for the (obvious extension of the) ramified language $\mathcal{L}(\omega_1^{CK}, \mathcal{G})$ defined in Sacks [1990, III.4], i.e. for every formula $\mathcal{F}$ there is a condition satisfied by the generic that decides $\mathcal{F}$. Also note that there is a term of the language $\hat{x}\Phi(x)$ for each ranked formula Φ that denotes the set of numbers satisfying Φ which define all the elements of the structure $\mathcal{M}(\omega_1^{CK}, g)$. For generic g with $\omega_1^g = \omega_1^{CK}$, the structure consists of all sets hyperarithmetic in g as is shown there as well. Of course, the sets hyperarithmetic in the h_i are represented by terms $t(\mathcal{H}_i)$ built up from $\mathcal{H}_i$, i.e. ones $\hat{x}\Phi(x)$ where Φ (hereditarily) contains $\mathcal{H}_i$ but no other $\mathcal{H}_j$ or $\mathcal{G}$. The existence of functions g generic in this sense with $g \leq_h \mathcal{O}$ and so $\omega_1^g = \omega_1^{CK}$ as well as all the usual facts about generic objects can be found in Sacks [1990, IV.3]. We can now easily argue as in the Turing degrees that we have the desired (partial) lattice embedding.

Theorem 4.2. *Let $\mathcal{P}$ be a recursive partial lattice (with 0); let α_n be a recursive representation of $\mathcal{P}$ as in Theorem 4.1; and g be an $\mathcal{M}(\omega_1^{CK})$ generic. The map taking p_i to the hyperdegree of h_i as defined above preserves the partial lattice structure of $\mathcal{P}$.*

Proof. The argument is standard. As $h_0(n) = \alpha_{g(n)}(0) = 0$ for every n by 4.1.0, 0 is preserved. If $p_i \leq p_j$ then $h_i(n) = \alpha_n(i) = \alpha_m(i)$ for any m such that $\alpha_m(j) = \alpha_n(j) = h_j(n)$ and so h_i is even recursive in h_j as the array α_n is uniformly recursive. Thus the embedding preserves order. Similarly if $p_i \vee p_j = p_k$ we can compute $h_k(n) = \alpha_{g(n)}(k)$ recursively from $h_i(n)$ and $h_j(n)$ by finding any α_m such that $\alpha_m(i) = h_i(n) = \alpha_{g(n)}(i)$ and $\alpha_m(j) = h_j(n) = \alpha_{g(n)}(j)$ as by 4.1.1, $\alpha_m(k) = \alpha_{g(n)}(k) = h_k(n)$ for any such m. Thus the embedding preserves join when defined in $\mathcal{P}$. The arguments for preserving $\nleq$ and $\wedge$ depend on genericity.

Suppose $p_i \nleq p_j$ and consider any term $t(\mathcal{H}_j)$ of the language generated by $\mathcal{H}_j$. We wish to show that $t(h_j) \neq h_i$ as this implies that $h_1 \nleq_h h_j$. If it were otherwise, there would be a condition $q \Vdash t(\mathcal{H}_j) = \mathcal{H}_i$. By 4.1.2 there are n and m such that $\alpha_n(j) = \alpha_m(j)$ but $\alpha_n(i) \neq \alpha_m(i)$. Let r be an extension of q containing the formula $\mathcal{G}(z) = n$ which decides a values for $t(\mathcal{H}_j)(z)$ and forces $g(z) = n$ for some z not mentioned in q and let r' be the same condition as r except that it contains $\mathcal{G}(z) = m$ in place of $\mathcal{G}(z) = n$. As the interpretation of $\mathcal{H}_j$ is the same in any two generics extending r and r' which differ only at z both conditions force the same value for $t(\mathcal{H}_j)(z)$. On the other hand, $r \Vdash \mathcal{H}_i(z) = \alpha_n(i)$ while $r' \Vdash \mathcal{H}_i(z) = \alpha_m(i)$. As $\alpha_n(i) \neq \alpha_m(i)$ by our choice of n and m we have the desired contradiction. Thus the embedding preserves $\nleq$.

Finally, suppose that $p_i \wedge p_j = p_k$. We already know that h_i and h_j are hyperarithmetic (indeed recursive) in h_k by the preservation of order. We wish to show that any $f \leq_h h_i, h_j$ is hyperarithmetic in h_k. We take terms t_0 and t_1 and a condition q satisfied by g that forces $t_0(\mathcal{H}_i) = t_1(\mathcal{H}_j)$ and describe a procedure hyperarithmetic in h_k that defines $f = t_0(h_i) = t_1(h_j)$. The definition of f from h_k as a function on ω is given by the following procedure: to find $f(u)$ find any r extending q which forces some particular (necessarily common) value for $t_0(\mathcal{H}_i)(u)$ and $t_1(\mathcal{H}_j)(u)$ such that $h_k(z)(= \alpha_{g(z)}(k)) = \alpha_{r(z)}(k)$ for every z in the domain of r. We claim that this procedure is hyperarithmetic in h_k and provides the true value of these terms evaluated on h_i and h_j, respectively. It is, of course, recursive in h_k to check that a forcing condition satisfies the second requirement. As forcing for sentences of fixed rank is a hyperarithmetic relation, this procedure produces a hyperarithmetic reduction of f to h_k as long as it always produces the correct value.

To see that the procedure always produces the correct values, suppose there are $u, v \neq w \in \omega$ and some r extending q as described such that $r \Vdash t_0(\mathcal{H}_i)(u) = t_1(\mathcal{H}_j)(u) = v$ but $t_0(h_i) = w = t_1(h_j)$. As g is generic there is an s extending q satisfied by g that forces $t_0(\mathcal{H}_i)(u) = t_1(\mathcal{H}_j)(u) = w$. We now work for a contradiction. Assume without loss of generality that the domains of r and s are the same. For each z in this common domain, $\alpha_{r(z)}(k) = \alpha_{s(z)}(k)$ by our assumption on r. By 4.1.3, we can choose for each such z numbers $q_{z,l}$ for $l = 1, 2, 3$ witnessing the conclusion of 4.1.3 for $\alpha_{r(z)}$ and $\alpha_{s(z)}$ in place of α_n and α_m. We define forcing conditions q_l by $q_l(z) = q_{z,l}$. We extend these conditions to generics g^l (with g^0 extending r) by simply copying g when they are not defined. (Finite changes in a generic keep the function generic.) It is clear from the requirements of 4.1.3 that $h_i^0 = h_i^1$, $h_j^1 = h_j^2$, $h_i^2 = h_i^3$ and $h_j^3 = h_j$. Thus by our choice of q and the first equality, $v = t_0(h_i^0)(u) = t_0(h_i^1)(u)$. Our choice of q and the next equalities give $t_0(h_i^1)(u) = t_1(h_j^1)(u) = t_1(h_j^2)(u) = t_0(h_i^2)(u) = t_0(h_i^3)(u) = t_1(h_j^3)(u) = t_1(h_j)(u)$ but this last term has value $w \neq v$ by assumption for the desired contradiction. □

This theorem shows that the partial lattices described in the Section 3 can be embedded in $\mathcal{D}_h$ as required to implement the proof of rigidity described there and in Section 2. Thus we have our direct proof of rigidity.

Theorem 4.3. (Slaman and Woodin) *The hyperdegrees are rigid.*

5 Biinterpretability

Slaman and Woodin [2006] prove for $\mathcal{D}_T$ that rigidity implies biinterpretability by showing that one can describe their full analysis of persistence of automorphisms within the degree structure itself. As mentioned in Slaman [1991] their methods apply to many other degree structures. Based on our direct proof of rigidity for $\mathcal{D}_h$ we can derive biinterpretability using only the coding methods of Slaman and Woodin [1986] to code each countable relation (there on $\mathcal{D}_T$, here on $\mathcal{D}_h$) by finitely many parameters (uniformly in the arity of the relation) in the setting of the hyperarithmetic degrees. These methods will also apply to the coarser degree structures mentioned in Slaman [1991].

There is not much needed here beyond pointing out that the coding constructions of Slaman and Woodin [1986] work the same way for $\mathcal{D}_T$ as long as one uses Cohen-like forcing in the hyperarithmetic setting in place of the arithmetic one used in their work. Indeed, the arguments here are even a bit simpler since we do not need the careful calculations done there of how much genericity is needed. We give a brief description of the argument.

The first step is to prove that any antichain of degrees $\mathbf{a}_i$ is definable from three parameters $\mathbf{b}, \mathbf{g}_1$ and $\mathbf{g}_2$. We take $B \in \mathbf{b}$ to be an upper bound on the $\mathbf{a}_i$. The other parameters are defined by a forcing construction. One begins with a sequence of representatives $A_i \in \mathbf{a}_i$ such that A_i is recursive in any of its infinite subsets. The notion of forcing, $\mathcal{P}$, consists of triples $p = \langle p_1, p_2, p_3 \rangle$ where $p_1, p_2 \in 2^{<\omega}$ have the same length (which we also call the length of p) and $p_3 \in \omega$. Extension is defined by $q \leq p \Leftrightarrow q_1 \supseteq p_1 \;\&\; q_2 \supseteq p_2 \;\&\; q_3 \geq p_3 \;\&\; \forall k \leq p_3 \forall a \in A_k(|p_1| < \langle k, a \rangle \leq |q_1| \rightarrow q_1(\langle k, a \rangle) = q_2(\langle k, a \rangle))$. Let G be any generic for $\mathcal{P}$ in the sense of hyperarithmetic forcing analogous to the one in Sacks [1990] as described above relativized to B with basic terms $\mathcal{A}_k$ for the A_k, $\mathcal{B}$ for B and $\mathcal{G}_1, \mathcal{G}_2$ for the unions of first and second coordinates of $\mathcal{G}$ as well. If G_1, G_2 are the union of the first coordinates of G then the set $\{\mathbf{a}_i | i \in \omega\}$ of hyperdegrees is definable as the set of minimal solutions $\mathbf{x}$ in $\mathcal{D}_h$ below $\mathbf{b}$ to the equation $(\mathbf{g}_1 \vee \mathbf{x}) \wedge (\mathbf{g}_2 \vee \mathbf{x}) \neq \mathbf{x}$.

To see that each $\mathbf{a}_k$ satisfies the equation, consider a condition $p \in G$ with $p_3 \geq k$ and the set $C(k) = \{m \in A_k | \; |p_1| < \langle k, m \rangle \in G_1\}$. It is immediate from its definition that $C(k) \leq_T G_1 \oplus A_k$ and it follows from the choice of p and the definition of extension that $C(k) = \{m \in A_k | \; |p_1| < \langle k, m \rangle \in G_2\}$ and so $C(k) \leq_T G_2 \oplus A_k$ as well. To see that $C(k) \not\leq_h A_k$ suppose to the contrary and choose a term $t_1(\mathcal{G}_1)$ for $C(k)$ and consider a term t so that $t(\mathcal{A}_k)$ is a standard name for a given set hyperarithmetic in A_k and a condition $q \in G$ extending p such that $q \Vdash t(\mathcal{A}_k) = t_1(\mathcal{G}_1)$. Let $m \in A_k$ be larger than $|q_1|$. We can clearly find an extension r of q such that $r_1(\langle k, m\rangle) \neq t(A_k)(\langle k, m\rangle)$ as the right hand side is fixed independently of the choice of generic. This gives the desired contradiction.

Next, we suppose we have some $C \leq_h B$ and $D \leq_h G_1 \oplus C, G_2 \oplus C$ such that $D \not\leq_h C$ and prove that $A_k \leq_h C$ for some k. Choose terms $t_C(\mathcal{B}), t_D(\mathcal{B}), t_1(\mathcal{G}_1 \oplus t_c(\mathcal{B})), t_2(\mathcal{G}_2 \oplus t_c(\mathcal{B}))$ representing the relevant sets and a condition $p \in G$ such that $p \Vdash t_D(\mathcal{B}) = t_1(\mathcal{G}_1 \oplus t_c(\mathcal{B})) = t_2(\mathcal{G}_2 \oplus t_c(\mathcal{B}))$. There must be infinitely many n with conditions q^n, r^n of p of common length at least n extending p that force different values for $t_1(\mathcal{G}_1 \oplus t_c(\mathcal{B}))(n)$ as otherwise we could compute D hyperarithmetically in C by finding for each n (other than the finitely many assumed exceptions) any condition q of length at least n extending p that forced a value for $t_1(\mathcal{G}_1 \oplus t_c(\mathcal{B}))(n)$ and know that it is the correct value. As forcing for formulas of fixed rank is hyperarithmetic in the parameter C, this would contradict our assumption that $D \not\leq_h C$. We can thus find such q^n and r^n hyperarithmetically in C. We can interpolate a sequence of conditions $s^{n,1}, \ldots, s^{n,m}$ between q^n and r^n so that the successive $s^{n,i}$ differ at exactly one number. We can then extend the $s^{n,i}$ to $\hat{s}^{n,i}$ which also differ only at that same location and each force a value for $t_1(\mathcal{G}_1 \oplus t_c(\mathcal{B}))(n) = t_2(\mathcal{G}_2 \oplus t_c(\mathcal{B}))(n)$. As the values at the two ends are different there must be an i such that $\hat{s}^{n,i}$ and $\hat{s}^{n,i+1}$ force different values. If the one location $\langle j, m\rangle$ at which they differed were not such that $j < p_3$ and $m \in A_j$ then we could form a single condition $s = \langle \hat{s}_1^{n,i}, \hat{s}_2^{n,i+1}, u\rangle$ for u the maximum of the third coordinates of the two conditions which would extend p and have the same values for G_1 and G_2 as $\hat{s}^{n,i}$ and $\hat{s}^{n,i+1}$, respectively. Thus s would force the same value for $t_1(\mathcal{G}_1 \oplus t_c(\mathcal{B}))(n)$ as $\hat{s}_1^{n,i}$ while it would force the same value for $t_2(\mathcal{G}_2 \oplus t_c(\mathcal{B}))(n)$ as $\hat{s}^{n,i+1}$. As these are different and s extends p this would be a contradiction. Thus there are infinitely many pairs of conditions (extending p) differing only at one point $\langle j, m\rangle$ with $j < p_3$ which force different answers. Again, as the forcing relation for formulas of fixed rank is hyperarithmetic, we can find infinitely many such hyperarithmetically in the parameter C. As all of these must have $m \in A_j$ by our argument above and there must be infinitely many with the same $j < p_3$, we can find infinitely many $m \in A_j$ for this j hyperarithmetically in C. Thus by our choice of A_j, $A_j \leq_h C$ as required.

Finally, Slaman and Woodin [1986] show how to convert an arbitrary countable relation on the degrees into an antichain so that the original relation is definable from the antichain and parameters. Suppose R is an n-ary relation on the degrees less than $\mathbf{b}$, $\langle \mathbf{b}_j | j \in \omega\rangle$ lists the degrees below $\mathbf{b}$ with representatives B_j, and $G_{i,j}$ for $1 \leq i \leq n$ and $j \in \omega$ are mutually Cohen generics over $B \in \mathbf{b}$ in the sense of hyperarithmetic forcing of degrees $\mathbf{g}_i$ (for example the appropriate columns of a single generic). Now all of the sets $\{\mathbf{b}_j \vee \mathbf{g}_{i,j} | j \in \omega\} = S_i$ and $\{\mathbf{g}_{i,j} | j \in \omega\} = T_i$ are antichains in $\mathcal{D}_h$ and so definable as above as is $U = \{\mathbf{g}_{1,j_1} \vee \ldots \vee \mathbf{g}_{n,j_n} |\ R(\mathbf{b}_{j_1}, \ldots, \mathbf{b}_{j_n})\}$. Thus $R(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ if and only if there are $\mathbf{y}_i \in T_i$ and $\mathbf{z}_i \in S_i$ such that the join of the $\mathbf{y}_i$ is in U and for each i, $\mathbf{x}_i \vee \mathbf{y}_i = \mathbf{z}_i$.

Thus we can definably quantify over all countable relations on $\mathcal{D}_h$. In particular, we can definably describe a class of parameters that each define by some fixed scheme the standard model of arithmetic with a scheme for coding subsets of the model by degrees as well. This allows us to translate our proof of rigidity into biinterpretability. For any coding scheme for a model of arithmetic we can define a relation between degrees and codes of sets in such a model that associates degrees with sets of that degree. To be specific, we say that a degree $\mathbf{x}$ is associated with a set X coded in the model if for every set Y coded in the model there is a (partial) lattice isomorphic to $\mathcal{L}_Y$ in $\mathcal{D}_h$ with least element $\mathbf{x}$ if and only if (in the model) $Y \leq_h X$. As each $\mathcal{L}_Y$ is recursive in Y, it can be described in the model using the code for X and the apparatus of arithmetic in the model. The required images of $\mathcal{L}_Y$ in $\mathcal{D}_h$ and isomorphism between it and the version coded in the model of arithmetic can then be specified by other countable relations on the degrees of the model and ones above $\mathbf{x}$ and below some $\mathbf{z}$ representing the top of the lattice. Thus we have our direct proof of biinterpretability.

Theorem 5.1. (Slaman and Woodin) *The structure $\mathcal{D}_h$ of the hyperdegrees is biinterpretable with second order arithmetic.*

We close with a comment on another view of hyperarithmetic reducibility and its implications for some other degree structures.

We have mentioned two views of the hyperarithmetic sets. The first sees them as the sets recursive in $0^{(\alpha)}$ for some recursive ordinal α. The second as the Δ^1_1 sets. A third view sees them as the subsets of ω constructed in Gödel's L before the first nonrecursive ordinal, ω_1^{CK}. In the last view, we see X as hyperarithmetic in Y if $X \in L_{\omega_1^Y}[Y]$, i.e. X is constructed before the first ordinal not recursive in Y with the use of a predicate for Y in the language. This view of $\leq_h$ has a natural generalization when one sees ω_1^{CK} as the first Σ_1 admissible ordinal and $L_{\omega_1^{CK}}$ as the least admissible set containing ω. The relativization sees $L_{\omega_1^Y}[Y]$ as the least Σ_1 admissible set containing Y. The suggested reducibility generalizes Σ_1 to Σ_n and we say that $X \leq_{\Sigma_n} Y$ if X is a member of the least Σ_n admissible set containing Y. The associated degrees are called the Σ_n-admissible degrees in Slaman [1991]. As noted there, the methods of Slaman and Woodin carry over to these degrees as well. So do the ones presented here.

The notions of forcing to be considered are the same. The universes are now of the form $L_{\omega_n^X}[X]$ where ω_n^X is the first ordinal α such that $L_{\omega_n^X}[X]$ is Σ_n admissible. Cohen forcing in these settings has been considered in α-recursion theory. The crucial point for the forcing constructions is the preservation of Σ_n admissibility. Once this is established, the arguments for e.g. incomparability and infima requirements are the same. Discussions of the forcing and preservation of Σ_n admissibility can be found, for example, in Chong [1984] for $n = 1$ and for larger n in Shore [1974] albeit mixed in there with a more complicated priority argument. The essential ingredient for our decoding analysis was that the decoding of X (and so also $\bar{X}$) was given by a formula which was Σ_1 over the structure being considered and that X being Δ_1 over the structure for Y guarantees that X is reducible to Y. (Note that if X is Δ_1 over $L_{\omega_n^Y}[Y]$ then $X \in L_{\omega_n^Y}[Y]$ as even Σ_1 admissibility gives a bound β on the witnesses needed to demonstrate that $n \in X$ or

$n \notin X$ for each $n \in \omega$. We can then define X over $L_\beta[Y]$ and so see that $X \leq_{\Sigma_n} Y$.) The formulas decoding $n \in X$ (and $n \in \bar{X}$) were all of the form that there are various sets reducible to a fixed set Z with properties described by positive formulas in the orderings. Each set Σ_n-reducible to Z is given simply by an ordinal less than ω_n^Z and the relations of one set being constructible from another by a given ordinal and an ordinal being Σ_n admissible relative to a given set are certainly Δ_1 (over even any Σ_1 admissible) in the sets and ordinals. Thus the decoding of X from an embedding of $\mathcal{M}_X$ produces a set Σ_n-reducible to the image of $1_{\mathcal{M}_X}$ as required. Our forcing arguments then give a direct proof of the analogous results.

Theorem 5.2. (Slaman and Woodin) *The structures of the Σ_n-degrees are for $n > 1$ are rigid and biinterpretable with second order arithmetic.*

References

[1] Chong, C. T. [1984], *Techniques of Admissible Recursion Theory*, LNM **1106**, Springer-Verlag, Berlin.

[2] Feferman, S. [1965], Some applications of the notion of forcing and generic sets, *Fund. Math.* **56**, 325-45.

[3] Jockusch, C. G. Jr. and Shore, R. A. [1984], Pseudo-jump operators II: transfinite iterations, hierarchies and minimal covers, *J. Symb. Logic* **49**, 1205-1236.

[4] Jónsson, B. On the representations of lattices, *Math. Scand.* 1, 193-206.

[5] Lerman, M. [1971], Initial segments of the degrees of unsolvability, *Ann. Math.* **93**, 365-89.

[6] Lerman, M. [1983], *Degrees of Unsolvability*, Perspectives in Mathematical Logic, Springer-Verlag, Berlin.

[7] MacIntyre, J. [1977], Transfinite extensions of Friedberg's completeness criterion. *J. Symb. Logic* **42**, 1–10.

[8] Sacks, G. E. [1990], *Higher Recursion Theory*, Perspectives in Mathematical Logic, Springer-Verlag, Berlin.

[9] Abraham, U. and Shore, R. A. [1986], The degrees of constructibility below a Cohen real, *J. London Math. Soc.* (3) **53** (1986), 193-208.

[10] Shore, R. A. [1974], Σ_n sets which are Δ_n-incomparable (uniformly), *J. Symb. Logic* **39** (1974), 295-304

[11] Shore, R. A. [1981], The theory of the degrees below $0'$, *J. London Math. Soc.* (3) **24** (1981), 1-14.

[12] Shore, R. A. [1982], On homogeneity and definability in the first order theory of the Turing degrees, *J. of Symb. Logic* **47** (1982), 8-16.

[13] Shore, R. A. [1982a], Finitely generated codings and the degrees r.e. in a degree **d**, *Proc. Am. Math. Soc.* **84**, 256-263.

[14] Shore, R. A. and Slaman, T. A. [2000], Defining the Turing jump, *Math. Research Letters* **6** (1999), 711-722.

[15] Slaman, T. A. [1991], Degree structures, in *Proc. Int. Cong. Math., Kyoto 1990*, Springer-Verlag, Tokyo, 303-316.

[16] Slaman, T. A. and Woodin, H. W. [1986], Definability in the Turing degrees, *Illinois J. Math.* **30**, 320-334.

[17] Slaman, T. A. and Woodin, H. W. [2006], Definability in degree structures, this volume.