Low, Superlow, and Superduperlow Sets: An Exposition of a Known But Not Well-Known Result

William Gasarch
University of Maryland

1 Introduction

We use the following standard notation.

Notation 1.1.

1. $M_0, M_1, \ldots$ is be a standard list of Turing Machines.
2. W_e is the domain of M_e. Hence $W_0, W_1, \ldots$ is a list of all c.e. sets.
3. $M_0^{()}, M_1^{()}, \ldots$ is a standard list of oracle Turing Machines.
4. K is the set $\{e : M_e(e)\downarrow\}$.
5. If A is a set then $A' = \{e : M_e^A(e)\downarrow\}$.

We use the following definitions. *Low* and *Superlow* are standard; however, *Superduperlow* seems to be a new term.

Def 1.2.

1. A set A is *Low* if $A' \le_{\mathrm{T}} K$.
2. A set A is *Superlow* if $A' \le_{\mathrm{tt}} K$.
3. A set A is *Superduperlow* if $A' \le_{\mathrm{btt}} K$.

By a finite injury priority argument one can construct a noncomputable superlow c.e. set A (we include this proof in the appendix). This raises the question: Is there a noncomputable superduperlow set A?

We asked about this at a logic conference and found out:

1. Four prominent computability theorists thought that there was no such set; however, none knew of a proof or reference.

2. Carl Jockusch, also a prominent computability theorist, knew of unpublished proofs by: (1) Bickford and Mills, (2) Phillips, (3) himself, and (4) Stephan. He also knew of a more complicated published proof by Mohrerr [5]. She actually proved the stronger result that if $A^{\rm tt} \le_{\rm btt} B'$ then $A \le_{\rm T} B$, as did Bickford and Mills.

It is our opinion that the proofs of Jockusch and Stephan are beautiful and should be better known. Hence we present them.

We will need the following two standard Lemmas. The first one is *the Shoenfield Limit Lemma.*

Lemma 1.3. *$A \le_{\rm T} K$ iff there exists a computable function $h : \mathsf{N} \times \mathsf{N} \rightarrow \mathsf{N}$ such that*

$$A(x) = \lim_{s \rightarrow \infty} h(x, s).$$

Lemma 1.4. *If $A \le_{\rm T} B$ then $A \le_{\rm m} B'$.*

2 Bi-immune and Hyperimmune Sets

Def 2.1.

1. A set C is *immune* if C is infinite and has no infinite c.e. subsets

2. A set C is *bi-immune* if both C and $\overline{C}$ are immune.

3. If B is an infinite set then *the principal function of B*, denoted p_B, is defined as
$$p_B(s) = \text{ the } s^{\rm th} \text{ element of } B.$$

4. A function f is *majorized* by a function g if, for all x, $f(x) < g(x)$.

5. A set B is *hyperimmune* if p_B is not majorized by any computable function. (There is an alternative definition of hyperimmune which is a variant of immune, hence the name. We do not need that alternative definition.)

The following Lemma is due to Miller and Martin [4]. We include the proof since the original article is behind a paywall.

Lemma 2.2. *If $\emptyset <_{\rm T} A \le_{\rm T} K$ then there exists a hyperimmune set B such that $B \equiv_{\rm T} A$. (We will only use the $B \le_{\rm T} A$ part.)*

Proof. Since $A \leq_{\mathrm{T}} K$ there exists, by Lemma 1.3, a computable h such that

$$A(x) = \lim_{s\to\infty} h(x,s).$$

Let

$$f(x) = \text{ the least } s \geq \max\{x, f(x-1)\} \text{ such that } (\forall y \leq x)[A(y) = h(y,s)].$$

Let $s = f(x)$. Then, for all $y \leq x$, $h(y,s) = A(y)$. One might think that the limit has settled down on all $y \leq x$; however, it may be that there is some $y \leq x$ and $t > s$ such that $h(y,t) \neq A(y)$. If so then there will be some $t' > t$ with $h(y,t') = A(y)$. And $h(y,-)$ may even change its mind again! But eventually it will always be $A(y)$.

Let B be the image of f. Clearly $B \leq_{\mathrm{T}} A$. We show B is hyperimmune. Since $f = p_B$ is the principal function of B, it suffices to show that f cannot be majorized by any computable function.

Assume, by way of contradiction, that there is a computable g such that, for all x, $f(x) < g(x)$. We use this to obtain an algorithm for A. Given x, we want to determine $A(x)$.

Note that, for all y,

$$y \leq f(y) < g(y)$$

Let $y \geq x$. Imagine what would happen if

$$h(x,y) = h(x,y+1) = \cdots = h(x,f(y)) = \cdots = h(x,g(y)) = b.$$

We would have:

$$A(x) = h(x,f(y)) = b.$$

Hence we would know $A(x)$. Therefore we need to find such a y. If we knew that one existed we could just look for it.

One does exist! Let y be such that

$$h(x,y) = h(x,y+1) = \cdots =$$

Such a y exists since h reaches a limit. This y clearly suffices. We cannot find this particular y but we do not need to. We need only find *some* y such that

$$h(x,y) = h(x,y+1) = \cdots = h(x,g(y)) = b.$$

Here is the formal algorithm for A.

1. Input(x)
2. Find a $y \geq x$ such that
$$h(x, y) = h(x, y+1) = \cdots = h(x, g(y)) = b$$
3. Output the value b.

Thus A is computable— a contradiction. Hence f cannot be majorized by any computable function.

Therefore we have a set $B \leq_{\mathrm{T}} A$ such that B is hyperimmune.

We leave the proof that $A \leq_{\mathrm{T}} A$ to the reader.

$\square$

The following is a result of Carl Jockusch [2].

Lemma 2.3. *For every hyperimmune B there exists a bi-immune $C \leq_{\mathrm{T}} B$.*

Proof. Let B be hyperimmune. Since B is hyperimmune, p_B is not majorized by any computable function. We use this to construct a bi-immune $C \leq_{\mathrm{T}} B$. To ensure that C is bi-immune we make sure that C satisfies the following requirements:

$$R_e : W_e \text{ infinite } \rightarrow (W_e \cap C \neq \emptyset \wedge W_e \cap \overline{C} \neq \emptyset).$$

CONSTRUCTION
Stage 0: For all e, R_e is not satisfied.

Stage s: Find the least $e \leq s$, if it exists, such that R_e is not satisfied and $W_{e,p_B(s)}$ has at least two elements $x_1, x_2 \geq s$ which have not yet been put into C or $\overline{C}$. Put x_1 into C, x_2 into $\overline{C}$, and declare R_e *satisfied.* (it will never become unsatisfied). We also say that R_e has *acted.* If there is no such e, do nothing.
END OF CONSTRUCTION

We have $C \leq_{\mathrm{T}} B$ since (1) the only noncomputable part of the construction is computing $p_B \leq_{\mathrm{T}} B$, and (2) $C(n)$ is decided by stage n . (For definiteness, a number is in C iff the construction puts it into C.)

We show that C is bi-immune by showing that it satisfies each requirement. We assume that $R_1, \ldots, R_{e-1}$ are satisfied and show that R_e is satisfied. There are two cases.

1. W_e is finite. Then clearly R_e is satisfied.

2. W_e is infinite. Assume, by way of contradiction, that W_e is not satisfied. From this we will construct a computable function g that majorizes f which will be the contradiction. Let s_0 be such that by state s_0 all of $R_1, \ldots, R_{e-1}$ that are going to act have acted. So for all $s \geq s_0$ R_e is not satisfied yet fails to act! Why!?

   Let $g(s)$ be the least $t \geq s_0$ such that $W_{e,t}$ has at least $2s+2$ elements $\geq s$. Note that g is computable. Lets look at the construction at a stage $s \geq s_0$. R_e did not act. Why? It must be that *there is no* x_1, x_2 such that (1) $x_1, x_2 \in W_{e,p_B(s)}$, (2) $x_1, x_2 \geq s$, and (3) x_1, x_2 were not used by any of $P_0, \ldots, P_{e-1}$. By the definition of g *there is* x_1, x_2 such that (1) $x_1, x_2 \in W_{e,g(s)}$, (2) $x_1, x_2 \geq s$, and (3) x_1, x_2 were not used by any of $P_0, \ldots, P_{e-1}$. Hence we have that,

$$(\forall s \geq s_0)[p_B(s) < g(s)].$$

   Recall that g is computable. We want to say g *majorizes* p_B but this is not quite true; however, a finite variant of g majorizes p_B and is clearly computable. Hence there is a computable function that majorizes p_B. This contradicts B being hyperimmune.

□

Lemma 2.4. *Let $\emptyset \leq_{\mathrm{T}} A \leq_{\mathrm{T}} K$.*

1. *If A is not computable then there exists C bi-immune such that $C \leq_{\mathrm{T}} A$.*

2. *If there is no bi-immune set $C \leq_{\mathrm{T}} A$ then A is computable (this is the contrapositive of Part 1).*

Proof. 1) By Lemma 2.2 there is a hyperimmune set $B \leq_{\mathrm{T}} A$. By Lemma 2.3 there is a bi-immune set $C \leq_{\mathrm{T}} B$. Hence there is a bi-immune set $C \leq_{\mathrm{T}} B \leq_{\mathrm{T}} A$. □

3 Superduperlow implies Decidable: Proof One

Def 3.1. Let $n \geq 0$. A set D is *weakly n-c.e.* if there exists a function h such that

- $D(x) = \lim_{s \to \infty} h(x,s)$
- $|\{s : h(x,s) \neq h(x,s+1)\}| \leq n$.

Note 3.2. Let D be weakly n-c.e. and $x \in \mathsf{N}$. Our view: initially x thinks $D(x) = h(x,0)$; however, it can change its mind $\le n$ times.

The following easy lemma we leave to the reader.

Lemma 3.3. *If $\emptyset <_{\rm T} D \le_{\rm btt} K$ then there exists an $n \ge 1$ such that D is n-c.e. but not $(n-1)$-c.e.*

The following is an unpublished Theorem of Jockusch.

Theorem 3.4. *If A is superduperlow then A is decidable.*

Proof. Since A is superduperlow $A' \le_{\rm btt} K$.

Let $D \le_{\rm T} A$. We will show that D is not bi-immune then apply Lemma 2.4 to deduce that A is computable. If D is decidable then D is not bi-immune; hence we assume D is undecidable.

Since $D \le_{\rm T} A$, by Lemma 1.4, $D \le_{\rm m} A'$. Since $A' \le_{\rm btt} K$ we have

$$D \le_{\rm m} A' \le_{\rm btt} K.$$

Hence $D \le_{\rm btt} K$. Since D is not decidable, by Lemma 3.3, there exists $n \ge 1$ such that D is weakly n-c.e. but not weakly $(n-1)$-c.e. Let h be such that

- $D(x) = \lim_{s\to\infty} h(x,s)$
- $|\{s : h(x,s) \ne h(x,s+1)| \le n$.

Let

$$E = \{x : |\{s : h(x,s) \ne h(x,s+1)| = n\}\}.$$

E is infinite, else D is weakly $(n-1)$-c.e. Let

$$E_0 = \{x : h(x,0) = 0 \wedge |\{s : h(x,s) \ne h(x,s+1)| = n\}\}.$$

$$E_1 = \{x : h(x,0) = 1 \wedge |\{s : h(x,s) \ne h(x,s+1)| = n\}\}.$$

Clearly both E_0 and E_1 are c.e: Clearly at least one of E_0 or E_1 is infinite. There are four cases depending on (1) which of E_0, E_1 is infinite, and (2) the parity of n. For all four cases keep in mind that E_0 and E_1 are c.e.

Case E_0 infinite, n even: Every $x \in E_0$ starts out thinking it's not in D and changes its mind an even number of times. Hence E_0 is an infinite c.e. subset of $\overline{D}$, so D is not bi-immune.

Case E_0 infinite, n odd: Every $x \in E_0$ starts out thinking it's not in D and changes its mind an odd number of times. Hence E_0 is an infinite c.e. subset of D, so D is not bi-immune.
Case E_1 infinite, n even: Every $x \in E_1$ starts out thinking it's in D and changes its mind an even number of times. Hence E_1 is an infinite c.e. subset of D, so D is not bi-immune.
Case E_1 infinite, n odd: Every $x \in E_1$ starts out thinking it's in D and changes its mind an odd number of times. Hence E_1 is an infinite c.e. subset of $\overline{D}$, so D is not bi-immune.

The upshot is that, for *every* set $D \leq_{\mathrm{T}} A$, D is not bi-immune. By Lemma 2.4, A is computable. □

4 Superduperlow implies Decidable: Proof Two

In this section we present a proof by Frank Stephan that uses concepts from Bounded Queries in Computability Theory. We will provide all that you need; however, for more information, see the survey by Gasarch [1].

Def 4.1. Let $A \subseteq \mathsf{N}$ and $n \in \mathsf{N}$.

1. $\chi_n^A : \mathsf{N}^n \rightarrow \{0,1\}^n$ is the following function:

$$\chi_n^A(x_1, \ldots, x_n) = A(x_1) \cdots A(x_n).$$

2. $\#_n^A : \mathsf{N}^n \rightarrow \mathsf{N}$ is the following function:

$$\#_n^A(x_1, \ldots, x_n) = |\{x_1, \ldots, x_n\} \cap A|.$$

We give an intuition for the next definition. Given 3 numbers a, b, c we want to know $\chi_3^K(a,b,c)$. We could just output (0,0,0), (0,0,1), …, (1,1,1) and be happy that *one of them is* $\chi_3^K(a,b,c)$. Can we output ≤ 7 tuples, one of which is $\chi_3^K(a,b,c)$? Yes if we are willing to not know when the process has output its last candidate[1]. We can do the following: Output (0,0,0) since it is certainly possible that $\chi_3^K(a,b,c) = (0,0,0)$. Then run $M_a(a)$, $M_b(b)$, and $M_c(c)$ at the same time until (if it happens) one of them halts: (1) if it's a then output $(1,0,0)$, (2) if it's b then output $(0,1,0)$, (3) if it's c then output $(0,0,1)$. Then run all those that haven't halted until (if it happens) one of them halts. We leave it to the reader to finish this up. Throughout this process you will output at most 4 tuples *one of which is* $\chi_3^K(a,b,c)$. We do not know when the process has output its last candidate. This motivates the next definition:

[1]It is known that there is no algorithm that will, on input (a,b,c), always output 7 3-tuples one of which is $\chi_3^K(a,b,c)$. The proof uses the recursion theorem.

Def 4.2. A function f is in $\mathrm{EN}(m)$ if there exists a Turing machine M that will, on input x, over time, output at most m numbers, one of which is $f(x)$.

Clearly, for all A, $\#_n^A \in \mathrm{EN}(n+1)$. Kummer [3] (see Gasarch [1] for an alternative proof) showed that, for undecidable sets, this is the best one can do.

Theorem 4.3. *For all A, if there exists n such that $\#_n^A \in \mathrm{EN}(n)$, then A is computable.*

We use Theorem 4.3 to show that all superduperlow sets are decidable.
We need a known lemma. We give the proof for completeness.

Lemma 4.4. $\chi_n^K \in \mathrm{EN}(n+1)$.

Proof. On input $(x_1, \ldots, x_n)$:

1. Output $(0, \ldots, 0)$ as a possible answer. We call $(0, \ldots, 0)$ *the current tuple.*
2. Run $M_{x_1}(x_1), \ldots, M_{x_n}(x_n)$ at the same time. If the ith one halts then change the ith bit of the current tuple to 1, output the new current tuple, and keep running the machines until another one halts (which may never happen).

If j of them halt then this process will output $j+1$ tuples, of which the last one is $\chi_n^K(x_1, \ldots, x_n)$. Hence this process outputs at most $n+1$ tuples, one of which is $\chi_n^K(x_1, \ldots, x_n)$. $\square$

Theorem 4.5. *If A is superduperlow then A is decidable.*

Proof. Assume that A is superduperlow. Let k be such that $A' \le_{\text{k-tt}} K$. We show that, for some (large enough) n, $\#_{2^n-1}^A \in \mathrm{EN}(2^n-1)$, hence A is decidable.

Let $A_1, \ldots, A_n$ be the following sets.

$$A_i = \{(x_1, \ldots, x_{2^n-1}) : \text{ the } i\text{th bit of } \#_{2^n-1}^A(x_1, \ldots, x_n) \text{ is } 1\ \}.$$

For each i, $A_i \le_{\mathrm{T}} A$ (actually $A_i \le_{\mathrm{tt}} A$ but we do not need this). Hence, by Lemma 1.4, $A_i \le_{\mathrm{m}} A'$. Since $A' \le_{\text{k-tt}} K$ we get $A_i \le_{\text{k-tt}} K$. We use this to obtain a procedure for

$$\#_{2^n-1}^A \in \mathrm{EN}(kn+1).$$

1. Input $(x_1, \ldots, x_{2^n-1})$.
2. For $1 \le i \le n$ do the following: Using $A_i \le_{\text{k-tt}} K$ find k numbers $y_{i1}, \ldots, y_{ik}$ such that if we knew $\chi_{ik}^K(y_{i1}, \ldots, y_{ik})$ then we would know if $(x_1, \ldots, x_{2^n-1})$ is in A_i.

3. We now have kn numbers $(y_{11}, \ldots, y_{1k}, y_{21}, \ldots, y_{2k}, \ldots, y_{n1}, \ldots, y_{nk})$ such that if we knew $\chi^K_{kn}(y_{11}, \ldots, y_{1k}, y_{21}, \ldots, y_{2k}, \ldots, y_{n1}, \ldots, y_{nk})$ we would, for $1 \le i \le n$, know if $(x_1, \ldots, x_{2^n-1}) \in A_i$.

4. By Lemma 4.4 $\chi^K_{kn} \in \mathrm{EN}(kn+1)$. Run this enumeration algorithm on $(y_{11}, \ldots, y_{1k}, y_{21}, \ldots, y_{2k}, \ldots, y_{n1}, \ldots, y_{nk})$. Every time a candidate is enumerated, use it to obtain a candidate for $\#^A_{2^n-1}$. Output that candidate and continue.

By the above enumeration algorithm $\#^A_{2^n-1} \in \mathrm{EN}(kn+1)$. Take n large enough so that $kn+1 \le 2^n - 1$ to obtain that $\#^A_{2^n-1} \in \mathrm{EN}(2^n-1)$. By Theorem 4.3 A is computable. $\square$

5 Acknowledgment

We thank Carl Jockusch and Frank Stephan for supplying most of the material for this paper and for helpful discussions.

A There exists an undecidable c.e. Superlow Set

We give the standard construction of a noncomputable low c.e. set A; however, we analyze the construction carefully to show that A is actually superlow.

Theorem A.1. *There exists an undecidable c.e. superlow set.*

Proof. We construct a c.e. set A that satisfies the following requirements:

$P_e : W_e \text{ infinite } \implies W_e \cap A \neq \emptyset.$

These are called *positive requirements* since they act by *putting numbers into A*. It is easy to show that, if A is co-infinite and all of the P_e's are satisfied, then A is undecidable. (We will also make A co-infinite, though we do not state it as a formal requirement.)

$N_e : (\exists^\infty s)[M^{A_s}_{e,s}(e)\downarrow] \implies M^A_e(e)\downarrow.$

These are called *negative requirements* since they act by *keeping numbers out of A*. We will show that, if all N_e are satisfied, then A is superlow. These requirements protect a computation from being injured. By this we mean that, if $M^{A_s}_{e,s}(e)\downarrow$ then this requirement will try to make sure that no numbers enter A that might make this computation diverge (N_e will try to keep *all* numbers $\le$ the max number queried from going into A). Associated to every N_e will be a restraint function $r(e,s)$. This is N_e saying *you cannot put an element into A* *that is* $\le r(e,s)$. This restraint will be respected by the lower priority positive requirements (P_e, P_{e+1}, etc.) but not by the higher priority positive requirements (P_0, P_1, …, P_{e-1}).

The requirements are in the following priority ordering

$$N_0, P_0, N_1, P_1, \ldots$$

CONSTRUCTION
Stage 0: $A_0 = \emptyset$. $(\forall e)[r(e,0) = 0]$. For all e, P_e is not satisfied.

Stage s: Visit each requirement in turn, via the priority ordering, up to P_s.
Case 1: A positive requirement P_e. If (a) P_e is not satisfied, and (b) there exists $x \in W_{e,s}$ such that $x \ge 2e$ and $x > \max_{i \le e} r(e,s)$, then P_e acts by putting x into A. P_e is declared satisfied. We say that N_i is *injured.* Note that P_e will never become unsatisfied.
Case 2: A negative requirement N_e. If $M_{e,s}^{A_s}(e)\downarrow$ then set $r(e,s)$ to be the largest number that is queried in this computation. Note that if no number ever enters A that is $\le r(e,s)$ then $M_e^A(e)\downarrow$.
END OF CONSTRUCTION

Claim 1: Every P_e acts finitely often.
Proof of Claim 1: If P_e never acts then clearly P_e acts finitely often.

If P_e ever acts then it is satisfied, will never be injured, and will never acts again. Hence P_e acts finitely often.
End of Proof of Claim 1:

Claim 2: For all e, N_e is satisfied and $\lim_{s\to\infty} r(e,s) < \infty$.
Proof of Claim 2: Let s_0 be such that, for all $i < e$, P_e will never act past stage s_0. Note that s_0 exists by Claim 1.

If $(\forall^\infty s)[M_{e,s}^{A_s}(e)\uparrow]$ then N_e is satisfied since its premise is false. Past some point N_e will never act, hence its restraint changes only finitely often, so
$\lim_{s\to\infty} r(e,s) < \infty$.

If $(\exists^\infty s)[M_{e,s}^{A_s}(e)\downarrow]$ then there is some $s_1 \ge s_0$ such that $M_{e,s_1}^{A_s}(e)\downarrow$. When that happens N_e will act and a restraint $r(e,s_1)$ will be set. Since no higher priority positive requirement ever acts, N_e is never injured, and hence is satisfied. Since N_e never acts again, $\lim_{s\to\infty} r(e,s) = r(e,s_1) < \infty$.
End of Proof of Claim 2:

Claim 3: Every P_e is satisfied.
Proof of Claim 3: If W_e is finite then P_e is satisfied. Hence we assume that W_e is infinite. Let s_0 be such that for all $i < e$ $\lim_{s\to\infty} r(i,s) = r(i,s_0)$. Let

$R(e) = \max_{i \le e} r(i, s_0)$. Since W_e is infinite there will be an $x > \max\{2e, R(e)\}$ that is enumerated into W_e at some stage $s > \max\{s_0, e\}$. If P_e is not yet satisfied then P_e will act at stage s and be satisfied.
End of Proof of Claim 3:

Claim 4: A is co-infinite.
Proof of Claim 4: Look at the numbers $S_e = \{1, 2, \ldots, 2e, 2e+1\}$. Since P_{e+1} only uses numbers $\ge 2e+2$, the only positive requirements that will use elements of S_e are $P_0, \ldots, P_e$. Hence at most $e+1$ of the elements of S_e will enter A. Hence at least e of the elements of S_e will not enter A. Since this is true for all e, A is co-infinite.
End of Proof of Claim 4:

Claim 5: A is superlow.
Proof of Claim 5:

Note that the only requirements that can injure N_e are $P_0, P_1, \ldots, P_{e-1}$. These requirements act at most once. Hence N_e is injured at most e times. We can determine $e \in A'$ by asking the following questions at the same time (which is why we get $A' \le_{\rm tt} K$). For each $i \le e$ we have the following two questions to K.

- Is N_e injured at least i times?
- Is there a stage s that occurs after N_e is injured i times where $M_{e,s}^{A_s}(e)\downarrow$?

From the answers (1) determine the largest i_0 such that N_e is injured exactly i_0 times, and (2) determine if there is some stage s after the i_0 injuries such that $M_{e,s}^{A_s}(e)\downarrow$. If the answer to (2) is YES then $e \in A'$ since that computation will never be injured. If the answer is NO then $e \notin A'$ since for almost all s, $M_{e,s}^{A_s}(e)\uparrow$. □

References

[1] W. Gasarch. Gems in the field of bounded queries. In Cooper and Goncharov, editors, *Computability and Models*, 2003. `http://www.cs.umd.edu/~gasarch/papers/papers.html`.

[2] C. Jockusch. The degrees of bi-immune sets. *Zeitschrift für logik and Grundlagen d. Math (Has changed its name to Math Logic Quarterly)*, 15, 1986, pp.135–140. The article is behind a paywall.

[3] M. Kummer. A proof of Beigel's cardinality conjecture. *Journal of Symbolic Logic*, 57(2), June 1992, pp.677–681. `http://www.jstor.org/action/showPublication?journalCode=jsymboliclogic`.

[4] W. Miller and D. A. Martin. The degree of hyperimmune sets. *Zeitschrift für logik and Grundlagen d. Math (Has changed its name to Math Logic Quarterly)*, 14, 1968, pp. 159–166. The article is behind a paywall.

[5] J. Mohrherr. A refinement of low$_n$ and high$_n$ for the r.e. degrees. *Zeitschrift für logik and Grundlagen d. Math (Has changed its name to Math Logic Quarterly)*, 32, 1986, pp.5–12 The article is behind a paywall.