Learning languages with queries

Steffen Lange [*], **Jochen Nessel** [*,**] **Sandra Zilles** [***]

[*] Deutsches Forschungszentrum für Künstliche Intelligenz, Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany, lange@dfki.de

[**] Medizinische Universität zu Lübeck, Institut für Theoretische Informatik, Wallstraße 40, 23560 Lübeck, Germany, jochen@tcs.mu-luebeck.de

[***] Universität Kaiserslautern, Fachbereich Informatik, Postfach 3049, 67653 Kaiserslautern, Germany, zilles@informatik.uni-kl.de

Abstract. The following deals with learning indexable classes of recursive languages by asking different kinds of queries about them. Among the types of queries considered here are superset, subset, equivalence and membership queries.
The resulting models of learning with queries are compared to one another and to standard learning models like finite learning, conservative inference as well as learning in the limit. We elaborate a complete picture concerning the relation of all these learning models and characterize all query learning models introduced.

Keywords. Formal languages, computational learning theory, formal learning models

1 Introduction

When you want to learn something, asking questions is one of the most natural things to do. It has been done in teaching for thousands of years by students and teachers alike. For example, Socrates preferred to teach his pupils with a series of questions and answers.

In machine learning, learning by "asking questions" was first modeled and investigated by Angluin in [3]. Since then learning with queries has been intensively studied; see the survey paper [4] for a detailed report of what has been achieved so far. An example for the use of queries in machine learning systems is Shapiro's Algorithmic Debugging System, cf. [10].

In contrast to Gold's [5] model of learning in the limit (see below), Angluin's [3] model deals with "one-shot" learning. Here, a learning algorithm (henceforth called query learner) receives information about a target concept by asking queries which will truthfully be answered by an oracle. After asking at most finitely many queries, the learner is required to make up its mind and to output its one and only hypothesis. If this hypothesis correctly describes the target concept, learning took place.

Angluin's work and ensuing work in this area mainly addresses two aspects: learning of finite classes of concepts and its efficiency. In contrast, we focus our attention on the learnability of infinite classes of concepts, thereby neglecting complexity issues.

Regarding finite classes of concepts, it is quite obvious that every concept class can be learned with the usual types of queries (membership, subset, superset and equivalence queries as well as the restricted versions thereof, see below). This situation changes, if infinite classes of concepts form the object of learning. For illustration, consider the case of learning all extended pattern languages[1]. Restricted superset queries do not suffice to learn this language class, while, for instance, restricted equivalence queries will do, see [8]. This motivates a detailed analysis of the power and the limitations of the different models of learning with the various types of queries.

This paper addresses query learning of a particular type of infinite classes of concepts, namely indexable classes of recursive languages (indexable classes, for short). In machine learning, again following Angluin's work, see [1], the learnability of indexable classes has been intensively studied within different learning models, see the survey paper [11] for more information. This is motivated by the fact that many interesting and natural classes, including regular, context free, context sensitive, and pattern languages, constitute in-

[1] A pattern is a word consisting of constant symbols from some finite alphabet Σ and variables from an enumerable set X of variables. The extended pattern language generated by a pattern π is the set of all strings over Σ that can be obtained by substituting strings over Σ for the variables in π.

dexable classes.

Angluin [1] provides a very useful criterion that is necessary and sufficient for an indexable class to be Gold-style learnable from only positive data[2] – namely the existence of specific uniformly recursively enumerable sets, so-called "tell-tales". A tell-tale for a language L' in a class $\mathcal{C}$ of languages is a finite subset of L' not contained in any proper "sub-language" of L' in $\mathcal{C}$. In Gold's learning model, such tell-tales provide a firm basis on which a universal learning method M can be established, see [1]. Intuitively, M realizes the following idea: if the tell-tale for some L' shows up in the data about some target language L and if L' is consistent with all data seen so far, M guesses L'. If $L' \neq L$, M must receive data proving that L' is definitely a wrong guess, i. e. elements from L not contained in L'.

Both from a practical and a theoretical point of view it is very useful to find universal learning methods for different learning models, i. e. universal in the sense that they can handle all learning problems that are solvable in this particular model. Moreover, tell-tales and variants thereof turned out to be quite useful in order to characterize the principal learning power of various models of Gold-style language learning. This leads to the idea to compare these learning models with the query learning models and to find out, whether tell-tales are of a similar importance in these settings. In particular, it would be interesting to know, whether any kind of tell-tale-like property

(a) is sufficient and/or necessary for successful query learning (using queries of some specified type);

(b) provides a universal method for successful query learning (using queries of some specified type).

In the first part of our formal study, we investigate the following query learning models: learning of indexable classes with membership and equivalence queries as well as with subset, superset, restricted subset, and restricted superset queries. The learning power of the resulting models is studied, the models are compared to one another and to standard learning models like finite learning and learning in the limit, see [5, 1, 11]. In doing so, a complete picture concerning the relation of all these learning models is obtained. It is rather obvious that learning with equivalence queries coincides with Gold's model of limit learning from positive *and* negative data (so-called counterexamples), see [5], while learning with membership queries equals finite learning from positive and negative data, see [5]. Our analysis now shows that any query learner using superset queries can be simulated by a Gold-style language learner that receives exclusively positive data. In contrast to that, there are classes that are learnable using subset queries, but that are not learnable in Gold's models of learning from positive data only. This can be traced back to a dual relation between superset and subset queries: the relevance of positive data for simulating a superset query learner matches the relevance of negative data for simulating a subset query learner. Specifically, it turns out that any query learner using subset queries can be replaced by an equally powerful Gold-style learner that receives negative data only. Furthermore, characterizations of all introduced models of learning with queries are presented. In particular, some kind of tell-tale property can be used to characterize learning with subset or superset queries. As desired, this provides a universal method of how to design query learners.

The second part of our formal study deals with another aspect. In the usual model, it is assumed that the query learner is constrained to query languages it is supposed to learn. As it turns out, this is a severe restriction. Permitting to query languages not contained in the target class increases the capabilities of the relevant learners. Now, as there are classes, which can only be learned with the help of queries concerning additional languages, it is quite conceivable, that these classes have just been chosen too small, i. e. that some proper superclass is learnable in the current query model, because it contains all languages needed for the queries. In other words, this would suggest that negative results are just caused by badly stated problems. But in general this is not true, as our last result will show: there are indeed classes only learnable, if it is allowed to query additional languages, although *none* of the superclasses themselves are learnable using the specified type of queries. Finally the relevance of this extended model is demonstrated by applying it to the problem of learning extended pattern languages.

2 Notions and Notations

Familiarity with standard mathematical and recursion theoretic notions and notations as well as with basic language theoretic concepts is assumed, cf. [9, 6].

From now on, a fixed finite alphabet Σ with $\{a, b\} \subseteq \Sigma$ is assumed. A *word* is any element from Σ^* and a *language* is any subset of Σ^*. The *complement* $\overline{L}$ of a language L is the set $\Sigma^* \setminus L$. Subsequently, $(F_j)_{j \in \mathbb{N}}$ denotes any fixed, repetition-free, and recursively generable family of all finite subsets of $\mathbb{N}$, i. e. there is a computable function that, given any $j \in \mathbb{N}$, enumerates all elements of F_j and stops.

[2] In this model, a learner receives successively more and more elements from a target language L and outputs, from time to time, hypotheses about L. Learning took place, if the sequence of hypotheses stabilizes on a correct hypothesis for L.

Let $\mathcal{C}$ be a class of recursive languages over Σ^*. $\mathcal{C}$ is said to be an *indexable class of recursive languages* (*indexable class*, for short), if there is an effective enumeration $(L_j)_{j\in\mathbb{N}}$ of all and only the languages in $\mathcal{C}$ that has uniformly decidable membership, i. e. there is a computable function that, for any $w \in \Sigma^*$ and $j \in \mathbb{N}$, returns 1, if $w \in L_j$, and 0, otherwise. Such an enumeration will subsequently be called an *indexing* of $\mathcal{C}$.

In the query learning model, a learner has access to an oracle that truthfully answers queries of a specified kind. Any such oracle considered has to be computable. This constraint restricts the class of languages that may form the object of learning.

A *query learner* M is an algorithmic device that, depending on the reply on the previous queries, computes either a new query or a hypothesis and halts. M uses natural numbers to denote its queries and hypotheses; both will be interpreted with respect to an underlying hypothesis space. When learning a target indexable class $\mathcal{C}$, any indexing $\mathcal{H} = (L_j)_{j\in\mathbb{N}}$ of $\mathcal{C}$ may form a *hypothesis space*. So, as in the original query learning model, cf. [3], when learning $\mathcal{C}$, M is only allowed to query languages belonging to $\mathcal{C}$.

More formally speaking, let $\mathcal{C}$ be an indexable class, let $L \in \mathcal{C}$, let $\mathcal{H} = (L_j)_{j\in\mathbb{N}}$ be an indexing of $\mathcal{C}$, and let M be a query learner. *M learns L with respect to $\mathcal{H}$ using a certain type of queries* provided that it eventually halts and that its one and only hypothesis, say j, correctly describes L, i. e. $L_j = L$. In other words, M returns its one and only correct guess j after asking only finitely many queries. Furthermore, *M learns $\mathcal{C}$ with respect to $\mathcal{H}$ using a certain type of queries*, if it learns every $L' \in \mathcal{C}$ with respect to $\mathcal{H}$ using queries of the specified type.

The query types considered in this paper are:

Membership queries. The input is a string w and the answer is ‘yes’ and ‘no’, respectively, depending on whether or not w belongs to the target language L.

Equivalence queries. The input is an index j of some language $L' \in \mathcal{C}$. If $L = L'$, the answer is ‘yes’. Otherwise, together with the answer ‘no’ a counterexample from the symmetrical difference of L and L' is supplied.

Subset queries. The input is an index j of some language $L' \in \mathcal{C}$. If $L' \subseteq L$, the answer is ‘yes’. Otherwise, together with the answer ‘no’ a counterexample from $L' \setminus L$ is supplied.

Superset queries. The input is an index j of some language $L' \in \mathcal{C}$. If $L \subseteq L'$, the answer is ‘yes’. Otherwise, together with the answer ‘no’ a counterexample from $L \setminus L'$ is supplied.

Equivalence, subset, and superset queries are also studied in a *restricted form*, for which the answer ‘no’ is no longer supplemented by a counterexample.

$MemQ$, $EquQ$, $SubQ$, and $SupQ$ denote the collection of all indexable classes $\mathcal{C}'$ for which there are a query learner M' and a hypothesis space $\mathcal{H}'$ such that M' learns $\mathcal{C}'$ with respect to $\mathcal{H}'$ using membership, equivalence, subset, and superset queries, respectively. For learning with equivalence (subset, superset) queries, it is required that $\mathcal{H}'$ has a decidable equivalence problem (inclusion problem). Moreover, if the learner uses only the restricted form of queries, this is indicated by the prefix “*Res*” connected with the notations $EquQ$, $SubQ$, or $SupQ$.

Comparing query learning with the standard models in Gold-style language learning requires some more notions. These will be kept short, see [5, 1, 11] for more details.

Let L be a language. Any infinite sequence $(w_j)_{j\in\mathbb{N}}$ with $\{w_i \mid i \in \mathbb{N}\} = L$ is called a *text* for L. Any infinite sequence $((w_j, b_j))_{j\in\mathbb{N}}$ with $b_i \in \{+,-\}$, $\{w_i \mid b_i = +\} = L$ and $\{w_i \mid b_i = -\} = \overline{L}$ is said to be an *informant* for L.

Let $\mathcal{C}$ be an indexable class, $\mathcal{H} = (L_j)_{j\in\mathbb{N}}$ a hypothesis space, and $L \in \mathcal{C}$. An *inductive inference machine* (*IIM*) is an algorithmic device, that reads longer and longer initial segments of a text (informant) and, from time to time, outputs numbers as its hypotheses. As above, an IIM M returning some j is construed to hypothesize the language L_j. Given a text t (an informant i) for L, *M identifies L from t (i) with respect to $\mathcal{H}$*, if the sequence of hypotheses output by M, when fed t (i), stabilizes on a number j (i. e. past some point M always outputs the hypothesis j) with $L_j = L$. *M identifies $\mathcal{C}$ from text (informant)* with respect to $\mathcal{H}$, if it identifies every $L' \in \mathcal{C}$ from every corresponding text (informant). As above, $LimTxt$ ($LimInf$) denotes the collection of all indexable classes $\mathcal{C}'$ for which there are an IIM M' and a hypothesis space $\mathcal{H}'$ such that M' identifies $\mathcal{C}'$ from text (informant) with respect to $\mathcal{H}'$.

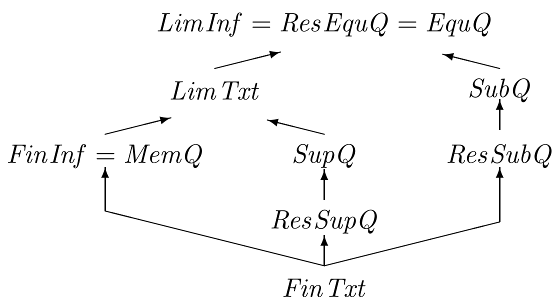
In contrast to the query learning model, an IIM is allowed to change its mind finitely many times before returning its final and correct hypothesis. In general, it is not decidable whether or not an IIM has already output its final hypothesis. Adding this requirement to the above definition results in *finite learning*, see [5]. Similar to the query learning model, finite learning can be understood as a kind of “one-shot” learning, where the first hypothesis already has to be correct. The corresponding models $FinTxt$ and $FinInf$ are defined analogously as above.

3 Results

3.1 Separations

The figure below summarizes the observed separations and coincidences.

Each learning type is represented as a vertex in a digraph. A directed edge (or path) from A to B indicates that the learning type B outperforms type A, i. e. B is a proper superset of A. Moreover, no edge (or path) between A and B indicates that the learning types A and B are incomparable.



All separations of learning models in this figure can be achieved using indexable classes that admit an indexing with a decidable inclusion problem.

This figure shows, that learning with (restricted) equivalence queries is exactly as powerful as Gold's models of limit learning from positive and negative data (so-called counterexamples), while learning with membership queries equals finite learning from positive and negative data. Learning with superset queries will only work for indexable class that are Gold-style learnable from only positive data. Hence, the power of these superset query learners is rather limited. In contrast, there are indexable classes that cannot be learned from only positive data, but that can be learned with subset queries. Thus, subset query learners may use the kind of additional information which counterexamples normally provide. Interestingly, the characterizations in the next section will even show that any class in $SubQ$ is learnable in the limit from only negative data.

The following theorems provide the formal evidence for the results displayed above. We start with two rather obvious results that are somehow known.

Theorem 1
(a) $LimInf = ResEquQ = EquQ$
(b) $FinInf = MemQ$

In contrast to the case when learning with equivalence queries is considered, learners that are allowed to ask subset queries are more powerful than learners that are constrained to ask restricted subset queries. The next example illustrates this phenomenon.

Let $(F'_j)_{j \in \mathbb{N}^+}$ be an enumeration of all non-empty finite subsets of $\{a\}^+$. Now $\mathcal{C}_{rsub}$ contains the language $L = \{b\}^+$ as well as all languages $L_k = F'_k \cup \{b, \ldots, b^k\}$, $k \in \mathbb{N}^+$. On the one hand, subset queries suffice to learn $\mathcal{C}_{rsub}$. This is caused by the following three facts. First, the reply 'yes' to a subset query corresponding to L implies that L equals the target language. Second, any counterexample that is provided together with the reply 'no' to a subset query corresponding to L allows a learner to delete all but finitely many candidate languages from further considerations. Third, every indexable class that consists of only finitely many languages is obviously learnable with subset queries. On the other hand, a learner that witnesses $\mathcal{C}_{rsub} \in ResSubQ$ must have the ability to learn the indexable class that contains all and only all finite non-empty subsets of $\{a\}^+$. The latter is even impossible in case when a learner is allowed to ask subset queries.

Next, consider the indexable class $\mathcal{C}_{rsup}$ that contains all languages $L_k = \{a^{kn} \mid n \in \mathbb{N}^+\}$, $k \in \mathbb{N}^+$. Applying similar arguments as above, one sees that superset queries are sufficient to learn $\mathcal{C}_{rsup}$. On the other hand, it is not hard to verify that restricted superset queries are not sufficient.

Theorem 2
(a) $FinTxt \subset ResSubQ \subset SubQ \subset LimInf$
(b) $FinTxt \subset ResSupQ \subset SupQ \subset LimTxt$

Proof: First, let $\mathcal{C} \in FinTxt$. From the characterization of the learning type $FinTxt$ (cf. [7]) it is known that $L \not\subseteq L'$ and $L' \not\subseteq L$ for all distinct languages L and L' in $\mathcal{C}$. Hence, a restricted subset (superset) query corresponding to some $L \in \mathcal{C}$ receives the answer 'yes' iff L equals the target language, and therefore $\mathcal{C} \in ResSubQ \cap ResSupQ$. Next, let $\mathcal{C}'$ be an indexable class that contains only two languages, say L and L'. Moreover, suppose that L is a proper subset of L'. Obviously, by the argument stressed above, we obtain $\mathcal{C}' \notin FinTxt$. On the other hand, since $\mathcal{C}'$ contains only finitely many languages, we directly obtain $\mathcal{C}' \in ResSubQ \cap ResSupQ$, and thus $FinTxt \subset ResSubQ$ and $FinTxt \subset ResSupQ$.

Second, by definition, $ResSubQ \subseteq SubQ$ and $ResSupQ \subseteq SupQ$. Moreover, the indexable classes $\mathcal{C}_{rsub}$ and $\mathcal{C}_{rsup}$ witness that both inclusions are proper.

Third, $SubQ \subseteq LimInf$ holds, since every indexable class is $LimInf$–identifiable (cf. [5]). Comparing the characterizations of the learning types $SupQ$ (cf. Theorem 5 in Section 3.2) and $LimTxt$ (cf. [1]), one directly sees that $SupQ \subseteq LimTxt$. Finally, the missing parts $LimInf \setminus SubQ \neq \emptyset$ and $LimTxt \setminus SupQ \neq \emptyset$ follow directly via Assertions (c) and (e) of Theorem 3. □

Theorem 3
(a) $ResSubQ \setminus LimTxt \neq \emptyset$
(b) $FinInf \setminus SubQ \neq \emptyset$
(c) $ResSupQ \setminus SubQ \neq \emptyset$
(d) $ResSupQ \setminus FinInf \neq \emptyset$
(e) $FinInf \setminus SupQ \neq \emptyset$

Proof: We provide the separating indexable classes only. The corresponding verification is left to the reader.

Assertion (a) is witnessed by the following indexable class $\mathcal{C}_a$ that contains $L = \{a\}^+$ as well as $L_k = L \setminus \{a^k\}$ for all $k \in \mathbb{N}^+$.

To verify Assertion (b), one may choose the indexable class $\mathcal{C}_b$. Recall that $(F'_j)_{j\in\mathbb{N}^+}$ is an enumeration of all finite non-empty subsets of $\{a\}^+$. Now $\mathcal{C}_b$ contains all languages $L_k = F'_k \cup \{b, \ldots, b^k\}$, $k \in \mathbb{N}^+$.

Next, one directly sees that the class $\mathcal{C}_c$ of all finite non-empty subsets of $\{a\}^+$ witnesses Assertion (c). Moreover, it is well-known that $\mathcal{C}_c \notin FinInf$, and thus Assertion (d) is verified, as well.

Finally, Assertion (e) can be shown by means of the indexable class $\mathcal{C}_e$ containing $L = \{a\}^+ \cup \{b\}$ and all languages $L_k = \{a, \ldots, a^k\}$, $k \in \mathbb{N}^+$. □

3.2 Characterizations

By Theorem 2, regarding learning with subset or superset queries, it is necessary to study restricted queries separately. The characterization of $ResSubQ$ expresses the following idea: by asking subset queries, until the reply 'yes' is received for some query j, the learner may find a candidate L_j for the language to be learned. Then a finite number of subset queries must suffice to find out whether this candidate is a proper subset of the target language L or not. For that purpose the learner is able to generate a suitable finite list $F_{h(j)}$ of queries, such that L equals L_j iff none of the queries in $F_{h(j)}$ describe subsets of L.

Theorem 4 *Let $\mathcal{C}$ be an indexable class.*

(a) $\mathcal{C} \in ResSubQ$ *iff there is an indexing* $(L_j)_{j\in\mathbb{N}}$ *of* $\mathcal{C}$ *and some recursive function* $h : \mathbb{N} \mapsto \mathbb{N}$ *such that, for all* $i, j \in \mathbb{N}$,

  (i) if $i \in F_{h(j)}$, *then* $L_i \not\subseteq L_j$,

  (ii) if $L_j \subset L_i$, *there is some* $z \in F_{h(j)}$, *such that* $L_z \subseteq L_i$.

(b) $\mathcal{C} \in ResSupQ$ *iff there is an indexing* $(L_j)_{j\in\mathbb{N}}$ *of* $\mathcal{C}$ *and some recursive function* $h : \mathbb{N} \mapsto \mathbb{N}$ *such that, for all* $i, j \in \mathbb{N}$,

  (i) if $i \in F_{h(j)}$, *then* $L_i \not\supseteq L_j$,

  (ii) if $L_j \supset L_i$, *there is some* $z \in F_{h(j)}$, *such that* $L_z \supseteq L_i$.

Proof: ad (a).
Necessity. Let $(L_j)_{j\in\mathbb{N}}$ be an indexing of $\mathcal{C}$ and let M be a query learner that learns $\mathcal{C}$ using restricted subset queries.

Let $j \in \mathbb{N}$ be fixed. Simulate M when learning L_j. Since M is supposed to identify L_j, it asks only finitely many queries before returning its one and only hypothesis. Let $q_1, \ldots, q_m$ be the queries answered with 'no'. Then set $h(j) = t$, where t is the unique index satisfying $F_t = \{q_1, \ldots, q_m\}$. Now it is not hard to verify the desired properties.

Sufficiency. We only provide an idea for the required query learner; the rest of the verification is omitted. Initially, let the learner compute step (1) with $z = 0$.

(1) Query $z, z+1, \ldots$ until the reply 'yes' is received for some query j. Goto (2).

(2) If $F_{h(j)} = \emptyset$, output j and stop. Otherwise, for all $i \in F_{h(j)}$, query i.

  If the reply to each of these queries is 'no', then output j and stop. Else set $z = j + 1$ and goto (1).

The success of this learner is based on the following observation: assume $L \in \mathcal{C}$ and $j \in \mathbb{N}$ such that $L_j \subseteq L$. Then L equals L_j iff $L_z \not\subseteq L$ for all $z \in F_{h(j)}$.

ad (b). The proof proceeds analogously. □

Note that the proof of sufficiency reveals some uniform strategy which must be successful for each indexable class in $ResSubQ$ resp. $ResSupQ$.

For the characterizations of $SubQ$ and $SupQ$ some kind of tell-tale property (cf. [1]) is used. Recall that a tell-tale for a language L' in a class $\mathcal{C}$ is a finite subset of L' which is not contained in any proper "sublanguage" of L' in $\mathcal{C}$. Theorem 5 shows that indexable classes learnable with superset queries can be characterized using tell-tale-like sets for the languages in the class, whereas tell-tale-like sets for the complements of the languages in the class are of relevance when learning with subset queries is considered. The proof is omitted, but as usual, the verification of sufficiency of the given properties provides universal methods of how to design appropriate query learners.

Theorem 5 *Let $\mathcal{C}$ be an indexable class.*

(a) $\mathcal{C} \in SubQ$ *iff there is an indexing* $(L_j)_{j\in\mathbb{N}}$ *of* $\mathcal{C}$ *and some recursively generable family* $(T_{j,n})_{j,n\in\mathbb{N}}$ *of finite non-empty sets such that*

- *(i) for all* $i, j, n \in \mathbb{N}$*, if* $T_{j,n} \subseteq \overline{L_i}$*, then* $\overline{L_i} \not\subset \overline{L_j}$*,*
- *(ii) for all* $j \in \mathbb{N}$ *and all* $W \subseteq \Sigma^*$*, if* $T_{j,n} \not\subseteq W$ *for all* $n \in \mathbb{N}$*, then there is some* $i \in \mathbb{N}$ *with* $L_i \setminus L_j \neq \emptyset$ *and* $W \cap (L_i \setminus L_j) \neq \emptyset$*.*

(b) $\mathcal{C} \in SupQ$ *iff there is an indexing* $(L_j)_{j\in\mathbb{N}}$ *of* $\mathcal{C}$ *and some recursively generable family* $(T_{j,n})_{j,n\in\mathbb{N}}$ *of finite non-empty sets such that*

- *(i) for all* $i, j, n \in \mathbb{N}$*, if* $T_{j,n} \subseteq L_i$*, then* $L_i \not\subset L_j$*,*
- *(ii) for all* $j \in \mathbb{N}$ *and all* $W \subseteq \Sigma^*$*, if* $T_{j,n} \not\subseteq W$ *for all* $n \in \mathbb{N}$*, then there is some* $i \in \mathbb{N}$ *with* $L_j \setminus L_i \neq \emptyset$ *and* $W \cap (L_j \setminus L_i) \neq \emptyset$*.*

Theorem 5 clearly shows a duality of subset and superset queries. The tell-tale property in Assertion (b) implies that all indexable classes in $SupQ$ are identifiable in the limit from text, i. e. from positive examples only. Analogously, Assertion (a) means that the class $\{\overline{L} \mid L \in \mathcal{C}\}$ of complements belongs to $LimTxt$, whenever $\mathcal{C}$ is learnable with subset queries. In other words, all indexable classes in $SubQ$ are identifiable in the limit from negative examples only.

3.3 Learning with extended queries

In the elementary model of learning via subset or superset queries the learner is only allowed to ask queries corresponding to languages in the target class. Yet it is quite conceivable that this constraint restricts the learning power, i. e. that more indexable classes would be identifiable, if it was permitted to query languages not contained in the target class. Indeed the class $\mathcal{C}_{extsub}$ consisting of the empty language and all cofinite subsets of Σ^+, which contain the word a, is not identifiable with subset queries. Yet permitting to query the language $\{a\}$ allows for learning $\mathcal{C}_{extsub}$ with restricted subset queries. Similarly, the class $\mathcal{C}_{extsup}$ consisting of the language Σ^+ and all finite subsets of Σ^+, which do *not* contain the word a, is not identifiable using superset queries. On the other hand, if it is allowed to query the language $\Sigma^+ \setminus \{a\}$, restricted superset queries are sufficient for learning $\mathcal{C}_{extsup}$.

If $\mathcal{C}$ is an indexable class, an *extended query learner* for $\mathcal{C}$ is permitted to query languages in $\mathcal{C} \cup \mathcal{C}'$ for some indexable class $\mathcal{C}'$. We say that $\mathcal{C}$ is learnable using extended subset (superset) queries with respect to $\mathcal{C}'$ iff there is an indexing $(L_j)_{j\in\mathbb{N}}$ of $\mathcal{C} \cup \mathcal{C}'$, such that

- the inclusion problem in $(L_j)_{j\in\mathbb{N}}$ is decidable,
- it is decidable, for any arbitrary index j, whether or not L_j belongs to $\mathcal{C}$,
- there is an extended query learner M identifying $\mathcal{C}$ with respect to $(L_j)_{j\in\mathbb{N}}$ using subset (superset) queries concerning $\mathcal{C} \cup \mathcal{C}'$.

$ExtSubQ$ ($ExtSupQ$) denotes the collection of all indexable classes $\mathcal{C}$ learnable with extended subset (superset) queries; the learning types $ExtResSubQ$ and $ExtResSupQ$ are defined analogously.

It remains to study the learning power of the extended models, i. e. to find out, to what extent the collections of learnable language classes increase, if additional queries are permitted. In contrast to the former model the concepts of learning with subset queries and learning with restricted subset queries coincide in the extended model (analogously concerning superset queries).

Theorem 6
(a) $SubQ \subset ExtResSubQ = ExtSubQ$
(b) $SupQ \subset ExtResSupQ = ExtSupQ$

Proof: ad (a). $SubQ \subseteq ExtSubQ$ is obvious. An example for a class in $ExtSubQ \setminus SubQ$ is the family $\mathcal{C}_{extsub}$ defined above.

By definition $ExtResSubQ \subseteq ExtSubQ$. To verify the opposite inclusion, suppose $\mathcal{C}$ is learnable with extended subset queries via some learner M. The idea for an $ExtResSubQ$-learner M' for $\mathcal{C}$ is to simulate the learner M, extended by a routine constructing counterexamples for every negative reply received by M. So suppose that M, having queried a language L, receives the reply ‘no’ from an oracle *not* providing counterexamples. If $(w_i)_{i\geq 0}$ is an effective enumeration of L, let M' query the languages $\{w_0\}$, $\{w_1\}$, $\{w_2\}, \ldots$, until the reply ‘no’ is received for some query concerning $\{w_q\}$. Then w_q is provided as a counterexample to the learner M. As soon as M returns its hypothesis, M' does the same and stops. It is easy to verify that M' learns $\mathcal{C}$ with extended restricted subset queries.

ad (b). By definition $SupQ \subseteq ExtSupQ$, moreover the class $\mathcal{C}_{extsup}$ belongs to $ExtSupQ \setminus SupQ$. The proof of $ResExtSupQ = ExtSupQ$ is similar to the proof of $ResExtSubQ = ExtSubQ$. □

Note that all the extra queries posed by the learner M' in the proof above can also be represented by membership queries. It turns out that subset queries in combination with additional membership queries are always sufficient for simulating $ExtSubQ$-learners (analogously for superset queries), see Theorem 7.

Res*Sub-MemQ* (Res*Sup-MemQ*) denotes the class of all language classes identifiable by a learner using membership and restricted subset (restricted superset) queries.

Theorem 7
(a) $ExtSubQ = Res$*Sub-MemQ*
(b) $ExtSupQ = Res$*Sup-MemQ*

Proof: ad (a). Replacing each membership query for some word w by a restricted subset query for $\{w\}$ verifies Res*Sub-MemQ* $\subseteq ExtSubQ$. The opposite inclusion is obtained by simulating any subset query corresponding to a language not in the target class via membership queries and restricted subset queries corresponding to languages in the target class only. For that purpose fix $\mathcal{C}$ and $\mathcal{C}'$, as well as some learner M, such that $\mathcal{C}$ is learnable by M with extended *restricted* subset queries with respect to $\mathcal{C} \cup \mathcal{C}'$ (note that, by Theorem 6, restricted queries do not decrease the learning power in the extended model).

Now let a new learner M' start simulating M. Whenever M poses a query corresponding to a language in $\mathcal{C}$, M' does the same. If M queries some language $L' \in \mathcal{C}'$, let M' in parallel (i) ask restricted subset queries for all supersets of L' in $\mathcal{C}$ and (ii) test all elements of L' for membership in the target language. This process stops with the reply 'yes', if the answer 'yes' is ever received for a query according to (i); it stops with the reply 'no', if the answer 'no' is ever received for a query according to (ii). As inclusion is decidable in $\mathcal{C} \cup \mathcal{C}'$, this method is effective. It is not hard to verify that M' learns $\mathcal{C}$ with membership queries and restricted subset queries, i. e. $\mathcal{C} \in Res$*Sub-MemQ*.

Part (b) follows similarly. □

Comparing query learning in the extended model with Gold-style language learning, the following results are easily obtained. The formal proof is omitted.

Theorem 8
(a) $ExtSubQ \subset LimInf$
(b) $FinInf \subset ExtSupQ \subset LimTxt$

$SupQ$ and $ExtSupQ$ are proper subclasses of $LimTxt$, but the extended model provides more learning power than the elementary model. Considering the tell-tale properties in Theorem 5, which hold similarly for extended queries, suggests a characterization of $ExtSupQ$ in terms of a particular type of Gold-style language learning, namely conservative inference (cf. [1, 7]).

Let $\mathcal{C}$ be an indexable class and $(L_j)_{j\in\mathbb{N}}$ an indexing for $\mathcal{C}$. We say that $\mathcal{C}$ can be *conservatively* identified with respect to $(L_j)_{j\in\mathbb{N}}$ iff there is an IIM M that identifies $\mathcal{C}$ from text with respect to $(L_j)_{j\in\mathbb{N}}$ and that performs exclusively justified mind changes, i. e. if M, on some text t, outputs hypotheses j and later j', then M must have seen some word $w \notin L_j$ before it outputs j'. In other words, M may only change its hypothesis when it has found hard evidence that it is wrong.

Theorem 9 *Let $\mathcal{C}$ be indexable. $\mathcal{C} \in ExtSupQ$ iff there is an indexing $(L_j)_{j\in\mathbb{N}}$ of $\mathcal{C}$ such that*

(1) the inclusion problem in $(L_j)_{j\in\mathbb{N}}$ is decidable,

(2) $\mathcal{C}$ can be conservatively identified with respect to $(L_j)_{j\in\mathbb{N}}$.

Proof: We only sketch the idea. If $\mathcal{C}$ belongs to $ExtSupQ$, it has recursively generable "tell-tales" similar to those in the characterization of $SupQ$ in Theorem 5 (b). Using a characterization of conservative inference from [7], $\mathcal{C}$ can then be conservatively identified with respect to $(L_j)_{j\in\mathbb{N}}$.

Given that $\mathcal{C}$ is identifiable conservatively with respect to $(L_j)_{j\in\mathbb{N}}$, "tell-tales" are recursively generable (cf. again [7]). This helps to show $\mathcal{C} \in Res$*Sup-MemQ* $= ExtSupQ$. Restricted superset queries are used to find candidate languages; membership queries suffice to test whether the tell-tale corresponding to a candidate is contained in the target language. □

It is also possible to characterize *ExtSubQ* and *ExtSupQ* similarly to Theorem 4.

Theorem 10 *Let $\mathcal{C}$ be an indexable class.*

(a) $\mathcal{C} \in ExtSubQ$ iff there exist an indexable class $\mathcal{C}'$, an indexing $(L_j)_{j\in\mathbb{N}}$ of $\mathcal{C} \cup \mathcal{C}'$, and recursive functions $h, g : \mathbb{N} \mapsto \mathbb{N}$ such that, for all $i, j \in \mathbb{N}$ with $L_j \in \mathcal{C}$,

(i) if $i \in F_{h(j)}$ then $L_i \in \mathcal{C}$ and if $i \in F_{g(j)}$ then $L_i \in \mathcal{C}'$,

(ii) if $i \in F_{h(j)} \cup F_{g(j)}$, then $L_i \not\subseteq L_j$,

(iii) if $L_j \subset L_i$, then there is some $z \in F_{h(j)} \cup F_{g(j)}$, such that $L_z \subseteq L_i$.

(b) $\mathcal{C} \in ExtSupQ$ *iff there exist an indexable class* $\mathcal{C}'$*, an indexing* $(L_j)_{j\in\mathbb{N}}$ *of* $\mathcal{C} \cup \mathcal{C}'$*, and recursive functions* $h, g : \mathbb{N} \mapsto \mathbb{N}$ *such that, for all* $i, j \in \mathbb{N}$ *with* $L_j \in \mathcal{C}$*,*

(i) *if* $i \in F_{h(j)}$ *then* $L_i \in \mathcal{C}$ *and if* $i \in F_{g(j)}$ *then* $L_i \in \mathcal{C}'$*,*

(ii) *if* $i \in F_{h(j)} \cup F_{g(j)}$*, then* $L_i \not\supseteq L_j$*,*

(iii) *if* $L_j \supset L_i$*, then there is some* $z \in F_{h(j)} \cup F_{g(j)}$*, such that* $L_z \supseteq L_i$*.*

The classes $\mathcal{C}_{extsub}$ and $\mathcal{C}_{extsup}$ are evidence to the fact that extending the set of permitted queries may increase the learning power. Still it is conceivable, that this phenomenon results from restricting the target class $\mathcal{C}$ too much by excluding the “relevant” languages, i. e. adding the additionally queried languages to $\mathcal{C}$ yields an indexable class that itself is learnable in the elementary query model. Theorem 11 shows that this is not true.

Theorem 11
(a) There is some $\mathcal{C} \in ExtSubQ$ *such that* $\mathcal{C} \cup \mathcal{C}' \notin SubQ$ *for all classes* $\mathcal{C}'$*.*
(b) There is some $\mathcal{C} \in ExtSupQ$ *such that* $\mathcal{C} \cup \mathcal{C}' \notin SupQ$ *for all classes* $\mathcal{C}'$*.*

Proof: ad (a). The class $\mathcal{C}_{extsub}$ is learnable with extended subset queries with respect to $L_a = \{a\}$, although $\mathcal{C}_{extsub} \cup \{L_a\} \notin SubQ$.

ad (b). Consider the class $\mathcal{C}_{extsup}$ and the language $L_b = \Sigma^+ \setminus \{a\}$. $\mathcal{C}_{extsup}$ is learnable with extended superset queries with respect to L_b, yet the whole class $\mathcal{C}_{extsup} \cup \{L_b\}$ does not belong to $SupQ$. □

So the differences between learning with subset (superset) queries and the corresponding extended versions are not the result of choosing the target classes in an inadequate manner.

For further illustration, consider the more natural example of learning extended pattern languages. Recall the corresponding definition. According to [2], a pattern is a word consisting of constant symbols from Σ and variables from an enumerable set X of variables. The extended pattern language generated by a pattern π is the set of all strings in Σ^* that can be obtained by substituting strings from Σ^* for the variables in π.[3] The term *one-variable extended pattern language* refers to the case that the admissible patterns contain at most one variable.

The following non-learnability result is from [8].

[3] In Angluin’s [2] definition it is only allowed to replace variables by non-empty strings, i. e. variables could not be simply erased.

Theorem 12 ([8]) *The class of all one-variable extended pattern languages is not learnable with restricted superset queries.*

The following positive result provides a typical example that it may be advantageous to allow additional queries:

Theorem 13 *The class of all one-variable extended pattern languages is learnable with restricted superset queries, if it is additionally permitted to query arbitrary extended pattern languages.*

The proof is omitted. Note that, as already mentioned in the introduction, still the class of all extended pattern languages is *not* learnable with restricted superset queries, again see [8].

References

1. D. Angluin. Inductive inference of formal languages from positive data. *Information and Control*, 45:117–135, 1980.
2. D. Angluin. Finding patterns common to a set of strings. *Journal of Computer and System Sciences*, 21:46–62, 1980.
3. D. Angluin. Queries and concept learning, *Machine Learning* 2:319–342, 1988.
4. D. Angluin. Queries revisited. *Proc. Int. Conference on Algorithmic Learning Theory (ALT’01)*, LNAI 2225:12–31, Springer-Verlag, 2001.
5. E. M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.
6. J. E. Hopcroft, J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley Publishing Company, 1979.
7. S. Lange, T. Zeugmann. Types of monotonic language learning and their characterization. *Proc. 5th Annual ACM Workshop on Computational Learning Theory*, 377–390. ACM Press, 1992.
8. J. Nessel, S. Lange. Learning erasing pattern languages with queries. *Proc. Int. Conference on Algorithmic Learning Theory (ALT’00)*, LNAI 1968:86–100, Springer-Verlag, 2000.
9. H. Rogers, Jr. *Theory of Recursive Functions and Effective Computability*, MIT Press, 1987.
10. E. Shapiro. *Algorithmic Program Debugging*, MIT Press, 1983.
11. T. Zeugmann, S. Lange. A guided tour across the boundaries of learning recursive languages. *Algorithmic Learning for Knowledge-Based Systems*, LNAI 961:190–258, Springer-Verlag, 1995.