A general comparison of language learning from examples and from queries

Sanjay Jain [a,1], Steffen Lange [b], and Sandra Zilles [c,*]

[a] *School of Computing, National University of Singapore, Singapore 117543*

[b] *Fachbereich Informatik, Hochschule Darmstadt, Haardtring 100, 64295 Darmstadt, Germany*

[c] *DFKI GmbH, Erwin-Schrödinger-Straße, 67663 Kaiserslautern, Germany*

Abstract

In language learning, strong relationships between Gold-style models and query models have recently been observed: in some quite general setting Gold-style learners can be replaced by query learners and vice versa, without loss of learning capabilities. These 'equalities' hold in the context of learning indexable classes of recursive languages.

Former studies on Gold-style learning of such indexable classes have shown that, in many settings, the enumerability of the target class and the recursiveness of its languages are crucial for learnability. Moreover, studying query learning, non-indexable classes have been mainly neglected up to now. So it is conceivable that the recently observed relations between Gold-style and query learning are not due to common structures in the learning processes in both models, but rather to the enumerability of the target classes or the recursiveness of their languages.

In this paper, the analysis is lifted onto the context of learning arbitrary classes of recursively enumerable languages. Still, strong relationships between the approaches of Gold-style and query learning are proven, but there are significant changes to the former results. Though in many cases learners of one type can still be replaced by learners of the other type, in general this does not remain valid vice versa. All results hold even for learning classes of recursive languages, which indicates that the recursiveness of the languages is not crucial for the former 'equality' results. Thus we analyze how constraints on the algorithmic structure of the target class affect the relations between two approaches to language learning.

Key words: Inductive inference, query learning, formal languages, recursion theory

1 Introduction

In order to model different aspects of human learning and machine learning, different abstract approaches have to be considered. Each model analyzed within the scope of learning theory addresses only special facets of learning. For example, in Gold's [9] model of *identification in the limit* learning is interpreted as a limiting process of generating and improving hypotheses about a target concept. These hypotheses are built upon instances of the target concept offered to the learner. In the limit, the output of the learner is supposed to stabilize on a correct guess, but during the learning process one never knows whether or not the current hypothesis is already correct. The potential of changing its mind is a crucial quality of the learner.

In contrast to that, Angluin's [3,4] model of *query learning* is concerned with learning as a finite process in which a learner and a teacher interact. The learner asks questions of a specified type about the target concept and the teacher answers these reliably. After finitely many steps the learner is required to return a single hypothesis, which then correctly describes the target concept. Here the crucial characteristics of the learner are its access to special information on the target concept and its restrictions in terms of mind changes. Since a query learner identifies the target concept with just a single hypothesis, we allude to this scheme as *one-shot learning.* [2]

Recently, the combination of these two approaches, see Jain and Kinber [11,12] as well as the common features of learners in either model, see Lange and Zilles [17,18] have gained interest in the learning theory community. Lange and Zilles [17,18] contributes a systematic analysis of common features of both approaches, thereby focusing on the identification of formal languages, ranging over indexable classes of recursive languages, as target concepts, see Angluin [2], Lange and Zeugmann [16], and Zeugmann and Lange [25]. Characterizing different types of Gold-style language learning in terms of query learning has pointed out correspondences between the two models. In particular, Lange and Zilles [17,18] demonstrate how learners identifying languages in the limit can be replaced by one-shot learners without loss of learning power—and vice versa. That means, under certain circumstances the capabilities of limit learners are equal to those of one-shot learners using queries. An important parameter in this context is the range of possible hypothesis spaces/query

* corresponding author
Email addresses: `sanjay@comp.nus.edu.sg` (Sanjay Jain),
`s.lange@fbi.h-da.de` (Steffen Lange), `sandra.zilles@dfki.de` (Sandra Zilles).
[1] Sanjay Jain was supported in part by NUS grant numbers R252–000–127–112 and R252–000–212–112.
[2] Most studies on query learning mainly deal with the efficiency of query learners, whereas, in what follows, we are only interested in qualitative learnability results.

spaces used during the learning process. Despite the fundamental differences in the definitions of the two learning paradigms, there are strong relations—at least in the case of learning indexable families of recursive languages.

The latter restriction had initially been made, since many natural language classes are indexable. Former studies on Gold-style learning of indexable classes of languages, cf. Zeugmann and Lange [25], have shown that, in many settings, the enumerability of the target class may be the crucial reason for positive learnability results. Moreover, when studying query learning, non-indexable classes have been mainly neglected up to now. So it is conceivable that the strong relationships between Gold-style and query learning observed in [17,18] are not caused by common structures in the learning processes in both models, but rather by the enumerability of the target classes or maybe at least by the recursiveness of the target languages themselves. In order to determine the actual cause for the relationships observed before, we now lift the analysis thereof onto more complex classes of languages.

Therefore the current paper concerns the relationships of Gold-style learning and query learning for the case that arbitrary classes of recursively enumerable (r.e.) languages form the target. This is additionally based on the following observation: when trying to learn a class of recursive languages, a certain type of learner may sometimes be successful only in case the learner uses a hypothesis space comprising more than the languages to be learned—such as for instance a hypothesis space given by an r.e. indexing of r.e. languages. Then a natural question might be whether it is possible to learn not only the initial target class, but additionally the languages represented by further queries a learner asks or further hypotheses a learner states during learning the initial target languages. This again leads to the problem of learning r.e. languages. Literature, see e.g. de Jongh and Kanazawa [7], offers more examples of lifting results on learning recursive languages, as contributed by Angluin [2], to learning r.e. languages. Note that non-proper learning, i.e., learning with hypothesis spaces strictly comprising the class of target concepts, has been studied extensively in different fields of Algorithmic Learning Theory, such as in query learning, see Angluin [4], and in probably approximately correct (PAC) learning as introduced by Valiant [24], see Kearns and Vazirani [15]. For instance, Kearns et al. [14] as well as Pitt and Valiant [21] have proven that the class of all k-term DNFs

- cannot be PAC-learned efficiently using k-term DNFs as hypotheses, if RP $\neq$ NP (see Motwani and Raghavan [20] for more details on complexity classes like RP and NP and their relations), but
- can be PAC-learned efficiently using the strictly class-comprising hypothesis space of all k-CNFs.

From now on assume that arbitrary classes of r.e. languages form the target classes. Below we prove that in almost all cases, where equivalences between two learning models A and B had been witnessed for learning indexable classes of recursive languages, learners of type A can be replaced by learners of type B without loss of learning power—but no longer vice versa. So, although most of the equivalences between Gold-style models and query models no longer hold, at least some of the inclusions hold, thereby forming a hierarchy of inference types. This shows that huge parts of the relationships shown for learning indexable classes of recursive languages are maintained; the cause must be common structures of learning processes in Gold-style and query learning! An important parameter in the final hierarchy is again the underlying hypothesis space/query space.

Interestingly, all separations of inference types in the final hierarchy can be witnessed even by (non-indexable) classes of *recursive* languages. This raises the question whether the main reason for the equivalence results in [17,18] is the fact that the classes considered are enumerable and not that the languages themselves are recursive. So we analyzed whether the results in [17,18] can be lifted to the case of learning enumerable classes of r.e. languages. The relationships observed are somewhat dismal: several of the equivalence results do *not* hold for learning enumerable classes of r.e. languages, but at least one of them does. That means that in most but not in all cases, the main reason for the equivalence results in [17,18] lies not only in the enumerability of the target classes.

A preliminary version of this paper appeared in [13].

2 Preliminaries

Familiarity with standard recursion theoretic and language theoretic notions is assumed, see Rogers [22] and Hopcroft and Ullman [10]. $\mathbb{N}$ denotes the set of natural numbers $\{0, 1, 2, \ldots\}$. From now on, a fixed finite alphabet Σ with $\{a, b\} \subseteq \Sigma$ is given. A *word* is any element from the set Σ^* of all finite strings over Σ and a *language* any subset of Σ^*. The *complement* of a language L, denoted $\overline{L}$, is the set $\Sigma^* \setminus L$. Any total function $t : \mathbb{N} \to \Sigma^*$ with $\{t(i) \mid i \in \mathbb{N}\} = L$ is called a *text* for L. A text t is often identified with an infinite sequence $(t(i))_{i\in\mathbb{N}}$. Then, given $n \in \mathbb{N}$, t_n is the initial segment $(t(0), \ldots, t(n))$ and $content(t_n)$ denotes the set $\{t(0), \ldots, t(n)\}$. An initial segment of text for L is also called a finite text segment of L.

For any set X, $\mathrm{card}(X)$ denotes the cardinality of X, where $\mathrm{card}(X) = \infty$ for infinite X. Given two sets X and Y, the notion $X\#Y$ is used to indicate that X and Y are incomparable, i.e., neither $X \subseteq Y$ nor $Y \subseteq X$ holds.

In the sequel, φ is a Gödel numbering of all partial recursive functions and $K = \{i \in \mathbb{N} \mid \varphi_i(i) \text{ is defined}\}$. The language family $(W_i)_{i\in\mathbb{N}}$ is given by $W_i = \{w_j \mid \varphi_i(j) \text{ is defined}\}$ for all $i \in \mathbb{N}$, where $(w_j)_{j\in\mathbb{N}}$ is a repetition-free effective enumeration of Σ^*. Then $W_{i,s}$, $s \in \mathbb{N}$, is the set of all words w_j, such that $j < s$ and $\varphi_i(j)$ terminates within s steps. Given $A \subseteq \mathbb{N}$, an A-recursive function is a function recursive using an oracle for the set A.

A family $(A_i)_{i\in\mathbb{N}}$ of languages is *uniformly recursive* (*uniformly r.e.*) if there is a recursive (partial recursive) function f such that $A_i = \{w \in \Sigma^* \mid f(i,w) = 1\}$ for all $i \in \mathbb{N}$. For uniformly recursive families membership is uniformly decidable. A family $(A_i)_{i\in\mathbb{N}}$ is *uniformly K-r.e.*, if there is a recursive function g such that $A_i = \{w \in \Sigma^* \mid g(i,w,n) = 1 \text{ for all but finitely many } n\}$ for all $i \in \mathbb{N}$. A class $\mathcal{C}$ of recursive languages over Σ^* is called an *indexable class of recursive languages* (or *indexable class* for short), if there is a uniformly recursive family $(L_i)_{i\in\mathbb{N}}$ of all and only the languages in $\mathcal{C}$.

2.1 Gold-style language learning

Let $\mathcal{C}$ be a class of r.e. languages, $\mathcal{H} = (A_i)_{i\in\mathbb{N}}$ a language family (a *hypothesis space*). An *inductive inference machine* (*IIM* for short) M is an algorithmic device that reads longer and longer initial segments σ of a text and outputs numbers $M(\sigma)$. Returning i, M is construed to hypothesize the language A_i.

The following definition of learning in the limit is based on Gold [9]. Given a text t for $L \in \mathcal{C}$, M *identifies L from t with respect to $\mathcal{H} = (A_i)_{i\in\mathbb{N}}$ in the limit*, 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 $A_i = L$. M *identifies $\mathcal{C}$ in the limit from text* with respect to $\mathcal{H}$, if it identifies every $L' \in \mathcal{C}$ from every text for L'.

In what follows, we focus our studies on uniformly r.e. families as hypothesis spaces. $\mathit{LimTxt}_{\text{r.e.}}$ denotes the collection of all classes $\mathcal{C}'$ for which there is a uniformly r.e. hypothesis space $\mathcal{H}$ and an IIM M' identifying $\mathcal{C}'$ in the limit from text with respect to $\mathcal{H}$.

A quite natural and often studied modification of $\mathit{LimTxt}_{\text{r.e.}}$ is defined by the model of *conservative inference*, see Angluin [2] and Lange and Zeugmann [16] for this concept in the context of learning recursive languages. M is a *conservative* IIM for $\mathcal{C}$ with respect to $\mathcal{H} = (A_i)_{i\in\mathbb{N}}$, 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 $w \notin A_i$ before returning j. The collection of all classes identifiable from text by a conservative IIM with respect to some uniformly r.e. hypothesis space is denoted by $\mathit{ConsvTxt}_{\text{r.e.}}$.

Note that $ConsvTxt_{\text{r.e.}} \subset LimTxt_{\text{r.e.}}$, as witnessed by the indexable class used by Zeugmann and Lange [25] to separate *LimTxt*-learnable indexable classes from *ConsvTxt*-learnable indexable classes.

Another often studied version of Gold-style language learning is behaviorally correct learning, cf. Case and Lynes [6]: If $\mathcal{C}$ is a class of r.e. languages, $\mathcal{H} = (A_i)_{i \in \mathbb{N}}$ any hypothesis space, M an IIM, then M is a *behaviorally correct* learner for $\mathcal{C}$ from text with respect to $\mathcal{H}$, if for each $L \in \mathcal{C}$ and each text t for L, for all but finitely many n, $A_{M(t_n)} = L$ is fulfilled. Here M may alternate different correct hypotheses arbitrarily often instead of converging to a single hypothesis. Defining the notion $BcTxt_{\text{r.e.}}$ as usual yields $BcTxt_{\text{r.e.}} \supset LimTxt_{\text{r.e.}}$, see Case and Lynes [6].

Since we analyze learning from text, we assume in the sequel that all target languages are *non-empty*.

One main aspect of human learning is modeled in the approach of learning in the limit: the ability to change one's mind. Thus learning is a process in which the learner may change its hypothesis arbitrarily often until reaching its final correct guess. In particular, it is in general impossible to find out when the final hypothesis has been reached, i.e., when a success in learning has eventuated.

The main concern of our analysis will be comparisons of such inference types to query learning models resulting in a hierarchy reflecting the capabilities of the corresponding learners.

Finally, note that each class in $LimTxt_{\text{r.e.}}$, $ConsvTxt_{\text{r.e.}}$, $BcTxt_{\text{r.e.}}$ can be learned using the hypothesis space $(W_i)_{i \in \mathbb{N}}$. This is possible because for any uniformly r.e. hypothesis space $(A_i)_{i \in \mathbb{N}}$ there is a recursive 'compiler' function c, such that $A_i = W_{c(i)}$ for all $i \in \mathbb{N}$. Thus hypotheses with respect to $(A_i)_{i \in \mathbb{N}}$ can be easily transformed into hypotheses with respect to $(W_i)_{i \in \mathbb{N}}$. We will use this property in our proofs below.

Our notions of learning are closely related to the notion of *stabilizing sequences*, see Fulk [8], which will be crucial for our results as well. If $\mathcal{H} = (A_i)_{i \in \mathbb{N}}$ is a hypothesis space, M an IIM, and L a language, then any finite text segment σ of L is called a *LimTxt*-stabilizing sequence (a *BcTxt*-stabilizing sequence) for M, L, and $\mathcal{H}$, if $M(\sigma) = M(\sigma\sigma')$ ($A_{M(\sigma)} = A_{M(\sigma\sigma')}$) for all finite text segments σ' of L. If L is *LimTxt*-learned by M (*BcTxt*-learned by M) with respect to $\mathcal{H}$, then there exists a *LimTxt*-stabilizing sequence (a *BcTxt*-stabilizing sequence) for M, L, and $\mathcal{H}$.

2.2 Language learning via queries

In the query learning model, a learner has access to a teacher 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, see Angluin [3]. Its queries and hypotheses are coded as natural numbers; both will be interpreted with respect to an underlying *hypothesis space*. We adapt Angluin's original definition here for learning r.e. languages as follows: when learning a class $\mathcal{C}$ of r.e. languages, any family $(A_i)_{i\in\mathbb{N}}$ of languages may form a hypothesis space.

More formally, let $\mathcal{C}$ be a class of r.e. languages, let $L \in \mathcal{C}$, let $\mathcal{H} = (A_i)_{i\in\mathbb{N}}$ be a hypothesis space, 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, represents L, i.e., $A_i = L$. So M returns its unique and correct guess i after finitely many queries. Moreover, *M learns $\mathcal{C}$ with respect to $\mathcal{H}$ using some type of queries*, if it learns every language in $\mathcal{C}$ with respect to $\mathcal{H}$ using queries of the specified type. If L is a target language, a query learner M may ask:

Restricted superset queries. The input is an index of a language L' in $\mathcal{H}$. The answer is 'yes' or 'no', depending on whether or not L' is a superset of L.

Restricted disjointness queries. The input is an index of a language L' in $\mathcal{H}$. The answer is 'yes' or 'no', depending on whether or not L' and L are disjoint.

The term 'restricted' is used to distinguish these inference types from learning with superset (disjointness) queries, where, with each negative reply to a query j the learner is provided a counterexample, i.e., a word in $L \setminus A_j$ (in $L \cap A_j$).

$SupQ_{\text{r.e.}}$ and $DisQ_{\text{r.e.}}$ denote the collections of all classes $\mathcal{C}'$ for which there is a uniformly r.e. hypothesis space $\mathcal{H}$ and a query learner M' learning $\mathcal{C}'$ with respect to $\mathcal{H}$ using restricted superset and restricted disjointness queries, respectively. In the sequel we will again without loss of generality assume that $SupQ_{\text{r.e.}}$-learners and $DisQ_{\text{r.e.}}$-learners always use the hypothesis space $\mathcal{H} = (W_i)_{i\in\mathbb{N}}$.[3] In the literature, see Angluin [3,4], more types of queries, such as (restricted) subset queries, membership queries, and (restricted) equivalence queries have been analyzed, but in what follows we concentrate on the two types explained above. Obviously, restricted superset and restricted disjointness queries are in general not decidable, i.e., the teacher may be non-computable.

[3] Again note that for any uniformly r.e. hypothesis space $(A_i)_{i\in\mathbb{N}}$ there is a recursive 'compiler' function c, such that $A_i = W_{c(i)}$ for all $i \in \mathbb{N}$. Thus queries and hypotheses with respect to $(A_i)_{i\in\mathbb{N}}$ can be transformed into queries and hypotheses with respect to $(W_i)_{i\in\mathbb{N}}$.

Note that, in contrast to the models of Gold-style language learning introduced above, learning via queries focuses the aspect of one-shot learning, i.e., it is concerned with scenarios in which learning occurs without mind changes.

3 Learning indexable classes of recursive languages

Numerous studies of language learning restrict their focus to indexable classes, since, first, these include many natural classes of languages, and second, many conceptions can be simplified in this context. In particular, uniformly recursive families may be considered as hypothesis spaces in the approaches of both Gold-style and query learning (indicated by a subscript 'rec' instead of 'r.e.'). In this section, all results referred to hold for indexable classes only. Recent studies [17,18] have shown astonishing relations between the two approaches witnessed by equivalences of pairs of inference types, such as $SupQ_{\text{rec}} = DisQ_{\text{rec}} = ConsvTxt_{\text{r.e.}}$ ($= ConsvTxt_{\text{rec}}$, see a result by Jain in [19]) and $DisQ_{\text{r.e.}} = LimTxt_{\text{r.e.}}$. In these equalities, all inference types are considered restricted to indexable classes.

Concerning characterizations of $SupQ_{\text{r.e.}}$ and $BcTxt_{\text{rec}}$ by similar means, oracle-IIMs as well as more general hypothesis spaces have been useful. Firstly, an oracle-IIM is an IIM which is recursive relative to an arbitrary oracle, i.e. its computation depends on according to which oracle it currently accesses, see e.g. Stephan [23]. For instance, using a K-oracle, such an IIM M becomes a K-recursive IIM M^K. Thus, e.g., $ConsvTxt_{\text{r.e.}}[K]$ denotes the collection of classes $ConsvTxt_{\text{r.e.}}$-learnable with the help of a K-oracle. Restricting such inference types to indexable classes, one obtains for instance $ConsvTxt_{\text{r.e.}}[K] = SupQ_{\text{r.e.}}$.

Secondly, in order to characterize $BcTxt_{\text{r.e.}}$, uniformly K-r.e. hypothesis spaces have been introduced for query learning, indicated by a subscript 'K-r.e.' as in $SupQ_{K\text{-r.e.}}$. This has lead to the result $SupQ_{K\text{-r.e.}} = DisQ_{K\text{-r.e.}} = BcTxt_{\text{r.e.}}$.

Figure 1 gives a summary of the relationship of these inference types restricted to indexable classes.

4 Learning classes of r.e. languages

In the sequel, a hierarchy as in Figure 1 is established for arbitrary classes of r.e. languages. A section on query learning with uniformly r.e. hypothesis spaces is followed by a section dealing with uniformly K-r.e. hypothesis spaces. Note that the usage of uniformly recursive families as hypothesis spaces is, in general, inapplicable here and are thus not considered further.

$$
\begin{gathered}
\mathit{BcTxt}_{\text{r.e.}} = \mathit{LimTxt}_{\text{rec}}[K] = \mathit{LimTxt}_{\text{r.e.}}[K] = \\
\mathit{SupQ}_{K\text{-r.e.}} = \mathit{DisQ}_{K\text{-r.e.}} \\
\uparrow \\
\mathit{ConsvTxt}_{\text{r.e.}}[K] = \mathit{SupQ}_{\text{r.e.}} \\
\uparrow \\
\mathit{LimTxt}_{\text{r.e.}} = \mathit{LimTxt}_{\text{rec}} = \\
\mathit{ConsvTxt}_{\text{rec}}[K] = \mathit{DisQ}_{\text{r.e.}} \\
\uparrow \\
\mathit{ConsvTxt}_{\text{r.e.}} = \mathit{SupQ}_{\text{rec}} = \mathit{DisQ}_{\text{rec}}
\end{gathered}
$$

Fig. 1. This graph illustrates the relations between different inference types restricted to indexable classes as studied in [17,18]. Arrows indicate proper inclusions of inference types.

4.1 Results for uniformly r.e. hypothesis spaces

Our first comparison already yields a change to the former hierarchy: when learning arbitrary classes of r.e. languages, $\mathit{ConsvTxt}_{\text{r.e.}}$-learners can in general no longer be replaced by $\mathit{DisQ}_{\text{r.e.}}$-learners.

Theorem 1 $\mathit{DisQ}_{\text{r.e.}} \# \mathit{ConsvTxt}_{\text{r.e.}}$.

Proof. Two statements have to be verified:

Part 1. $\mathit{DisQ}_{\text{r.e.}} \setminus \mathit{ConsvTxt}_{\text{r.e.}} \neq \emptyset$.
Part 2. $\mathit{ConsvTxt}_{\text{r.e.}} \setminus \mathit{DisQ}_{\text{r.e.}} \neq \emptyset$.

Proof of Part 1. This follows immediately from Figure 1.

Proof of Part 2. In order to prove that $\mathit{ConsvTxt}_{\text{r.e.}} \setminus \mathit{DisQ}_{\text{r.e.}} \neq \emptyset$, consider the following class $\mathcal{C}$.

Let $L_i = \{a^i b^x \mid x \in \mathbb{N}\}$. Let $L_i^S = \{a^i b^x \mid x \in S\}$ for any set $S \subseteq \mathbb{N}$.

Let $\mathcal{C}_1 = \{L_i^S \mid i \in \mathbb{N}, \text{card}(S) < \infty, (\exists e)\ [\text{card}(S \cap \{x \mid x \leq 2e\}) > e + 1]\}$.

Below, we will define a recursive function f such that, for all i, the following two properties hold:

(a) $W_{f(i)} \subseteq L_i$ and $W_{f(i)}$ is recursive (although an index for the characteristic function of $W_{f(i)}$ in general cannot be obtained from i);
(b) For all e, $\text{card}(W_{f(i)} \cap \{a^i b^x \mid x \leq 2e\}) \leq e + 1$.

Let $\mathcal{C}_2 = \{W_{f(i)} \mid i \in \mathbb{N}\}$. Let

$$\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2 \,.$$

Thus, $\mathcal{C}$ is uniformly r.e. and consists only of recursive languages (however, $\mathcal{C}$ is not an indexed family).

$\mathcal{C} \in \mathit{ConsvTxt}_{\text{r.e.}}$ is witnessed by an IIM M which, on target L, first acquires an i with $L \subseteq L_i$. M outputs $f(i)$, until an e with

$$\mathrm{card}(\{x \mid a^i b^x \in L, x \leq 2e\}) > e + 1$$

is found. In the latter case M runs the learning procedure for finite sets.

Let $(M_i)_{i\in\mathbb{N}}$ be an enumeration of all $\mathit{DisQ}_{\text{r.e.}}$-learners. We now define f, such that (a) and (b) above are fulfilled and for each i, M_i either does not $\mathit{DisQ}_{\text{r.e.}}$-identify $W_{f(i)}$, or it does not $\mathit{DisQ}_{\text{r.e.}}$-identify $\mathcal{C}_1$.

For any $i, s \in \mathbb{N}$, let $W^s_{f(i)}$ denote the subset of $W_{f(i)}$ enumerated before stage s. Let $W^0_{f(i)} = \{a^i\}$, i.e., the word a^i is enumerated in $W_{f(i)}$ *before* stage 0. Go to stage 0.

In general, stage s reads as follows.

– Step 1:
  Run M_i for s steps, where each restricted disjointness query j (representing W_j) of M_i is answered 'yes', if $W_{j,s} \cap W^s_{f(i)} = \emptyset$; 'no' otherwise.
– Step 2:
  If M_i does not output a hypothesis within s steps, go to stage $s + 1$.
  Else dovetail steps 2.1 and 2.2 until one of them succeeds. If 2.1 succeeds before 2.2, then go to stage $s + 1$, else if 2.2 succeeds, then go to step 3.
  (2.1) Find a restricted disjointness query j from step 1 which was answered 'yes', but $W_j \cap W^s_{f(i)} \neq \emptyset$.
  (2.2) Find a restricted disjointness query j from step 1 which was answered 'yes', and $a^i b^y \in W_j$ for some $y > 2s$.
- Step 3:
  Let j, y be as found in step 2.2. Enumerate $a^i b^y$ in $W_{f(i)}$ and go to stage $s + 1$ (otherwise stage s never ends).

Fix i. By construction, $W_{f(i)}$ fulfills the conditions (a) (as either $W_{f(i)}$ is finite, or $a^i b^s \in W_{f(i)}$, iff it is enumerated in $W_{f(i)}$ before stage s) and (b) (as at most $s + 1$ elements are enumerated before stage s, and every element enumerated at or after stage s is of form $a^i b^y$ for some $y > 2s$). We consider two cases.

Case 1: Stage s starts but does not finish.

In this case clearly $W_{f(i)}$ is finite. Now, since step 2.1 did not succeed, all questions of M_i in step 1 above for the input being $W_{f(i)} = W^s_{f(i)}$, are answered correctly at stage s, and M_i outputs a hypothesis on $W_{f(i)}$. Furthermore, all questions j of M_i on $W_{f(i)}$ which are answered 'yes', have the property that $W_j \cap L_i$ is finite (since step 2.2 did not succeed).

Thus, there exists a finite set S with $L_i^S \in \mathcal{C}_1$ such that M_i behaves the same way on L_i^S as it does on $W_{f(i)}$. To see this, let $S = \{x \mid a^i b^x \in W_{f(i)}\} \cup \{z \mid e \leq z \leq 2e\}$, where $e = 1 + \max(\{y \mid a^i b^y \in W_j$ for some query j asked by M_i on input $W_{f(i)}$, and answered 'yes'$\})$ (* note that for each question j asked by M_i on input $W_{f(i)}$ and answered 'yes', $W_j \cap L_i$ is finite *).

Now, M_i can $DisQ_{\text{r.e.}}$-identify at most one of the languages $W_{f(i)}$ and L_i^S, both of which are in $\mathcal{C}$.

Case 2: Every stage s ends.

Consider the following subcases:

Case 2.1: M_i on $W_{f(i)}$ asks infinitely many questions or never outputs a hypothesis.

In this case clearly, M_i does not $DisQ_{\text{r.e.}}$-identify $W_{f(i)} \in \mathcal{C}$.

Case 2.2: Not case 2.1.

In this case let stage s be large enough so that, if j is a question asked by M_i on $W_{f(i)}$ (when all the questions are answered correctly), and $W_j \cap W_{f(i)} \neq \emptyset$, then $W_{j,s} \cap W^s_{f(i)} \neq \emptyset$. Note that then beyond stage s all questions of M_i are answered correctly in step 1. Now step 2.1 and 2.2 cannot succeed. Thus the only way infinitely many stages can exist is by M_i not returning any hypothesis. The latter yields a contradiction.

From the above cases it follows that M_i does not $DisQ_{\text{r.e.}}$-identify $\mathcal{C}$.

Thus we obtain $DisQ_{\text{r.e.}} \# ConsvTxt_{\text{r.e.}}$. □

This incoherency holds since presently $DisQ_{\text{r.e.}}$ no longer equals $LimTxt_{\text{r.e.}}$. However, an inclusion as in Theorem 2 still indicates a relation between Gold-style and query learning, albeit weaker than when restricted to indexable classes.

Theorem 2 $DisQ_{\text{r.e.}} \subset LimTxt_{\text{r.e.}}$.

Proof. For the inclusion the corresponding proof for indexable classes from Theorem 5 in [19] can be adopted. The inequality follows from Theorem 1 and

$ConsvTxt_{\text{r.e.}} \subseteq LimTxt_{\text{r.e.}}$. □

The relationship between $LimTxt_{\text{r.e.}}$ and $SupQ_{\text{r.e.}}$ remains unchanged from the former hierarchy, as Theorem 3 shows.

Theorem 3 $LimTxt_{\text{r.e.}} \subset SupQ_{\text{r.e.}}$.

Proof. Two statements have to be verified:

Part 1. $LimTxt_{\text{r.e.}} \subseteq SupQ_{\text{r.e.}}$.
Part 2. $SupQ_{\text{r.e.}} \setminus LimTxt_{\text{r.e.}} \neq \emptyset$.

Proof of Part 1. In order to prove $LimTxt_{\text{r.e.}} \subseteq SupQ_{\text{r.e.}}$, suppose $\mathcal{C}$ is a class of r.e. languages in $LimTxt_{\text{r.e.}}$. Let M be an IIM identifying $\mathcal{C}$ in the limit with respect to $(W_i)_{i\in\mathbb{N}}$.

Suppose $L \in \mathcal{C}$ is the target language. A *SupQ*-learner M' for L with respect to $(W_i)_{i\in\mathbb{N}}$ is defined to carry out the following instructions, starting in stage 0. Note that Gödel numbers (representations in $(W_i)_{i\in\mathbb{N}}$) can be computed for all queries to be asked. Stage n reads as follows:

- Ask superset queries for $\Sigma^* \setminus \{w_0\}, \ldots, \Sigma^* \setminus \{w_n\}$. Let $L_{[n]}$ be the set of words w_x, $x \leq n$, for which the corresponding query is answered with 'no'. (* Note that $L_{[n]} = L \cap \{w_x \mid x \leq n\}$. *)
- Let $(\sigma^n_x)_{x\in\mathbb{N}}$ be an effective enumeration of all finite segments of texts for $L_{[n]}$. For all $x, y \leq n$ pose a superset query for $W_{M(\sigma^y_x)}$ and thus build $\text{Cand}_n = \{\sigma^y_x \mid x, y \leq n \text{ and } W_{M(\sigma^y_x)} \supseteq L\}$ from the queries answered with 'yes'.
- For all $\sigma \in \text{Cand}_n$, pose a superset query for the r.e. language

  $$W'_\sigma = \begin{cases} \Sigma^*, & \text{if } M(\sigma\sigma') \neq M(\sigma) \text{ for some text segment } \sigma' \text{ of } W_{M(\sigma)}, \\ \emptyset, & \text{otherwise}. \end{cases}$$

  (* $W'_\sigma \not\supseteq L$ iff σ is a *LimTxt*-stabilizing sequence for M and $W_{M(\sigma)}$. *)
  If all these queries are answered 'yes', then go to stage $n+1$. Otherwise, if $\sigma \in \text{Cand}_n$ is minimal with $W'_\sigma \not\supseteq L$, then hypothesize $M(\sigma)$ and stop.

M' identifies L with superset queries with respect to $(W_i)_{i\in\mathbb{N}}$, because (i) M' eventually returns a hypothesis and (ii) this hypothesis is correct for L. To prove (i), note that M is a *LimTxt*-learner for L with respect to $(W_i)_{i\in\mathbb{N}}$. So there are i, x, y such that $M(\sigma^y_x) = i$, $W_i = L$, and σ^y_x is a *LimTxt*-stabilizing sequence for M and L. Then $W'_{\sigma^y_x} = \emptyset$ and the corresponding superset query is answered with 'no'. Thus M' returns a hypothesis. To prove (ii), assume M' returns a hypothesis representing $W_{M(\sigma)}$ for some text segment σ of L. Then, by definition of M', $L \subseteq W_{M(\sigma)}$ and σ is a *LimTxt*-stabilizing sequence for M and $W_{M(\sigma)}$. In particular, σ is a *LimTxt*-stabilizing sequence for M and L.

Since M learns L in the limit from text, this implies $L = W_{M(\sigma)}$. Hence the hypothesis M' returns is correct for L.

Therefore $\mathcal{C} \in \mathit{SupQ}_{\text{r.e.}}$ and $\mathit{LimTxt}_{\text{r.e.}} \subseteq \mathit{SupQ}_{\text{r.e.}}$.

Proof of Part 2. $\mathit{SupQ}_{\text{r.e.}} \setminus \mathit{LimTxt}_{\text{r.e.}} \neq \emptyset$ is even witnessed by a uniformly recursive family of languages, see the proof of Theorem 7 in [19]. □

Interestingly, the characterization $\mathit{SupQ}_{\text{r.e.}} = \mathit{ConsvTxt}_{\text{r.e.}}[K]$ persists when learning classes of r.e. languages. Here the proof for indexable classes, see Theorem 9 in [19], applies.

Theorem 4 $\mathit{SupQ}_{\text{r.e.}} = \mathit{ConsvTxt}_{\text{r.e.}}[K]$.

Though $\mathit{SupQ}_{\text{r.e.}} \subset \mathit{LimTxt}_{\text{r.e.}}[K]$ persists (Theorem 5), the relation between $\mathit{SupQ}_{\text{r.e.}}$ and $\mathit{BcTxt}_{\text{r.e.}}$ changes significantly for arbitrary classes of r.e. languages, see Theorem 7. The reason is that $\mathit{LimTxt}_{\text{r.e.}}[K]$ no longer equals $\mathit{BcTxt}_{\text{r.e.}}$ (Theorem 6).

Theorem 5 $\mathit{SupQ}_{\text{r.e.}} \subset \mathit{LimTxt}_{\text{r.e.}}[K]$.

Proof. $\mathit{SupQ}_{\text{r.e.}} \subseteq \mathit{LimTxt}_{\text{r.e.}}[K]$ follows from Theorem 4, since $\mathit{LimTxt}_{\text{r.e.}}[K]$ comprises $\mathit{ConsvTxt}_{\text{r.e.}}[K]$. As $\mathit{BcTxt}_{\text{r.e.}} \setminus \mathit{ConsvTxt}_{\text{r.e.}}[K] \neq \emptyset$ (see Theorem 10 in [19]), Theorem 4 yields $\mathit{BcTxt}_{\text{r.e.}} \setminus \mathit{SupQ}_{\text{r.e.}} \neq \emptyset$. Theorem 6 then implies $\mathit{LimTxt}_{\text{r.e.}}[K] \setminus \mathit{SupQ}_{\text{r.e.}} \neq \emptyset$. □

Theorem 6 $\mathit{BcTxt}_{\text{r.e.}} \subset \mathit{LimTxt}_{\text{r.e.}}[K]$.

Proof. The proof concerns two statements:

Part 1. $\mathit{BcTxt}_{\text{r.e.}} \subseteq \mathit{LimTxt}_{\text{r.e.}}[K]$.
Part 2. $\mathit{LimTxt}_{\text{r.e.}}[K] \setminus \mathit{BcTxt}_{\text{r.e.}} \neq \emptyset$.

Proof of Part 1. To show $\mathit{BcTxt}_{\text{r.e.}} \subseteq \mathit{LimTxt}_{\text{r.e.}}[K]$ suppose $\mathcal{C}$ is a class of r.e. languages in $\mathit{BcTxt}_{\text{r.e.}}$. Let M be an IIM identifying $\mathcal{C}$ behaviorally correctly in $(W_i)_{i\in\mathbb{N}}$.

The following oracle-IIM M' *LimTxt*-identifies $\mathcal{C}$ with respect to $(W_i)_{i\in\mathbb{N}}$ using an oracle for K:

> Given a text segment t_n of length $n+1$, M' first computes $M(t_n)$.
> If $n = 0$, then $M'(t_n) = M(t_n)$.
> If $n > 0$, then M' uses the K-oracle to determine whether or not there is a word w_x for some $x \leq n$, such that $w_x \in W_{M(t_n)} \setminus W_{M'(t_{n-1})}$ or $w_x \in W_{M(t_{n-1})} \setminus W_{M'(t_n)}$. If no such word exists, then $M'(t_n) = M'(t_{n-1})$. Else $M'(t_n) = M(t_n)$.

It is not hard to prove that M' learns all languages in $\mathcal{C}$ in the limit with respect to $(W_i)_{i\in\mathbb{N}}$. Thus $BcTxt_{\text{r.e.}} \subseteq LimTxt_{\text{r.e.}}[K]$.

Proof of Part 2. $LimTxt_{\text{r.e.}}[K] \setminus BcTxt_{\text{r.e.}} \neq \emptyset$ is witnessed by the class

$$\mathcal{C}_R = \{L_f \mid f \text{ is a recursive function}\} ,$$

where for each partial recursive function f we define $L_f = \{a^x b^{f(x)} \mid x \in \mathbb{N}\}$. (* $\mathcal{C}_R$ consists only of recursive languages. *)

If $\mathcal{C}_R$ was $BcTxt$-learnable, then the class of all recursive functions would be Bc-learnable as defined by Barzdins [5]. The latter contradicts a result in [5]. On the other hand, $\mathcal{C}_R$ is $SupQ_{\text{r.e.}}$-learnable: if $L \in \mathcal{C}_R$ is the target language, a learner can find the least i with $L_{\varphi_i} \supseteq L$. Then L_{φ_i} must equal L. By Theorem 4, then $\mathcal{C}_R \in ConsvTxt_{\text{r.e.}}[K] \subseteq LimTxt_{\text{r.e.}}[K]$.[4] This establishes $BcTxt_{\text{r.e.}} \subset LimTxt_{\text{r.e.}}[K]$. □

Theorem 7 $BcTxt_{\text{r.e.}} \# SupQ_{\text{r.e.}}$.

Proof. For $BcTxt_{\text{r.e.}} \setminus SupQ_{\text{r.e.}} \neq \emptyset$ see Theorem 10 in [19]. The class $\mathcal{C}_R$ used to prove Theorem 6 witnesses $SupQ_{\text{r.e.}} \setminus BcTxt_{\text{r.e.}} \neq \emptyset$. □

4.2 Results for uniformly K-r.e. hypothesis spaces

Finally, we consider K-r.e. hypothesis spaces for query learning as in [17,18]. A family $(A_i)_{i\in\mathbb{N}}$ is *uniformly K-r.e.*, if there is a recursive function g with $A_i = \{w \in \Sigma^* \mid g(i,w,n) = 1 \text{ for all but finitely many } n\}$ for all $i \in \mathbb{N}$. As it turns out, all the equality results from former studies, as illustrated in Figure 1, now turn into proper inclusions. So, though there are strong relations between the corresponding inference types, these are not as strong as in the context of learning indexable classes. Theorem 8 and Theorem 9 state this formally.

Theorem 8 $LimTxt_{\text{r.e.}}[K] \subset DisQ_{K\text{-r.e.}}$.

Proof. The proof concerns two statements:

Part 1. $LimTxt_{\text{r.e.}}[K] \subseteq DisQ_{K\text{-r.e.}}$.
Part 2. $DisQ_{K\text{-r.e.}} \setminus LimTxt_{\text{r.e.}}[K] \neq \emptyset$.

[4] $\mathcal{C}_R \in LimTxt_{\text{r.e.}}[K]$ also follows from a result by Adleman and Blum [1], which proves that the access to an oracle for K permits to learn the class of all recursive functions in the limit, as defined by Gold [9]. This also yields $LimTxt_{\text{r.e.}}[K]$-learnability of $\mathcal{C}_R$.

Proof of Part 1. In order to prove $\mathit{LimTxt}_{\text{r.e.}}[K] \subseteq \mathit{DisQ}_{K\text{-r.e.}}$, suppose that $\mathcal{C}$ is a class of r.e. languages in $\mathit{LimTxt}_{\text{r.e.}}[K]$. Let M be an oracle-IIM identifying $\mathcal{C}$ in the limit with respect to $(W_i)_{i\in\mathbb{N}}$, using a K-oracle.

Suppose $L \in \mathcal{C}$ is the target language. Let $(V_i)_{i\in\mathbb{N}}$ be a uniformly K-r.e. family, in which grammars for all queries as posed in the instructions below can be computed (* such a family exists *). A DisQ-learner M' for L with respect to $(V_i)_{i\in\mathbb{N}}$ is defined as follows. Suppose $V_e = \emptyset$ and $V_{e'} = \Sigma^*$. If $V_{e'}$ is disjoint with L, then output e. Otherwise, M' acts on the following instructions, starting in stage 0.

Stage n reads as follows:

- Ask restricted disjointness queries for $\{w_0\}, \ldots, \{w_n\}$. Let $L_{[n]}$ be the set of words w_x, $x \leq n$, for which the corresponding query is answered with 'no'. (* Note that $L_{[n]} = L \cap \{w_x \mid x \leq n\}$. *)
- Let $(\sigma_x^n)_{x\in\mathbb{N}}$ be an effective enumeration of all finite segments of texts for $L_{[n]}$. For all $x, y \leq n$ compute $M(\sigma_x^y)$ (* note that for these computations a K-oracle must be simulated *) as follows: whenever M wants to access a K-oracle in order to determine whether $k \in K$ for some $k \in \mathbb{N}$, then pose a restricted disjointness query for the language

$$V'_k = \begin{cases} \Sigma^*, & \text{if } k \in K, \\ \emptyset, & \text{otherwise}. \end{cases}$$

  If the answer is 'yes', then transmit the answer 'no' to M and vice versa.
- For each $x, y \leq n$, pose a restricted disjointness query for the K-r.e. language $\overline{W_{M(\sigma_x^y)}}$. Let $\text{Cand}_n = \{\sigma_x^y \mid x, y \leq n \text{ and } \overline{W_{M(\sigma_x^y)}} \cap L = \emptyset\}$ be the set of those segments, for which the query has been answered with 'yes'. (* Note that $\text{Cand}_n = \{\sigma_x^y \mid x, y \leq n \text{ and } L \subseteq W_{M(\sigma_x^y)})\}$. *)
- For all $\sigma \in \text{Cand}_n$, pose a restricted disjointness query for the K-r.e. language

$$V'_\sigma = \begin{cases} \Sigma^*, & \text{if, given access to a } K\text{-oracle as requested,} \\ & M(\sigma\sigma') \neq M(\sigma) \text{ for some text segment } \sigma' \text{ of } W_{M(\sigma)}, \\ \emptyset, & \text{otherwise}. \end{cases}$$

  (* $V'_\sigma \cap L = \emptyset$ iff σ is a LimTxt-stabilizing sequence for M and $W_{M(\sigma)}$. *) If all these queries are answered 'no', then go to stage $n+1$. Else, if $\sigma \in \text{Cand}_n$ is minimal with $V'_\sigma \cap L = \emptyset$, then hypothesize a j with $V_j = W_{M(\sigma)}$ and stop.

M' identifies L with disjointness queries with respect to $(V_i)_{i\in\mathbb{N}}$, because (i) M' eventually returns a hypothesis and (ii) this hypothesis is correct for L. To prove (i), note that M is a LimTxt-learner for L with respect to $(W_i)_{i\in\mathbb{N}}$. So there are i, x, y such that $M(\sigma_x^y) = i$, $W_i = L$, and σ_x^y is a *LimTxt*-stabilizing sequence for M and L. Then $V'_{\sigma_x^y} = \emptyset$ and the corresponding disjointness query is answered with 'yes'. Thus M' returns a hypothesis. To prove (ii), assume M' returns a hypothesis representing $W_{M(\sigma)}$ for some segment σ of a text for L. Then, by definition of M', $L \subseteq W_{M(\sigma)}$ and σ is a *LimTxt*-stabilizing sequence for M and $W_{M(\sigma)}$. In particular, σ is a *LimTxt*-stabilizing sequence for M and L. Since M learns L, this implies $L = W_{M(\sigma)}$. Hence the hypothesis M' returns is correct for L.

Therefore $\mathcal{C} \in DisQ_{K\text{-r.e.}}$ and $LimTxt_{\text{r.e.}}[K] \subseteq DisQ_{K\text{-r.e.}}$.

Proof of Part 2. This proof requires a further definition:

> We say that an oracle-IIM M is *nice*, if for all oracles A and all languages L, [if M^A has a stabilizing sequence on L, then every text for L starts with a stabilizing sequence for M^A on L]. Note that from any oracle-IIM M, one can effectively find an oracle-IIM M' such that M' is nice, and for all A, $LimTxt_{\text{r.e.}}[A]$-identifies at least as much as M (Theorem 13 in Fulk [8] can be seen to easily relativize).

Now, $LimTxt_{\text{r.e.}}[K] \neq DisQ_{K\text{-r.e.}}$ can be verified as follows. Let $M_0, M_1, \ldots$ be a recursive sequence of nice oracle-IIMs, such that any class in $LimTxt_{\text{r.e.}}[K]$ is $LimTxt_{\text{r.e.}}[K]$-identified by some M_i.

Let $X_i = \{a^i b^x \mid x \in \mathbb{N}\}$. Let t^i be the canonical text $a^i b^0, a^i b^1, a^i b^2, \ldots$ for X_i. Let $X_i^n = content(t_n^i) = \{a^i b^x \mid x \leq n\}$.

Define L_i as follows. If there is no stabilizing sequence for M_i^K on X_i, then let $L_i = X_i$. Else, let t_n^i be a stabilizing sequence for M_i^K on X_i (where n is the least non-zero number such that t_n^i is a stabilizing sequence for M_i^K on X_i). Then if $W_{M_i^K(t_n^i)} \supset X_i^n$, then let $L_i = X_i^n$; else let $L_i = X_i^{n+1}$.

Let

$$\mathcal{C} = \{L_i \mid i \in \mathbb{N}\}.$$

(* Note that $\mathcal{C}$ consists only of recursive languages. *)

Note that M_i^K does not $LimTxt_{\text{r.e.}}[K]$-identify L_i. Thus $\mathcal{C} \notin LimTxt_{\text{r.e.}}[K]$.

We now show how to get a K-r.e. grammar for L_i from i. This is clearly enough to verify $\mathcal{C} \in DisQ_{K\text{-r.e.}}$ (as i can be obtained by asking restricted disjointness queries for L_0, L_1, ..., until the unique i to cause a 'no'-reply is found).

Now $a^i b^n \in L_i$ iff:

(i) $n = 0$ or
(ii) for all $y \leq n$ [t_y^i is not a stabilizing sequence for M_i^K on X_i] or

(iii) for all $y < n$ [[t^i_y is not a stabilizing sequence for M^K_i on X_i] and $W_{M^K_i(t^i_n)} \not\supseteq X^n_i$].

This is a K-r.e. predicate, hence one can obtain a K-r.e. grammar for L_i. □

Theorem 9 $\mathit{DisQ}_{K\text{-r.e.}} \subset \mathit{SupQ}_{K\text{-r.e.}}$.

Proof. The following two statements have to be verified:

Part 1. $\mathit{DisQ}_{K\text{-r.e.}} \subseteq \mathit{SupQ}_{K\text{-r.e.}}$.
Part 2. $\mathit{SupQ}_{K\text{-r.e.}} \setminus \mathit{DisQ}_{K\text{-r.e.}} \neq \emptyset$.

Proof of Part 1. Let $\mathcal{C}$ be a class of r.e. languages which is DisQ-learnable by some M with respect to a uniformly K-r.e. hypothesis space $(V_i)_{i\in\mathbb{N}}$. Let $(V'_i)_{i\in\mathbb{N}}$ be a uniformly K-r.e. family, in which grammars for all restricted superset queries needed below can be computed (* such a family exists *). For a target language L, an IIM M' is defined as follows. If $V'_e \supseteq L$, where e is the least index for $\emptyset$ in $(V'_i)_{i\in\mathbb{N}}$, then output e. Otherwise, M' executes stage 0.

Stage n reads as follows:

- Simulate M. If M poses a restricted disjointness query for V_j, determine the set Cand_n of all $w \in \{w_0,\dots,w_n\}$, for which a restricted superset query concerning
$$V'_w = \begin{cases} \Sigma^*, & \text{if } w \in V_j\,, \\ \emptyset, & \text{if } w \notin V_j\,, \end{cases}$$
is answered with 'yes'. (* $\mathrm{Cand}_n = \{w_0,\dots,w_n\} \cap V_j$. *)
- Then pose a restricted superset query for all languages $\Sigma^* \setminus \{w\}$ with $w \in \mathrm{Cand}_n$.
  · If all the answers are 'yes' (* $\mathrm{Cand}_n \cap L = \emptyset$ *), then transmit the answer 'yes' to M.
  · If at least one answer is 'no', then transmit the answer 'no' to M. (* 'no'-answers are always correct. *)
- If M has not returned a hypothesis within n steps, then go to stage $n+1$.
- Else, if M within n steps has returned a hypothesis for the language V_i, pose a restricted superset query representing V_i in $(V'_s)_{s\in\mathbb{N}}$.
  · If the answer is 'no', then go to stage $n+1$.
  · If the answer is 'yes', then let J be the set of indices of queries of M, which have been answered with 'yes'. For all $j \in J$ pose a restricted superset query for
$$V' = \begin{cases} \Sigma^*, & \text{if } V_j \cap V_i \neq \emptyset\,, \\ \emptyset, & \text{if } V_j \cap V_i = \emptyset\,. \end{cases}$$
    If all these queries are answered 'no', then return a grammar for V_i in $(V'_k)_{k\in\mathbb{N}}$ (* because all queries are then answered correctly for the

language $V_i \supseteq L$ and for L—so the hypothesis V_i of M must be correct for L *).

If at least one of these queries is answered 'yes', then go to stage $n+1$.

It is not hard to show that M' learns L with respect to $(V_i')_{i\in\mathbb{N}}$. So $\mathcal{C} \in SupQ_{K\text{-r.e.}}$.

Proof of Part 2. A class in $SupQ_{K\text{-r.e.}} \setminus DisQ_{K\text{-r.e.}}$ can be defined as follows.

Let A be a Π_3-complete set. Let $L_i = \{a^i b^{j+1} a^{x+1} \mid j, x \in \mathbb{N}\}$ and $L_i^s = \{a^i b^{j+1} a^{x+1} \mid j \in \mathbb{N},\ x \le s\}$ for all $i, s \in \mathbb{N}$. Finally, let

$$\mathcal{C} = \{L_i \mid i \in A\} \cup \{L_i^s \mid i \notin A,\ s \in \mathbb{N}\}\,.$$

(* Note that $\mathcal{C}$ consists only of recursive languages. *)

We first show $\mathcal{C} \notin DisQ_{K\text{-r.e.}}$. Suppose by way of contradiction that M witnesses $\mathcal{C} \in DisQ_{K\text{-r.e.}}$ with respect to some uniformly K-r.e. hypothesis space $(V_i)_{i\in\mathbb{N}}$. We establish a contradiction by concluding $A \in \Sigma_3$, though A is Π_3-complete. For that purpose, fix recursive sets Q, R with

$$\begin{aligned} i \in A \ &\text{iff}\ (\forall x)(\exists y)(\forall z)[Q(i,x,y,z)]\,, \\ w \in V_i \ &\text{iff}\ (\exists y)(\forall z)\ [R(i,w,y,z)]\,. \end{aligned}$$

Define a Σ_3-procedure P on input $i \in \mathbb{N}$ to begin in stage 0.

Stage n reads as follows:

(a) Test whether or not $(\exists y)(\forall z)[Q(i,n,y,z)]$ is true. If not, then stop with the output '$i \notin A$'. Else go to (b).

(b) Simulate M for $n+1$ steps. Therein, whenever M poses a restricted disjointness query k, transmit
   - the answer 'yes' to M in case $(\exists j,x)(\exists y)(\forall z)[R(k, a^i b^{j+1} a^{x+1}, y, z)]$ is *not* true (* i.e., if $L_i \cap V_k = \emptyset$ *);
   - the answer 'no', otherwise.

   In case M does not return any hypothesis within $n+1$ steps of computation, go to stage $n+1$. Else stop with the output '$i \in A$'.

   (* If $i \notin A$, then there would be some s, such that, in the scenario above, all answers transmitted to M would be correct for the languages $L_i^s,\ L_{i+1}^s,\ L_{i+2}^s, \ldots$, which would all belong to $\mathcal{C}$. Since M returns a hypothesis, M would fail for infinitely many languages in $\mathcal{C}$—a contradiction. *)

Note that if $i \notin A$, then step (a) will succeed for appropriate stage n (for which not $(\exists y)(\forall x)[Q(i,n,y,z)]$ holds). On the other hand, if $i \in A$, then step

(b) will eventually succeed, as M is supposed to $DisQ_{K\text{-r.e.}}$-identify L_i. Thus, P decides A in Σ_3. This contradiction implies that $\mathcal{C} \notin DisQ_{K\text{-r.e.}}$.

We now show that $\mathcal{C} \in SupQ_{K\text{-r.e.}}$.

Let $q_x^i(t) = \min(\{t\} \cup \{r \leq t \mid (\forall z \leq t)\ [Q(i,x,r,z)]\})$. Note that $q_x^i(t)$ is recursive, and $\lim_{t\to\infty} q_x^i(t)$ exists iff $(\exists y)(\forall z)[Q(i,x,y,z)]$ is true. Let

$$f_i(u,t) = \begin{cases} 1\,, & \text{if } (\forall x \leq u)[q_x^i(t) = q_x^i(t+1)]\,, \\ 0\,, & \text{otherwise}\,. \end{cases}$$

It is easy to verify that $X_i = \{w_u \mid \lim_{t\to\infty} f_i(u,t) = 1\}$, is finite if $i \notin A$, and equal to Σ^*, if $i \in A$. Moreover, f_i is a K-r.e. function for X_i. Thus, a K-r.e. grammar for X_i can be obtained effectively from i.

Now define M as follows.

- For a target language L, query $\Sigma^* \setminus \{a^i b^{j+1} a^{x+1}\}$, for various values of i, j, x, until i, j, x are found such that $\Sigma^* \setminus \{a^i b^{j+1} a^{x+1}\} \not\supseteq L$.
  (* By definition of $\mathcal{C}$, this implies that L is of the form L_i or L_i^s for some $s \in \mathbb{N}$. *)
- Now, pose a restricted superset query for X_i.
  (* Note that if $i \in A$, then $X_i = \Sigma^* \supseteq L$, and if $i \notin A$, then $X_i \not\supseteq L$ (as X_i would then be finite, while L is infinite). Thus M can determine whether or not $i \in A$. *)
  · If the answer is 'yes' (* $i \in A$ *), then output a grammar for L_i.
  · If the answer is 'no' (* $i \notin A$ *), then search for the minimal $s \in \mathbb{N}$ such that $L_i^s \supseteq L$ (* now $L = L_i^s$ *) and output a grammar for L_i^s.

With the help of the remarks in the definition of M, it is not hard to verify that M learns $\mathcal{C}$ with restricted superset queries.

Thus $\mathcal{C} \in SupQ_{K\text{-r.e.}} \setminus DisQ_{K\text{-r.e.}}$. □

5 Discussion

Above we have seen that many of the equivalences of Gold-style and query inference types in the context of learning indexable classes no longer hold, if arbitrary target classes of r.e. languages are considered. Nevertheless, these two approaches of learning reveal strong relations, expressed in an inclusion hierarchy of inference types. Altogether, this shows that in many cases all learners of the one kind of inference types can be transformed into learners of the other kind without loss of learning power, though in general not vice versa.

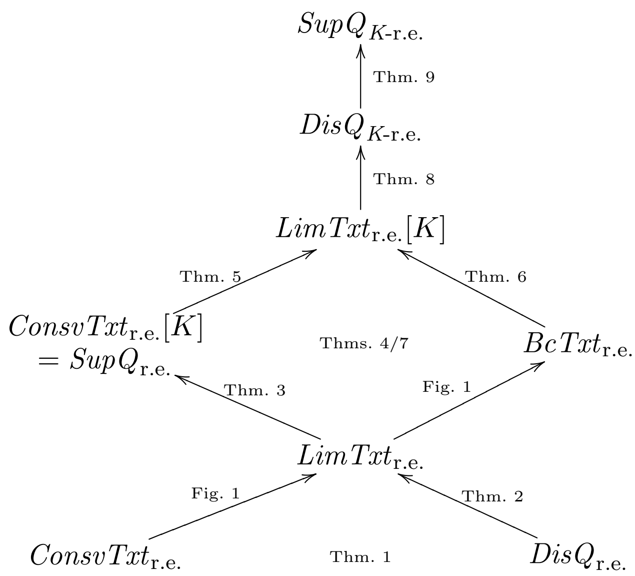

Fig. 2. This graph illustrates the relations between different inference types studied above. Arrows indicate proper inclusions of inference types. Two inference types which are not connected by a path of arrows are incomparable. References to the theorems corresponding to the results claimed in the graph are given.

Interestingly, our proofs for the inclusions are constructive, i.e. the transformations of learners can be done uniformly and indicate the essential reasons for the strong relations.

Another outcome is that all our separation results are witnessed by classes of recursive languages: $SupQ_{\text{r.e.}} \setminus LimTxt_{\text{r.e.}} \neq \emptyset$, $BcTxt_{\text{r.e.}} \setminus SupQ_{\text{r.e.}} \neq \emptyset$, and $LimTxt_{\text{r.e.}}[K] \setminus SupQ_{\text{r.e.}} \neq \emptyset$ are obtained in [17,18] using indexable classes of recursive languages; the other separations displayed in Figure 2 have been verified with non-indexable classes of recursive languages. For the latter, note that all classes used in our proofs above consist only of recursive languages. Of course these proofs would not have worked with indexable classes of recursive languages, since the corresponding separations do not hold for indexable classes, see Figure 1. So the equalities are not due to the recursiveness of the target languages alone. The fact that the target classes are indexable is crucial. This raises the question whether the new inequalities obtained above hold for uniformly r.e. classes. As it turns out, at least one of them does not, while some of them do.

When restricting the focus to learning indexable classes, [17] has shown that the capabilities of $DisQ_{K\text{-r.e.}}$-learners and $LimTxt_{\text{r.e.}}[K]$-learners are equal, which does not hold for general classes of recursive languages, as witnessed in the proof of Theorem 8. Interestingly, the enumerability of the target class is the crucial reason for the equality result in [17], as the following theorem illustrates.

Theorem 10 *Let $\mathcal{C}$ be a uniformly r.e. class of languages. Then $\mathcal{C} \in DisQ_{K\text{-r.e.}}$ iff $\mathcal{C} \in LimTxt_{\text{r.e.}}[K]$.*

Proof. $LimTxt_{\text{r.e.}}[K] \subseteq DisQ_{K\text{-r.e.}}$ holds by Theorem 8. So suppose $\mathcal{C}$ is a uniformly r.e. class of languages which is learnable according to the definition of $DisQ_{K\text{-r.e.}}$. Let f be a recursive function such that

$$\mathcal{C} = \{W_{f(i)} \mid i \in \mathbb{N}\}.$$

Let M be a $DisQ_{K\text{-r.e.}}$-learner for $\mathcal{C}$ with respect to a K-r.e. hypothesis space $(V_i)_{i\in\mathbb{N}}$. Let g be a recursive function such that $w \in V_i$ iff $\lim_{t\to\infty} g(i, w, t) = 1$.

The idea is to construct a $LimTxt_{\text{r.e.}}[K]$-learner M' for $\mathcal{C}$ by simulating M. Given a text segment t_n, M' searches for a language in $\mathcal{C}$, which is consistent with t_n and for which the behavior of the known learner M seems reasonable, at least when taking t_n into consideration. The length of the given text segment serves as a bound for M' when trying to analyze whether the behavior of M is reasonable.

Formally, define $M'(t_n)$ (using an oracle for K) as follows:

– If there exists a $j \leq n$ such that the following three conditions are satisfied:
(1) $content(t_n) \subseteq W_{f(j)}$.
(2) M outputs a hypothesis if the questions k of M are answered as follows:
  · 'no', if there exists a $w \in content(t_n)$ with $g(k, w, n') = 1$ for all $n' \geq n$.
  · 'yes', otherwise.
(3) For any query k made by M in the simulation in 2 above:
  · if there exists a $w_1 \in W_{f(j),n}$ with $g(k, w_1, n') = 1$ for all $n' \geq n$, then there also exists a $w_2 \in content(t_n)$ with $g(k, w_2, n') = 1$ for all $n' \geq n$.
  (* i.e., the seeming 'yes'-answers for $content(t_n)$ as a target language also seem to be 'yes'-answers for $W_{f(j)}$ as a target language *).

then output $f(j)$ for the least such j. Else output 0.

Note that these simulations can be done using an oracle for K. We now claim that M' $LimTxt_{\text{r.e.}}$-identifies $\mathcal{C}$ using a K-oracle. To see this, suppose $L \in \mathcal{C}$ and t is a text for L. Let j be minimal with $W_{f(j)} = L$. Fix n large enough such that:

A. If all the questions of M are answered correctly in the simulation made by $M'(t_n)$, then M outputs a hypothesis.
B. $j \leq n$.
C. Let Q denote the set of questions asked as in A above. Then, for all $k \in Q$, if $V_k \cap L \neq \emptyset$, then for some $w \in content(t_n)$, for all $n' \geq n$, $g(k, w, n') = 1$ (* i.e., answers can be given correctly based on t_n *).

D. For all $j' < j$, if $W_{f(j')} \not\supseteq L$, then $W_{f(j')} \not\supseteq content(t_n)$.
E. For all $j' < j$, if $W_{f(j')} \supset L$, then for some $k \in Q$ with $V_k \cap L = \emptyset$ and $x \in W_{f(j'),n}$, for all $n' \geq n$, $g(k, x, n') = 1$.

Claim. Such an n exists.

Proof of Claim. The existence of an n fulfilling (A–D) is obvious. Note that, if n_0 is the least such n, then (A–D) is fulfilled for all $n > n_0$ as well. In order to see that (E) can be fulfilled simultaneously with (A–D), assume that for some $j' < j$ no $n > n_0$ satisfying (E) existed. Then t_{n_0} could be extended to a text for $W_{f(j)}$ and $W_{f(j')}$, with answers as in (A) being correct for both $W_{f(j)}$ and $W_{f(j')}$. So M would fail to $DisQ_{\text{r.e.}}$-learn at least one of the languages $W_{f(j)}$ and $W_{f(j')}$, both of which belong to $\mathcal{C}$. This contradicts the choice of M.

qed Claim.

Now, $M'(t_{n'}) = f(j)$ follows for all $n' \geq n$. □

For the other separations, except for $SupQ_{K\text{-r.e.}} \setminus DisQ_{K\text{-r.e.}} \neq \emptyset$, we will now prove that even enumerability of the target class is not sufficient for achieving the equality results from [17,18]. Whether or not a similar result holds for the separation of $SupQ_{K\text{-r.e.}}$ and $DisQ_{K\text{-r.e.}}$, remains an open question.

Theorem 11 *(1) There exists a uniformly r.e. class $\mathcal{C}$ of languages such that $\mathcal{C} \in ConsvTxt_{\text{r.e.}} \setminus DisQ_{\text{r.e.}}$ (and thus $\mathcal{C} \in LimTxt_{\text{r.e.}} \setminus DisQ_{\text{r.e.}}$).*
(2) There exists a uniformly r.e. class $\mathcal{C}$ of languages such that $\mathcal{C} \in SupQ_{\text{r.e.}} \setminus BcTxt_{\text{r.e.}}$.
(3) There exists a uniformly r.e. class $\mathcal{C}$ of languages such that $\mathcal{C} \in LimTxt_{\text{r.e.}}[K] \setminus BcTxt_{\text{r.e.}}$.

Proof.

ad 1. This is already witnessed by the proof of Theorem 1, where a uniformly r.e. class of recursive languages is used for the separation.

ad 2 and 3. The idea is to use a class comprising the class $\mathcal{C}_R$ used in the proof of Theorem 6. (Recall that $\mathcal{C}_R = \{L_f \mid f \text{ is a recursive function}\}$, where $L_f = \{a^x b^{f(x)} \mid x \in \mathbb{N}\}$ for each partial recursive function f.) For that purpose, choose a uniformly r.e. family $(L_i)_{i \in \mathbb{N}}$ of recursive languages satisfying the following demands:

A. for all $i \in \mathbb{N}$, L_i is either finite or $L_i \in \mathcal{C}_R$,
B. for all $L \in \mathcal{C}_R$ there is some $i \in \mathbb{N}$ such that $L = L_i$,
C. $L_0 = \emptyset$,
D. for all $i, j \in \mathbb{N}$ with $L_i \neq \emptyset$, if $L_i \subset L_j$, then $i < j$.

Claim. There exists a family satisfying the properties (A–D).

Proof of Claim. First, we define a numbering ψ of partial recursive functions as follows:

$$\psi_e(x) = \begin{cases} \varphi_e(x)\,, & \text{if } \varphi_e(y) \text{ is defined for all } y \le x\,, \\ \text{undefined}\,, & \text{otherwise}\,. \end{cases}$$

Obviously, each function ψ_e is either total or has the domain $\{y \mid y < x\}$ for some $x \in \mathbb{N}$.

Second, we have to define L_i for $i \in \mathbb{N}$. If $i = 0$, simply let $L_i = L_0 = \emptyset$. If $i > 0$, let $e_0, \dots, e_k \in \mathbb{N}$ be coefficients such that $e_0 < \cdots < e_k$ and $i = 2^{e_0} + \cdots + 2^{e_k}$, i.e., $(e_0, \dots, e_k)$ are the non-null coefficients in the binary representation of i. Now L_i is constructed as the union of sets $L_{i,t}$, i.e., $L_i = \bigcup_{t \in \mathbb{N}} L_{i,t}$, as follows:

- Initially, search for some bound $s \in \mathbb{N}$, such that for each $j < k$ there is some $z_j \le s$ satisfying
  · both $\psi_{e_j}(z_j)$ and $\psi_{e_k}(z_j)$ are defined within s steps of computation,
  · $\psi_{e_j}(z_j) \neq \psi_{e_k}(z_j)$.
  If such a bound s is found, then let

  $$L_{i,0} = \cdots = L_{i,s} = \{a^x b^{\psi_{e_k}(x)} \mid x \le s \text{ and } \psi_{e_k}(x) \text{ is defined within } s \text{ steps}\}$$

  and define the other sets $L_{i,t}$ by stages, starting in stage s. Otherwise, if no such bound s is found, $L_{i,t} = \emptyset$ for all $t \in \mathbb{N}$ and thus $L_i = \emptyset$.
- *Stage t for $t \in \mathbb{N}$:* Note that $L_{i,t}$ has already been defined. Now, for each $e \in \mathbb{N}$, let

  $$L^t_{\psi_e} = \{a^x b^{\psi_e(x)} \mid x \le t \text{ and } \psi_e(x) \text{ is defined within } t \text{ steps}\} \text{ and}$$
  $$\mathrm{Cand}_{i,t} = \{e \le e_k \mid L_{i,t} \subseteq L^t_{\psi_e}\}$$

  Each of the sets $L^t_{\psi_e}$ is finite. Thus choose the least number $e \in \mathrm{Cand}_{i,t}$ such that $\mathrm{card}(L^t_{\psi_e}) \ge \mathrm{card}(L^t_{\psi_{e'}})$ for all $e' \in \mathrm{Cand}_{i,t}$. Define $L_{i,t+1} = L^t_{\psi_e}$ and go to stage $t+1$.

Note that, by definition, for each $i \in \mathbb{N}$ and the corresponding coefficient e_k there is some $e \le e_k$ such that $L_i = L_{\psi_e}$. It remains to verify the properties (A–D).

A. Let $i \in \mathbb{N}$. Then $L_i = L_{\psi_e}$ for some $e \le e_k$. Since ψ_e is either total or a function with a finite domain, L_i is finite or $L_i \in \mathcal{C}_R$.
B. Let $L \in \mathcal{C}_R$, i.e., $L = L_f$ for some recursive function f. Let $c \in \mathbb{N}$ be minimal such that $\psi_c = f$. Now all ψ-indices less than c correspond either to proper subfunctions of f or to functions disagreeing with f in a defined value. We collect indices of the latter functions in a set

   $$E = \{e < c \mid \text{there is a } z \text{ such that } \psi_e(z) \text{ is defined and } \psi_e(z) \neq \psi_c(z)\}\,.$$

Let $E = \{e_0, \dots, e_{k-1}\}$ with $e_0 < \dots < e_{k-1}$ and let $i = 2^{e_0} + \dots + 2^{e_{k-1}} + 2^c$. Now it is not hard to verify that $L_i = L_{\psi_c} = L_f = L$.

C. This holds by definition.

D. Assume to the contrary that there are $i, j \in \mathbb{N}$ such that $L_i \neq \emptyset$, $L_i \subset L_j$, and $j < i$. As above, choose e_k maximal with $2^{e_k} \leq i$ and choose a suitable initial bound s. Then there exist $e, e' \leq e_k$ with $L_i = L_{\psi_e}$ and $L_j = L_{\psi_{e'}}$. Since the domain of each function ψ_e either is finite or equals $\mathbb{N}$, we obtain that L_i is a finite set. Thus choose some $t > s$ satisfying

$$L_{i,t} = L_i = L_{\psi_e} \subset L^t_{\psi_{e'}} .$$

Such a t exists by definition. From $e' \leq e_k$ we obtain $e' \in \mathrm{Cand}_{i,t}$. But, since $L_{i,t+1} = L_{i,t} = L_i$ (i.e., L_i is no longer increased from stage t on), the cardinality of $L^t_{\psi_{e'}}$ must be no greater than the cardinality of $L_{i,t+1} = L_{i,t}$. The latter is a contradiction to $L_{i,t} \subset L^t_{\psi_{e'}}$.

qed Claim.

Now define

$$\mathcal{C} = \{L_i \mid i \in \mathbb{N}\} .$$

By property (B) $\mathcal{C}$ obviously comprises $\mathcal{C}_R$ and thus $\mathcal{C} \notin BcTxt_{\text{r.e.}}$. In contrast to that, $\mathcal{C} \in SupQ_{\text{r.e.}}$ can be verified: for identifying some $L \in \mathcal{C}$ a query learner can find the least i such that $L_i \supseteq L$ and then return an index representing L_i in the underlying hypothesis space. Property (D) then implies $L_i = L$ and therefore L is successfully identified.

Thus, $\mathcal{C} \in SupQ_{\text{r.e.}}$ and, by Theorem 5, $\mathcal{C} \in LimTxt_{\text{r.e.}}[K]$ as well. □

Thus we have seen that in several cases the equivalence results for indexable classes from previous work are weakened to strict inclusions, regardless of whether or not the target class (i) consists of recursive languages only, or (ii) is enumerable (the latter with one exception, see Theorem 10). This shows that indexable target classes yield a specific situation for Gold-style and query learning. Strong relationships between the two models are already witnessed in the general case of learning arbitrary classes of r.e. languages; however, these relationships can intensify when restricting the focus to target classes of lower algorithmic complexity, but in general only when restricting the focus to *indexable* target classes.

Acknowledgment

We thank the anonymous referees of this journal as well as the anonymous referees of a preliminary version [13] of this paper, especially, for suggesting

the simplified proof of Theorem 11–(2).

References

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

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

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

[4] D. Angluin. Queries revisited. *Theoret. Comput. Sci.*, 313:175–194, 2004.

[5] J. Barzdins. Two theorems on the limiting synthesis of functions. *Theory of Algorithms and Programs*, Latvian State University, Riga 210:82–88, 1974.

[6] J. Case, C. Lynes. Machine inductive inference and language identification. In: *Proceedings of the 9th International Colloquium on Automata, Languages, and Programming*, Lecture Notes in Computer Science 140, pp. 107–115, Springer, 1982.

[7] D. De Jongh, M. Kanazawa. Angluin's theorem for indexed families of r.e. sets and applications. In: *Proceedings of the 9th Annual Conference on Computational Learning Theory*, pp. 193–204, ACM Press, 1996.

[8] M. Fulk. Prudence and other conditions for formal language learning. *Inform. Comput.*, 85:1–11, 1990.

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

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

[11] S. Jain, E. Kinber. Learning languages from positive data and negative counterexamples. In: *Proceedings of the 15th International Conference on Algorithmic Learning Theory*, Lecture Notes in Artificial Intelligence 3244, pp. 54–68, Springer, 2004.

[12] S. Jain, E. Kinber. Learning languages from positive data and a finite number of queries. In: *Proceedings of the 24th Conference on Foundations of Software Technology and Theoretical Computer Science*, Lecture Notes in Artificial Intelligence 3328, pp. 360–371, Springer, 2004.

[13] S. Jain, S. Lange, S. Zilles. Gold-style and query learning under various constraints on the target class. In: *Proceedings of the 16th International Conference on Algorithmic Learning Theory*, Lecture Notes in Artificial Intelligence 3734, pp. 226–240, Springer, 2005.

[14] M. Kearns, M. Li, L. Pitt, L. Valiant. Recent results on Boolean concept learning, In: *Proceedings of the 4th International Workshop on Machine Learning*, pp. 337–352, Morgan Kaufmann, 1987.

[15] M. Kearns, U. Vazirani. *An Introduction to Computational Learning Theory*, MIT Press, 1994.

[16] S. Lange, T. Zeugmann. Language learning in dependence on the space of hypotheses. In: *Proceedings of the 6th Annual Conference on Computational Learning Theory*, pp. 127–136, ACM Press, 1993.

[17] S. Lange, S. Zilles. Replacing limit learners with equally powerful one-shot query learners. In: *Proceedings of the 17th Annual Conference on Learning Theory*, Lecture Notes in Artificial Intelligence 3120, pp. 155–169, Springer, 2004.

[18] S. Lange, S. Zilles. Comparison of query learning and Gold-style learning in dependence of the hypothesis space. In: *Proceedings of the 15th International Conference on Algorithmic Learning Theory*, Lecture Notes in Artificial Intelligence 3244, pp. 99–113, Springer, 2004.

[19] S. Lange, S. Zilles. Relations between Gold-style learning and query learning. *Inform. Comput.*, 203:211–237, 2005. (Summarizes [17] and [18].)

[20] R. Motwani, P. Raghavan. *Randomized Algorithms*, Cambridge University Press, 1995.

[21] L. Pitt, L. Valiant. Computational limitations on learning from examples. *Journal of the ACM*, 35:965–984, 1988

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

[23] F. Stephan. *Degrees of Computing and Learning.* Habilitationsschrift, Ruprecht-Karls-Univ., Heidelberg, 1999.

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

[25] T. Zeugmann, S. Lange. A guided tour across the boundaries of learning recursive languages. In: *Algorithmic Learning for Knowledge-Based Systems*, Lecture Notes in Artificial Intelligence 961, pp. 190–258, Springer, 1995.