Learning via Queries and Oracles

Frank Stephan *
Universität Karlsruhe

Abstract

Inductive inference considers two types of queries: Queries to a teacher about the function to be learned and queries to a non-recursive oracle. This paper combines these two types — it considers three basic models of queries to a teacher, namely QEX[Succ], QEX[<] and QEX[+], together with membership queries to some oracle.

The results for these three models of query-inference are very similar: If an oracle is already omniscient for query-inference, then it is already omniscient for EX. There is an oracle of trivial EX-degree, which allows nontrivial query-inference. Furthermore, queries to a teacher can not overcome differences between oracles and the query-inference degrees are a proper refinement of the EX-degrees.

1 Introduction

One famous example of learning via queries to a teacher is the game Mastermind. The teacher first selects the code – a quadruple of colours – that should be learned. Then the learner tries to figure out the code. In each round, the learner makes one guess what the code might be and the teacher answers, how many colours and positions of the guess are correct. The learner succeeds if the correct code is found after a given number of rounds. The learner may in addition consult an oracle, e.g., a book or a database. This book is of course ignorant of the current code to be learned since it was pressed before the teacher elected the code. But nevertheless the book may be helpful if it contain an algorithm of how to generate the queries. The power of this help depends on the quality of the algorithm. In this paper both phenomena, the teacher and the oracle, are combined and the power of the oracles is measured w.r.t. different concepts of learning from a teacher.

Angluin [2] introduced learning via queries to a teacher, who knows the function (or in her case: regular set) to be learned. Gasarch and Smith [8] considered logical queries about functions as $(\exists x, y)[f(x) = 0 \wedge f(y) = 1]$ and $(\exists x)(\forall y)[f(x) = f(y) \Rightarrow x = y]$. In addition to some logical standard symbols, the logical queries may use extra symbols to denote arithmetic operations or relations on the variables and numerical constants; the power of the query language mainly depends on these symbols. Gasarch and Smith [8] looked at the symbols Succ, $<$ and $+$; they found out that these facilities increase the learning power. But it is impossible to infer the set REC of all recursive functions, so the learning power can still be increased by additional concepts.

Gasarch and Pleszkoch [6] brought the notion of queries to an oracle into the field of learning theory. The machine may only make membership queries to the oracle, i.e., it may ask whether a given number x is in A or not in A. Membership queries are more restricted than logical queries, but in the case of oracles they are more suitable since they enforce that the learning power of the various oracles inside a Turing degree is the same. Furthermore, almost everywhere in the literature, oracles are accessed via membership queries.

Fortnow et al. [5, Section 7] first mentioned the combined concept with queries to a teacher and to an oracle, i.e., the IIM M may ask either the teacher a query about f or make a membership query to an oracle A. QEX[Succ; A], QEX[$<$; A] and QEX[$+$; A] denote these combined classes of inference; they satisfy the inclusions EX[A] $\subseteq$ QEX[Succ; A] $\subseteq$ QEX[$<$; A] $\subseteq$ QEX[$+$; A].

The topic of this paper is, how much the oracle helps at a given query language. Gasarch was particularly

*Institut für Logik, Komplexität und Deduktionssysteme, Universität Karlsruhe, 76128 Karlsruhe, Germany, (Email: `fstephan@ira.uka.de`). Supported by the Deutsche Forschungsgemeinschaft (DFG) grant Me 672/4-2.

interested when an oracle is omniscient, i.e., allows to infer all recursive functions, and when it is trivial, i.e., the learner can learn without the oracle the same and the oracle therefore provides no help for the learner. Fortnow et al. [5] treat this question for many models of learning but left it open for models where queries of *both* types were allowed. The results depend of course on the query languages, e.g., the query-language $L[+,\times]$ already codes a K-oracle and therefore $\mathrm{QEX}[+,\times;A]$ is omniscient for every oracle A. In this paper, the omniscient degree of the three query-languages $L[\mathrm{Succ}]$, $L[<]$ and $L[+]$ is characterized and partial results on inclusion structure and the trivial degree are obtained.

2 Preliminaries

The set of all natural numbers is denoted by ω. The set of all finite sequences of natural numbers is ω^*. The Greek letters σ and τ denote elements of ω^*, the Latin letters f, g and h denote functions. Some functions can be identified with infinite strings, e.g., $\sigma 0^\infty$ is the function which takes the value $\sigma(x)$ for $x < |\sigma|$ and 0 otherwise.

REC is the set of all total and recursive functions and $\mathrm{REC}_{0,1} = \{f \in \mathrm{REC} : (\forall x)[f(x) \leq 1]\}$. A set A is Turing reducible to B ($A \leq_T B$) if A can be computed via a machine which knows B, i.e., which has an infinite database which supplies for each x the information whether x belongs to B or not. Such a database is called an oracle and the query whether $x \in B$ a membership query to B. The class $\{A : A \equiv_T B\}$ is called the Turing degree of B where $A \equiv_T B$ means that both, $A \leq_T B$ and $B \leq_T A$ hold. Given two sets A and B, the Turing degree of the join $A \oplus B = \{2x : x \in A\} \cup \{2x+1 : x \in B\}$ is the least upper bound of the Turing degrees of A and B. K denotes the halting problem, i.e., the set $\{x : \varphi_x(x)\downarrow\}$. This notion can be relativized to $A' = \{x : \varphi_x^A(x)\downarrow\}$ is the halting problem relative to A where φ_x^A is the x-th recursive function equiped with an oracle A. A set A is called high if $K' \leq_T A'$, i.e., if the halting problem relative to K can be solved using the halting problem relative to A. Further explanations on recursion theoretic and its notation can be found in the books of Odifreddi [13] and Soare [16].

A further recursion theoretic notion, which is important in the context of this paper, is that of the 1-generic sets which are those that either meet or strongly avoid each recursive set of strings: If A is 1-generic and W is a recursive set of strings, then there is either some $\sigma \in W$ with $\sigma \preceq A$ ("A meets W") or there is some $\sigma \notin W$ with $\sigma \preceq A$ and $\tau \notin W$ for all $\tau \succeq \sigma$ ("A strongly avoids W"). In this definition, "recursive" may be replaced by "r.e." without changing the concept. The interested reader may find more information on 1-generic sets in Jockusch's paper [10] or Soare's book [16, A.VI.3.6-9].

Gold [9] introduced the first inference criterion EX: He called a total recursive function $M : \omega^* \to \omega$ an inductive inference machine (IIM) for a set $S \subseteq \mathrm{REC}$, if M EX-*infers* each $f \in S$, i.e., if M converges on each $f \in S$ to an index e for f: $\varphi_e = f$ and $M(\sigma) = e$ for almost all $\sigma \preceq f$. Osherson, Stob and Weinstein [14] give an overview on this and many other inference criteria.

M $\mathrm{EX}[A]$-*infers* a function f, if M during the inference process may make membership queries to A. $\mathrm{EX}[A]$ denotes the class of all sets of recursive functions which are $\mathrm{EX}[A]$-learnable via some IIM. The oracles are ordered by the ability to infer classes of functions, so if $\mathrm{EX}[A] \subseteq \mathrm{EX}[B]$ then the oracle A supports more powerful inference processes as B. This relation induces an ordering on the oracles and $\{B : \mathrm{EX}[B] = \mathrm{EX}[A]\}$ is called the *inference degree* of an oracle A.

While Turing degrees are a measure for the power of oracles relative to computation, inference degrees are a measure for the power of oracles relative to learning classes of recursive functions. It is easy to see that if $A \leq_T B$ then $\mathrm{EX}[A] \subseteq \mathrm{EX}[B]$: A can be computed relative to B and an IIM having B-oracle can simulate these computations during the inference process. Therefore if A is recursive then $\mathrm{EX}[A] \subseteq \mathrm{EX}[B]$ for all oracles B and there is a least inference degree, called the *trivial inference degree*, which contains all recursive sets; indeed $\mathrm{EX}[A] = \mathrm{EX}$ for all oracles in this trivial degree. Slaman and Solovay [15] gave the following characterization of the trivial degree:

Fact 2.1 [15] *A has trivial* EX*-degree iff either A is recursive or A has the Turing degree of a 1-generic set $B \leq_T K$.*

Obviously if an IIM $\mathrm{EX}[A]$-infers the whole class REC then A belongs to the greatest inference degree. This degree is called the *omniscient inference degree*. Adleman and Blum [1] showed that for the criterion EX there exists an omniscient inference degree and that it has the following easy characterization:

Fact 2.2 [1] *A has omniscient* EX*-degree iff A is high.*

A *query language* is a language that has the usual logical symbols (equality, constants, variables, quantifiers and logical operations to link the atom) and a special symbol f. Query languages may contain further symbols, in particular Succ, $<$ and $+$ which denote the successor-function $\mathrm{Succ}(x) = x+1$ on ω, the less-than relation $<$ on ω and the addition $+$ on ω. These three languages – with one of the additional symbols – are denoted by $L[\mathrm{Succ}]$, $L[<]$ and $L[+]$. A *query* in the language is a formula without free variables; such formulas without free variables are called sentences. The intention is that a query $\phi(f)$ is asking about f. For example in the language $L[<]$ the query

$$(\exists x)(\forall y)\,[y < x \vee f(y) = 0]$$

is asking whether at some point the function becomes the constant 0. $S \in \mathrm{QEX}[\star]$ means, that some machine infers S using the query-language $L[\star]$ where $\star$ denotes in this paper — in contrast to other papers on this field — one of the symbols Succ, $<$, $+$. Note that by introducing a new variable and one quantifier, Succ and $<$ can be expressed by $<$ and $+$, respectively:

$$\begin{aligned} x = \mathrm{Succ}(y) \quad &\Leftrightarrow \quad y < x \wedge (\forall z)\,[z < y \vee \\ &\qquad z = y \vee z = x \vee z > x]; \\ x < y \quad &\Leftrightarrow \quad (\exists z)\,[y = x + z + 1]. \end{aligned}$$

Indeed EX $\subset$ QEX[Succ] $\subset$ QEX[$<$, Succ] = QEX[$<$] $\subset$ QEX[$+,<$] = QEX[$+$] holds.

This paper deals with combinations of both concepts. So a QEX[$\star; A$] IIM queries either the teacher in the language $L[\star]$ about f or makes a membership query to the oracle A. But the IIM can not make combined queries, e.g., $(\exists x)\,[f(x) = 0 \wedge x \in A]$ is not permitted. So it is possible to define inference degrees for the criteria QEX[$\star$] = QEX[Succ], QEX[$<$], QEX[$+$]. The inference degree of A under the criterion QEX[$\star$] is

$$\{B : \mathrm{QEX}[\star; B] = \mathrm{QEX}[\star; A]\}.$$

There exists a trivial QEX[$\star$] and an omniscient QEX[$\star$] degree. So it is the main topic of this paper, to look at the inclusion structure of the QEX[$\star$]-degrees:

omniscient degree: The omniscient QEX[$\star$]-degree

$$\{A : \mathrm{REC} \in \mathrm{QEX}[\star; A]\}$$

consists of all high oracles, i.e., REC $\in$ QEX[$\star; A$] iff $A' \geq_T K'$.

trivial degree: The trivial QEX[$\star$]-degree

$$\{A : \mathrm{QEX}[\star; A] = \mathrm{QEX}[\star]\}$$

contains all recursive sets, but is a proper subset of the trivial EX-degree $\{A : \mathrm{EX}[A] = \mathrm{EX}\}$.

inclusion structure: The QEX[$\star$]-degrees are a refinement of the EX-degrees: if EX[A] $\not\subseteq$ EX[B] then also QEX[$\star; A$] $\not\subseteq$ QEX[$\star; B$] but not the other way round. There are fewer QEX[$\star$] inclusions than EX inclusions.

complete families: There are complete families S and S' which witness the following two non-inclusions for every non-high oracle A:

$$S \in \mathrm{QEX}[+] \Leftrightarrow \mathrm{QEX}[<; A]$$

and

$$S' \in \mathrm{QEX}[<] \Leftrightarrow \mathrm{QEX}[\mathrm{Succ}; A].$$

These results require the following two important facts – the first one of them due to Gasarch, Pleszkoch and Solovay [7, Theorem 11]; the second one can be obtained via a modification of their proof.

Fact 2.3 [7, Theorem 11] *There are recursive functions k_+ and $truth_+$ such that for every sentence $\phi \in L[+]$ and every $\{0,1\}$-valued function f: If f is $k_+(n)$-good and $n \geq |\phi|$ then*

$$\phi(f) \;\Leftrightarrow\; truth_+(n, \phi, f(0), f(1), \ldots, f(n))$$

where f is $k_+(n)$-good iff

$$\begin{array}{ll} (\text{I}) & (\exists^\infty x)\,[f(x) = 1]; \\ (\text{II}) & (\forall x > n)\,[f(x) = 1 \\ & \quad\Rightarrow\; (k_+(n))!\ divides\ x]; \\ (\text{III}) & (\forall x > n)(\forall y > x)\,[f(x) = f(y) = 1 \\ & \quad\Rightarrow x \cdot k_+(n) \leq y]. \end{array}$$

The formula $y!$ denotes the product $1 \cdot 2 \cdot \ldots \cdot y$.

Gasarch, Pleszkoch and Solovay stated Fact 2.3 a little bit differently; they computed for each $\sigma = f(0)\, f(1) \ldots f(n)$ and for each sentence ϕ a value $k(\phi, \sigma)$ depending on ϕ and σ instead of n. So the function k_+ used here is derived from their function k by the formula $k_+(n) = \max\{k(\phi,\sigma) : |\phi| \leq n \wedge |\sigma| \leq n+1\}$ where ϕ ranges over all sentences in $L[+]$ and σ over all strings in $\{0,1\}^*$. It is easy to see that every $k_+(n)$-good function f is also $k(\phi,\sigma)$-good for all sentences ϕ and strings σ with $|\phi| \leq n$ and $|\sigma| \leq n+1$; the formulation of Fact 2.3 given above is more suitable for this paper.

Fact 2.4 *There are recursive functions $k_<$ and $truth_<$ such that for any sentence $\phi \in L[<]$ and any $\{0,1\}$-valued function f: If f is $k_<(n)$-good and $n \geq |\phi|$ then*

$$\phi(f) \;\Leftrightarrow\; truth_<(n, \phi, f(0), f(1), \ldots, f(n))$$

where f is $k_<(n)$-good iff

$$\begin{array}{ll} (\text{IV}) & (\exists^\infty x)\,[f(x) = 1]; \\ (\text{V}) & (\forall x > n)\,[f(x) = 1 \Rightarrow k_<(n) \leq x]; \\ (\text{VI}) & (\forall x > n)(\forall y > x)\,[f(x) = f(y) = 1 \\ & \quad\Rightarrow x + k_<(n) \leq y]. \end{array}$$

For ease of argumentation, the functions k_+ and $k_<$ satisfy w.l.o.g. the following: k_+ and $k_<$ are increasing functions, i.e., $k_+(n) < k_+(n+1)$ for all n. Furthermore, $n < k_<(n) \leq k_+(n)$ for all n, i.e., k_+ majorizes $k_<$.

3 The Omniscient Degree

Fortnow et al. [5] left open the problem to determine the omniscient inference degree of the criteria QEX[Succ], QEX[$<$] and QEX[$+$]. Since all EX-omniscient oracles are QEX[$\star$]-omniscient, only the other direction is interesting: whether there are QEX[$\star$]-omniscient oracles, which are not EX-omniscient. The following theorem gives a negative answer: if EX[A] $\not\subseteq$ EX[B] then also QEX[$\star; A$] $\not\subseteq$ QEX[$\star; B$]. Thus the QEX[$\star$]-degrees are a refinement of the EX-degrees and their omniscient degree is a subset of the omniscient EX-degree; indeed these omniscient degrees are equal.

Theorem 3.1 EX[A] $\not\subseteq$ EX[B] $\Rightarrow$ EX[A] $\not\subseteq$ QEX[$+; B$].

Proof: The basic idea of this proof is to define a recursive transformation , such that for every $S \subseteq \mathrm{REC}_{0,1}$ the transformed set $\tilde{S} = \{, (g) : g \in S\}$ has the following properties:

$$\begin{array}{ll} (1) & \tilde{S} \subseteq \mathrm{REC}_{0,1}. \\ (2) & S \in \mathrm{EX}[A] \Rightarrow \tilde{S} \in \mathrm{EX}[A]. \\ (3) & S \notin \mathrm{EX}[B] \Rightarrow \tilde{S} \notin EX[B]. \\ (4) & \tilde{S} \notin EX[B] \Rightarrow \tilde{S} \notin \mathrm{QEX}[+;B]. \end{array}$$

Choosing $S \in \mathrm{EX}[A] \Leftrightarrow \mathrm{EX}[B]$ gives directly a witness $\tilde{S} \in \mathrm{EX}[A] \Leftrightarrow \mathrm{QEX}[+;B]$.

The transformation makes use of Fact 2.3 in the way that on one hand there is an increasing recursive sequence $a_0, a_1, \ldots$ such that the transformed function , (g) is $k_+(a_n)$-good for all n and on the other hand , (g) codes g in a very easy way so that (2) and (3) can be guaranteed. The sequence $a_0, a_1, \ldots$ and the transformation , are defined as follows:

$$\begin{array}{rcl} a_0 & = & 0; \\ a_{n+1} & = & (k_+(a_n))!; \\ f(x) & = & , (g)(x) \\ & = & \begin{cases} g(m) & \text{if } (\exists m)\,[x = a_{2m}]; \\ 1 & \text{if } (\exists m)\,[x = a_{2m+1}]; \\ 0 & \text{otherwise.} \end{cases} \end{array}$$

Let $S \subseteq \mathrm{REC}_{0,1}$. Obviously $\tilde{S} \subseteq \mathrm{REC}_{0,1}$ and (1) holds. (2) follows since g can be recovered from $f = , (g)$ by $g(m) = f(a_{2m})$: Let M witness $S \in \mathrm{EX}[A]$. The new IIM N to witness $\tilde{S} \in \mathrm{EX}[A]$ computes on input $f(0), f(1), ..., f(a_{2m})$ the values $g(0), g(1), ..., g(m)$, then N simulates M to compute $e = M(g(0), g(1), \ldots, g(m))$ and finally N outputs an index $h(e)$ for , (φ_e); such a recursive function h exists since the sequence $a_0, a_1, \ldots$ is recursive. It is easy to see that N converges on , (g) to $h(e)$ iff M converges on g to e; thus if M infers S then N infers $\tilde{S}$. So (2) holds. Similarly one can show that there is also an inverse translation of the inductive inference machines and (3) follows by contraposition. It remains to show (4):

First it is shown that every $f = , (g)$ is $k_+(a_n)$-good for all n: Since $f(a_{2m+1}) = 1$ for all m, (i) holds. Furthermore, if $x > a_n$ and $f(x) = 1$, then $x = (k_+(a_m))!$ for some $m \geq n$ and $(k_+(a_n))!$ divides x since $a_m \geq a_n$ and $k_+(a_m) \geq k_+(a_n)$. Thus (ii) holds. For $m \geq n$, $a_{m+1} = (k_+(a_m))! \geq k_+(a_m) \cdot (k_+(a_m) \Leftrightarrow 1) \geq k_+(a_m) \cdot a_m \geq k_+(a_n) \cdot a_m$, thus also (iii) is satisfied. So f is $k_+(a_n)$-good for all n and since $a_{|\phi|} \geq |\phi|$, the relation

$$\phi(f) \Leftrightarrow truth_+(a_{|\phi|}, \phi, f(0), f(1), \ldots, f(a_{|\phi|}))$$

enables to answer any query ϕ to f by analizing the prefix $f(0)f(1)\ldots f(a_{|\phi|})$ of f. So the $\mathrm{QEX}[+;B]$ inference algorithm can be translated into an $\mathrm{EX}[B]$ algorithm, that is, $\tilde{S} \in \mathrm{QEX}[+;B] \Rightarrow \tilde{S} \in \mathrm{EX}[B]$ and $\tilde{S} \notin \mathrm{EX}[B] \Rightarrow \tilde{S} \notin \mathrm{QEX}[+;B]$. This finishes the proof of (4).

If $\mathrm{EX}[A] \not\subseteq \mathrm{EX}[B]$, then this is witnessed by some $S \subseteq \mathrm{REC}_{0,1}$. By (2) and (3), also $\tilde{S} \in \mathrm{EX}[A] \Leftrightarrow \mathrm{EX}[B]$. From (4) it follows that $\tilde{S} \notin \mathrm{QEX}[+;B]$. Therefore the non-inclusion $\mathrm{EX}[A] \not\subseteq \mathrm{QEX}[+;B]$ holds. ▮

The inclusions $\mathrm{EX}[C] \subseteq \mathrm{QEX}[\mathrm{Succ};C] \subseteq \mathrm{QEX}[<;C] \subseteq \mathrm{QEX}[+;C]$ hold for all oracles C. Therefore every non-inclusion in the EX-degrees induces a non-inclusion in the $\mathrm{QEX}[\star]$-degrees. Furthermore the omniscient degree of the criteria EX, QEX[Succ], QEX[<] and QEX[+] coincides:

Theorem 3.2 *If* $\mathrm{EX}[A] \not\subseteq \mathrm{EX}[B]$ *then*
- $\mathrm{QEX}[+;A] \not\subseteq \mathrm{QEX}[+;B]$*;*
- $\mathrm{QEX}[<;A] \not\subseteq \mathrm{QEX}[<;B]$*;*
- $\mathrm{QEX}[\mathrm{Succ};A] \not\subseteq \mathrm{QEX}[\mathrm{Succ};B]$*.*

Furthermore $\mathrm{QEX}[+;A]$, $\mathrm{QEX}[<;A]$ *and* $\mathrm{QEX}[\mathrm{Succ};A]$ *are omniscient iff* A *is high.*

4 The Trivial Degree

To characterize the trivial $\mathrm{QEX}[\star]$-degrees still is an open problem, but a new partial result is obtained by showing that the trivial degrees of $\mathrm{QEX}[\star]$ and EX are different. The result is based on the idea of coding some nonrecursive r.e. set B uniformly into the graph of each $f \in S$ for some $S \notin \mathrm{QEX}[\star;B]$. Since the IIM can recover B with queries about f, it can also $\mathrm{QEX}[\star;A]$ infer S whenever $A \oplus B \equiv_T K$. There is even an 1-generic set $A \leq_T K$ such that $A \oplus B \equiv_T K$. This A has nontrivial $\mathrm{QEX}[\star]$-degree but trivial EX-degree. On the other hand Theorem 3.2 implies that if A has trivial $\mathrm{QEX}[\star]$-degree then A has also trivial EX-degree.

Again the basic idea of the construction is defining a transformation , ^{B} depending on an r.e. non-high and non-recursive oracle B. Starting with the set $\mathrm{REC}_{0,1}$ of all recursive $\{0,1\}$-valued functions, the transformed set S defined as $\{, ^B(g) : g \in \mathrm{REC}_{0,1}\}$ has the following properties:

$$\begin{array}{ll} (1) & S \subseteq \mathrm{REC}_{0,1}. \\ (2) & A \oplus B \geq_T K \Rightarrow S \in \mathrm{QEX}[\star;A]. \\ (3) & S \notin EX[B]. \\ (4) & S \notin \mathrm{QEX}[\star;B] \text{ and } S \notin \mathrm{QEX}[\star]. \end{array}$$

Posner and Robinson [12] showed

Fact 4.1 *Given any nonrecursive set* $B \leq_T K$ *there is an 1-generic set* $A \leq_T K$ *such that* $A \oplus B \equiv_T K$*.*

Using this 1-generic set A, it will come out that $A \oplus B \equiv_T K$ and therefore $S \in \mathrm{QEX}[\star;A]$ but on the other hand $S \notin \mathrm{QEX}[\star]$, so A does not have trivial $\mathrm{QEX}[\star]$-degree, but A has trivial EX-degree.

Now the proof in detail. Lemma 4.2 is the first step in obtaining such a set B and transformation , ^{B}. B is chosen as some kind of deficiency set of non-high nonrecursive degree; h is an auxiliary function used below. This special form of B makes it easier to define $,^B$ in the proofs of Theorems 4.3 and 4.4.

Lemma 4.2 *In every r.e. Turing degree there is an r.e. set B and a recursive function h such that*
- $B = \{n : (\exists m > n)\,[h(m) \le n]\}$ *and*
- $(\forall n)\,[h(n) \le n]$.

Proof: Let f be a recursive 1-1 enumeration of some given r.e. set C. For each s define

$$\begin{aligned} g(s) &= \text{the } f(s)\text{-th element of } \overline{B_s};\\ h(s) &= \min\{g(s), s\};\\ B_s &= \{n \le s : (\exists m > n)\\ &\qquad [m \le s \wedge h(m) \le n]\}.\end{aligned}$$

$B = \bigcup_s B_s$ satisfies the first condition, the second follows from the definition of h. The third follows from the choice of C and the fact, that $B \equiv_T C$:

$B \le_T C$ since B is permitted by C. Any new element n enters B at stage s only if $h(s) \le n < s$ and $h(s)$ is the $f(s)$-th element of $\overline{B_s}$, i.e., n is permitted by the new element $f(s)$ of C with $f(s) \le h(s) \le n$.

$\overline{B}$ is infinite, since for any c there is some s with $f(n) \ge c$ for all $n \ge s$. Since B_s is finite, $\overline{B_s}$ is infinite and its least c elements do never enter B.

$C \le_T B$ since $c \in C \Leftrightarrow c \in \{f(0), f(1), \ldots, f(s+c+1)\}$ where s is the first stage such that the $c+1$ least elements of $\overline{B}$ and $\overline{B_s}$ are identical. If $c \notin C$ then also $c \notin \{f(0), f(1), \ldots, f(s+c+1)\}$ since f is an enumeration of C. Otherwise $c \in C$ and there is some t with $c = f(t)$. Let d be the $c+1$-st element of $\overline{B}$ and $\overline{B_s}$. Since B_s contains only elements below s, $d \le s + c + 1$. If $t > s + c + 1$ then $h(t) = g(t) < d < t$ and d would be enumerated to B in contradiction to the choice of d. Thus $t \le s + c + 1$. ▮

Using Lemma 4.2 and Theorems 4.3 and 4.4, it is possible to show that there is a set A of nontrivial query inference degree but of trivial EX-degree. Theorem 4.3 has a full proof for the case of $L[+]$ while Theorem 4.4 deals with the remaining cases $L[\mathrm{Succ}]$ and $L[<]$ and its proof shows only the differences to the proof of Theorem 4.3.

Theorem 4.3 *There is an 1-generic set $A \le_T K$ such that* $\mathrm{QEX}[+;A] \not\subseteq \mathrm{QEX}[+]$.

Proof: Let B and h be as in Lemma 4.2 and let B have non-high and nonrecursive Turing degree. Further let k be as in Theorem 3.1 and let A be as in Fact 4.1. The modification from $,$ to $,^B$ is more implicit by modifying the sequence $a_0, a_1, \ldots$ from Theorem 3.1 while $,^B$ is analog to $,$:

$$\begin{aligned} a_0 &= 0;\\ a_{n+1} &= \begin{cases} (k_+(a_n))! + a_{2h(m)} \\ \quad \text{if } n = 2m \wedge h(m) < m; \\ (k_+(a_n))! \\ \quad \text{otherwise.} \end{cases}\\ f(x) &= ,^B(g)(x) \\ &= \begin{cases} g(m) & \text{if } (\exists m)\,[x = a_{2m}]; \\ 1 & \text{if } (\exists m)\,[x = a_{2m+1}]; \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

The basic idea of the construction is that the queries in $L[+]$ to f – independent of g – have the same Turing-degree as B. For each sentence ϕ there is some $m > |\phi|$ such that $m \notin B$. For these $m \notin B$ (but not for all m), f is $k_+(a_{2m})$-good and $\phi(f)$ is equivalent to $\mathit{truth}_+(a_{2m}, \phi, f(0), f(1), f(2), \ldots, f(a_{2m}))$. On the other hand every membership query to B can be answered via logical queries about any $f = ,^B(g)$.

The item (1) of the four items mentioned before Fact 4.1 follows directly from the construction, so it remains to show the other three items for proving the theorem:

(2): Let $A \oplus B \ge_T K$. Then $S \in \mathrm{EX}[A \oplus B]$ since K is an omniscient oracle. Any membership query to B can be calculated via logical queries about $f \in S$; these queries use only the fact that $f \in S$ but not which particular function f is:

$$\begin{aligned} & m \in B \\ \Leftrightarrow\ & (\exists n > m)\,[h(n) \le m] \\ \Leftrightarrow\ & (\exists n > m)\,[a_{2n+1} \not\equiv 0 \text{ modulo } a_{2m+1}] \\ \Leftrightarrow\ & (\exists x > a_{2m+1})\,[f(x) = 1 \text{ and} \\ & \quad x \not\equiv 0 \text{ modulo } a_{2m+1}] \\ \Leftrightarrow\ & (\exists x, y, z)\,[f(x + a_{2m+1} + 1) = 1 \wedge \\ & \quad 0 < y < a_{2m+1} \wedge \\ & \quad x + a_{2m+1} + 1 = z \cdot a_{2m+1} + y]. \end{aligned}$$

Multiplications with constants can be expressed in the language $L[+]$, e.g., $5 \cdot x$ as $x + x + x + x + x$. The constant a_{2m+1} can be computed from m. Since a_{2m+1} is a constant, the relation $0 < y < a_{2m+1}$ can be replaced by a disjunction over equalites, i.e. $y = 1 \vee y = 2 \vee \ldots \vee y = (a_{2m+1} \Leftrightarrow 1)$. Thus the whole query is effectively equivalent to a sentence in $L[+]$ and the $\mathrm{EX}[A \oplus B]$ inference process can be simulated by a $\mathrm{QEX}[+;A]$ inference machine. Therefore $S \in \mathrm{QEX}[+;A]$.

(3): Assume that $S \in \mathrm{EX}[B]$ via M. Then a new $\mathrm{EX}[B]$-IIM N translates the input function g to $f = ,^B(g)$, computes $e = M(f(0), f(1), \ldots, f(a_{2m}))$ and outputs an index $u(e)$ for $\varphi_{u(e)}(x) = \varphi_e(a_{2x})$. It is easy to see that N witnesses $\mathrm{REC}_{0,1} \in \mathrm{EX}[B]$ since $,^B(g) \in S$ for every $g \in \mathrm{REC}_{0,1}$. Then B has to be high in contradiction to the choice of B, thus $S \notin \mathrm{EX}[B]$.

(4): This part is based on the following observation: If $m \notin B$ then $,(g)$ is $k_+(a_{2m+1})$-good. Since $f(a_n) = 1$ for all odd n, f takes infinitely often the value 1 and satisfies (I). Since $h(n) > m$ for all $n > m$, all $n \ge 2m + 1$ either satisfy $a_{n+1} = (k_+(a_n))!$ or $a_{n+1} = (k_+(a_n))! + a_{2l}!$ for some $l > m$; note that $a_{2l} = (k_+(a_{2l-1}))!$. In both cases $(k_+(a_{2m+1}))!$ divides a_{n+1} and (II) holds. Furthermore, $a_{n+1} \geq k_+(a_n))! \geq (k_+(a_{2m+1}) \cdot a_n$ for all $n \geq 2m+1$, thus also (III) holds.

Therefore for any sentence $\phi \in L[+]$ there is $m \notin B$ such that $|\phi| \leq a_{2m+1}$. Since $f = \, , ^B(g)$ is $k_+(a_{2m+1})$-good,

$$\phi(f) \Leftrightarrow \mathit{truth}_+(a_{2m+1}, \phi, f(0), \ldots, f(a_{2m+1}))$$

and the query can be answered from $f(0), \ldots, f(a_{2m+1})$. Thus $S \in$ QEX[+] $\Rightarrow S \in$ EX[B]. Note that the oracle B can not be removed since it provides the information to find an index $2m+1$ such that f is $k_+(a_{2m+1})$-good.

So all 4 items hold. By Fact 4.1 it is possible to select a 1-generic set $A \leq_T K$ such that $A \oplus B \equiv_T K$. Then $S \in$ QEX[+; A] $\Leftrightarrow$QEX[+] and A does not have trivial QEX[+]-degree. ▮

Theorem 4.4 *There is an 1-generic set $A \leq_T K$ such that* QEX[Succ; A] $\not\subseteq$ QEX[<].

Proof: This proof is similar to that of Theorem 4.3. The sets A and B as well as the function h are the same, but it is necessary to use Fact 2.4 instead of Fact 2.3 and to adapt $, ^B$.

$$\begin{array}{rcl} a_0 & = & 0; \\ a_{n+1} & = & \begin{cases} a_n + k_<(a_{3h(m)+1}) \\ \quad \text{if } (\exists m)[n = 3m+1]; \\ a_n + k_<(a_n) \\ \quad \text{otherwise.} \end{cases} \\ f(x) & = & , ^B(g)(x) \\ & = & \begin{cases} g(m) & \text{if } (\exists m)\,[x = a_{3m}]; \\ 1 & \text{if } (\exists m)\,[x = a_{3m+1}]; \\ 1 & \text{if } (\exists m)\,[x = a_{3m+2}]; \\ 0 & \text{otherwise.} \end{cases} \end{array}$$

Again f is $k_<(a_{3m})$-good whenever $m \notin B$. So any $L[<]$-query to f can be answered using B-oracle and a sufficiently long prefix of f. On the other hand

$$\begin{array}{l} m \in B \;\Leftrightarrow\; (\exists x > a_{3m+1})\,[f(x) = 1 \;\wedge \\ \quad (f(x+k_<(a_0)) = 1 \vee f(x+k_<(a_1)) = 1 \;\vee \\ \quad\; f(x+k_<(a_2)) = 1 \vee \ldots \vee f(x+k_<(a_{3m+2})) = 1)\,]. \end{array}$$

The values $k_<(a_i)$ are constants, thus each term $x + k_<(a_i)$ can be expressed by using sufficiently often the term Succ, e.g., $x+3$ equals Succ(Succ(Succ(x))). So B can be recovered via queries in L[Succ] to f. The rest of the proof follows the lines of Theorem 4.3. ▮

The following Theorem summarizes the results of this section:

Theorem 4.5 *There is a set A such that*
- EX[A] = EX*;*
- QEX[+; A] $\not\subseteq$ QEX[+]*;*
- QEX[<; A] $\not\subseteq$ QEX[<]*;*
- QEX[Succ; A] $\not\subseteq$ QEX[Succ].

5 Complete Families

The notion of a complete family S is normally used with respect to some notion of reduction such that all families inferable under a given criterion are reducible to S. The complete families considered here are not complete w.r.t. an inference criterion but w.r.t. a non-inclusion, i.e., a family S is called complete w.r.t. the non-inclusion QEX[+] $\not\subseteq$ QEX[<] iff $S \in$ QEX[+] and $S \notin$ QEX[<; A] for every oracle A of non-omniscient QEX[<]-degree. In this section it is shown that there are complete families w.r.t. to the non-inclusions QEX[+] $\not\subseteq$ QEX[<], QEX[<] $\not\subseteq$ QEX[Succ] and QEX[Succ] $\not\subseteq$ EX. As a corollary one obtains that EX[A] $\subset$ QEX[Succ; A] $\subset$ QEX[<; A] $\subset$ QEX[+; A] for all non-high oracles A.

Theorem 5.1 *There is a family $S \in QEX[+]$ which is not* QEX[<; A]*-learnable for any non-omniscient A.*

Proof: This family S is the same as in Theorem 4.3 with the only difference that B is chosen to have the same degree as K. The proof of (2) does not require that B is non-high and therefore can be adapted to show $S \in$ QEX[+]. The oracle B can be recovered by the same existential queries about f as in Theorem 4.3:

$$\begin{array}{l} m \in B \;\;\Leftrightarrow \\ (\exists x, y, z)\,[\,(f(x+a_{2m+1}+1) = 1) \;\wedge \\ \qquad (y = 0 \vee y = 1 \vee \ldots \vee y = a_{2m+1} \Leftrightarrow 2) \;\wedge \\ \qquad (x + a_{2m+1} = z \cdot a_{2m+1} + y)\,] \end{array}$$

Note that a_{2m+1} in this sentence is a constant. So inferring any $f \in S$, the IIM has access to B via the above coded membership queries and since B belongs to the omniscient EX-degree, the IIM identifies f. Thus $S \in$ QEX[+].

Now it is shown that every $f \in S$ is $k_<(a_n)$-good for all n: The functions $f \in S$ take the value 1 exactly on some infinite subset of $\{a_0, a_1, \ldots\}$. So (IV) holds, for (V) and (VI) it is necessary to look at the sequence $a_0, a_1, \ldots$; by construction all $m \geq n$ satisfy: $a_{m+1} \geq (k_+(a_m))! \geq (k_+(a_m) \Leftrightarrow 1) \cdot k_+(a_m) \geq a_m \cdot k_+(a_m) \geq a_m + k_+(a_m) \geq a_m + k_<(a_m) \geq a_m + k_<(a_n)$. This implies (IV) since $a_m \geq a_{n+1} \geq a_n + k_<(a_n)$ and (V) since for all $x > a_n$ and $y > x$ with $f(x) = 1 \wedge f(y) = 1$ there is an m with $x = a_m$ and $y \geq a_{m+1} \geq x + k_<(x)$.

So each $f \in S$ is $k_<(a_n)$-good for all n and therefore each $L[<]$-query ϕ to f can be decided after reading $f(0), \ldots, f(a_{|\phi|})$. Thus $S \in$ QEX[<; A] $\Rightarrow S \in$ EX[A]. Furthermore, $S \in$ EX[A] $\Rightarrow$ REC$_{0,1} \in$ EX[A]. Therefore if $S \in$ QEX[<; A] then A is high, i.e., A has omniscient QEX[<]-degree. ▮

The next theorem separates the class QEX[<] from the class QEX[Succ; A]. It has a more complicated proof. In Theorem 5.1 only existential queries are necessary to recover B from f. But every existential query in L[<

to a $\{0,1\}$-valued function f can be replaced by a sequence of existential queries in $L[\mathrm{Succ}]$ to f, or in other words, if $S \in REC_{0,1} \cap \mathrm{Q_1EX}[<]$ then $S \in \mathrm{Q_1EX}[\mathrm{Succ}]$ where $\mathrm{Q_1EX}[\star]$ means that queries of the inference process may only contain existential quantifiers but not universal quantifiers. There are two ways to overcome this difficulty: either by dropping the requirement that $S \subseteq \mathrm{REC}_{0,1}$ and considering a class like

$$\{f : f = \varphi_{f(0)}\} \cup$$
$$\{f : f = \varphi_x \text{ where } x = \max\{y : f(y) > y\}\}$$

or by coding the oracle K in f such that it can not be recovered via existential queries in $L[<]$. The proof of Theorem 5.2 goes the second way.

Theorem 5.2 *There is a set* $S \in \mathrm{QEX}[<]$ *with* $S \notin \mathrm{QEX}[\mathrm{Succ}; A]$ *for all non-high oracles* A.

Proof: Again let K_s be an enumeration of $\overline{K}$; K_s defines the following function h approximating $\overline{K}$: Let $h(x,s) = 0$ if $x \in K_s$ and $h(x,s) = 1$ otherwise, i.e., if $x \notin K_s$. Given any function $g \in \mathrm{REC}_{0,1}$ the function $f = \,, (g)$ is defined as $\sigma_0\sigma_1\sigma_2\ldots$ with

$$\sigma_s \;=\; 0^s\,1\,0^s\,g(s)\,0^s\,10^0h(0,s)\,0^s\,10^1h(1,s)\,0^s$$
$$10^2h(2,s)\,0^s \ldots 10^sh(s,s)\,0^s.$$

This construction is used to code g and K into $f = \,, (g)$. Since the length of the σ_s does not depend on g but only on s and since $g(s) = \sigma_s(2s{+}1)$, the σ_s – and therefore also g – can be recovered from f. Furthermore, each σ_s contains for all $x \leq s$ the substring $10^xh(x,s)$: Thus if all $h(x,s) = 1$, i.e., if $x \notin K$, then 10^s1 occurs in all strings σ_s with $s \geq x$. Otherwise $h(x,s) = 0$ for almost all s and 10^x1 is not a substring of σ_s for almost all s. So $x \in K$ iff 10^x1 is only finitely often a substring of f iff either $f(y) = 0$ or $f(y+x+1) = 0$ for almost all y. In short

$$x \in K \;\Leftrightarrow$$
$$(\exists z)(\forall y > z)[f(y) = 0 \vee f(y+x+1) = 0].$$

This query is in $L[\mathrm{Succ};<]$ since x is a constant, e.g., if $x = 3$ then $y+x+1$ means $\mathrm{Succ}(\mathrm{Succ}(\mathrm{Succ}(\mathrm{Succ}(y))))$. Now let

$$S = \{,(g) : g \in \mathrm{REC}_{0,1}\}.$$

$S \in \mathrm{QEX}[\mathrm{Succ};<] = QEX[<]$ since queries to K can be coded as queries in $L[\mathrm{Succ};<]$ about f. So it remains to show that $S \notin \mathrm{QEX}[\mathrm{Succ}; A]$ for every non-high oracle A. This is done by showing that if $S \in \mathrm{QEX}[\mathrm{Succ}; A]$ then $S \in \mathrm{EX}[A]$. In particular it is sufficient to show that every $L[\mathrm{Succ}]$-query ϕ to any $f \in S$ can be decided effectively by analizing a sufficiently long prefix of f.

For ease of notation, the additional expressions $x = y+c$, $x \neq y+c$ and $f(x+c) = d$ for integer constants c and $d \in \{0,1\}$ are allowed using the convention $f(z) = 0$ for $z < 0$. But the variables still range only over ω. The given formula ϕ w.l.o.g. does not contain complicated atoms as e.g. $f(f(x)+2) = 1$ or $f(x) \neq y$ since these can be replaced by $(f(x) = 0 \wedge f(2) = 1) \vee (f(x) = 1 \wedge f(3) = 1)$ and $(f(x) = 0 \wedge y \neq 0) \vee (f(x) = 1 \wedge y \neq 1)$, respectively. So if f occurs in the formula, then always in atoms of the form $f(x+c) = d$ for a variable x, an integer constant c and a Boolean constant $d \in \{0,1\}$.

The algorithm step-wise eliminates all quantifiers from the formula. It proceeds as follows, where ψ, ψ_1 and ψ_2 always denote quantifier-free subformulas of ϕ.

(a) $(\forall u)\,[\psi(u, x_1, \ldots, x_n)]$ occurs in ϕ:
Then the formula $(\forall u)\,[\phi(u, x_1, \ldots, x_n)]$ is replaced by $\neg((\exists u)\,[\neg\phi(u, x_1, \ldots, x_n)])$.

(b) $(\exists u)\,[\psi(u, x_1, \ldots, x_n)]$ occurs in ϕ and ψ is not in disjunctive normal form:
ψ is transformed into disjunctive normal form.

(c) $(\exists u)\;[\psi_1(u, x_1, \ldots, x_n) \;\vee\; \psi_2(u, x_1, \ldots, x_n)]$ occurs in ϕ.
Then the subformula is replaced by the formula $(\exists u)[\psi_1(u, x_1, \ldots, x_n)] \;\vee\; (\exists u)[\psi_2(u, x_1, \ldots, x_n)]$.

(d) $(\exists u)\,[u = x_1 + c \wedge \psi(u, x_1, \ldots, x_n)]$ occurs in ϕ.
If $c \geq 0$ then only u is replaced by $x_1 + c$ and the subformula becomes $\psi(x_1{+}c, x_1, \ldots, x_n))$. Otherwise $c < 0$ and the subformula contains the additional information $x_1 \geq \Leftrightarrow c$. Thus the subformula becomes $(x_1 \neq 0 \wedge x_1 \neq 1 \wedge \ldots \wedge x_1 \neq \Leftrightarrow c \Leftrightarrow 1 \wedge \psi(x_1{+}c, x_1, \ldots, x_n))$.

(e) $(\exists u)[\psi(u, x_1, \ldots, x_n)]$ occurs in ϕ, but none of the above cases holds.
W.l.o.g. $\psi(u, x_1, \ldots, x_n) = (f(u + c_0) = d_0) \wedge \ldots \wedge (f(u + c_i) = d_i) \wedge (u \neq c_{i+1}) \wedge \ldots \wedge (u \neq c_j) \wedge (u \neq x_{l_{j+1}} + c_{j+1}) \wedge \ldots \wedge (u \neq x_{l_k} + c_k) \wedge \eta(x_1, \ldots, x_n)$.
Let $c = 2 + k + |c_0| + \ldots + |c_i|$ and $d = |\sigma_0| + |\sigma_1| + \ldots + |\sigma_c|$. The formula $(\exists u)\,[\psi(u, x_1, \ldots, x_n)]$ is replaced by $\psi(0, x_1, \ldots, x_n) \vee \psi(1, x_1, \ldots, x_n) \vee \ldots \vee \psi(d, x_1, \ldots, x_n)$.

If several of the above rules are applicable, then the first one is executed. Thus ψ is always in disjunctive normal form, if rule (c), (d) or (e) is applied. Furthermore, ψ does not contain $\vee$'s if rule (d) or (e) is applied. So it remains to show that (e) works, i.e., that the truth-value of its formula does not change by this operation.

For ease of argumentation assume that $c_0 = 0, c_1 = 1, \ldots, c_i = i$. Let τ denote $d_0d_1\ldots d_i$. If τ contains more than two 1's, then τ is a substring of $\sigma_0\sigma_1\ldots\sigma_i$. So the formula is correct in this case. If τ contains 0 or 1 1's, i.e., if $\tau = 0^i$ or τ is of the form 0^*10^*, then τ occurs in σ_i, in σ_{i+1}, $\ldots$, and in σ_c. So there are more than k occurrences of τ in f before d and since there are only $k \Leftrightarrow i$ inequalities in ψ, the subformula $(f(u)f(u{+}1)\ldots f * u{+}i \Leftrightarrow 1) = \tau) \wedge (u \neq c_{i+1}) \wedge \ldots \wedge (u \neq c_j) \wedge (u \neq x_{l_{j+1}} + c_{j+1}) \wedge \ldots \wedge (u \neq x_{l_k} + c_k)$ holds. In the last case τ has the form $0^*10^a10^*$ with $i > a + 2$. Either $a \notin K_c$ and then τ occurs in each of the strings $\sigma_i, \sigma_{i+1}, \ldots, \sigma_c$, so the new formula is correct by the same argument as before. Or $a \in K_c$ and then τ does not occur in f beyond d.

So the formula is iteratively transformed into an equivalent one without variables and quantifiers. This formula consists only of atoms of the form $f(c) = d$ and can be decided by knowing a sufficiently long prefix of f. ∎

One standard example to show that the union of EX-learnable families can be outside EX is the family

$$\begin{aligned} S &= \{f :\ (\exists e)\,[f = \varphi_e \wedge 0^e 1 \preceq f] \ \vee \\ &\qquad (\exists \sigma \in \{0,1\}^*)\,[f = \sigma 0^\infty]\,\} \end{aligned}$$

which consists of all functions which code their index plus all which are almost everywhere 0 This family S has also other practical applications: S is not EX[A]-learnable for any non-omniscient A [11, Proof of Theorem 8.1]. But S is QEX[Succ]-learnable [8, Theorem 6]: An IIM conjectures φ_e but continuously checks for $c = 0, 1, \ldots$ whether $(\exists x > c)\,[f(x) \neq 0]$. These queries can be formulated without the symbol $<$ since c is a constant, so if $f = \sigma 0^\infty$, then the IIM will discover this fact after making $c = |\sigma|$ queries. The IIM makes a mind change to $\sigma 0^\infty$ and terminates. Thus one obtains the following fact:

Fact 5.3 *There is a family* $S \in$ QEX[Succ] *such that* $S \notin$ EX[A] *for all non-high oracles* A*.*

So the families given in Theorem 5.1, Theorem 5.2 and Fact 5.3 show that the following corollary is true:

Corollary 5.4 EX[A] $\subset$ QEX[Succ; A] $\subset$ QEX[$<$; A] $\subset$ QEX[$+$; A] *for all non-high oracles* A*.*

6 Conclusion

The omniscient degrees of EX, QEX[Succ], QEX[<] and QEX[+] are the same and consist exactly of the high oracles. The trivial degrees of QEX[Succ], QEX[<] and QEX[+] are a proper subset of the trivial EX-degree. Their structure is unknown, but it might be that the trivial degree of these three query inference criteria consists only of the recursive sets. Whenever EX[A] $\not\subseteq$ EX[B] then the query inference degree of A is not contained in that of B; but still it is unknown whether the inclusion structure of the query inference degrees is the same for the three criteria QEX[Succ], QEX[<] and QEX[+]. The noninclusions QEX[+] $\not\subseteq$ QEX[<] $\not\subseteq$ QEX[Succ] $\not\subseteq$ EX are strong in the sense that there exist "complete problems" S, S', S'' such that $S \in$ QEX[+]$\Leftrightarrow$QEX[$<$; A], $S' \in$ QEX[<]$\Leftrightarrow$QEX[Succ; A] and $S'' \in$ QEX[Succ]$\Leftrightarrow$EX[A] for all non-high oracles A.

Acknowledgments
The author would like to thank Bill Gasarch, Susanne Kaufmann and Martin Kummer for proofreading and comments.

References

[1] L. Adleman, M. Blum. Inductive inference and unsolvability. *Journal of Symbolic Logic*, 56:891–900, 1991.

[2] D. Angluin. Learning Regular Sets from Queries and Counterexamples. *Information and Computation*, 75:87–106, 1987.

[3] R. Beigel, W. Gasarch, J. Gill, J. Owings. Terse, superterse, and verbose sets. *Information and Computation*, 103:68–85, 1993.

[4] J. Case, C. Smith. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science*, 25:193–220, 1983.

[5] L. Fortnow, W. Gasarch, S. Jain, E. Kinber, M. Kummer, S. Kurtz, M. Pleszkoch, T. Slaman, R. Solovay, F. Stephan. Extremes in the degrees of inferability. *Annals of Pure and Applied Logic*, 66:231–276, 1994.

[6] W. Gasarch, M. Pleszkoch. Learning via queries to an oracle. In: *Proceedings of the Second Annual ACM Conference on Computational Learning Theory*– COLT'89, pp. 214–229, Morgan Kaufmann, Los Altos, 1989.

[7] W. Gasarch, M. Pleszkoch, R. Solovay. Learning via Queries in $[+,<]$. *Journal of Symbolic Logic*, 57(1):53–81, 1992.

[8] W. Gasarch, C. Smith. Learning via Queries. *Journal of the Association of Computing Machinery*, 39(3):649–676, 1992.

[9] E. M. Gold. Language Identification in the limit. *Information and Control*, 10:447–474, 1967.

[10] C. Jockusch. Degrees of generic sets. In: *London Mathematical Societey Lecture Notes*, 45:110–139, 1981.

[11] M. Kummer, F. Stephan. On the structure of degrees of inferability. In: *Proceedings of the Sixth Annual ACM Conference on Computational Learning Theory*–COLT'93, pp. 117–126, ACM-Press, New York, 1993.

[12] D. B. Posner, R. W. Robinson. Degrees Joining to $\mathbf{0}'$. *Journal of Symbolic Logic*, 46(4):714–722, 1981.

[13] P. Odifreddi. *Classical recursion theory.* North-Holland, Amsterdam, 1989.

[14] D. Osherson, M. Stob, S. Weinstein. *Systems that learn.* MIT Press, Cambridge (MA), 1986.

[15] T. Slaman, R. Solovay. When oracles do not help. In: *Proceedings of the Fourth Annual ACM Conference on Computational Learning Theory*–COLT'91, pp. 379–383, Morgan Kaufmann, San Mateo, 1991.

[16] R. I. Soare. *Recursively enumerable sets and degrees.* Springer-Verlag, Berlin, 1987.