Formal language identification: Query learning vs. Gold-style learning

Steffen Lange
FH Darmstadt
FB Informatik, Haardtring 100
64295 Darmstadt, Germany
s.lange@fbi.fh-darmstadt.de

Sandra Zilles
TU Kaiserslautern
FB Informatik, Postfach 3049
67653 Kaiserslautern, Germany
zilles@informatik.uni-kl.de

Abstract

A natural approach towards powerful machine learning systems is to enable options for additional machine/user interactions, for instance by allowing the system to ask queries about the concept to be learned. This motivates the development and analysis of adequate formal learning models.

In the present paper, we investigate two different types of query learning models in the context of learning indexable classes of recursive languages: Angluin's original model and a relaxation thereof, called learning with extra queries. In the original model the learner is restricted to query languages belonging to the target class, while in the new model it is allowed to query other languages, too. As usual, the following standard types of queries are considered: superset, subset, equivalence, and membership queries.

The learning capabilities of the resulting query learning models are compared to one another and to different versions of Gold-style language learning from only positive data and from positive and negative data (including finite learning, conservative inference, and learning in the limit). A complete picture of the relation of all these models has been elaborated. A couple of interesting differences and similarities between query learning and Gold-style learning have been observed. In particular, query learning with extra superset queries coincides with conservative inference from only positive data. This result documents the naturalness of the new query model.

1 Introduction

In machine learning, the quite natural approach of learning by 'asking questions' was first modeled and investigated by Angluin [2]. An example for its use in machine learning systems is Shapiro's Algorithmic Debugging System, cf. [10]. Since Angluin's pioneering paper [2], the query learning model has been receiving a lot of attention (see [3] for a quite recent overview).

Angluin's [2] 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 be truthfully answered by an oracle. After at most finitely many queries, the learner is required to stop this process and to output its one and only hypothesis. The learning process is considered successful, if this hypothesis correctly describes the target concept.

Angluin's work and ensuing work in this area mainly address the aspect of efficiency of query learning, measured in terms of the number of queries maximally needed to satisfy the learning goal, see [3] and the references therein. Thus several interesting polynomial-time query learners for different concept classes have been designed. In particular, this has revealed close relations between query learning and PAC-learning, see [11, 2].

In the present paper, we study the pros and cons of Angluin's [2] query learning model in the context of learning indexable classes of recursive languages (indexable classes, for short). The learnability of indexable classes has been intensively studied within different formal frameworks (see [12] for a survey). This is motivated by the fact that many interesting and natural classes, including regular, context free, context sensitive, and pattern languages, constitute indexable classes.

We investigate learning of indexable classes with superset, subset, equivalence, and membership queries, comparing the learning capabilities of the resulting query learners to one another. In contrast to former studies, we neglect complexity issues. Regarding finite classes of concepts, apparently, every class can be learned with the usual types of queries. This is no longer valid, if infinite classes form the learning task. For illustration, membership queries do not suffice to learn the class of all extended pattern languages, see [8]. This motivates a detailed analysis of the power and the limitations of the query learning models.

Moreover, the resulting query learning models are compared to models of Gold-style language learning such as finite learning, conservative inference, and learning in the limit from only positive data as well as from positive and negative data. In Gold-style learning, the learner may change its recent hypothesis when more information is provided, but it receives only 'local' information about the object to be learned. In contrast, in the query model the learner receives rather 'global' information and can affect the sample of information it receives, but it may never revise its hypothesis. So a comparison between query inference and Gold-style learning may help to explain the relevance of these different features of learning. Again, for this purpose it is useful to neglect efficiency issues in query learning.

A complete picture displaying the relations between all discussed versions of query learning and Gold-style learning is obtained. For example, our analysis shows that any query learner using superset queries can be simulated by a Gold-style learner receiving only positive data. In contrast to that, there are classes learnable using subset queries, but not Gold-style learnable from positive data only. This can be traced back to a duality of 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.

From a theoretical point of view, Angluin's [2] query learning model has the drawback that learnable classes may possess non-learnable subclasses. Moreover, a couple of quite simple indexable classes are not learnable with superset or subset queries. The observed weakness is often caused by the fact that the learners are constrained to query exclusively languages belonging to the target class. In many cases, the learners are simply not allowed to make the 'appropriate' queries. In the present paper, we therefore modify Angluin's [2] original model by relaxing this constraint and introduce a new model, called learning with extra queries. We analyze the learning power of the resulting query learners by comparing them to the capabilities of query learners in the original model and to Gold-style language learners. As it turns out, an indexable class is learnable using extra superset queries if and only if there is a conservative Gold-style language learner that identifies this class from only positive examples. To a certain extent, this result proves the naturalness of the new model of query inference.

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, 5].

From now on, a fixed finite alphabet Σ with $\{a, b\} \subseteq \Sigma$ is assumed. By Σ^* we denote the set of all words over Σ and any subset $L \subseteq \Sigma^*$ is called a *language*. The *complement* $\overline{L}$ of a language L is the set $\Sigma^* \setminus L$. If $\mathcal{C}$ is a class of languages, then we denote by $\overline{\mathcal{C}}$ the class $\{\overline{L} \mid L \in \mathcal{C}\}$ of all complements of languages in $\mathcal{C}$.

Let $\mathcal{C}$ be a class of recursive languages over Σ^*. $\mathcal{C}$ is said to be an *indexable class*, if there is an effective enumeration $(L_i)_{i\in\mathbb{N}}$ of all and only the languages in $\mathcal{C}$ such that membership is uniformly decidable, i. e., there is a computable function that, for any $w \in \Sigma^*$ and $i \in \mathbb{N}$, returns 1, if $w \in L_i$, 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. A *query learner* M is an algorithmic device that, depending on the reply on the previous queries, either computes a new query or returns a hypothesis and halts. Its queries and hypotheses are coded as natural numbers; both will be interpreted with respect to an underlying *hypothesis space*. When learning an indexable class $\mathcal{C}$, any indexing $\mathcal{H} = (L_i)_{i\in\mathbb{N}}$ of $\mathcal{C}$ may form a hypothesis space, such that M, when learning $\mathcal{C}$, is only allowed to query languages belonging to $\mathcal{C}$, see [2].

More formally, let $\mathcal{C}$ be an indexable class, let $L \in \mathcal{C}$, let $\mathcal{H} = (L_i)_{i\in\mathbb{N}}$ be an indexing of $\mathcal{C}$, and let M be a query learner. *M learns L with respect to $\mathcal{H}$ using some type of queries* if it eventually halts and its only hypothesis, say i, correctly describes L, i. e., $L_i = L$. So M returns its unique and correct guess i after only finitely many queries. Moreover, *M learns $\mathcal{C}$ with respect to $\mathcal{H}$ using some type of queries*, if it learns every $L' \in \mathcal{C}$ with respect to $\mathcal{H}$ using queries of the specified type. Below we consider:

Membership queries. The input is a string w and the answer is 'yes' or 'no', 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 $(L' \setminus L) \cup (L \setminus L')$ 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.

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.

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

MemQ, *EquQ*, *SupQ*, and *SubQ* denote the collections 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, superset, and subset queries, respectively. If the learner uses only the restricted form of queries, this is indicated by the prefix '*res*' connected to *EquQ*, *SupQ*, or *SubQ*. Note that, by definition, *resEquQ* $\subseteq$ *EquQ*, *resSupQ* $\subseteq$ *SupQ*, and *resSubQ* $\subseteq$ *SubQ*.

Moreover, it will be helpful to notice the following simple observation stating that superset and subset queries yield dual learning models.

Proposition 1 *Let $\mathcal{C}$ be an indexable class.*
(a) $\mathcal{C} \in$ SupQ iff $\overline{\mathcal{C}} \in$ SubQ   (b) $\mathcal{C} \in$ resSupQ iff $\overline{\mathcal{C}} \in$ resSubQ

Comparing query learning with the standard models in Gold-style language learning requires some more notions explained in brief below, see also [4, 1, 12]. Let L be a language. Any infinite sequence $t = (w_i)_{i \in \mathbb{N}}$ with $\{w_i \mid i \in \mathbb{N}\} = L$ is called a *text* for L. Then, for any $n \in \mathbb{N}$, t_n denotes the initial segment $(w_0, \ldots, w_n)$ and *content*(t_n) denotes the set $\{w_0, \ldots, w_n\}$. Any infinite sequence $((w_i, b_i))_{i \in \mathbb{N}}$ 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, $\mathcal{H} = (L_i)_{i \in \mathbb{N}}$ a hypothesis space (i. e., an indexing possibly comprising a proper superclass of $\mathcal{C}$), and $L \in \mathcal{C}$. An *inductive inference machine* (*IIM*) is an algorithmic device, that reads longer and longer initial segments σ of a text (informant) and outputs numbers $M(\sigma)$ as its hypotheses. As above, an IIM M returning some i is construed to hypothesize the language L_i. Given a text (an informant) t for L, M *identifies* L *from* t *with respect to* $\mathcal{H}$, if the sequence of hypotheses output by M, when fed t, stabilizes on a number i (i. e., past some point M always outputs the hypothesis i) with $L_i = L$. 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 $\mathcal{H}'$. A quite natural and often studied modification of *LimTxt* is defined by the model of *conservative inference*. M is a *conservative* IIM for $\mathcal{C}$ with respect to $\mathcal{H}$, if M performs only justified mind changes, i. e., if M, on some text t for some $L \in \mathcal{C}$, outputs hypotheses i and later j, then M must have seen some element $w \notin L_i$ before returning j. The collection of all indexable classes identifiable from text by a conservative IIM is denoted by *ConsvTxt*.

In contrast to query learners, 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. In case this is decidable we allude to *finite learning*, see [4]. Similar to query learning, 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 as above. Some helpful results on Gold-style learning are summarized in the following proposition, see [4, 6, 12] for the details.

Proposition 2 *FinTxt* $\subset$ *FinInf* $\subset$ *ConsvTxt* $\subset$ *LimTxt* $\subset$ *LimInf*.

3 Comparison results

The scope of this paper is to compare the learning capabilities of Angluin's query learning models to one another and to the different versions of Gold-style learning defined above. First, note that learning with (restricted) equivalence queries coincides with Gold's model of limit learning from positive and negative data, while learning with membership queries equals finite learning from positive and negative data.[1] Second, the power of superset query learners is rather limited, because they will only be successful for classes that are Gold-style learnable from only positive data. In contrast, there are indexable classes not learnable from only positive data, but learnable with subset queries. Thus, subset query learners may exploit the kind of additional information which negative data normally provide. Still, the potential of subset query learners is severely limited: there are indexable classes in *LimTxt*, even in *FinInf*, which are not learnable using subset queries. In contrast to the case when learning with equivalence queries is considered, learners that are allowed to ask superset (subset) queries are more powerful than learners that are constrained to ask restricted superset (subset) queries. The formal details are summarized in Theorem 3 and Theorem 4.

Theorem 3 *(a) FinTxt* $\subset$ *resSupQ* $\subset$ *SupQ* $\subset$ *LimTxt* $\subset$ *LimInf*

(b) FinTxt $\subset$ *resSubQ* $\subset$ *SubQ* $\subset$ *LimInf*

Proof. (a) First, *FinTxt* $\subset$ *resSupQ* can be verified using the following fact known from [12]: if $\mathcal{C} \in$ *FinTxt*, then $L \# L'$ for all distinct languages L and L' in $\mathcal{C}$.

Second, we verify *SupQ* $\setminus$ *resSupQ* $\neq \emptyset$. Consider the class $\mathcal{C}_{rsup}$ containing $L_0 = \{a\}^*$ and all languages $L_i = (\{a\}^* \setminus \{a, a^i\}) \cup \{b^z \mid z \geq i\}$, $i \geq 1$. To learn $\mathcal{C}_{rsup}$, a query learner M starts with a query corresponding to L_0. If the answer is 'yes', M knows that the target language equals L_0. If the answer is 'no', an element from $\overline{L_0}$, say b^z, $z \geq 1$, is supplied as a counterexample. Since $b^z \notin L_i$ for all $i > z$, M can easily identify the target language by posing queries corresponding to $L_1, L_2, \ldots, L_z$. To see that $\mathcal{C}_{rsup} \notin$ *resSupQ*, suppose there is a superset query learner M' for $\mathcal{C}_{rsup}$. Consider M' when

[1] These results are somehow folklore and will not be proven here.

learning L_1. If M' queries L_1, provide the answer 'yes'. In all other cases provide the answer 'no'. Let z be the maximal index of a language M' queries before returning its sole hypothesis. Since the answers provided are consistent with the two distinct languages L_1 and L_{z+1}, M' fails for at least one of them. So $\mathcal{C}_{rsup} \notin$ *resSupQ* and hence *SupQ* $\setminus$ *resSupQ* $\neq \emptyset$.

Third, *LimTxt*$\setminus$*SupQ* $\neq \emptyset$ follows via Theorem 4(e) and *FinInf* $\subseteq$ *LimTxt*. To verify *SupQ* $\subseteq$ *LimTxt*, let $\mathcal{C} \in$ *SupQ* and M a query learner identifying $\mathcal{C}$ with respect to an indexing $(L_i)_{i\in\mathbb{N}}$. The following IIM M' *LimTxt*-identifies $\mathcal{C}$. Given a text segment σ of length n, M' interacts with M simulating a learning process for n steps. In step k, $k \leq n$, depending on how M' has replied to the previous queries posed by M, the learner M computes either (i) a new query i or (ii) a hypothesis i. In case (ii), M' returns the hypothesis i and stops simulating M. In case (i), M' checks whether there is a word in *content*$(\sigma) \setminus L_i$. If such a word exists, M' replies 'no' using one such word as a counterexample; else M' replies 'yes'. If $k < n$, M executes step $k+1$, else M' returns any auxiliary hypothesis and stops simulating M. Given segments σ of a text for some target language, if their length n is large enough, M' answers all queries of M correctly and M returns its sole hypothesis within n steps. So, the hypotheses returned by M' stabilize on this correct guess.

(b) First, *FinTxt* $\subset$ *resSubQ* holds, since $L \# L'$ for all distinct languages L and L' in any $\mathcal{C} \in$ *FinTxt*. The remaining assertions follow from *(a)* and Proposition 1. □

Theorem 4 *(a) resSupQ* $\not\subseteq$ *SubQ* *(b) resSubQ* $\not\subseteq$ *SupQ* *(c) resSupQ* $\not\subseteq$ *FinInf* *(d) resSubQ* $\not\subseteq$ *LimTxt* *(e) FinInf* $\not\subseteq$ *SupQ* *(f) FinInf* $\not\subseteq$ *SubQ*

Proof. For (d) and (f) we provide the separating classes only. The class that contains $L = \{a\}^*$ and $L_i = L \setminus \{a^i\}$ for all $i \geq 1$ proves (d). Assertion (f) can be shown using the class consisting of all languages $\{a, a^i\} \cup \{b, \ldots, b^i\}$, $i \geq 1$. Now (b) follows from (d), because *SupQ* $\subseteq$ *LimTxt*. Moreover, (a) is obtained from (b) via Proposition 1. Assertions (c) and (e) follow from (d) and (f), respectively, via Proposition 1, taking into account that *FinInf* $\subseteq$ *LimTxt* and that an indexable class $\mathcal{C}$ is in *FinInf* iff $\overline{\mathcal{C}}$ is in *FinInf*. □

4 Learning with extra queries

Angluin's [2] query learning model has one drawback: learnable classes may possess non-learnable subclasses. For instance, $\mathcal{C}_{rsup}$ can be identified using superset queries, while its subclass $\mathcal{C}'_{rsup} = \mathcal{C}_{rsup} \setminus \{\{a\}^*\}$ cannot. This drawback is a direct consequence of Angluin's assumption that the learner may only ask queries referring to languages in the target class. As the class $\mathcal{C}'_{rsup}$ illustrates, this assumption may bar the learner from asking the 'appropriate' queries. By Proposition 1, an analogous result holds for learning with subset queries.

To overcome these difficulties, we introduce a modified version of Angluin's [2] model. Let $\mathcal{C}$ be an indexable class. An *extra query learner* for $\mathcal{C}$ is permitted to query languages in any indexing $(L'_i)_{i\in\mathbb{N}}$ of a superclass $\mathcal{C}'$ of $\mathcal{C}$. We say that $\mathcal{C}$ is learnable using extra superset (subset) queries with respect to $(L'_i)_{i\in\mathbb{N}}$ iff there is an extra query learner M learning $\mathcal{C}$ with respect to $(L'_i)_{i\in\mathbb{N}}$ using superset (subset) queries concerning $\mathcal{C}'$. Then *xSupQ* (*xSubQ*) denotes the collection of all indexable classes $\mathcal{C}$ learnable with extra superset (subset) queries; the notions *xresSupQ* and *xresSubQ* are defined analogously.

The above discussion immediately yields the following result.

Theorem 5 *(a) SupQ $\subset$ xSupQ (b) SubQ $\subset$ xSubQ*

In contrast to the original model, learning with superset queries and learning with restricted superset queries now coincide (analogously for subset queries).

Theorem 6 *(a) xresSupQ $=$ xSupQ (b) xresSubQ $=$ xSubQ*

Proof. (a) Obviously, it suffices to verify *xSupQ* $\subseteq$ *xresSupQ*. So let $\mathcal{C} \in$ *xSupQ* and let M be a corresponding query learner. In order to learn some target language L, the required *xresSupQ*-learner M' uses M as a subroutine. In the long run, M' is supposed to adopt M's hypothesis. For this purpose, M' asks the same queries as M does and hands over the obtained replies to M. In order to make M work, M' has to find supplementing counterexamples in case a superset query of M', referring to some L', receives the reply 'no'. But this is easily done. M' simply asks, for all elements $w \in \overline{L'}$, a superset query corresponding to the language $\Sigma^* \setminus \{w\}$, until it receives the reply 'no'. If this happens, M' has found the counterexample needed, since $\Sigma^* \setminus \{w\} \not\supseteq L$ iff $w \in L$.

(b) Using (a) and an adaptation of Proposition 1 for learning with extra queries yields the desired proof. □

Although the learning power of query learners increases with the permission to ask extra queries, the learners are not always able to identify also the extra languages they query.

Theorem 7 *(a) There is some $\mathcal{C} \in$ xSupQ such that $\mathcal{C}' \notin$ SupQ for all superclasses $\mathcal{C}'$ of $\mathcal{C}$.*

(b) There is some $\mathcal{C} \in$ xSubQ such that $\mathcal{C}' \notin$ SubQ for all superclasses $\mathcal{C}'$ of $\mathcal{C}$.

Proof. (a) Let $\mathcal{C}_{xsup}$ be the class consisting of Σ^* and all finite subsets of $\Sigma^* \setminus \{a\}$.

First, to verify $\mathcal{C}_{xsup} \in$ *xSupQ*, choose an indexing of $\Sigma^* \setminus \{a\}$ and all languages in $\mathcal{C}_{xsup}$. A learner M for $\mathcal{C}_{xsup}$ may initially make a query for $\Sigma^* \setminus \{a\}$. If the answer is 'no', then the target language L must be Σ^*. If the answer is 'yes', M can determine L by (i) asking finite subsets of $\Sigma^* \setminus \{a\}$ until some finite $L_i \supseteq L$ is found and (ii) asking subsets of L_i until the minimal subset of L_i forcing the answer 'yes' is known.

Second, no superclass of $\mathcal{C}_{xsup}$ belongs to *SupQ*. Assuming the contrary, suppose $\mathcal{C}'$ is a superclass of $\mathcal{C}_{xsup}$ in *SupQ*. Then distinguish two cases.

Case 1. $\Sigma^* \setminus \{a\} \in \mathcal{C}'$: Since $\mathcal{C}'$ also contains all finite subsets of $\Sigma^* \setminus \{a\}$, arguments used already in [4] imply $\mathcal{C}' \notin$ *LimTxt* and thus $\mathcal{C}' \notin$ *SupQ* by Theorem 3.

Case 2. $\Sigma^* \setminus \{a\} \notin \mathcal{C}'$: Assume some learner M identifies $\mathcal{C}'$ with superset queries. Consider a special scenario S of M when learning Σ^*. If M queries Σ^*, let M get the answer 'yes'. If M queries some $L_i \neq \Sigma^*$, let M get the answer 'no' together with some counterexample $w_i \neq a$. Note that some $w_i \in \Sigma^* \setminus L_i$ with $w_i \neq a$ exists, because $L_i \neq \Sigma^* \setminus \{a\}$. After finitely many queries M guesses Σ^*.

Let $L_{i_1}, \ldots, L_{i_n}$ be those languages queried in the scenario S, which have been answered 'no'. Let $L = \{w_{i_1}, \ldots, w_{i_n}\}$ be the set of the corresponding counterexamples. Notice that L is a finite subset of $\Sigma^* \setminus \{a\}$ and hence belongs to $\mathcal{C}'$. But, since all queries in the scenario S are answered truthfully with respect to L, M fails to learn L. So $\mathcal{C}' \notin$ *SupQ*.

(b) This follows from *(a)*, as Proposition 1 holds analogously for learning with extra queries. □

Considering the classes $\mathcal{C}'_{rsup}$ and $\mathcal{C}_{xsup}$ in *xSupQ* $\setminus$ *SupQ*, it is remarkable that for both classes there is a successful learner which uses restricted superset queries and which is additionally allowed to ask membership queries. So one might suspect that the full capabilities of *xSupQ*-learners can already be achieved by learners using restricted superset and membership queries. But this is not the case. Indeed, [7] provides a class in *xSupQ* which is not learnable with superset and membership queries. An adaptation of Proposition 1 for extra query learning implies an analogous result concerning subset queries.

It remains to compare learning with extra queries to the Gold-style learning types. A method similar to that used in the proof of Theorem 3 yields *xSupQ* $\subseteq$ *LimTxt*. But interestingly, there is an even stronger relation between Gold-style language learning from text and learning with extra superset queries. The learning capabilities of *conservative* IIMs and of extra superset query learners coincide. Note that, with Proposition 2 and Theorem 8, then *xSupQ* $\subset$ *LimTxt*.

Theorem 8 *xSupQ* $=$ *ConsvTxt*

Proof. '*ConsvTxt* $\subseteq$ *xSupQ*': Fix $\mathcal{C} \in$ *ConsvTxt*. Then there is an indexing $(L_i)_{i\in\mathbb{N}}$ comprising $\mathcal{C}$ and an IIM M, such that M *ConsvTxt*-identifies $\mathcal{C}$ with respect to $(L_i)_{i\in\mathbb{N}}$. Note that, if $L \in \mathcal{C}$ and t is a text for L, then M never returns an index i with $L \subset L_i$ on any initial segment of t.

Now the indexable class used for the queries contains all languages in $(L_i)_{i\in\mathbb{N}}$ and all languages $\Sigma^* \setminus \{w\}$ for $w \in \Sigma^*$. An *xSupQ*- learner M' identifying any $L \in \mathcal{C}$ may work as follows. M' uses queries concerning languages of the form $\Sigma^* \setminus \{w\}$ to construct a text for L and simulates M on the obtained text segments until M returns an index of a superset of L. This index is then returned by M'.

First, to effectively enumerate a text t for L, M' determines the set T of all words $w \in \Sigma^*$, for which the query representing $\Sigma^* \setminus \{w\}$ is answered with 'no'. Since $T = L$ and T is recursively enumerable, any recursive enumeration of T yields a text for L.

Second, to compute its hypothesis, M' executes steps 0, 1, 2, …until it receives a stop signal. In general, step n, $n \in \mathbb{N}$, consists of the following instructions:

> Determine $i := M(t_n)$, where t is a recursive enumeration of the set T. Pose a query referring to L_i.[2] If the answer is 'no', execute step $n+1$. Otherwise hypothesize the language L_i and stop. (* In the latter case, as M never hypothesizes a proper superset of L, M' returns an index for L. *)

Further details are omitted.

'*xSupQ* $\subseteq$ *ConsvTxt*': Fix some $\mathcal{C} \in$ *xSupQ*. Then there is an indexing $(L_i)_{i\in\mathbb{N}}$ comprising $\mathcal{C}$ and a query learner M, such that M learns $\mathcal{C}$ with extra superset queries in $(L_i)_{i\in\mathbb{N}}$. Define a new indexing $(L'_i)_{i\in\mathbb{N}}$:

- L'_0 is the empty language.
- If $i \geq 1$ and i is the canonical index[3] of the non-empty set $\{i_1, \ldots, i_n\}$, then $L'_i = L_{i_1} \cap \cdots \cap L_{i_n}$.

An IIM M' identifying $\mathcal{C}$ in the limit from text with respect to the hypothesis space $(L'_i)_{i\in\mathbb{N}}$, given a text t, may work as follows: $M'(t_0) := 0$. To compute $M'(t_{n+1})$, the learner M' interacts with M simulating a learning process for n steps. In step k, $k \leq n$, depending on how M' has replied to the previous queries posed by M, the learner M computes either (i) a new query i or (ii) a hypothesis i.

If case (i) occurs, then M' checks whether there is a word in *content*$(t_n) \setminus L_i$. If such a word, say w, is found, then M' provides the reply 'no', accompanied by w as a counterexample, to the query learner M. If no such word is found, then M' provides the reply 'yes'. If $k < n$, then M' proceeds with step $k+1$, otherwise M' returns the hypothesis $M'(t_{n+1}) := M'(t_n)$ and stops.

If case (ii) occurs, then M' computes the hypothesis $M'(t_{n+1})$ according to the following directives:

- Let $L_{i_1^+}, \ldots, L_{i_m^+}$ be the languages represented by the queries answered with 'yes' in the currently simulated scenario.
- Compute the canonical index i' of the set $\{i, i_1^+, \ldots, i_m^+\}$ and return the hypothesis $M'(t_{n+1}) = i'$.

It is not hard to verify that M' learns $\mathcal{C}$ in the limit from text; the relevant details are omitted. Moreover, as we will see next, M' avoids overgeneralized hypotheses, that means, if t is a text for some $L \in \mathcal{C}$, $n \in \mathbb{N}$, and $M'(t_n) = i'$, then $L'_{i'} \not\supset L$. Therefore, M' can easily be transformed into a learner M'' which identifies the class $\mathcal{C}$ conservatively in the limit from text.[4]

To prove that M' avoids overgeneralizations, assume to the contrary, that there is an $L \in \mathcal{C}$, a text t for L, and an $n \in \mathbb{N}$, such that the hypothesis $i' = M'(t_n)$ fulfills $L'_{i'} \supset L$. Then $i' \neq 0$. By definition of M', there must be a learning scenario S for M, in which

[2] Since one cannot ensure that L_i belongs to $\mathcal{C}$, this indicates why restricted superset and membership queries do not suffice for achieving the power of *xSupQ*.

[3] See [9] for the definition of canonical indices of finite sets.

[4] Note that a result in [12] states that any indexable class, which is learnable in the limit from text without overgeneralizations, belongs to *ConsvTxt*.

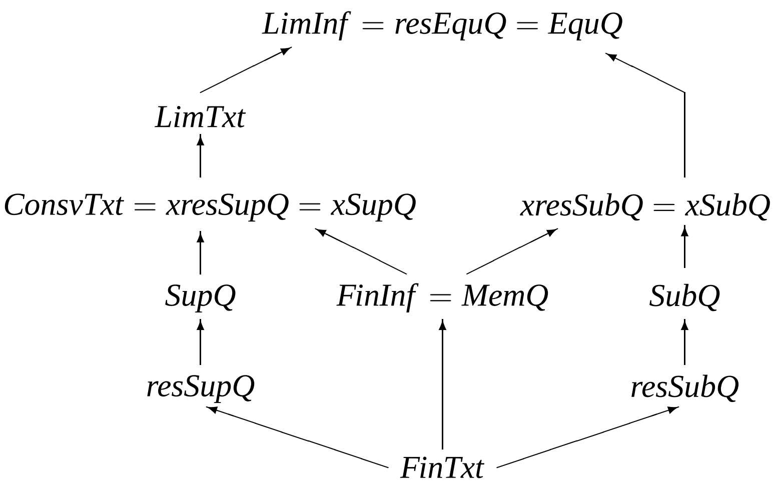
- M poses queries representing $L_{i_1^-}, \dots, L_{i_k^-}, L_{i_1^+}, \dots, L_{i_m^+}$ (in some particular order);
- the answers are 'no' concerning $L_{i_1^-}, \dots, L_{i_k^-}$ and 'yes' concerning $L_{i_1^+}, \dots, L_{i_m^+}$;
- afterwards M returns the hypothesis i.

Hence i' is the canonical index of $\{i, i_1^+, \dots, i_m^+\}$. This implies $L'_{i'} = L_i \cap L_{i_1^+} \cap \dots \cap L_{i_m^+}$. As by assumption $L'_{i'} \supset L$, each of the languages $L_{i_1^+}, \dots, L_{i_m^+}$ is a superset of L. By definition of M', $L_{i_j^-} \not\supseteq$ *content*(t_n) for $1 \leq j \leq k$. Therefore none of the languages $L_{i_1^-}, \dots, L_{i_k^-}$ are supersets of L. So the answers in the learning scenario S above are truthful respecting the language L. As M learns $\mathcal{C}$ with extra superset queries, the hypothesis i must be correct for L, i. e., $L_i = L$. This yields $L'_{i'} \subseteq L$ in contradiction to $L'_{i'} \supset L$.

So M' learns $\mathcal{C}$ in the limit from text without overgeneralizations, which—by the argumentation above—implies $\mathcal{C} \in$ *ConsvTxt*. □

5 Summary

The following figure summarizes the observed comparison results. Concerning the indicated results not proved yet, note that *MemQ* $\subseteq$ *xSupQ* and *MemQ* $\subseteq$ *xSubQ* follow from the fact that each membership query for a word w can be simulated by a superset query for $\Sigma^* \setminus \{w\}$ or a subset query for $\{w\}$. An example for a class in *resSupQ* $\setminus$ *xSubQ* is the class of all finite languages.



Each learning type is represented as a vertex in a digraph. A directed path from A to B indicates that the learning type B outperforms type A, i. e., B is a proper superset of A. A missing path between A and B means that A and B are incomparable, i. e., there are learning problems solvable with respect to the learning type A, but not solvable with respect to the learning type B, and vice versa.

References

[1] D. Angluin. Inductive inference of formal languages from positive data. *Information and Control* 45:117–135, 1980.

[2] D. Angluin. Queries and concept learning, *Machine Learning* 2:319–342, 1988.

[3] D. Angluin. Queries revisited. In: *Proc. 12th Int. Conference on Algorithmic Learning Theory*, LNAI 2225, 12–31, Springer-Verlag, 2001.

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

[5] J. E. Hopcroft, J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley Publishing Company, 1979.

[6] S. Lange. *Algorithmic Learning of Recursive Languages*, Mensch und Buch Verlag, 2000.

[7] S. Lange, S. Zilles. Replacing limit learners with equally powerful one-shot query learners. In: *Proc. 17th Annual Conference on Learning Theory*, LNAI, Springer-Verlag, 2004, to appear.

[8] J. Nessel, S. Lange. Learning erasing pattern languages with queries. *In: Proc. 11th Int. Conference on Algorithmic Learning Theory*, LNAI 1968, 86–100, Springer-Verlag, 2000.

[9] H. Rogers. *Theory of Recursive Functions and Effective Computability*, MIT Press, 1987.

[10] E. Shapiro. *Algorithmic Program Debugging*, MIT Press, 1983.

[11] L. Valiant. A theory of the learnable. *Communications of the ACM* 27:1134–1142, 1984.

[12] T. Zeugmann, S. Lange. A guided tour across the boundaries of learning recursive languages. In: *Algorithmic Learning for Knowledge-Based Systems*, LNAI 961, 190–258, Springer-Verlag, 1995.