ON THE STRUCTURE OF GAMMA DEGREES[1]

ALAN P. SILVER

In [4], Ladner, Lynch and Selman defined a non-deterministic polynomial time bounded version of many-one reducibility, denoted by $\leq_m^{\mathcal{NP}}$, as follows:

> For any sets A and B, we say $A \leq_m^{\mathcal{NP}} B$ if and only if there is a non-deterministic Turing transducer, M, that runs in polynomial time such that, $x \in A$ just in case there is a y, computed by M on input x, with $y \in B$.

However, their definition does not completely capture the essence of a many-one reducibility due to the fact that, given some $x \in A$, there only needs to be *some* y output by M such that $y \in B$. It may be the case that all computation branches of M halt on input x, but only one of the output values is actually in B. Seen in this light, their reducibility is obviously a candidate for a polynomial time bounded singleton reducibility rather than a many-one reducibility. This intuitive idea is borne out by the fact that, if we define $\leq_s^{\mathcal{P}}$, a polynomial time bounded version of singleton reducibility (see [7] for details), then $\leq_m^{\mathcal{NP}} \equiv \leq_s^{\mathcal{P}}$. We claim that gamma reducibility is the correct notion of a non-deterministic polynomial time bounded many-one reducibility and study its properties.

Gamma reducibility, $\leq_\gamma$, was first defined by Adleman and Manders [1] in 1977, and was later studied by Long in [5] and [6]. Its relevance to the world of time bounded computations comes, in part, from the fact that $\mathcal{P} = \mathcal{NP} \Rightarrow [\leq_\gamma \equiv \leq_m^{\mathcal{P}}] \Rightarrow \mathcal{P} = \mathcal{NP} \cap \text{co-}\mathcal{NP}$ and that if A is an $\mathcal{NP}$-complete set (with respect to $\leq_\gamma$), then $A \in \mathcal{NP} \cap \text{co-}\mathcal{NP} \Leftrightarrow \mathcal{NP} = \text{co-}\mathcal{NP}$.

DEFINITION 1 (Adelman and Manders [1]). $A \leq_\gamma B$ if there is a non-deterministic polynomial time bounded transducer, M, such that if

$$G(M) = \{\langle x, y\rangle \mid \text{On input } x,\ M \text{ outputs } y \text{ on some computation branch}\}$$

then:

(I) $\quad (\forall x)(\exists y)[\langle x, y\rangle \in G(M)]$

(II) $\quad (\forall x)(\forall y)[\langle x, y\rangle \in G(M) \Rightarrow [x \in A \Leftrightarrow y \in B]]$

Trivially, $A \leq_\gamma B \Leftrightarrow \overline{A} \leq_\gamma \overline{B}$.

---

[1] Research undertaken as part of the author's doctoral thesis [7], supported by the UK Science and Engineering Research Council, award number 88002274. The author wishes to thank his supervisor, Dr. Barry Cooper, and also Prof. Klaus Ambos-Spies for much useful advice in the preparation of this work.

We note without proof that $\langle \mathcal{D}_\gamma, \leq_\gamma \rangle$ is an upper semilattice with $A \oplus B$ as the least upper bound of A and B, that the $\mathcal{NP}$ γ-degrees form an ideal of $\langle \mathcal{D}_\gamma, \leq_\gamma \rangle$ and that the zero degree, $\mathbf{0}_\gamma$, is exactly $\mathcal{NP} \cap$ co-$\mathcal{NP}$.

DEFINITION 2.

i) For any two sets, A and B, we say that A and B *differ finitely*, $A =^* B$, if and only if $A - B \cup B - A$ is finite.

ii) A class of sets $\mathcal{C}$ is *closed under finite variations* (c.f.v.) if, for $A \in \mathcal{C}$ and $B =^* A$, we have $B \in \mathcal{C}$.

iii) A class of recursive sets, $\mathcal{C}$, is *recursively presentable* (r.p.) if $\mathcal{C}$ is empty or there exists a recursive set U, called the *universal set for* $\mathcal{C}$, such that

$$\mathcal{C} = \{U^e \mid e \in \mathbb{N}\} \quad \text{where } U^e = \{x \mid \langle e, x \rangle \in U\}.$$

Note that if $\mathcal{C}$ is any recursively presentable class of sets, say $\mathcal{C} = \{A_i\}_{i \in \mathbb{N}}$, then for each i, $A_i \leq^{\mathcal{P}}_{\mathrm{m}} U$ via $\lambda x[\langle i, x \rangle]$, where U is the universal set for $\mathcal{C}$. Thus $A_i \leq_\gamma U$.

It is simple to show that if A is a recursive set, then $deg_\gamma(A)$ contains only recursive sets, and that the recursive γ-degrees form an ideal of the γ-degrees and are c.f.v. and r.p.

Let f be a strictly increasing function where $f(0) \neq 0$, and define f^n by $f^0(m) = m$ and $f^{n+1}(m) = f(f^n(m))$. The $(n+1)th$ *f-interval* is $I^f_n = \{x \mid f^n(0) \leq |x| < f^{n+1}(0)\}$, and we note that $\{I^f_n\}_{n \in \mathbb{N}}$ partitions $\{0,1\}^*$. Furthermore, if A is recursive, then let $I^f_A = \bigcup \{I^f_n \mid n \in A\}$.

Ambos-Spies defines a function as *polynomially honest* if f is recursive and there is some polynomial, p, such that for all x, we can compute $f(x)$ in less than $p(f(x))$ steps. This is clearly *not* equivalent to Homer's original definition of polynomial honesty. To avoid confusion, we will refer to a function satisfying Ambos-Spies's definition as *polynomially honest II*. Clearly, if f is polynomially honest II and $A \in \mathcal{P}$ (so A is recursive), then $I^f_A \in \mathcal{P}$. A function, g, *dominates* f if $(\forall n)[f(n) < g(n)]$. Note that if f is recursive, then there is some strictly increasing polynomially honest II function, g, that dominates f (see [2]).

A recursive set $A \notin \mathcal{P}$ is *super sparse II* if there is a strictly increasing polynomially honest II function f such that $A \subseteq Z_f = \{0^{f(n)} \mid n \in \mathbb{N}\}$ and "$0^{f(n)} \in A$?" can be answered in less than $f(n+1)$ steps. As above, we call this super sparse II to distinguish it from Hartmanis's well-known notion of super sparseness.

Finally, let

$$\begin{aligned} A^{(n)} &= \{x \mid \langle n, x \rangle \in A\} \\ A^{(\leq n)} &= \{\langle m, x \rangle \mid m \leq n \ \& \ \langle m, x \rangle \in A\} \\ kA &= \{kx \mid x \in A\} \\ kA + i &= \{kx + i \mid x \in A\} \end{aligned}$$

LEMMA 3 (Ambos-Spies [2]). *There exists a super sparse II set in EXP-TIME.*

PROOF. Let $f(0) = 1$ and $f(n+1) = 2^{f(n)}$. We will construct A as a subset of $\{0^{f(2n+1)} \mid n \in \mathbb{N}\}$.

Let $\{R_i\}_{i\in\mathbb{N}}$ be an enumeration of the sets in $\mathcal{P}$. Given some n, perform 2^n steps of the following algorithm:

i) Find the largest $m \leq n$ such that $f(2m+1) \leq n$.
ii) If $f(2m+1) < n$ then $0^n \notin A$. Otherwise;
iii) Set $0^n \in A \Leftrightarrow 0^n \notin R_k$ where $m = \langle k, l\rangle$.

If the computation does not terminate in 2^n steps, then $0^n \notin A$.

It is clear that $A \in$ EXPTIME. Suppose, if possible, that $A \in \mathcal{P}$, then there is some $k \in \mathbb{N}$ such that $A = R_k$.

Now, consider n such that there is some m where $f(2m+1) = n$. Clearly, since there is a polynomial, p, such that $f(n+1)$ can be computed in less than $p(f(n))$ steps, then $f(2m+1)$ can be computed in less than $p(f(2m))$ steps. Thus, for sufficiently large n, $f(2m+1)$ can be computed in less than 2^n steps.

So, for large n we will have that $0^n \in A \Leftrightarrow 0^n \notin R_k$ where $m = \langle k, l\rangle$ and $f(2m+1) = n$.

Therefore, given k we will have, for sufficiently large l:

$$0^{f(2\langle k,l\rangle+1)} \in A \quad \Leftrightarrow \quad 0^{f(2\langle k,l\rangle+1)} \notin R_k$$

Thus $A \neq R_k$ and so $A \notin \mathcal{P}$.

Finally we note that $(\forall n)[\ 2^n \leq f(n+1)\ ]$, so "$0^{f(n)} \in A$?" can be answered in less than $f(n+1)$ steps.

Thus A is super sparse II. □

THE JOIN LEMMA (Ambos-Spies [2]). *Let C_0 and C_1 be any recursive sets and $\mathcal{C}_0, \mathcal{C}_1$ be r.p. and c.f.v. classes such that $C_0 \cup C_1 \notin \mathcal{C}_0$ and $C_1 \notin \mathcal{C}_1$. Then there is a recursive function, g_0, such that if g is a strictly increasing function that dominates g_0, and A is an infinite coinfinite recursive set, then $(C_0 \cap I_A^g) \cup C_1 \notin \mathcal{C}_0 \cup \mathcal{C}_1$.*

THE DENSITY THEOREM. *Let* $\mathbf{a}$ *and* $\mathbf{b}$ *be the γ-degrees of recursive sets such that* $\mathbf{b} < \mathbf{a}$. *Then there is a recursive γ-degree,* $\mathbf{d}$ *such that* $\mathbf{b} < \mathbf{d} < \mathbf{a}$.

PROOF. Fix sets $A = \{2x \mid x \in A_0\}$ and $B = \{2x+1 \mid x \in B_0\}$ for some $A_0 \in \mathbf{a}$ and $B_0 \in \mathbf{b}$. Clearly A and B are recursive and γ-equivalent to A_0 and B_0 respectively.

Now suppose, if possible, that $A \cup B \in \mathbf{b}$, i.e., $A \cup B \leq_\gamma B$. Then there is some non-deterministic polynomial time bounded transducer, M, such that:

(I) $(\forall x)(\exists y)[\langle x, y\rangle \in G(M)]$

(II) $(\forall x)(\forall y)[\langle x, y\rangle \in G(M) \Rightarrow [x \in A \cup B \Leftrightarrow y \in B]]$

Note that $x \in A \cup B \Leftrightarrow (x = 2m \,\&\, x \in A) \vee (x = 2m+1 \,\&\, x \in B)$.

Define a non-deterministic machine, N, such that on input x, N checks if $x = 2m+1$. If so then N outputs some fixed $b \in \overline{B}$, otherwise N simulates M on input x and outputs $M(x)$.

Clearly, as M is polynomial time bounded, then N is too, and we have $(\forall x)(\exists y)[\langle x, y\rangle \in G(N)]$.

Now, suppose $\langle x, y\rangle \in G(N)$, then

i) $x = 2m+1$. Then $y = b$ and $x \notin A$ as $A \subseteq 2\mathbb{N}$.

Thus $x \notin A$ & $y \notin B$.

ii) $x = 2m$. Then if $x \in A$ we must have $y \in B$, and if $x \notin A$ then as $x \notin B \subseteq 2\mathbb{N}+1$, we must have $y \notin B$.

Thus $x \in A \Leftrightarrow y \in B$, i.e., $A \leq_\gamma B$ via N. This contradicts the fact that $A <_\gamma B$, so we must have $A \cup B \notin \mathbf{b}$.

Also, since $B \notin \mathbf{a}$, then we can apply the Join Lemma with $C_0 = A$, $C_1 = B$, $\mathcal{C}_0 = \mathbf{b}$ and $\mathcal{C}_1 = \mathbf{a}$. This will give the function g_0, so consider some strictly increasing polynomially honest II function, g, that dominates g_0. Clearly $2\mathbb{N}$ is infinite, coinfinite and recursive, so the Join Lemma gives that:

$$(A \cap I^g_{2\mathbb{N}}) \cup B \;\notin\; \mathbf{b} \cup \mathbf{a}$$

Now, let $D = (A \cap I^g_{2\mathbb{N}}) \cup B$.

If $B \leq_\gamma D \leq_\gamma A$, then we have $B <_\gamma D <_\gamma A$ as the Join Lemma ensures that $D \notin \mathbf{a}, \mathbf{b}$. Thus we have $\mathbf{b} < \mathbf{d} < \mathbf{a}$.

Claim. $B \leq_\gamma (A \cap I^g_{2\mathbb{N}}) \cup B \leq_\gamma A$.

Proof. We show that $B \leq^{\mathcal{P}}_{m} (A \cap I^g_{2\mathbb{N}}) \cup B \leq^{\mathcal{P}}_{m} A$, which will prove the claim. Pick some $a \in 2\mathbb{N} \cap I^g_{2\mathbb{N}+1}$.

Now, define $f(x) = \begin{cases} x & \text{if } x = 2m+1 \text{ some } m \\ a & \text{if } x = 2m \text{ some } m \end{cases}$

Clearly f is polynomial time computable. We consider three cases:

i) $x \in B \Rightarrow x = 2m+1 \Rightarrow f(x) = x \Rightarrow f(x) \in B$
$$\Rightarrow f(x) \in (A \cap I^g_{2\mathbb{N}}) \cup B$$

ii) $x \notin B$ & $x = 2m+1 \Rightarrow f(x) = x = 2m+1$
$$\Rightarrow f(x) \notin B \;\&\; f(x) \in 2\mathbb{N}+1 \Rightarrow f(x) \notin B \;\&\; f(x) \notin A$$
$$\Rightarrow f(x) \notin (A \cap I^g_{2\mathbb{N}}) \cup B$$

iii) $x \notin B$ & $x = 2m \Rightarrow f(x) \in 2\mathbb{N}$ & $f(x) \in I^g_{2\mathbb{N}+1}$
$$\Rightarrow f(x) \notin B \;\&\; f(x) \notin I^g_{2\mathbb{N}} \Rightarrow f(x) \notin (A \cap I^g_{2\mathbb{N}}) \cup B$$

Thus, $B \leq^{\mathcal{P}}_{m} (A \cap I^g_{2\mathbb{N}}) \cup B$ via f.

Further, pick some $b \notin A \oplus B$ and define

$$g(x) = \begin{cases} 2x & \text{if } x = 2m \;\&\; x \in I^g_{2\mathbb{N}} \\ b & \text{if } x = 2m \;\&\; x \in I^g_{2\mathbb{N}+1} \\ 2x+1 & \text{if } x = 2m+1 \end{cases}$$

We consider three cases:

i) $x \in 2\mathbb{N}+1$
So $g(x) = 2x+1$, and so $x \in B \Leftrightarrow g(x) \in A \oplus B$. Since $A \subseteq 2\mathbb{N}$ then $x \in B \Leftrightarrow x \in (A \cap I^g_{2\mathbb{N}}) \cup B$. Thus $x \in (A \cap I^g_{2\mathbb{N}}) \cup B \Leftrightarrow g(x) \in A \oplus B$.

ii) $x \in 2\mathbb{N} \cap I^g_{2\mathbb{N}}$
So $g(x) = 2x$ and thus $x \in A \Leftrightarrow g(x) \in A \oplus B$. Again, as $B \subseteq 2\mathbb{N}+1$, then for $x \in 2\mathbb{N} \cap I^g_{2\mathbb{N}}$, we have $x \in A \Leftrightarrow x \in (A \cap I^g_{2\mathbb{N}}) \cup B$.
So $x \in (A \cap I^g_{2\mathbb{N}}) \cup B \Leftrightarrow g(x) \in A \oplus B$.

iii) $x \in 2\mathbb{N}$ & $x \in I^g_{2\mathbb{N}+1}$
So $g(x) = b$. Clearly $x \notin (A \cap I^g_{2\mathbb{N}}) \cup B$ and $g(x) \notin A \oplus B$.
Thus, $x \in (A \cap I^g_{2\mathbb{N}}) \cup B \Leftrightarrow g(x) \in A \oplus B$ follows immediately.

Thus, $(A \cap I^g_{2\mathbb{N}}) \cup B \leq^{\mathcal{P}}_{m} A \oplus B$ via g. Since $\mathbf{b} < \mathbf{a}$, then $A \equiv^{\mathcal{P}}_{m} A \oplus B$.
Thus $(A \cap I^g_{2\mathbb{N}}) \cup B \leq_\gamma A$ and this completes the proof. □

We can extend the proof of the Density Theorem to show that between any two distinct comparable recursive γ-degrees there exist two incomparable γ-degrees whose supremum is the higher degree. This result can then be further extended to give infinitely many incomparable degrees whose supremum is the higher one. Together with repeated applications of the Density Theorem this shows that, given recursive γ-degrees $\mathbf{b} < \mathbf{a}$, there are infinitely many distinct comparable and incomparable γ-degrees between them.

COMBINED SPLITTING AND DENSITY THEOREM. *Given any two recursive γ-degrees $\mathbf{a}$ and $\mathbf{b}$ with $\mathbf{b} < \mathbf{a}$, there exist recursive γ-degrees $\mathbf{d}_0$ and $\mathbf{d}_1$ such that:*

i) $\mathbf{b} < \mathbf{d}_i < \mathbf{a}$ *for* $i = 0, 1$
ii) $\mathbf{a}$ *is the least upper bound of* $\mathbf{d}_0$ *and* $\mathbf{d}_1$ *i.e.,* $\mathbf{a} = \mathbf{d}_0 \oplus \mathbf{d}_1$.

PROOF. As in the Density Theorem, fix A and B to be in $\mathbf{a}$ and $\mathbf{b}$ respectively, where $A \subseteq 2\mathbb{N}$ and $B \subseteq 2\mathbb{N}+1$. As before, $A \cup B \notin \mathbf{b}$ and $B \notin \mathbf{a}$, so apply the Join Lemma with $C_0 = A$, $C_1 = B$, $\mathcal{C}_0 = \mathbf{b}$ and $\mathcal{C}_1 = \mathbf{a}$.

We will obtain a recursive function, g_0, so let g be any strictly increasing polynomially honest II function that dominates g_0.

Let $D_0 = (A \cap I^g_{2\mathbb{N}}) \cup B$ and $\mathbf{d}_0 = deg_\gamma(D_0)$
$D_1 = (A \cap I^g_{2\mathbb{N}+1}) \cup B$ and $\mathbf{d}_1 = deg_\gamma(D_1)$.

Proof of i). By a similar argument to the Density Theorem we have that $B \leq_\gamma D_i \leq_\gamma A$ for $i = 0, 1$. By the Join Lemma, $D_i \notin \mathbf{a}, \mathbf{b}$, so $\mathbf{b} < \mathbf{d}_0$, $\mathbf{d}_1 < \mathbf{a}$.

Proof of ii). By our choice of A and B, $A \cup B \equiv_\gamma A \oplus B$, so for any set D,

$$(A \cap D) \cup B \equiv_\gamma (A \cap D) \oplus B.$$

Thus we have that $D_0 \equiv_\gamma (A \cap I^g_{2\mathbb{N}}) \oplus B$ and $D_1 \equiv_\gamma (A \cap I^g_{2\mathbb{N}+1}) \oplus B$.

We need to show that $[(A \cap I^g_{2\mathbb{N}}) \oplus B] \oplus [(A \cap I^g_{2\mathbb{N}+1}) \oplus B] \equiv_\gamma A \cup B$.

Pick some $a \notin A \cup B$. Define

$$f(x) = \begin{cases} a & \text{if } x = 2m \ \&\ m = 2n \ \&\ n \in I^g_{2\mathbb{N}+1} \\ a & \text{if } x = 2m+1 \ \&\ m = 2n \ \&\ n \in I^g_{2\mathbb{N}} \\ \left[\frac{\lfloor \frac{x}{2} \rfloor}{2}\right] & \text{otherwise} \end{cases}$$

Clearly f is polynomial time computable as $I^g_{2\mathbb{N}}, I^g_{2\mathbb{N}+1} \in \mathcal{P}$, and by definition, $D_0 \oplus D_1 \leq^{\mathcal{P}}_{m} A \cup B$ via f. Similarly, we can find a function to witness the reverse reduction and thus $D_0 \oplus D_1 \equiv_\gamma A \cup B$. However, $A \cup B \equiv_\gamma A \oplus B$ by our choice of A and B and since $\mathbf{b} < \mathbf{a}$ then we have $A \oplus B \equiv_\gamma A$. Thus we have $D_0 \oplus D_1 \equiv_\gamma A$, i.e., $\mathbf{d}_0 \oplus \mathbf{d}_1 = \mathbf{a}$. □

COROLLARY 7. *Given any two recursive γ-degrees* $\mathbf{a}$ *and* $\mathbf{b}$ *where* $\mathbf{b} < \mathbf{a}$, *and some* $n \in \mathsf{N}$, *there exist* $n+1$ *pairwise incomparable recursive γ-degrees* $\mathbf{d}_i$ *(for* $0 \leq i \leq n$*) such that:*

i) $(\forall i \leq n)[\ \mathbf{b} < \mathbf{d}_i < \mathbf{a}\ ]$,

ii) $\mathbf{a}$ *is the least upper bound of the* $\mathbf{d}_i$.

PROOF. Pick some $n \in \mathsf{N}$. Follow the proof of Theorem 6, and define for all $i < n$ the set $D_i = (A \cap I^g_{n\mathsf{N}+i}) \cup B$.

Now, pick $a \notin 2\mathsf{N} \cap I^g_{n\mathsf{N}+j}$ where $j = (i+1) \bmod n$. Define

$$f_i(x) = \begin{cases} x & \text{if } x = 2m+1 \text{ some } m \\ a & \text{if } x = 2m \text{ some } m \end{cases}$$

Clearly, f_i is polynomial time bounded and, by a similar proof to before, we can see that $(\forall x)[\ x \in B \ \Leftrightarrow\ f_i(x) \in (A \cap I^g_{n\mathsf{N}+i}) \cup B\ ]$. Thus $B \leq_\gamma D_i$ via f_i. Now, pick $b \notin A \oplus B$. Further define

$$g_i(x) = \begin{cases} 2x+1 & \text{if } x = 2m+1 \text{ some } m \\ 2x & \text{if } x = 2m \text{ some } m \text{ and } x \in I^g_{n\mathsf{N}+i} \\ b & \text{if } x = 2m \text{ some } m \text{ and } x \notin I^g_{n\mathsf{N}+i} \end{cases}$$

Again, g_i is polynomial time bounded as $I^g_{n\mathsf{N}+i} \in \mathcal{P}$. Further, $(A \cap I^g_{n\mathsf{N}+i}) \cup B \leq^{\mathcal{P}}_{m} A \oplus B$ via g_i. Thus, $B \leq_\gamma D_i \leq_\gamma A \oplus B$ so, as before, $B <_\gamma D_i <_\gamma A$. Now, by adapting the proof of Theorem 6 in the obvious way, we see that $A \equiv_\gamma D_0 \oplus D_1 \oplus \cdots \oplus D_n$.

So, given any $n \in \mathsf{N}$, there exist $n+1$ incomparable recursive γ-degrees satisfying Theorem 6. □

It is immediate from the Density Theorem that minimal γ-degrees do not exist, and so we turn our attention to the question of minimal pairs of γ-degrees. We first show that minimal pairs do exist, and then show that every non-zero γ-degree below a super sparse II set is half of a minimal pair of γ-degrees.

THEOREM 8. *If* A *is super sparse II, then the γ-degrees of* A *and* $\overline{A}$ *form a minimal pair of γ-degrees.*

PROOF. Let A be a super sparse II set via the function f, and suppose that $B \leq_\gamma A$ via M_{e_0} and $B \leq_\gamma \overline{A}$ via M_{e_1}. Thus:

$$(\forall x)(\forall y)(\forall z)[[\ \langle x,y\rangle \in G(M_{e_0})\ \&\ \langle x,z\rangle \in G(M_{e_1})\ ] \Rightarrow y \neq z\ ] \qquad [*]$$

To test membership of B, use the following algorithm:

i) On input x, compute y such that $\langle x,y\rangle \in G(M_{e_0})$.

ii) If $y \neq 0^{f(n)}$ for any $n \in \mathsf{N}$ then $y \notin A$ and so $x \notin B$. Halt.
Note that to test if "$y \notin Z_f$?", we know that M_{e_0} is non-deterministic polynomial time bounded, so if $\langle x,y\rangle \in G(M_{e_0})$, then $|y| \leq p_{e_0}(|x|)$. Thus, $f(n) \leq p_{e_0}(|x|)$ and, since f is strictly increasing we have $n \leq f(n)$. So to check if $y = 0^{f(n)}$ requires calculating $f(k)$ for $0 \leq k \leq f(n) \leq p_{e_0}(|x|)$.

Thus, we must make a maximum of $p_{e_0}(|x|)$ calculations. Now, f is polynomially honest II, so $f(k)$ can be calculated in less than $r(f(k))$ steps, for some polynomial r.
Since $f(k) \leq f(n) \leq p_{e_0}(|x|)$, then $f(k)$ can be calculated in less than $r(p_{e_0}(|x|))$ steps.
Thus, testing if "$y \notin Z_f$?" can be performed in less than $p_{e_0}(|x|)\cdot r(p_{e_0}(|x|))$ steps.

iii) Otherwise compute z such that $\langle x, z\rangle \in G(M_{e_1})$.

iv) If $z \neq 0^{f(m)}$ for any $m \in \mathbb{N}$ then $z \notin A$ and so $x \in B$. Halt. As in step ii), this can be performed in polynomial time.

v) So now we know that $y = 0^{f(n)}$ and $z = 0^{f(m)}$. By [*], we know that $n \neq m$.

a) *Case 1.* $n < m$.
Now $x \in B \Leftrightarrow y \in A \Leftrightarrow 0^{f(n)} \in A$. Since A is super sparse II then "$0^{f(n)} \in A$?" can be answered in less than $f(n+1)$ steps. f is strictly increasing and $n < m$, so this can be done in less than $f(m) = |z|$ steps. Now M_{e_1} is polynomial time bounded, so for $\langle x, z\rangle \in G(M_{e_1})$ we must have $|z| \leq p_{e_1}(|x|)$. Thus, "$x \in B$?" can be answered in less than $p_{e_1}(|x|)$ steps.

b) *Case 2.* $m < n$.
Now, $x \in B \Leftrightarrow z \in \overline{A} \Leftrightarrow 0^{f(m)} \in \overline{A}$. As above, since A is super sparse II and $m < n$, then "$0^{f(m)} \in A$?" can be answered in less than $f(n) = |y| \leq p_{e_0}(|x|)$ steps. Thus "$x \in \overline{B}$?" can be answered in less than $p_{e_0}(|x|)$ steps, so "$x \in B$?" can too.

Clearly, steps i) and iii) can be performed in non-deterministic polynomial time, and as shown, the other steps can be performed in deterministic polynomial time. Thus $B \in \mathcal{NP}$. Now $B \leq_\gamma A \Leftrightarrow \overline{B} \leq_\gamma \overline{A}$, so by following the above analysis, we see that $\overline{B} \in \mathcal{NP}$ too.

Thus $B \in \mathcal{NP} \cap \text{co-}\mathcal{NP} = \mathbf{0}_\gamma$. □

COROLLARY 9. *If A and $\overline{A}$ form a minimal pair, then for any B such that $\mathbf{0}_\gamma <_\gamma B \leq_\gamma A$, we have that B and $\overline{B}$ form a minimal pair in the γ-degrees. Thus, if A is super sparse II and $B \leq_\gamma A$, then B and $\overline{B}$ form a minimal pair.*

PROOF. Let A and $\overline{A}$ be minimal and $B \leq_\gamma A$ for some $B \notin \mathcal{NP} \cap \text{co-}\mathcal{NP}$.
Suppose $C \leq_\gamma B$ and $C \leq_\gamma \overline{B}$. Then as $\leq_\gamma$ is transitive, we have that $C \leq_\gamma A$ and $C \leq_\gamma \overline{A}$. Since A and $\overline{A}$ are minimal, then $C \in \mathbf{0}_\gamma$.
Thus $deg_\gamma(B)$ and $deg_\gamma(\overline{B})$ are a minimal pair. □

Finally, we show that $\mathcal{D}_\gamma$ is not a lattice. This is achieved by showing that exact pairs of γ-degrees exist. Corollaries to this result will also show that every non-zero γ-degree is one half of a minimal pair and one half of an exact pair.

THEOREM 10. *Let $\mathcal{C}$ be a recursively presentable class of recursive sets and let B be a recursive set. Then there is a recursive set A such that:*

i) $A \notin \mathcal{C}$

ii) $(\forall n)[\ A^{(n)} =^* B^{(n)}\ ]$

iii) $(\forall D)[[\ D \leq_\gamma A\ \&\ D \leq_\gamma B\ ] \Rightarrow (\exists n)[\ D \leq_\gamma B^{(\leq n)}\ ]]$

PROOF. For any set X and number n, we define $X|n = \{x \mid |x| \leq n \;\&\; x \in X\}$. Furthermore, for the purposes of clarity, we will use N to denote $\{0,1\}^*$, so that we have:

$$\mathsf{N}^{(\leq n)} = \{0,1\}^{*(\leq n)} = \{\langle m, x\rangle \mid m \leq n\}$$

Given $\mathcal{C}$ and B, we construct A in stages so that, at stage $s+1$ of the construction, we only add strings of length s to A. Thus, by the end of stage $s+1$ we will have completely determined $A|s$. The construction will be effective, thus ensuring that A is recursive.

In order to satisfy clause i), we will attempt to satisfy, for all e, the requirements:

$$R_{2e} \;:\; A \neq U^{(e)}$$

where U is a universal set for $\mathcal{C}$. This will ensure that $A \notin \mathcal{C}$.

We assume a recursive enumeration, $\{M_i, N_i\}_i$, of pairs of non-deterministic polynomial time bounded Turing transducers where M_i and N_i are time bounded by the polynomial $p_i(n)$. Now, if we have some set C such that $C \leq_\gamma A$ via M_i and $C \leq_\gamma B$ via N_i then $C = M_i(A) = N_i(B)$. Thus, in order to satisfy clause iii), it is sufficient to ensure that we satisfy, for all e, the requirements:

$$R_{2e+1} \;:\; M_e(A) = N_e(B) \;\;\Rightarrow\;\; N_e(B) \leq_\gamma B^{(\leq e)}$$

We will ensure that the construction used to meet these two requirements also satisfies clause ii).

Strategy.

We say that requirement R_n has a *higher priority* than requirement R_m if $n < m$. The action taken at stage $s+1$ of the construction will be designed to satisfy the highest priority requirement, R_n for $n \leq 2s+1$, which has not yet been satisfied at some previous stage and which can be satisfied by appropriately determining membership of A for strings of length s.

We say that R_{2e} is satisfied at stage $s+1$ if $A|s \neq U^{(e)}|s$. Recall that $A|s$ is completely determined by the end of stage $s+1$. If R_{2e} is satisfied for every e, then clause i) will be satisfied.

If R_{2e} is not satisfied at stage $s+1$, then we can satisfy it by setting $A(x) = 1 - U^{(e)}(x)$ for some string x of length s. Clearly, if R_{2e} is satisfied at stage $s+1$, then it will remain so at all later stages.

Note that clause ii) requires that $A^{(n)} =^* B^{(n)}$, so we must ensure that our action to satisfy R_{2e} does not cause this to fail. We do this by insisting that any string x of length s that is added to A to satisfy R_{2e} does not come from $\mathsf{N}^{(\leq e)}$. Then only finitely many strings z can be added to $A^{(n)}$ to make $A^{(n)} \neq B^{(n)}$. This will ensure that clause ii) is satisfied.

We say that R_{2e} *requires attention at stage* $s+1$ if $A|s = U^{(e)}|s$ (so R_{2e} is not yet satisfied) and there is some string $x \notin \mathsf{N}^{(\leq e)}$ such that $|x| = s$. In this case we will be able to satisfy R_{2e} by setting $A(x) = 1 - U^{(e)}(x)$.

In order to satisfy R_{2e+1}, we will try to construct A such that $M_e(A) \neq N_e(B)$. We will show that if we fail to do this then $N_e(B) \leq_\gamma B^{(\leq e)}$ and so the requirement will be satisfied anyway.

If $(\exists x)[\ |M_e(x)| < s\ \&\ A(M_e(x)) \neq B(N_e(x))\ ]$ then R_{2e+1} is already satisfied. Otherwise we say that R_{2e+1} *requires attention via* x *at stage* $s+1$ if $(\exists x)[\ |M_e(x)| = s\ \&\ M_e(x) \notin \mathbf{N}^{(\leq e)}\ ]$.

The problem with this is that to check the above statement requires performing an unbounded search for an appropriate string x. This will cause the construction to be ineffective making A non-recursive. We can avoid this problem by bounding the search by some time bound, $t(n)$, for A. In this case then R_{2e+1} will require attention via x at stage $s+1$ if

$$(\exists x)[\ |x| \leq t(s)\ \&\ |M_e(x)| = s\ \&\ M_e(x) \notin \mathbf{N}^{(\leq e)}\ ] \qquad [*]$$

If this holds then, for all strings y of length s, we can set:

$$A(y) = \begin{cases} B(y) & \text{if } y \neq M_e(x) \\ 1 - B(N_e(x)) & \text{if } y = M_e(x) \end{cases}$$

Then $A(M_e(x)) \neq B(N_e(x))$ so we have $M_e(A) \neq N_e(B)$. Thus R_{2e+1} is satisfied.

We consider two possible cases:

1) $M_e(A) \neq N_e(B)$.
   In this case R_{2e+1} will eventually require attention as we will find some x to witness this difference. Since there are only finitely many requirements of a higher priority and we will satisfy one at every stage, then R_{2e+1} will eventually be satisfied. Clearly, once satisfied R_{2e+1} will never again require attention and will remain satisfied forever.
2) Now there are two possible reasons why R_{2e+1} may not require attention at stage $s+1$. Firstly, R_{2e+1} may already be satisfied, in which case the above holds, or secondly there is no x to act as witness. In this case the search will be unsuccessful at every stage, and so there is some stage s_0 beyond which R_{2e+1} will never require attention.

   Thus $(\forall s \geq s_0)\neg(\exists x)[\ |x| \leq t(s)\ \&\ |M_e(x)| = s\ \&\ M_e(x) \notin \mathbf{N}^{(\leq e)}\ ]$

   i.e., $(\forall s \geq s_0)(\forall x)[\ |x| \leq t(s) \Rightarrow [\ |M_e(x)| \neq s\ \vee\ M_e(x) \in \mathbf{N}^{(\leq e)}\ ]]$.

   So, if this is true, then we need to show that there is a non-deterministic polynomial time bounded Turing transducer T such that $M_e(A) \leq_\gamma B^{(\leq e)}$ via T. Pick some $y_0 \in B^{(\leq e)}$ and some $y_1 \notin B^{(\leq e)}$. We construct T as follows; on input x, compute $M_e(x)$. We consider three cases:

   i) $|M_e(x)| < s_0$.
      Now, $A|s_0$ is finite and fixed by this stage, so information about it can be stored on one of T's worktapes and can be queried in polynomial time. Thus $A(M_e(x))$ can be computed in polynomial time. If $M_e(x) \in A$ then T outputs y_0, and if $M_e(x) \notin A$ then T outputs y_1.
      Then $x \in M_e(A) \Leftrightarrow T(x) = y_0$
      $\Leftrightarrow T(x) \in B^{(\leq e)}$.

   ii) $|M_e(x)| \geq s_0$ and $M_e(x) \in \mathbf{N}^{(\leq e)}$.
      We are assuming that $(\forall n)[\ A^{(n)} =^* B^{(n)}\ ]$, so by careful choice of s_0 we can ensure that for strings of length $\geq s_0$ we have $A^{(\leq e)} = B^{(\leq e)}$.

Now, $A^{(\leq e)}(M_e(x)) = B^{(\leq e)}(M_e(x))$, so we will have $A(M_e(x)) = B^{(\leq e)}(M_e(x))$. Thus, in this case, T outputs $M_e(x)$.
Then $x \in M_e(A) \Leftrightarrow T(x) = M_e(x)$
$\Leftrightarrow T(x) \in B^{(\leq e)}$.

iii) $|M_e(x)| \geq s_0$ and $M_e(x) \notin \mathbf{N}^{(\leq e)}$.
Since R_{2e+1} does not require attention at stage $s = |M_e(x)|$ then we have from [$*$] that
$(\forall y)[\ |M_e(y)| = s \Rightarrow [\ M_e(y) \in \mathbf{N}^{(\leq e)} \ \vee \ |y| > t(s)\ ]]$
In particular, since $M_e(x) \notin \mathbf{N}^{(\leq e)}$ and $|M_e(x)| = s$ we have $|x| > t(s) = t(|M_e(x)|)$. Now, $M_e(x)$ can be computed in less than $p_e(|x|)$ steps, and $A(M_e(x))$ can be computed in less than $t(|M_e(x)|) = t(s) \leq |x|$ steps, so $A(M_e(x))$ can be computed in non-deterministic polynomial time. Now, if $M_e(x) \in A$ then T outputs y_0 and if $M_e(x) \notin A$ then T outputs y_1.
Then $x \in M_e(A) \Leftrightarrow T(x) = y_0$
$\Leftrightarrow T(x) \in B^{(\leq e)}$

It is clear that, on input x, T always outputs some string y, so:

$$(\forall x)(\exists y)[\ \langle x, y\rangle \in G(T)\ ]$$

Furthermore, we have shown that:

$$(\forall x)(\forall y)[\ \langle x, y\rangle \in G(T) \Rightarrow [\ x \in M_e(A) \Leftrightarrow y \in B^{(\leq e)}\ ]]$$

Finally, since T was constructed to run in non-deterministic polynomial time, then $N_e(B) = M_e(A) \leq_\gamma B^{(\leq e)}$ via T as required.

Thus R_{2e+1} is satisfied in both cases.

We will construct the time bounds for A as we proceed through the construction. At any given stage, let the variable *count* record the *total* number of steps made during the whole construction so far. The time bounds will be defined in terms of *count*, where $t_e(s)$ will bound R_{2e+1} in its search at stage s. Then t, the final time bound for A will be defined by $t(n) = \Sigma_{e=0}^{2n+1} t_e(n)$.

The Construction.

Do nothing at stage 0.

Stage $s+1$.

1) For all $e \geq s+1$ let $t_e(s) := count$.
2) **for** $n := s$ **downto** 0

begin

We need to ascertain whether or not R_n requires attention at this stage and if so, what action it requires. Note that we do not perform this action yet, we are merely checking which requirements require attention. In the step 3) we will go back and take action to satisfy the requirement of highest priority that was found in this step.

i) $n = 2e$.
If R_n is not already satisfied then we note that R_n requires attention at stage $s+1$ and that it wants us to set, for all strings y of length s:

$$A(y) = \begin{cases} B(y) & \text{if } y \in \mathsf{N}^{(\leq e)} \\ 1 - U^{(e)}(y) & \text{if } y \notin \mathsf{N}^{(\leq e)} \end{cases}$$

Let $t_n(s) := count$.

ii) $n = 2e + 1$.

Check if R_n is already satisfied, i.e.:

$(\exists x)[\ |x| \leq t_{2e+2}(s-1)\ \&\ |M_e(x)| < s\ \&\ A(M_e(x)) \neq B(N_e(x))\ ]$

if not then check if

$(\exists x)[\ |x| \leq t_{2e+2}(s)\ \&\ |M_e(x)| = s\ \&\ M_e(x) \notin \mathsf{N}^{(\leq e)}\ ]$

If so, then note that R_n requires attention at stage $s+1$ and that it wants us to set, for the least string y that satisfies the above :-

$$A(y) = \begin{cases} B(y) & \text{if } y \neq M_e(x) \\ 1 - B(N_e(x)) & \text{if } y = M_e(x) \end{cases}$$

Let $t_n(s) := count$.

end

3) Act on the requirement of highest priority (assuming that there is one) as decided in step 2).

End of stage $s+1$.

It is clear that if R_n is satisfied at stage s then it remains satisfied at all later stages. Also, at every stage we satisfy the requirement of highest priority that requires attention (assuming that there is one). Thus, every requirement that requires attention will eventually be satisfied as there are only a finite number of requirements of higher priority that need to be satisfied first. It is clear from the construction that A is built so as to meet R_n for all n. Thus the construction works. □

COROLLARY 11. *Let* **b** *and* **c** *be recursive* γ*-degrees such that* $\mathbf{c} < \mathbf{b}$. *Then there exists a recursive* γ*-degree* $\mathbf{a} > \mathbf{c}$ *such that* **c** *is the infimum of* **a** *and* **b**.

Thus, by setting $\mathbf{c} = \mathbf{0}_\gamma$ we have that every non-zero recursive γ-degree is half of a minimal pair.

PROOF. Pick some $\hat{B} \in \mathbf{b}$ and $C \in \mathbf{c}$. Define B by

$$B(\langle n, x\rangle) = \begin{cases} \hat{B}(x) & \text{if } |x| \leq n \\ C(x) & \text{if } |x| > n \end{cases}$$

Clearly B is recursive as "$|x| \leq n$" is a recursive test and $\hat{B}$ and C are recursive sets.

Trivially we have that $B \leq_m^p \hat{B} \oplus C$ via $\Theta(\langle n, x\rangle) = \begin{cases} 2x & \text{if } |x| \leq n \\ 2x+1 & \text{if } |x| > n \end{cases}$

and $\hat{B} \oplus C \leq_m^p B$ via $\Theta(x) = \begin{cases} \langle |m|, m\rangle & \text{if } x = 2m \\ \langle 0, m\rangle & \text{if } x = 2m+1 \end{cases}$

Thus $B \equiv_\gamma \hat{B} \oplus C$ and so $B \in \mathbf{b}$. Furthermore $(\forall n)[\ B^{(n)} =^* C\ ]$ and so we have $(\forall n)[\ B^{(\leq n)} \equiv_\gamma B^{(n)} \equiv_\gamma C\ ]$.

Now, set $\mathcal{C} = \{D \mid D \leq_\gamma C\}$ and apply Theorem 10 with this B. This gives a recursive set A. By the theorem $(\forall n)[\ A^{(n)} =^* B^{(n)}\ ]$ so $(\forall n)[\ C \equiv_\gamma B^{(n)} \equiv_\gamma A^{(n)} \leq_\gamma A\ ]$. Furthermore, $A \notin \mathcal{C}$ by the theorem so $C <_\gamma A$.

Finally let $D \leq_\gamma A$ and $D \leq_\gamma B$. By the theorem $(\exists n)[\ D \leq_\gamma B^{(\leq n)}\ ]$ so $D \leq_\gamma C$ as $(\forall n)[\ C \equiv_\gamma B^{(\leq n)}\ ]$. Thus $\mathbf{a} = deg_\gamma(A)$ has the desired properties.□

DEFINITION 12.

i) A sequence, $\{\mathbf{c}_i\}_{i\in\mathbf{N}}$, of γ-degrees is *ascending* if $(\forall n)[\ \mathbf{c}_n \leq \mathbf{c}_{n+1}\ ]$ and $(\forall n)(\exists m)[\ \mathbf{c}_n < \mathbf{c}_m\ ]$.

ii) A sequence, $\{\mathbf{c}_i\}_{i\in\mathbf{N}}$, of γ-degrees is *recursive* if there is a recursive set, C, such that $\mathbf{c}_n = deg_\gamma(C^{(n)})$.

iii) γ-degrees $\mathbf{a}$ and $\mathbf{b}$ are an *exact pair* for $\{\mathbf{c}_i\}_{i\in\mathbf{N}}$ if $(\forall n)[\ \mathbf{c}_n \leq \mathbf{b},\mathbf{c}\ ]$ and $(\forall \mathbf{d})[\ \mathbf{d} \leq \mathbf{b},\mathbf{c} \Rightarrow (\exists n)[\ \mathbf{d} \leq \mathbf{c}_n\ ]]$.

It is clear from the Density Theorem that recursive ascending sequences of γ-degrees exist between any two comparable recursive γ-degrees. The next lemma will enable us to show that any upper bound for a recursive ascending sequence of γ-degrees is half of an exact pair for that sequence.

LEMMA 13 (Ambos-Spies [2]). *Let* $\mathbf{a}$ *and* $\mathbf{b}$ *be recursive* γ*-degrees. Then the following are equivalent:*

i) $\mathbf{a}$ *and* $\mathbf{b}$ *have no infimum.*

ii) There is an ascending sequence of γ*-degrees for which* $\mathbf{a},\mathbf{b}$ *is an exact pair.*

iii) There is a recursive ascending sequence of γ*-degrees for which* $\mathbf{a},\mathbf{b}$ *is an exact pair.*

PROOF. Clearly iii)⇒ii)⇒i), so it remains to show that i)⇒iii).

Assume $\mathbf{a}$ and $\mathbf{b}$ are recursive γ-degrees without infimum and let $\mathcal{C}$ be defined as $\mathcal{C} = \{D \mid deg_\gamma(D) \leq \mathbf{a}\ \&\ deg_\gamma(D) \leq \mathbf{b}\}$. It is obvious that $\mathcal{C}$ is recursively presentable, so let U be a universal set for $\mathcal{C}$ and let $C_n = U^{(\leq n)}$ and $\mathbf{c}_n = deg_\gamma(C_n)$. The sequence $\{\mathbf{c}_i\}_{i\in\mathbf{N}}$ is obviously recursive and $(\forall n)[\ \mathbf{c}_n \leq \mathbf{c}_{n+1} \leq \mathbf{a},\mathbf{b}\ ]$.

Furthermore, for any $\mathbf{d} \in C = \{\mathbf{c} \mid \mathbf{c} \leq \mathbf{a}\ \&\ \mathbf{c} \leq \mathbf{b}\}$ we have $\mathbf{d} \leq \mathbf{c}_n$ for some n. Thus $\mathbf{a}$ and $\mathbf{b}$ are an exact pair for $\{\mathbf{c}_i\}_{i\in\mathbf{N}}$, and since $\mathbf{a}$ and $\mathbf{b}$ have no infimum then C has no greatest element. Therefore, the sequence $\{\mathbf{c}_i\}_{i\in\mathbf{N}}$ is ascending. □

LEMMA 14. *Let* $\{\mathbf{c}_i\}_{i\in\mathbf{N}}$ *be a recursive ascending sequence of* γ*-degrees such that* $(\forall n)[\ \mathbf{c}_n \leq \mathbf{b}\ ]$ *for some recursive* γ*-degree* $\mathbf{b}$*. Then there exists a recursive* γ*-degree* $\mathbf{a}$ *such that* $\mathbf{a}$ *and* $\mathbf{b}$ *are an exact pair for* $\{\mathbf{c}_i\}_{i\in\mathbf{N}}$.

Thus, every non-zero recursive γ-degree is half of an exact pair.

PROOF. Pick recursive sets C and $\hat{B}$ such that $(\forall n)[\ C^{(n)} \in \mathbf{c}_n\ ]$ and $\hat{B} \in \mathbf{b}$. Define B by

$$B(\langle n, x\rangle) = \begin{cases} \hat{B}(x) & \text{if } |x| \leq n \\ C^{(n)}(x) & \text{if } |x| > n \end{cases}$$

Clearly B is recursive as "$|x| \leq n$" is a recursive test and $\hat{B}$ and C are recursive sets. Since, by definition, $B^{(\leq n)} =^* C_n$, we have $B^{(\leq n)} \in \mathbf{c}_n$. Furthermore, $\hat{B} \leq_m^{\mathcal{P}} B$ via $\Theta(x) = \langle |x|, x \rangle$, so we have that $\hat{B} \leq_\gamma B$.

Now apply Theorem 10 with $\mathcal{C} = \emptyset$. This gives a recursive set A such that $(\forall n)[\ A^{(n)} =^* B^{(n)}\ ]$. Thus $A^{(\leq n)} =^* B^{(\leq n)}$ so $A^{(\leq n)} \in \mathbf{c}_n$ for all n. Also, $(\forall D)[[\ D \leq_\gamma A\ \&\ D \leq_\gamma B\ ]\ \Rightarrow\ (\exists n)[\ D \leq_\gamma B^{(\leq n)}\ ]]$, so for any recursive γ-degree $\mathbf{d} \leq \mathbf{a}, \mathbf{b}$ we have $\mathbf{d} \leq \mathbf{c}_n$ for some n.

Thus $\mathbf{a}$ and $deg_\gamma(B)$ are an exact pair for $\{\mathbf{c}_i\}_{i \in \mathbf{N}}$, and since $\mathbf{b} \leq deg_\gamma(B)$ then $\mathbf{a}$ and $\mathbf{b}$ are also an exact pair for $\{\mathbf{c}_i\}_{i \in \mathbf{N}}$. □

It is clear from the above that every recursive ascending sequence of γ-degrees possesses an exact pair and that, given recursive γ-degrees $\mathbf{c} < \mathbf{b}$, there is some recursive γ-degree $\mathbf{a} > \mathbf{c}$ such that $\mathbf{a}$ and $\mathbf{b}$ have no infimum. Thus $\langle \mathcal{D}_\gamma, \leq_\gamma \rangle$ is not a lattice.

Remarks. There has been much debate, in the unbounded case, about the question of $\leq_r$-degrees within some $\leq_R$-degree, where $\leq_r \Rightarrow \leq_R$. In particular, Zakharov [10] proved that any non-zero $\leq_e$-degree contains at least two $\leq_s$-degrees. He also showed that there are infinitely many $\leq_m$-degrees within any Σ_2 $\leq_e$-degree. This was followed by Watson [8] who showed that any $\leq_e$-degree that is Δ_2 or Σ_2-high contains an infinite number of $\leq_s$-degrees. Finally, we know from Copestake [3] that all 1-generic $\leq_e$-degrees contain infinitely many $\leq_s$-degrees. However, she also showed that there exists a Σ_2 1-generic $\leq_e$-degree that is not in Δ_2, so it is still an open question as to whether there exists a non-zero $\leq_e$-degree that only contains finitely many $\leq_s$-degrees.

It is clear from the Density Theorem that any recursive $\leq_c^{\mathcal{NP}}$-degree that contains at least two distinct $\leq_\gamma$-degrees actually contains infinitely many $\leq_\gamma$-degrees. Since $\leq_c^{\mathcal{NP}}$ is the polynomial time bounded version of $\leq_e$, then this result is clearly of some interest. However, none of the known proofs of Zakharov's theorem (e.g., [10] and [8,9]) carry over to the polynomial time bounded case. We next show that the analogue of this result is actually false and that there are non-zero $\leq_c^{\mathcal{NP}}$-degrees that contain exactly one γ-degree. Thus any $\leq_c^{\mathcal{NP}}$-degree contains either exactly one or infinitely many γ-degrees. This gives an interesting contrast to the unbounded case. It should be noted that the reason that this result works is due to the fact that we can construct sets such that any reduction to them requires only one oracle question. This depends upon the fact that we are working with time bounded reducibilities as this result would obviously contradict Zakharov's theorem if it carried over to the unbounded case. Finally, we note that since $\leq_\gamma \Rightarrow \leq_s^{\mathcal{P}}$ then there are non-zero $\leq_c^{\mathcal{NP}}$-degrees that contain exactly one $\leq_s^{\mathcal{P}}$-degree.

THEOREM 15. $(\exists A \in EXPTIME)[deg_c^{\mathcal{NP}}(A)$ *contains exactly one* γ*-degree.*$]$

PROOF. Let A be the super sparse II set constructed in Lemma 3. Recall that A is super sparse II via the function f defined by:

$$f(0) = 1 \quad \text{and} \quad f(n+1) = 2^{f(n)}$$

and that $A \in \text{EXPTIME}$.

We claim (following [2]) that in any oracle reduction to A, there is only one relevant oracle question, all of the others being redundant.

Recall that $A \subseteq Z_f = \{0^{f(n)} \mid n \in \mathbb{N}\}$ and let $B \leq_r A$ for some (non-deterministic) polynomial time bounded reduction $\leq_r$. We assume that the reduction is bounded by the polynomial $p(n)$. Now, to compute $B(x)$ we will perform some (non-deterministic) computation on x and ask at most $p(|x|)$ oracle questions of A. Clearly, since the reduction is bounded by p then the largest oracle question that can be asked is of length $p(|x|)$, and since we can check in polynomial time whether or not some y is in Z_f then the largest *relevant* oracle question is "$0^{f(n)} \in A$?", where $n = \max\{m \mid f(m) \leq p(|x|)\}$.

Now, given n, to compute $A(0^{f(n)})$ takes less than $2^{|0^{f(n)}|}$ steps, ie less than $2^{f(n)} = f(n+1)$ steps. To compute $A(0^{f(m)})$ for any $m < n$ takes $\leq f(m+1) \leq f(n) \leq p(|x|)$ steps. Thus all oracle questions with the exception of "$0^{f(n)} \in A$?" can be answered in polynomial time.

In other words, the only relevant oracle question is "$0^{f(n)} \in A$?", and so $B \leq_R A$ where $\leq_R$ is a polynomial time bounded reduction procedure that only allows one oracle question.

Thus, given $B \leq_c^{\mathcal{NP}} A$, we can replace the non-deterministic polynomial time bounded tt-condition generator by some other generator, g, that outputs 1-tt conditions, and we can replace the conjunctive evaluator by one, e, defined on $\alpha c\{0,1\}^*$ such that $B \leq_c^{\mathcal{NP}} A$ via g and e.

Now to check "$x \in B$?", we compute $g(x) = \alpha c y$ and then we know that since we are dealing with a conjunctive reducibility, then:

$$\begin{aligned} x \in B \;&\Leftrightarrow\; e(\alpha c C_A(y)) = 1 \\ &\Leftrightarrow\; C_A(y) = 1 \\ &\Leftrightarrow\; y \in A \end{aligned}$$

Now, define a non-deterministic polynomial time bounded Turing transducer, M, such that on input x, M first computes $g(x) = \alpha c y$ and then outputs y. Immediately we have that:

$$(\forall x)(\exists y)[\ \langle x, y\rangle \in G(M)\ ]$$

and

$$(\forall x)(\forall y)[\ \langle x, y\rangle \in G(M) \Rightarrow [x \in B \Leftrightarrow y \in A]]$$

Thus $B \leq_\gamma A$ via M. We have shown that $(\exists A \in \text{EXPTIME})(\forall B)[\ B \leq_c^{\mathcal{NP}} A \ \Rightarrow\ B \leq_\gamma A\ ]$. We now show the complement to this result. Assume that $A \leq_c^{\mathcal{NP}} B$ via a reduction bounded by polynomial q and $B \leq_\gamma A$. This latter fact gives that $B \leq_{1\text{-tt}}^{\mathcal{NP}} A$, so using the well-known characteristic of $\leq_{1\text{-tt}}^{\mathcal{P}/\mathcal{NP}}$ (see [2] for example), we know that there exist non-deterministic polynomial time bounded functions $g : \Sigma^* \mapsto \Sigma^*$ and $h : \Sigma^* \times \{0,1\} \mapsto \{0,1\}$ and a polynomial, p, such that

$$(\forall x)[\ B(x) = h(x, A(g(x)))\ ] \quad \& \quad |g(x)| \leq p(|x|)$$

Define $mf(x) = max\{n \mid f(n) \leq p(q(|x|))\}$ and say that a string y (where $|y| \leq q(|x|)$) is *x-relevant* if $g(y) = 0^{f(mf(x))}$ and $h(y,0) \neq h(y,1)$.

If y is not x-relevant then "$y \in B$?" can be answered in polynomial time as follows:

i) If $h(y,0) = h(y,1)$ then $B(y) = h(y,0)$.

ii) If $g(y) \notin \{0^{f(0)}, \ldots, 0^{f(mf(x))}\}$ then $g(y) \notin A$ so $B(y) = h(y,0)$.

iii) If $g(y) \in \{0^{f(0)}, \ldots, 0^{f(mf(x)-1)}\}$ then $B(y) = h(y, A(g(y)))$ where $A(g(y))$ can be computed in $\leq p(q(|x|))$ steps. This fact follows since A is super sparse II, so $A(0^{f(m)})$ can be computed in $\leq f(m+1)$ steps. Thus $A(g(y))$ can be computed in $\leq f(mf(x))$ steps. By definition of $mf(x)$ this is $\leq p(q(|x|))$ steps.

So, assuming that we have strings y and y', both x-relevant oracle questions, we note that $B(y) = h(y, A(0^{f(mf(x))}))$, $h(y,0) \neq h(y,1)$ and $B(y') = h(y', A(0^{f(mf(x))}))$.

Clearly, if we are given $B(y)$, then we can compute

$$A(0^{f(mf(x))}) = \begin{cases} 0 & \text{if } h(y,0) = B(y) \\ 1 & \text{if } h(y,1) = B(y) \end{cases}$$

in non-deterministic polynomial time, and consequently we can compute $B(y') = h(y', A(0^{f(mf(x))}))$ in non-deterministic polynomial time.

Thus, given the answer to any one oracle question, we can efficiently compute the answer to all others. Thus $A \leq_{1\text{-}tt}^{\mathcal{NP}} B$ and so we can find non-deterministic polynomial time bounded functions g_1 and h_1 such that $A(x) = h_1(x, B(g_1(x)))$, and a machine M, bounded by polynomial p_1, such that $B \leq_\gamma A$ via M.

Thus, for all x, $B(x) = A(M(x))$, so $A(x) = h_1(x, A(M(g_1(x))))$. [*]

Define $mf'(x) = max\{n \mid f(n) \leq p_1(|x|)\}$ and fix $x_0 \notin B$ and $x_1 \in B$. Construct a non-deterministic machine as follows; on input x:

i) If $h_1(x,0) = h_1(x,1)$ then output $x_{h(x,0)}$ and halt.

ii) If $M(g_1(x)) \neq 0^{f(mf'(x))}$ then

   a) if $h_1(x,i) = i$ output $x_{A(M(g_1(x)))}$ and halt;

   b) if $h_1(x,i) = 1-i$ output $x_{1-A(M(g_1(x)))}$ and halt.

iii) If $M(g_1(x)) = 0^{f(mf'(x))}$ then

   a) if $x \neq M(g_1(x))$ then output $x_{A(x)}$ and halt;

   b) if $x = M(g_1(x))$ then output $g_1(x)$ and halt.

Note that in step iii) b), we have the case where $x = M(g_1(x))$, so by [*] we have that $h_1(x,i) = i$ and so $A(x) = B(g_1(x))$. The machine clearly runs in non-deterministic polynomial time as all steps are polynomially bounded (note that $A(M(g_1(x)))$ and $A(x)$ can be computed in $\leq p_1(|x|)$ steps as A is super sparse II), and from [*] it follows immediately that $A \leq_\gamma B$ via this machine.

Thus $(\forall B)[\ A \leq_c^{\mathcal{NP}} B \ \Rightarrow\ A \leq_\gamma B\ ]$ and so $deg_\gamma(A) = deg_c^{\mathcal{NP}}(A)$ as required. □

REFERENCES

[1] LEONARD ADLEMAN and K. MANDERS, *Reducibility, randomness and intractibility*, ***Proc. 9th Annual ACM Symposium on the Theory of Computing***, 1977, pp. 151–163.

[2] KLAUS AMBOS-SPIES, *On the structure of the polynomial time degrees of recursive sets*, Habilitationsschrift, Forschungsbericht Nr. 206, Universität Dortmund, 1985.

[3] KATE COPESTAKE, *Enumeration degrees of* Σ_2 *sets*, Ph.D. thesis, Leeds University, 1987.

[4] RICHARD LADNER, NANCY LYNCH and ALAN L. SELMAN, *A comparison of polynomial time reducibilities*, ***Theoretical Computer Science***, **1** (1975), pp. 103–123.

[5] TIMOTHY J. LONG, *On* γ*-reducibility versus polynomial time many-one reducibility*, ***Theoretical Computer Science***, **14** (1981), pp. 91–101.

[6] TIMOTHY J. LONG, *Strong non-deterministic polynomial-time reducibilities*, ***Theoretical Computer Science***, **21** (1982), pp. 1–25.

[7] ALAN P. SILVER, *On polynomial time bounded reducibilities*, Ph.D. thesis, Leeds University, May 1991.

[8] PHIL WATSON, *Concerning D-recursively enumerable sets and s-reducibility of* Σ_2 *Sets*, Ph.D. thesis, Leeds University, 1988.

[9] PHIL WATSON, *On restricted forms of enumeration reducibility*, ***Annals of Pure and Applied Logic***, **49** (1990), pp. 75–96.

[10] S. D. ZAKHAROV, *e- and s-degrees*, ***Algebra and Logic***, **23** (1984), pp. 273–281.