REFINED QUERY INFERENCE

(Extended Abstract)

BY

Efim B. Kinber
Computing Centre
Latvian State University
29 Rainis Boulevard
226250 Riga
U.S.S.R.

Thomas Zeugmann[†]
Department of Mathematics
Humboldt University at Berlin
P.O. Box 1297
Berlin, 1086
G.D.R.

1. INTRODUCTION

The present paper deals with the theory of inductive inference which has attracted much attention of computer scientists (cf.,e.g., [5, 6, 10, 11]) and the references therein. Nowadays inductive inference is widely considered as a form of machine learning. Most of the work having been done in this field within the last two decades dealt with inference machines as *passive* recipients of data, i.e., the teacher fed more and more information concerning the object to be learned to the inference machine. Recently, Angluin [1] has introduced a learner and teacher paradigm in which the inference machine is additionally allowed to *ask a teacher questions*. Subsequently, various types of questions have been proven to be very helpful in efficiently learning appropriate concepts (cf. [2, 3, 4]). The recursion theoretic version of Angluin's teacher and learner paradigm has been introduced and studied by Gasarch/Smith [9] very recently. In [9] learning by asking questions is compared to learning by passively receiving data. Thereby many interesting and surprising results have been pointed out. In particular, Gasarch/ Smith [9] showed that the learning

† The results were obtained during the author's visit of the computing centre of the Latvian State University.

capabilities of inference machines asking questions are mainly determined by the language which the inference mechanism uses to phrase its questions. If the learner is allowed to ask first order questions with plus and times, then every recursive function can be learned by a single machine. On the other hand, allowing first order questions with plus and less, then questions containing a single quantifier are sufficient to learn more than what is possible by standard explanatory inference. But there is no longer a single machine that learns the entire set of recursive functions.
However, in [9] several problems remained open. What we like to present here is a refined comparison of learning via queries and learning via passively reading data. Thereby we deal with the number of allowed mind changes as well as with the number of allowed anomalies the final program may have. In order to achieve as sharp results as possible, we do not only consider the number of alternations of quantifiers, but even the number of quantifiers at all the posed questions are allowed to involve.

2. Basic Definitions and Notations

Unspecified notation follows Rogers [12]. In addition to or in contrast with [12] we use the following:
$\mathbb{N} = \{0, 1, 2, \dots\}$ denotes the set of all natural numbers. The set of all finite sequences of natural numbers is denoted by $\mathbb{N}^*$. The class of all partial recursive and recursive functions of one variable is denoted by $\mathbb{P}$, $\mathbb{R}$, respectively. By $\mathbb{R}_{\{0,1\}}$ we denote the set of all zero-one valued functions from $\mathbb{R}$ (recursive predicates). For f, g $\in \mathbb{P}$, and $x \in \mathbb{N}$ we write $f(x) = g(x)$ if both $f(x)$ and $g(x)$ are defined and equal. Let f, g $\in \mathbb{P}$, and let $a \in \mathbb{N}$, we write $f =_a g$ and $f =_* g$ iff $card(\{x \,/\, f(x) \neq g(x)\}) \leq a$ and $card(\{x \,/\, f(x) \neq g(x)\}) \leq \infty$, respectively. By φ we denote any fixed acceptable Gödel numbering of $\mathbb{P}$. Instead of $\lambda x \varphi(i,x)$ we write φ_i. Let $f \in \mathbb{P}$ and let $i \in \mathbb{N}$ such that $\varphi_i = f$. Then i is said to be a program for f. For convenience it is sometimes suitable to identify a function from $\mathbb{R}$ with the sequence of its values; so $0^i 1 0^\infty$ denotes the function f with $f(i) = 1$ and $f(x) = 0$ for all $x \neq i$.
We say that a sequence $(j_n)_{n \in \mathbb{N}}$ of natural numbers converges to a number j iff $j_n = j$ for almost all n. By $\subset$ we denote a proper set inclusion in contrast to $\subseteq$. Incomparability of sets is denoted by #.

An (standard, passive) *inductive inference machine* (abbr. IIM) is a total algorithmic device that successively takes as input the graph of a recursive function and produces (from time to time) programs as output.

Following Case/Smith [8] we define:

Definition 1 Let $a \in \mathbb{N} \cup \{*\}$, and let $f \in \mathbb{R}$. An IIM EX^a- identifies f iff the sequence of programs created by M converges to a number i such that $\varphi_i =_a f$.

If M does EX^a-identify f, we write $f \in EX^a(M)$. The collection of EX^a-inferrible sets is denoted by EX^a. If convergence is achieved after at most c mind changes of M we write $f \in EX^a_c(M)$, for $c \in \mathbb{N}$. By EX^a_c we denote the collection of EX^a-inferrible sets by IIMs restricted to at most c mind changes. For a = 0 we omit the upper index.

In the sequel we also deal with behaviorally correct inference which has been introduced by Barzdin [7], and which has been intensively studied in Case/Smith [8].

Definition 2 Let $f \in \mathbb{R}$. An IIM BC-identifies f iff the sequence $(i_k)_{k\in\mathbb{N}}$ of programs created by M satisfies $\varphi_{i_k} = f$, for almost all k.

We write $f \in BC(M)$, if M does BC-identify f and set $BC = \{U \subseteq \mathbb{R} \mid \exists M[U \subseteq BC(M)]\}$.

Following Gasarch/Smith [9] we define query inference machines as follows:

A *query inference machine* (abbr. QIM) M is a total algorithmic device taking as input a string of bits $\bar{b}$ and outputs ordered pairs (i,ψ). The bits of the string $\bar{b}$ correspond to the teacher's answers to previous queries. Furthermore, the first component of M's output denotes a program (possibly null) while the second component denotes the new question of M to the teacher. The questions are formulated in some language L. Without loss of generality we assume that all questions are in prenex normal form, i.e., quantifiers followed by a quantifier-free formula.

Definition 3 Let L be a language, let $f \in \mathbb{R}$, and let $a \in \mathbb{N} \cup \{*\}$. A QIM M EX^a-identifies f iff, when the teacher truthfully answers M's questions (formulated in L) about f, then the sequence of programs created by M converges to a number i such that $\varphi_i =_a f$.

If M does EX^a-identify f we write $f \in QEX^a[L](M)$. For a fixed language L the collection of EX^a-inferrible sets by some QIM is denoted by $QEX^a[L]$. Again, if convergence is achieved after at most c mind changes of M, we write $f \in QEX_c^a[L](M)$ and define $QEX_c^a[L] = \{U \subseteq \mathcal{R} / \exists QIM\ M\ [U \subseteq QEX_c^a[L](M)]\}$. Moreover, let $f \in QEX^a[L](M)$ for some QIM M and some language L. If M's questions involve at most k quantifiers and at most $d \geq 0$ alternations between blocks of existential and universal quantifiers, then we write $f \in Q_d^kEX^a[L](M)$. Please note that d = 0 means all questions are quantifier-free. The collections of sets $Q_d^kEX^a[L]$ and $Q_d^kEX_c^a[L]$ are analogously defined as above. Again, for a = 0 we omit the upper index.

Next we specify the languages that will be used. As in Gasarch/Smith [9], the base language $\mathfrak{L}$ that will be used allows the use of ¬, ∧, =, ∀, ∃, symbols for the natural numbers, variables ranging over $\mathbb{N}$, and a single function symbol $\mathfrak{F}$. The symbol $\mathfrak{F}$ will be used to represent the function being inferred. The base language $\mathfrak{L}$ consists only of these symbols. Furthermore, if $\mathfrak{L}$ is extended with additional symbols (e.g., a symbol for plus and a symbol for less), then we denote it just by these symbols, and include the other symbols implicitely. That means, instead of e.g. $\mathfrak{L} \cup \{+,<\}$ we shortly write [+,<]. Furthermore,we use the symbol [*] to denote an arbitrary extension of $\mathfrak{L}$.

Finally in this section we generalize the notion of team inference, originally introduced by Smith [13], to teams of QIMs. A team is a finite collection of QIMs using the same language L. A team $(M_1,\ldots,M_n)$ successfully QEX^a-infers a set $U \subseteq \mathcal{R}$, if, for each $f \in U$, some team member M_i successfully $QEX^a[L]$-identifies f. We set $QEX^a[L]_{team}(n) = \{U \subseteq \mathcal{R} / \exists (M_1,\ldots,M_n)$ [for every $f \in U$ there is some M_i, $1 \leq i \leq n$, that QEX^a-infers f]\}.

Now we are ready to present our results.

3. Results

First of all, we compare standard inference with an arbitrary but fixed number of mind changes to learning via queries, also restricted to at most a constant number of mind changes.

A careful analysis of the proof of Theorem 4 in [9] stating that $Q_0EX[*] = EX$ immediately yields the following result.

Theorem 1 Let $a \in \mathbb{N} \cup \{*\}$. Then for all $c \in \mathbb{N}$ we have $Q_0EX_c^a[*] = EX_c^a$.

Our next theorem shows that the situation considerably changes, if questions involving a single quantifier are allowed, even in case that only the base language $\mathfrak{L}$ is used.

Theorem 2 Let $a \in \mathbb{N} \cup \{*\}$. There is a $U \in Q^1_1EX_0[\mathfrak{L}] \setminus (\bigcup_{c\geq 0} EX^a_c)$.

Proof. The wanted class U is taken from Wiehagen [15], and defined as follows: $U = \{\alpha i p \ / \ \alpha \in \mathbb{N}^*,\ i \in \mathbb{N},\ i > 1,\ p \in \mathbb{R}_{(0,1)},\ \varphi_i = \alpha i p\}$.

First we show that $U \in Q^1_1EX_0[\mathfrak{L}](M)$, for some QIM M. The machine M successively asks $\forall x[\mathfrak{F}(x) = 0 \vee \mathfrak{F}(x) = 1]$

$$\forall x[x \neq 0 \rightarrowtail (\mathfrak{F}(x) = 0 \vee \mathfrak{F}(x) = 1)]$$

$$\forall x[(x \neq 0 \wedge x \neq 1) \rightarrowtail (\mathfrak{F}(x) = 0 \vee \mathfrak{F}(x) = 1)]$$

until the least k is found such that

$\forall x[(x \neq 0 \wedge x \neq 1 \wedge \ldots \wedge x \neq k) \rightarrowtail (\mathfrak{F}(x) = 0 \vee \mathfrak{F}(x) = 1)]$ is answered affirmitively. Then find the value f(k) by asking $\mathfrak{F}(k) = 0$, $\mathfrak{F}(k) = 1$, $\mathfrak{F}(k) = 2 \ldots$. Output f(k).

It remains to show that $U \notin \bigcup_{c\geq 0} EX^a_c$, for any $a \in \mathbb{N} \cup \{*\}$.

Suppose the converse, i.e., there is a $c \in \mathbb{N}$ such that $U \in EX^*_c(M)$. We shall construct a function $f \in U$ on which M fails. For the sake of simplicity of presentation, we present the case $c = 2$ only. The generalization is then straightforward. Using Smullyan's Recursion Theorem [14], we define recursive functions f_z, $1 \leq z \leq 6$ of seven variables. For short, let $f_z = \lambda x f_z(i,j,k,l,m,n,x)$, and set $f_z(i,j,k,l,m,n,x) = z$, if the numbers i, j, k, l, m, n are not pair-wise different, for $1 \leq z \leq 6$. Now let i, j, k, l, m, n be any pair-wise different numbers. Define

$f_z = i0^t$ for $t = 1, 2, 3, \ldots$ until the IIM M has produced its first hypothesis on the function $i0^\infty$ for $1 \leq z \leq 6$. Let t_0 be the least number such that M outputs its first hypothesis h_0 on the function $i0^\infty$. Set $\tau = i0^{t_0}$. Then define for $t = 1, 2, 3,$

$f_1 = \tau 0^{t+1}$ and $f_2 = \tau j 1^t$ until the IIM produces a second hypothesis on $i0^\infty$ or on $\tau j 1^\infty$. Suspend defining f_3, f_4, f_5, f_6.

Suppose M does produce a second hypothesis h_1 on one of the functions

$i0^\infty$ or $\tau j1^\infty$, and let σ be the extension of τ on which M behaves thus. Then define $f_3 = \tau\sigma$, $f_4 = \tau\sigma$, $f_1 = i0^\infty$, $f_2 = \tau j1^\infty$. Moreover, for $t = 1, 2, 3, \ldots$ define

$f_3 = \tau\sigma k0^t$ and $f_4 = \tau\sigma l1^t$ until the IIM M produces a third hypothesis on $\tau\sigma k0^\infty$ or on $\tau\sigma l1^\infty$. Suspend defining f_5 and f_6.

Suppose M produces a third hypothesis on $\tau\sigma k0^\infty$ or on $\tau\sigma l1^\infty$, and δ be the extension of $\tau\sigma$ on which M behaves thus. Then define

$f_5 = \tau\sigma\delta$, $f_6 = \tau\sigma\delta$, $f_3 = \tau\sigma k0^\infty$, and $f_4 = \tau\sigma l1^\infty$. Moreover, set $f_5 = \tau\sigma\delta m0^\infty$ and $f_6 = \tau\sigma\delta n1^\infty$.

Now let i, j, k, l, m, n be a tuple of fixed points due to Smullyan's Recursion Theorem [14]. It suffices to show that M does fail on a function f_z, for some $z \in \{1,\ldots,6\}$.

Case 1: M does not produce any hypothesis.

Then $\varphi_i = f_1 = \ldots = f_6 = i0^\infty \in U$, a contradiction.

Case 2: M produces a hypothesis h_0 on $i0^\infty$.

Then we have $\varphi_i = f_1 = i0^\infty$ and $\varphi_j = f_2 = i0^{t_0}j1^\infty$. Consequently, f_1, $f_2 \in U$. Since f_1 is not a finite variant of f_2, M does either fail on f_1 or f_2, or it must produce a second hypothesis h_1. Due to our construction we get that f_3, $f_4 \in U$. Again, since f_3 is not a finite variante of f_4 M either fails on f_3 or on f_4, or it produces a third hypothesis. Remember that M produces at least 3 hypothesis on both, f_3 and f_4, in case it identifies them. In this case, M also produces at least 3 hypotheses on f_5 and on f_6. Moreover, f_5 and f_6 belong to U, but f_5 is not a finite variante of f_6. Hence M either fails on f_5 or f_6, or it has to produce a fourth hypothesis. Since only two mind changes are allowed, this yields a contradiction.

q.e.d.

From the latter theorem it directly follows that inference via queries is more powerful than standard inference, if both are restricted to at most a fixed number of mind changes.

Corollary 3 For all $c \geq 0$ we have:

(1) $Q_1^1EX_0[*] \setminus EX_c \neq \emptyset$

(2) $EX_c \subset Q_1^1EX_c[*]$

On the other hand, Gasarch/Smith [9] have shown that $Q_1EX_0[*] \subseteq EX$. Now, applying Corollary 3 we see that their result cannot be improved. Allowing the query inference machine to make a single mind change considerably enlarges the learning potential, i.e., even function classes not contained in BC may become inferrible. Thus our next result sharpens Gasarch's and Smith's [9] Theorem 6 stating that $Q_1EX_1[*] \setminus EX \neq \emptyset$.

Theorem 4 $Q_1^1EX_1[*] \setminus BC \neq \emptyset$

Proof. (Sketch) Let U be defined as follows:
$U = \{f \mid f \in \mathbb{R},\ \varphi_{f(0)} = f\} \cup \{\alpha 0^\infty \mid \alpha \in \mathbb{N}^*\}$. Barzdin [7] has shown that $U \notin BC$. The QIM $\dot{M}$ inferring U first asks for the value f(0). Then it outputs f(0). Second it asks $\forall x[\mathfrak{F}(x) = 0]$, $\forall x[x \neq 0 \rightarrowtail \mathfrak{F}(x) = 0]$, etc., until eventually a k is found such that $\forall x[(x \neq 0 \wedge x \neq 1 \wedge \ldots \wedge x \neq k) \rightarrowtail \mathfrak{F}(x) = 0]$ is affirmatively answered. Next the QIM asks for the values f(0),...,f(k) and outputs an appropriate program for the function $f(0)\ldots f(k)0^\infty$.

q.e.d.

Keeping in mind that $Q_1EX_0[*] \subseteq EX$, one directly obtains the following corollary.

Corollary 5 $Q_1^1EX_0[*] \subset Q_1^1EX_1[*]$

We believe that the latter corollary is the base of a hierarchy defined in terms of mind changes. However, it is not known for which query languages the hierarchy extends.
Our next theorem establishes a hierarchy in terms of the number of quantifiers being all of the same type.

Theorem 6 For any $k \geq 1$ and any fixed $c \geq 0$ we have

$$Q_1^{r+1}EX_0[+,<] \setminus Q_1^rEX_c[+,<] \neq \emptyset$$

It is an interesting open problem whether or not Theorem 6 remains valid if the language $[+,<]$ is replaced by $[+,x]$, where x denotes a symbol for times. All that is known is that $\mathbb{R} \in Q_1EX\ [+,x]$ (cf. [9], Theorem 9).

Looking at Theorem 2 and 4, the following problem arises naturally: Does asking questions help to synthezise programs not having anomalies for functions which can only be inferred with anomalies by standard inference machines ? As we shall see, not even a single error can always be corrected, if we restrict ourselves to a query language not possessing the ability to ask undecidable questions.

Theorem 7 $EX^1 \setminus Q_1EX[+,<] \neq \emptyset$

However, it remained open whether or not more alternations of quantifiers could help. Nevertheless, we were able to generalize Theorem 7 as follows:

Theorem 8 For all $a \in \mathbb{N}$ we have

$$EX^{a+1} \setminus Q_1EX^a[+,<] \neq \emptyset$$

The proof uses a nontrivial extension of the proof techniques of [9].

The latter theorem again yields an infinite hierarchy.

$$Q_1EX[+,<] \subset Q_1EX^1[+,<] \subset \ldots \subset Q_1EX^a[+,<] \subset Q_1EX^{a+1}[+,<] \subset \ldots$$

Moreover, in connection with Theorem 4 the following corollary results.

Corollary 9 Let $a \in \mathbb{N}$. Then we have $EX^{a+1} \# Q_1EX^a[+,<]$.

Let us now consider the case $a = *$, i.e., the program synthesized in the limit is only required to compute a finite variant of the function to be identified. As it turns out, there are function classes being behaviorally correct identifiable, which cannot be EX^*-inferred by any QIM, that uses the language $[+,<]$ restricted to questions involving only one type of quantifier.

Theorem 10 $BC \setminus Q_1EX^*[+,<] \neq \emptyset$

As an immediate consequence of Theorem 4 and 10 we get:

Corollary 11 $BC \# Q_1EX^*[+,<]$

Next in this section we deal with questions involving more than one alternation of blocks of quantifiers. First of all we could sharpen

Theorem 12 of Gasarch/Smith [9] stating that $Q_2EX_0[\le] \setminus EX \ne \emptyset$ as follows:

Theorem 12 $Q_2^2EX_0[\le] \setminus BC \ne \emptyset$

The proof uses the same class U as the proof of Theorem 4 does.

Now let us have a closer look to the statement that $Q_1EX_0[*] \subseteq EX$ pointed out by Gasarch/Smith [9] (cf. Theorem 5). This result actually shows that the single blocks of quantifiers which the QIM uses to phrase its questions can be eliminated, if one does not further insist in restricting the inference machine to output a single guess only. After having made this observation, it is natural to ask whether or not such an elimination is also possible, if more than one alternation of quantifiers is allowed. Our next theorem answers this question affirmatively.

Theorem 13 For all $d \in \mathbb{N}$ we have

$$Q_{d+1}EX_0[*] \subseteq Q_dEX[*]$$

Proof. (sketch) Let $U \in Q_{d+1}EX_0[*](M)$, for some QIM M, and let $f \in U$. Then there is a finite sequence of questions such that, if $b_0,\dots,b_k$, where $b_i \in \{0,1\}$, $0 \le i \le k$, are the truth answers, then M on $b_0,\dots,b_k$ produces its correct guess a, i.e., $\varphi_a = f$.

Furthermore, due to the assumption M poses questions involving at most d+1 blocks of quantifiers. Since all questions are in prenex normal form, these blocks of quantifiers are followed by a quantifier-free formula ϕ. Any question involving at most d blocks of quantifiers remains unchanged. Let us now consider the questions exactly having d+1 blocks of quantifiers. Then two cases are possible, i.e., the first block (on the lefthand side) is either a block of universal quantifiers or a block of existential quantifiers. Let n be the length of the first block, i.e., the first block looks like $\forall x_1\forall x_2\dots\forall x_n$ in the first case, or like $\exists x_1\exists x_2\dots\exists x_n$ in the second case. Each of these questions is replaced by a potentially infinite sequence of questions as follows:

Case 1: $\psi = \forall x_1\forall x_2\dots\forall x_n\exists y_1\dots\exists y_z\dots\phi(x_1,\dots,x_n,y_1\dots)$

Let $(a_{11},a_{21},\dots,a_{n1})$, $(a_{12},a_{22},\dots,a_{n2})$, $(a_{13},a_{23},\dots,a_{n3}),\dots$ be any fixed computable enumeration of all n-tuples of natural numbers. Then ψ is replaced by

$\psi_1 = \exists y_1\dots\exists y_z\dots\phi(a_{11},a_{21},\dots,a_{n1},y_1,\dots)$,

$\psi_2 = \exists y_1\dots\exists y_z\dots\phi(a_{12},a_{22},\dots,a_{n2},y_1,\dots)$,

$\psi_3 = \exists y_1\dots\exists y_z\dots\phi(a_{13},a_{23},\dots a_{n3},y_1,\dots),\dots$etc.

Case 2: $\theta = \exists x_1\exists x_2\dots\exists x_n\forall y_1\dots\forall y_z\dots\phi(x_1,\dots,x_n,y_1,\dots)$

Then θ is replaced by

$\theta_1 = \forall y_1\dots\forall y_z\dots\phi(a_{11},a_{21},\dots,a_{n1},y_1,\dots)$,

$\theta_2 = \forall y_1\dots\forall y_z\dots\phi(a_{12},a_{22},\dots,a_{n2},y_1,\dots)$,

$\theta_3 = \forall y_1\dots\forall y_z\dots\phi(a_{13},a_{23},\dots a_{n3},y_1,\dots)$, etc.,

where the a_{ij} are defined as above.

At this point we make the following important observation.

If $\psi = \forall x_1\forall x_2\dots\forall x_n\exists y_1\dots\exists y_z\dots\phi(x_1,\dots,x_n,y_1\dots)$ is answered by "YES" then also every ψ_1, $\psi_2,\dots$ must be answered by "YES". If ψ is answered by "NO" then there is a tuple $(a_{1m},\dots,a_{nm})$ such that ψ_m has to be answered by "NO". Moreover, such an m must occur after finitely many steps.

Now let $\theta = \exists x_1\exists x_2\dots\exists x_n\forall y_1\dots\forall y_z\dots\phi(x_1,\dots,x_n,y_1,\dots)$ be answered by "NO". Then also every question θ_1, $\theta_2,\dots$ must be answered by "NO". If θ is affirmatively answered then there is a tuple $(a_{1m},\dots,a_{nm})$ such that θ_m has to be affirmatively answered.

In both cases we have the following situation. Either we always get the truth answer, i.e., the sequence of bits corresponding to the answers to ψ_1, $\psi_2,\dots$ or θ_1, $\theta_2,\dots$ is a constant one, or there is a point at which it steps from 1 to 0 or from 0 to 1. Having eventually reached this point, we stop asking questions from this sequence.

The wanted QIM M' works as follows:

Replace the first question ψ or θ of M as described above, and pose the question ψ_1 or θ_1, respectively. Fed the result obtained to M.

Output M's hypothesis, if any, and replace the new question ψ' or θ' produced by M as described above. Now ask ψ_2 or θ_2, and ψ'_1 or θ'_1, respectively. Check if the answer to ψ_2 or θ_2 coincides with the answer to ψ_1 or θ_1, respectively. In case it does, fed b_0 and b_1 to M, where b_0 corresponds to the answer to ψ_2 or θ_2, and b_1 corresponds to the answer to ψ'_1 or θ'_1, respectively.

In case it does not, fix the new obtained answer b_0 and remove the procedure generating ψ_1, $\psi_2,\ldots,$ or $\theta_1,\ldots,$ respectively. Fed b_0 to M. If M outputs a guess, output it. Otherwise fed again b_0 and b_1 to M. Then, if M produces a hypothesis, output it. Replace the next question ψ'' or θ'' as described above. Ask now ψ_3 or θ_3, ψ'_2 or θ'_2, and ψ''_1 or θ''_1, respectively, if no replaced sequence has been removed yet. Otherwise ask only the appropriate nonremoved questions. Proceed as described above, i.e., make the appropriate checks etc.

That means, in each stage of the work of M' we look for the shortest sequence b_0, $b_1,\ldots,b_\mu$ on which M produces a hypothesis. Applying the observation made above, now it is clear that we will find the shortest correct sequence on which M produces its correct guess. Consequently, after having reached this point, M' only outputs the correct hypothesis.

q.e.d.

If we allow the QIMs using d+1 alternations of quantifiers to formulate their questions to make a fixed number $c > 0$ of mind changes then the latter theorem does not seem to remain valid. However, all that is known is that then a team of c+1 QIMs suffices to achieve the desired elimination.

Theorem 14 Let $d \in \mathbb{N}$, and let $c > 0$. Then

$$Q_{d+1}EX_c[*] \subseteq Q_dEX\ [*]_{team}(c+1)$$

It is an interesting open problem to find out for which languages no team of c QIMs is powerful enough.

Our last theorem shows that there may be a certain trade off between the number of alternations of quantifiers and the number of quantifiers allowed at all.

Theorem 15 $Q_1^3EX_1[\Sigma] \setminus Q_2^2EX[\Sigma] \neq \emptyset$

It seems hard to generalize the latter theorem.

4. CONCLUSIONS

We have examined, in some more detail, inference via queries restricted to a fixed number of mind changes as well as inference via queries where the finally synthesized programs may have anomalies. Thereby we could show that even questions involving only one type of quantifier may aid the learning process in two directions. First, the number of mind changes can be reduced, and second, more function classes become inferrible. On the other hand, the Theorems 6 and 15 show that the learning potential of QIMs does mainly depend on the complexity of the allowed questions measured in the number of involved quantifiers.

SUMMARY

Mind changes

$$\begin{array}{ccccccccc} EX_0^a & \subset & EX_1^a & \subset & \subset & \bigcup_{c\geq 0} EX_c^a & \subset & BC \\ \| & & \| & & & \| & & \# \\ Q_0EX_0^a[*] & \subset & Q_0EX_1^a[*] & \subset & \subset & \bigcup_{c\geq 0} Q_0EX_c^a[*] & \subset & Q_1EX_0^a[\Sigma] \end{array}$$

Number of quantifiers

$$Q_1^1EX_c[+,<] \subset Q_1^2EX_c[+,<] \subset Q_1^3EX_c[+,<] \subset \ldots \subset Q_1^nEX_c[+,<] \subset \ldots$$

Number of allowed anomalies

$$Q_1EX[+,<] \subset Q_1EX^1[+,<] \subset \ldots \subset Q_1EX^a[+,<] \subset Q_1EX^{a+1}[+,<] \subset \ldots$$

REFERENCES

[1] Angluin, D., Learning regular sets from queries and counter-examples. *Information and Computation 75, 2 (1987), 87 - 106.*

[2] Angluin, D., Learning k-term DNF formulas using queries and counter-examples. *Dept. of Computer Science TR-559, Yale Univ., New Haven, CT, 1987.*

[3] Angluin, D., Queries and concept learning. *Machine learning 2 (1988), 319 - 342.*

[4] Angluin, D., Learning with hints. *Proc. First Workshop on Computational Learning Theory, Aug. 3 - 5. MIT, pp. 223 - 237.*

[5] Angluin, D.,and C.H. Smith, Inductive inference: theory and methods. *Computing Surveys 15 (1983), 237 - 269.*

[6] Angluin, D.,and C.H. Smith, Inductive inference. *In Encyclopedia of Artificial Intelligence, S. Shapiro, Ed., 1987.*

[7] Barzdin, Ya. M., Two theorems on the limiting synthesis of functions. *In Theory of Algorithms and Programs I, (Ya. M. Barzdin, Ed.), pp. 82 - 88 (in russian), Latvian State Univ., 1974.*

[8] Case, J., and C.H. Smith, Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science 25, 2 (1983), 193 - 220.*

[9] Gasarch, W.I., and C.H. Smith, Learning via queries. *Univ. of Maryland, Institute for Advanced Computer Studies, Dept. of Computer Science, TR-88-89.*

[10] Klette, R., and R. Wiehagen, Research in the theory of inductive inference by GDR-mathematiciens - a survey. *Information Science 22, 149 - 169.*

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

[12] Rogers, H.Jr., *Theory of Recursive Functions and Effective Computability.* (McGraw-Hill, New York 1967).

[13] Smith, C.H., The power of pluralism for automatic program synthesis.,*Journal of the ACM 29, 4 (1982), 1144 - 1165.*

[14] Smullyan, R.M., *Theory of formal systems.* Annals of Mathematical Studies 42, 7, (1962).

[15] Wiehagen, R., Limes-Erkennung rekursiver Funktionen durch spezielle Strategien. *Journal Inf. Processing and Cybernetics (EIK) 12 (1976),1/2, 93 - 99.*