Learning indexed families of recursive languages from positive data: A survey

Steffen Lange[a], Thomas Zeugmann[b], Sandra Zilles[c,*]

[a] *Fachbereich Informatik, Hochschule Darmstadt, Haardtring 100, 64295 Darmstadt, Germany*
[b] *Division of Computer Science, Hokkaido University, Sapporo 060–0814, Japan*
[c] *Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada T6G 2E8*

Abstract

In the past 40 years, research on inductive inference has developed along different lines, e.g., in the formalizations used, and in the classes of target concepts considered. One common root of many of these formalizations is Gold's model of identification in the limit. This model has been studied for learning recursive functions, recursively enumerable languages, and recursive languages, reflecting different aspects of machine learning, artificial intelligence, complexity theory, and recursion theory. One line of research focuses on indexed families of recursive languages — classes of recursive languages described in a representation scheme for which the question of membership for any string in any of the given languages is effectively decidable with a uniform procedure. Such language classes are of interest because of their naturalness. The survey at hand picks out important studies on learning indexed families (including basic as well as recent research), summarizes and illustrates the corresponding results, and points out links to related fields such as grammatical inference, machine learning, and artificial intelligence in general.
© 2008 Elsevier B.V. All rights reserved.

Keywords: Inductive inference; Formal languages; Recursion theory; Query learning

1. Introduction

Forty years ago, when the field of artificial intelligence was just emerging, many computer scientists were focused on understanding *learning*. For instance, the question of how humans learn to speak a language was a concern for the AI community as well as for researchers in computational linguistics and psycholinguistics.

Gold [33] can be called a pioneer in the field of inductive inference, since at that time he published his seminal paper on "Language identification in the limit". The question of how children learn languages was part of the motivation for his work; however, his focus was on theoretical investigations, i.e., his aim was

> [. . .] to construct a precise model for the intuitive notion "able to speak a language" in order to be able to investigate theoretically how it can be achieved artificially. [33], page 448.

---

* Corresponding author.
E-mail addresses: s.lange@fbi.h-da.de (S. Lange), thomas@ist.hokudai.ac.jp (T. Zeugmann), zilles@cs.ualberta.ca (S. Zilles).

1.1. A first model of language learning

In Gold's [33] model, a language is simply a set of strings over some fixed finite alphabet. From now on, we will use the term *target language* to refer to a language which has to be learned. As Gold states, we are in general not able to completely describe a language we speak in the form of rules, thus a first crucial observation is that languages must be learned from some kind of implicit information, namely examples. He considers the case where only positive examples (strings belonging to the target language) are available for the learner, as well as the case where both positive and negative examples (strings labeled according to whether or not they belong to the target language) are available.

Since human language learning is a process in which the learner revises hypothetical conjectures about the target language from time to time – never knowing whether or not the current conjectures completely and correctly reflect the target language – it is reasonable to model learning as a process in which information (in the form of examples) is presented step by step, and at each step the learner may revise its conjecture. In particular, the process is modeled such that it never ends, i.e., there are infinitely many steps. In each step, an example is presented. No assumptions are made concerning the order of examples, nor the multiplicity with which individual examples appear during the process.

To be able to judge whether or not the learner has successfully managed the task of learning the target language, one might demand that in each step of the process the learner explicates its internal conjecture about the target language in the form of some *finite* description meant to completely characterize a language — such descriptions are called *hypotheses*. To be able to interpret hypotheses, Gold [33] considered various representation schemes that might be used by the learner. For instance, if the target language L is recursive, then a program for a decision procedure (deciding for each string whether or not it belongs to L) might be a plausible hypothesis describing L. Similarly, grammars generating all and only the strings in L might be considered as sensible descriptions.

Assume some system of descriptions of languages, called a *hypothesis space*, is fixed, such that at least one correct description for the target language is contained in that system. Gold considers a learner to be an algorithmic device which is given examples, step by step, and which, in each of the infinitely many steps, returns a hypothesis. Gold's model of *identification in the limit* declares a learner successful if its sequence of hypotheses fulfills two requirements:

(1) it converges to some single hypothesis, i.e., after some step the learner returns the same hypothesis over and over again for all future steps;
(2) the hypothesis it converges to is a correct representation for the target language in the underlying hypothesis space.

So a complete and explicit description of a target language has to be inferred from incomplete and implicit information. However, such demands are only sensible in the case that some requirements are made concerning the sequence of examples presented. A *text* of a language L is an infinite sequence of strings that contains all strings of L. Alternatively, Gold [33] considers learning from an *informant*. An informant of a language L is an infinite sequence containing all strings over the underlying alphabet such that each string is labeled according to whether or not it belongs to L (see Section 3.1 for a detailed discussion).

Since for every hypothesis h it is easy to define a learner constantly returning h, learnability of a single language in this model is trivial. What is more interesting is to study classes of target languages in the sense that one learner is able to identify all the languages in the given class from examples. If such a learner exists for a class $\mathcal{C}$ of languages, then $\mathcal{C}$ is said to be identifiable (or learnable) in the limit.

1.2. Gold set the ball rolling . . .

Gold [33] studied structural properties of classes learnable and classes not learnable in this model. In particular, his examples of seemingly simple classes not identifiable in the limit from text may have made studies on language learning in the limit rather unattractive — at first! Nevertheless, he set the ball rolling, or one should better say several balls.

- Trakhtenbrot and Barzdin [88] studied the learnability of finite automata intensively and obtained, in particular, positive results for the case of learning from an informant.
- Gold's model allows not only for studying language learning in general, but also learning of recursive functions. Following Gold's [32,33] papers, much has been done on inductive inference of classes of recursive functions (see for instance Barzdin [13,11,9,10], Barzdin and Freivald [12], and Blum and Blum [17]). The reader is referred to Zeugmann and Zilles [103] for a survey on this branch of research.

- By focusing on general phenomena of learning in Gold's model, characterizations of classes of recursively enumerable languages learnable in the limit were studied by Wiehagen [93], thus opening a line of research focusing on learning recursively enumerable languages in the limit. Later studies thereon are concerned with many different variants of Gold's initial model, with aspects closely related to relevant questions in the area of machine learning, such as the role of the information offered, noisy information, and the mind change complexity of learners. The reader is referred to, for instance, Case and Lynes [22], Kinber and Stephan [45], de Jongh and Kanazawa [26], Stephan [87], Jain and Sharma [41], Case et al. [21], Case and Jain [19], and Jain et al. [36,37]. Still today this branch of research is very active.
- Another "ball" that was set rolling by the studies on inductive inference as initiated by Gold [33] is the one with which the present survey is concerned, namely learning indexable classes of recursive languages. The basics are due to Dana Angluin [2,3]. An *indexed family of recursive languages* is a family $L_0, L_1, L_2, \ldots$ of languages for which there is a decision procedure which, given any index i and any string w, decides whether or not w belongs to the language L_i. Indexable classes, i.e., classes that can be represented as such an indexed family, might be considered more natural than general classes of recursively enumerable languages. For instance, the classes of all non-empty regular or of all non-empty context-free languages are indexable; many computer systems deal with information represented as instances of some language contained in an indexed family.

Angluin [3] studied learning of indexed families in general, providing a very useful characterization of those learnable in the limit. Her results, also discussed in this survey, have been of high impact and still are so today.

These results encouraged people to study learnability of special indexable classes of recursive languages, such as, for instance, the pattern languages, or the k-reversible languages and generalizations thereof. Moreover, Angluin's [3] general characterization result and the sufficient conditions she discovered have given rise to many studies on general phenomena of learning with a focus on indexed families of recursive languages. Last but not least, another branch of research on learning indexed families in the limit has focused on relations to other learning models, such as PAC learning and learning from queries. Let us briefly summarize the history of these branches of research.

1.2.1. Learning in the limit of special indexed families

Angluin [2] initiated the study of special indexed families of recursive languages by defining the pattern languages and analyzing them in Gold's model. Roughly speaking, a pattern language is a set of strings that match a pattern consisting of terminal symbols over some given alphabet and variable symbols. "Match" here means that the string can be obtained from the pattern by replacing the variable symbols by non-empty strings over the given alphabet. The class of these languages continues to be of interest, especially in the learning theory community.

Shinohara [85] extended Angluin's [2] definition of pattern languages by allowing variables to be replaced by empty strings. The resulting class of languages is called the *extended pattern languages*, or *erasing pattern languages*.

While Angluin [2] had already shown that the pattern languages are learnable in the limit, the question of whether or not the erasing pattern languages are learnable kept scientists busy for many years, until Daniel Reidenbach finally proved their non-learnability in several cases of different alphabet sizes; see Reidenbach [76,78,79]. Studying erasing pattern languages also involved the analysis of interesting subclasses, such as the regular erasing pattern languages as studied by Shinohara [85], the quasi-regular erasing pattern languages analyzed by Mitchell [64], or erasing pattern languages with restrictions on the number of occurring variables (see Wright [97]).

The learnability of the pattern languages was also studied with a focus on efficient learning algorithms. For example, Kearns and Pitt [43] studied the learnability of k-variable pattern languages in the PAC model. Lange and Wiehagen [53] designed an iterative polynomial-time algorithm identifying the class of all pattern languages in the limit. Their algorithm was analyzed with respect to its average-case behavior by Zeugmann [100]. Rossmanith and Zeugmann [81] converted this learning algorithm into a stochastic finite learner (see also Zeugmann [101]). We refer the reader to Ng and Shinohara [72] for more information concerning the state of the art of learning various classes of pattern languages.

A further example of indexed families studied in the context of learning in the limit are the k-reversible languages. Here the initial analysis by Angluin [4] was of impact especially for the grammatical inference community (see for instance Pitt [75], Sakakibara [83], and Ciccello and Kremer [23]).

What is common to these approaches is that they enriched the studies of identification in the limit by new aspects such as: efficiency in learning, the problem of consistency in learning, the effect of additional information and special hypothesis spaces on learning, etc. (see Section 1.3 for a brief discussion of the relevance of these aspects).

1.2.2. General phenomena of learning in the limit of indexed families

From the early nineties on, learning theoreticians investigated more general questions in the context of learning indexed families of recursive languages, see Zeugmann and Lange [102] and Lange [50] and the references therein. Natural constraints on learning, reflecting demands for so-called conservative, monotonous, or incremental behavior (to name just a few), were formalized and analyzed systematically, with a focus on their impact on the capabilities of learners as well as characterizations of the structure of learnable classes of languages.

The role of hypothesis spaces is an important aspect in this context, again showing the relations to phenomena observed and studied in the discipline of machine learning (see Section 1.3).

These more general aspects of identification in the limit of indexed families are one of the main topics addressed in the present survey. Due to the manifold intuitive extensions of Gold's [33] model, each particularly addressing specific desirable or observable characteristics of learning processes, we shall only summarize a few of them. The reader should note that this selection is by no means all encompassing.

1.2.3. Relations to query, on-line, and PAC learning of indexed families

Identification in the limit, being just one of several formal models of concept learning examined in the field of algorithmic learning theory, was compared to other models of inference. Three prominent approaches in this context are:

- learning from queries as in the model defined by Angluin [5–7],
- on-line learning, see Littlestone [63], and
- probably approximately correct (PAC) learning introduced by Valiant [89] (see also Natarajan [69] and Kearns and Vazirani [44]).

Gold's model of learning from examples and Angluin's model of learning from queries have much in common, e.g., there are equivalences in the characteristic learning methods as well as in the limitations of both models. The reader is referred to Lange and Zilles [62] for a detailed discussion; however, this aspect will be addressed later in Section 5.2.

Learning a representation of a language from examples can be seen as a special case of learning to predict the strings in a language and thus of Littlestone's approach [63] of on-line learning. Hence it is not surprising that relations between Gold-style learning and on-line prediction were analyzed in the setting of learning recursive functions (see Barzdin [13] for the crucial definitions of on-line prediction in this setting). The strong relations between the two models are also inherent in studies of on-line learning for special indexed families of languages which are usually very common in the studies of Gold's model, such as for instance the pattern languages (see Kaufmann and Stephan [42]).

Finally, Gold-style learning of indexed families is also related to PAC learning. Though Gold-style and PAC learning are rarely compared directly in the literature (see Rossmanith and Zeugmann [81] and Zeugmann [101]), relations between the two approaches can be deduced from the relations they both have to learning from queries and to on-line learning. The reader is referred to the seminal paper by Angluin [6] for relations between PAC learning and query learning, and to Haussler et al. [34] and Littlestone [63] for relations between PAC learning and on-line learning.

1.3. The relevance and impact of the rolling balls

The analysis of learning indexed families of recursive languages in the limit is of greater interest than just to the field of algorithmic learning theory. Rather, such analysis has also influenced the fields of artificial intelligence, machine learning, and grammatical inference.

The relevance for artificial intelligence (see Russell and Norvig [82]) is evident in the analysis of natural properties of learners, such as monotonicity, incremental functioning, efficiency. Other essential AI methods, such as for problem solving by searching (see Mitchell [65]), also play an important role in learning in the limit. This will for instance become apparent in the characterization theorems discussed below in Section 3.5.

Focusing especially on machine learning issues, algorithmic learning theory is concerned with issues such as biasing, additional information in learning, and hypothesis spaces. Algorithmic learning theory addresses the handling of noisy data as well as the efficiency in terms of run-time or in terms of examples needed for achieving the

desired accuracy of hypotheses. For more background on machine learning, the reader is referred to Mitchell [66] or Bishop [16].

Efficiency, additional information, and hypothesis spaces are also of particular interest in the field of grammatical inference, thus accounting for strong relations of grammatical inference to the research in the scope of this survey. Indexed families are a main concern of grammatical inference; however there the focus is on structural aspects of target languages which should be represented in the hypotheses. Thus very special, most often class-preserving hypothesis spaces (i.e., hypothesis spaces containing only the languages in the target class) are used. Finite-state automata for instance might be considered as hypotheses when learning regular languages. For some background on grammatical inference, the reader is referred to the literature survey given by de la Higuera [28].

In the subsequent sections, these and other conceptualizations and phenomena will be addressed in the context of learning indexed families in the limit. Basic notions will be introduced in Section 2, followed by the introduction and discussion of Gold's [33] model of identification in the limit as well as some important variants thereof in Section 3. For illustration of the models and results, special classes of regular languages are chosen for a case study in Section 4. Sections 5 and 6 are then concerned with two additional aspects, namely related approaches to learning which differ essentially from Gold's model and the issue of efficiency in learning, respectively. Finally the relevance of the aspects and results discussed will be briefly summarized in Section 7.

2. Preliminaries

2.1. Notations

Familiarity with standard mathematical, recursion theoretic, and language theoretic notions and notations is assumed, see Odifreddi [73] and Hopcroft and Ullman [35].

Let Σ be a finite alphabet. Then Σ^* denotes the free monoid over Σ. We refer to the elements of Σ^* as *strings*. Furthermore, we set $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$, where ε denotes the empty string. A *language* is any subset of Σ^*, i.e., a set of strings. The *complement* $\overline{L}$ of a language L is the set $\Sigma^* \setminus L$.

By $\mathbb{N}$ we denote the set of all natural numbers, i.e., $\mathbb{N} = \{0, 1, \ldots\}$ and we set $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$. For any set Z we write $card(Z)$ in order to refer to the cardinality of Z. By *SEG* we denote the set of all finite sequences ("segments") of strings over Σ and we use a fixed effective one-one numbering $(\sigma_z)_{z \in \mathbb{N}}$ for *SEG*, such that σ_0 is the empty sequence.

In the following, let φ be any fixed Gödel numbering of all partial recursive functions over $\mathbb{N}$ and let Φ be an associated Blum [18] complexity measure. That is, $\varphi_j(x)$ is defined if and only if $\Phi_j(x)$ is defined, and the predicate "$\Phi_j(x) = y$" is uniformly recursive for all $j, x, y \in \mathbb{N}$. For each $j, x \in \mathbb{N}$, one can imagine $\Phi_j(x)$ to be the number of computational steps some fixed universal Turing machine (associated to φ) needs for computing $\varphi_j(x)$.

For $j, n \in \mathbb{N}$ we write $\varphi_j[n]$ for the initial segment $(\varphi_j(0), \ldots, \varphi_j(n))$ and say that $\varphi_j[n]$ is defined if all the values $\varphi_j(0), \ldots, \varphi_j(n)$ are defined. Moreover, let $Tot = \{j \in \mathbb{N} \mid \varphi_j \text{ is a total function}\}$ and $K = \{i \in \mathbb{N} \mid \varphi_i(i) \text{ is defined}\}$. The problem to decide whether or not $\varphi_i(i)$ is defined for any $i \in \mathbb{N}$ is called the *halting problem* with respect to φ. The halting problem with respect to φ is not decidable and thus the set K is not recursive (see, e.g., Odifreddi [73]). Note that *Tot* is not recursive, either.

The family $(W_j)_{j \in \mathbb{N}}$ of languages is defined as follows. For all $j \in \mathbb{N}$ we set $W_j = \{\omega_z \mid z \in \mathbb{N}, \; \varphi_j(z) \text{ is defined}\}$, where $(\omega_z)_{z \in \mathbb{N}}$ is some fixed computable enumeration of Σ^* without repetitions. Moreover, we use a bijective recursive function $\langle \cdot, \cdot \rangle \colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ for coding any pair (x, y) into a number $\langle x, y \rangle$.

If A is any (in general non-recursive) subset of $\mathbb{N}$, then an A-recursive (A-partial recursive) function is a function which is recursive (partial recursive) with the help of an oracle for the set A. That means, an A-recursive (A-partial recursive) function can be computed by an algorithm which has access to an oracle providing correct answers to any question of the type "does x belong to A?" for $x \in \mathbb{N}$.

For many statements, results, and proofs below, the algorithmic structure of language families will play an important role. This requires the following definition. We use r.e. to abbreviate *recursively enumerable*.

Definition 1. Let $(L_j)_{j \in \mathbb{N}}$ be a family of languages.

(i) $(L_j)_{j \in \mathbb{N}}$ is *uniformly recursive* iff there is a recursive function $f \colon \mathbb{N} \times \Sigma^* \to \{0, 1\}$ such that $L_j = \{w \in \Sigma^* \mid f(j, w) = 1\}$ for all $j \in \mathbb{N}$.

(ii) $(L_j)_{j \in \mathbb{N}}$ is *uniformly r.e.* iff there is a partial recursive function $f \colon \mathbb{N} \times \Sigma^* \to \{0, 1\}$ such that $L_j = \{w \in \Sigma^* \mid f(j, w) = 1\}$ for all $j \in \mathbb{N}$.

(iii) $(L_j)_{j \in \mathbb{N}}$ is *uniformly K-r.e.* iff there is a recursive function $g \colon \mathbb{N} \times \Sigma^* \times \mathbb{N} \to \{0, 1\}$ such that $L_j = \{w \in \Sigma^* \mid g(j, w, n) = 1$ for all but finitely many $n\}$ for all $j \in \mathbb{N}$.

The notion "K-r.e." is related to the notion of *A*-recursiveness defined above: if $(L_j)_{j \in \mathbb{N}}$ is uniformly K-r.e., then there is a K-partial recursive function $f \colon \mathbb{N} \times \Sigma^* \to \{0, 1\}$ such that $L_j = \{w \in \Sigma^* \mid f(j, w) = 1\}$ for all $j \in \mathbb{N}$. Note that for uniformly recursive families membership is uniformly decidable.

Throughout this survey we focus our attention on *indexable classes* defined as follows.

Definition 2. A class $\mathcal{C}$ of non-empty recursive languages over Σ^* is said to be *indexable* iff there is a uniformly recursive family $(L_j)_{j \in \mathbb{N}}$ such that $\mathcal{C} = \{L_j \mid j \in \mathbb{N}\}$. Such a family is called an *indexing* of $\mathcal{C}$.

We shall refer to such a class $\mathcal{C}$ as an *indexable class* for short. Note that for each infinite indexable class $\mathcal{C}$ there is a one-one indexing of $\mathcal{C}$, i.e., an indexing $(L_j)_{j \in \mathbb{N}}$ of $\mathcal{C}$, such that for each $L \in \mathcal{C}$ there is exactly one index j with $L = L_j$ (see for instance Lange et al. [52] for a corresponding proof).

Why do we restrict our analysis on such indexable classes? On the one hand, our restriction is not too severe in the sense that most target classes considered in application domains can be represented as indexable language classes. The exclusion of the empty language is mainly for technical reasons, since it simplifies the expositions. Note that many classes of formal languages which are of interest in algorithmics and in formal language theory are indexable, e.g., the class of all non-empty regular languages, the class of all non-empty context-free languages, the class of all non-empty context-sensitive languages, etc.

On the other hand, indexings may be used for representing hypotheses in a learning process (i.e., a learner may use an index i to state that its current hypothesis is L_i). The algorithmic structure of indexings then entails the advantage that hypotheses can be interpreted easier, since there is a uniform effective method for deciding whether or not some certain string is contained in the hypothesized language. This is reflected in the definition of indexed hypothesis spaces below.

Note that many interesting and surprising results were obtained also in the setting of learning classes of recursively enumerable languages (see for instance Osherson et al. [74], Jain et al. [40] and the references therein), many of these motivated by former studies on learning indexed families of recursive languages.

3. Gold-style learning

As mentioned in the Introduction, research on inductive inference was initiated by Gold [33] who defined the basic model of learning in the limit. This section is dedicated to Gold's model and variants thereof. We illustrate these models with basic examples and important necessary and sufficient conditions for learnability.

3.1. The learning model

To define a formal learning model, one must specify the following:

- the admissible *target concepts* and *classes of target concepts*,
- the *learners* to be considered,
- the kind of *information* a learner may receive about the target concept during the learning process,
- the *hypothesis space* the learner may use for communicating its conjectures about the target concept, and
- the *success criterion* by which a learning process is judged.

Throughout the present survey the classes of target concepts are indexable classes and the target concepts are thus recursive languages. The algorithmic structure of indexable classes suggests that they be used as hypothesis spaces. However, the minimal properties a hypothesis space should satisfy are a bit weaker and summarized in the following definition.

Definition 3. Let $\mathcal{C}$ be any indexable class. A *hypothesis space* for $\mathcal{C}$ is a family $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ which fulfills the following two properties.

(1) $\mathcal{H}$ comprises $\mathcal{C}$, i.e., $\mathcal{C} \subseteq \{L_j \mid j \in \mathbb{N}\}$.
(2) There is an effective algorithm which, given any $j \in \mathbb{N}$, enumerates all elements in L_j.

Furthermore, we need the following definition.

Definition 4. A family $(L_j)_{j \in \mathbb{N}}$ is said to be an *indexed hypothesis space* iff $(L_j)_{j \in \mathbb{N}}$ is uniformly recursive.

Note that an indexed hypothesis space *may* also contain *empty* languages in contrast to an indexing of an indexable class. But of course an indexing always constitutes an indexed hypothesis space.

We shall distinguish between the following types of hypothesis spaces for an indexable class $\mathcal{C}$: (i) a suitably chosen indexing of $\mathcal{C}$, (ii) a suitably chosen indexed hypothesis space $\mathcal{H}$ comprising $\mathcal{C}$, and (iii) the family $(W_j)_{j \in \mathbb{N}}$ induced by the Gödel numbering φ.[1]

So, for the remainder of this subsection, let $\mathcal{C}$ be any indexable class, let $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ be any hypothesis space for $\mathcal{C}$, and let $L \in \mathcal{C}$ be any target language.

Next, we consider the information from which the learner should perform its task. Gold [33] was primarily concerned with learning from both positive and negative examples and learning from positive examples only. In the first case, one considers any infinite sequence of all strings over the underlying alphabet that are labeled with respect to their containment in the target language L, while in the second case the source of information is any infinite sequence of strings containing eventually all strings from L. In both cases, the learner receives successively longer finite initial segments. Both approaches are of significant relevance for learning theory, however, due to the space constraints, this survey focuses mainly on learning from positive examples.

The sequences of strings containing eventually all strings from L are also called *texts* and are formally defined as follows.

Definition 5 (*Gold [33]*). Let L be any non-empty language. Every total function $t : \mathbb{N} \to \Sigma^*$ with $\{t(j) \mid j \in \mathbb{N}\} = L$ is called a *text* for L.

We identify a text t with the sequence of its values, i.e., $(t(j))_{j \in \mathbb{N}}$. Furthermore, for any $n \in \mathbb{N}$, the initial segment $(t(0), \ldots, t(n))$ is denoted by $t[n]$ and *content*$(t[n])$ denotes the set $\{t(0), \ldots, t(n)\}$. Note that there is no requirement concerning the computability of a text. Furthermore, a text for a language L may enumerate the elements in any order with any number of repetitions.

We continue with the specification of the learners. These learners, henceforth called *inductive inference machines*, are formally defined as follows.

Definition 6 (*Gold [33]*). An *inductive inference machine* (IIM) M is an algorithmic device that reads successively longer finite initial segments σ of a text and outputs numbers $M(\sigma) \in \mathbb{N}$ as its hypotheses.

Note that, given the hypothesis space $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$, an IIM M returning some j is construed to hypothesize the language L_j.

Finally, we have to specify the success criterion. This criterion is called *learning in the limit*[2] and requires that the sequence converges to a hypothesis correctly describing the target to be learned. Formally, a sequence $(j_n)_{n \in \mathbb{N}}$ is said to *converge* to a number j iff there is a number n_0 such that $j_n = j$ for all $n \geq n_0$.

Definition 7 (*Gold [33]*). Let $\mathcal{C}$ be any indexable class, $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ a hypothesis space, and $L \in \mathcal{C}$. An IIM M *LimTxt$_{\mathcal{H}}$-identifies* L iff

(1) for every text t for L there is a $j \in \mathbb{N}$ such that the sequence $M(t[n])_{n \in \mathbb{N}}$ converges to j, and
(2) $L = L_j$.

Furthermore, M *LimTxt$_{\mathcal{H}}$-identifies* $\mathcal{C}$ iff, for each $L' \in \mathcal{C}$, M *LimTxt$_{\mathcal{H}}$*-identifies L'.

Finally, let *LimTxt* be the collection of all indexable classes $\mathcal{C}'$ for which there is an IIM M' and a hypothesis space $\mathcal{H}'$ such that M' *LimTxt$_{\mathcal{H}'}$*-identifies $\mathcal{C}'$.

---

[1] Note that, in the literature, type (i) hypothesis spaces are called *class preserving* as opposed to *class comprising*, which used to be the term for hypothesis spaces of type (ii). However, since hypothesis spaces of type (iii) in fact also comprise the target class, we no longer use the latter term exclusively for type (ii) hypothesis spaces.

[2] In the literature, this criterion is also called explanatory learning or *Ex*-learning, for short.

In Definition 7 *Lim* stands for "limit" and *Txt* for "text," respectively. Thus, we also say that M learns L in the limit from text with respect to $\mathcal{H}$ if M *LimTxt*$_{\mathcal{H}}$-identifies L.

Since, by the definition of convergence, only finitely many strings in L have been seen by the IIM up to the (unknown) point of convergence, whenever an IIM identifies the possibly infinite language L, some form of learning must have taken place. For this reason, hereinafter the terms *infer*, *learn*, and *identify* are used interchangeably.

Furthermore, another main aspect of human learning is modeled in learning in the limit: the ability to change one's mind during learning. Thus learning is considered as a process in which the learner may change its hypothesis finitely often until reaching its final correct guess. Note that in general it is undecidable whether or not the final hypothesis has been reached.

3.2. Illustrating examples

One of the most straightforward examples of a class learnable in the limit is the class of all finite languages. Given any initial segment $t[n]$ of a text t for a finite language L, an IIM just has to conjecture the language *content*$(t[n])$. As soon as n is large enough such that *content*$(t[n]) = L$, this hypothesis is correct and will never be revised by the IIM. In contrast to this, Gold [33] has shown that any class containing all finite languages and at least one infinite language is *not* learnable in the limit from text.

This result may be considered disappointing, since it immediately implies that many well-known indexable classes are not in *LimTxt*, such as, e. g., the class of all non-empty regular languages or the class of all non-empty context-free languages.

Since these classes are relevant for many application domains, it is worth analyzing the learnability of interesting subclasses thereof. A first step in this direction was done by Angluin [3], who considered restrictions of regular expressions and thus the learnability of subclasses of regular languages. For stating her results, first recall that for any $X, Y \subseteq \Sigma^*$ the *product* of X and Y is defined as $XY = \{xy \mid x \in X, \ y \in Y\}$. Furthermore, we define $X^0 = \{\varepsilon\}$ and for all $i \geq 0$ we set $X^{i+1} = X^i X$. Then the *Kleene closure* of X is defined as $X^* = \bigcup_{i \geq 0} X^i$ and the *semi-group closure* of X is $X^+ = \bigcup_{i \geq 1} X^i$. We refer to the * and $^+$ operator as *Kleene star* and *Kleene plus*, respectively.

The restricted regular expressions considered by Angluin [3] can be defined as follows, where the special reserved symbol $^\times$ can be interpreted as Kleene plus or Kleene star (see below).

Definition 8. Let Σ be any finite alphabet not containing any of the symbols $^\times$, (, and). Then the set of *restricted regular expressions* over Σ is defined inductively as follows.

- For all $a \in \Sigma$, a is a restricted regular expression over Σ.
- If p, q are restricted regular expressions over Σ, then pq and $(p)^\times$ are restricted regular expressions over Σ.

  Note that, using standard techniques, one can effectively enumerate all restricted regular expressions.
  Interpreting $^\times$ as Kleene plus yields the following corresponding class of languages.

Definition 9. Let Σ be a finite alphabet. The *non-erasing language* $L_+(r)$ of a restricted regular expression r over Σ is defined inductively as follows.

- For all $a \in \Sigma$, $L_+(a) = \{a\}$.
- For all restricted regular expressions p and q over Σ, $L_+(pq) = L_+(p)L_+(q)$.
- For all restricted regular expressions p over Σ, $L_+((p)^\times) = L_+(p)^+$.

Let $RREG_+^{\Sigma}$ denote the class of all non-erasing languages of restricted regular expressions over Σ.

Analogously, interpreting $^\times$ as Kleene star we define the *erasing languages* as follows.

Definition 10. Let Σ be a finite alphabet. The *erasing language* $L_*(r)$ of a restricted regular expression r over Σ is defined inductively as follows.

- For all $a \in \Sigma$, $L_*(a) = \{a\}$.
- For all restricted regular expressions p and q over Σ, $L_*(pq) = L_*(p)L_*(q)$.
- For all restricted regular expressions p over Σ, $L_*((p)^\times) = L_*(p)^*$.

Let $RREG_*^\Sigma$ denote the class of all erasing languages of restricted regular expressions over Σ.

Given a finite alphabet Σ, since one can effectively enumerate all restricted regular expressions, both $RREG_+^\Sigma$ and $RREG_*^\Sigma$ are indexable classes and both are subclasses of the class of all regular languages. Next, we ask whether or not $RREG_+^\Sigma$ and $RREG_*^\Sigma$, respectively, are in *LimTxt*.

Angluin [3] has shown that, for any finite alphabet Σ, the class $RREG_+^\Sigma$ is learnable in the limit from text—in contrast to its superclass of all regular languages over Σ. One method to prove this is based on the following fact.

Proposition 11 (*Angluin [3]*). *Let Σ be a finite alphabet and $w \in \Sigma^*$ any string. Then there are only finitely many languages $L \in RREG_+^\Sigma$ such that $w \in L$.*

An IIM learning $RREG_+^\Sigma$ in the limit from text can use any fixed one-one indexing $(L_j)_{j\in\mathbb{N}}$ of $RREG_+^\Sigma$ in order to compute its hypotheses in $(W_j)_{j\in\mathbb{N}}$. Given a text segment $t[n]$, the learner may first determine all indices $j \le n$ such that $content(t[n]) \subseteq L_j$. If there is no such j, the learner may return some arbitrary auxiliary hypothesis. Else the learner may return some index k such that W_k is the intersection of all languages L_j with $j \le n$ and $content(t[n]) \subseteq L_j$. Since, by Proposition 11, there are only finitely many such indices for each $t[n]$, the conjectures returned by this IIM will eventually stabilize on the intersection of all languages in $RREG_+^\Sigma$ containing the target language. Obviously, this intersection must equal the target language. Note that this proof works independently from the size of the underlying alphabet Σ. Moreover, this is not the only successful method, as we will see in Section 3.3.1.

Angluin also considered the case where the symbol $^\times$ is interpreted as Kleene star. Here the main difference is that each substring of a regular expression $(p)^\times$ may be deleted, or, in other words, substituted by the empty string when generating strings in the corresponding regular language.

It was shown by Angluin [3] that the class $RREG_*^\Sigma$ is not learnable in the limit from text, if Σ contains at least two symbols. To prove that, Angluin showed that some characteristic criterion for learnability in the limit is not fulfilled for $RREG_*^\Sigma$. We will discuss this criterion later in Theorem 44.

However, if Σ is a singleton set, then $RREG_*^\Sigma \in LimTxt$ can be verified as follows.

Let $\Sigma = \{a\}$. Note that each $L \in RREG_*^{\{a\}}$ is generated by some restricted regular expression of the form

$$a^m (a^{k_1})^\times (a^{k_2})^\times \cdots (a^{k_s})^\times$$

with $m, s, k_1, \ldots, k_s \in \mathbb{N}$, such that $k_1 < k_2 < \cdots < k_s$ and

$$k_x \text{ is not a linear combination of the values in } \{k_1, \ldots, k_s\} \setminus \{k_x\}$$

for any $x \in \{1, \ldots, s\}$.

The reason is that (i) the order of subexpressions does not matter in case of a singleton alphabet, i.e., $L_*(pq) = L_*(qp)$, and (ii) $L_*(((p)^\times)^\times) = L_*((p)^\times)$. Based on this property, an IIM for $RREG_*^{\{a\}}$ may work according to the following instructions.

Input: $t[n]$ for some text t of a language $L \in RREG_*^{\{a\}}$ and some $n \in \mathbb{N}$
Output: restricted regular expression p

(1) Let m be the length of the shortest string in $content(t[n])$;
    Let $p = a^m$; let $z = 1$;
(2) While there is some $w \in content(t[n]) \setminus L_*(p)$ do {
    (a) Let $w \in content(t[n])$ be the shortest string such that $w \notin L_*(p)$;
        Let m' be the length of w and let $k_z = m' - m$;
    (b) Let $p = p(a^{k_z})^\times$ and $z = z + 1$;   }.
(3) Return p.

Note that this IIM returns regular expressions as its hypotheses and thus uses a hypothesis space canonically induced by an enumeration of such expressions. The proof of correctness of this procedure is left to the reader.

3.3. Sufficient conditions

3.3.1. Finite thickness, finite elasticity, and characteristic sets

Up to now, we have collected some examples of learnable and non-learnable classes for Gold's model of learning in the limit. However, these do not immediately yield any criteria distinguishing the learnable classes from those not learnable. Some distinguished properties of learnable classes would be helpful tools in this context.

For that purpose we will use sufficient or necessary criteria for a class being learnable in the limit from positive examples. A sufficient but not necessary condition is *finite thickness*.

Definition 12 (*Angluin [3]*). A class $\mathcal{C}$ is of *finite thickness* iff for each string $s \in \Sigma^*$ there are at most finitely many languages in $\mathcal{C}$ containing s, i. e., the class $\{L \in \mathcal{C} \mid s \in L\}$ is finite.

A simple example for a class of finite thickness is $RREG_+^\Sigma$ for an arbitrary finite alphabet Σ. The finite thickness property of $RREG_+^\Sigma$ is essentially what Proposition 11 states.

Theorem 13 (*Angluin [3]*). *Let $\mathcal{C}$ be an indexable class. If $\mathcal{C}$ is of finite thickness, then $\mathcal{C} \in LimTxt$.*

The converse is in general not true, i.e., there are classes in *LimTxt* which are not of finite thickness, such as, for instance, the class of all finite languages.

There are several methods by which an IIM may exploit the finite thickness property of the target class. The one used for the proof of $RREG_+^\Sigma \in LimTxt$ above can be generalized to the following learning method, if a one-one indexing $(L_j)_{j\in\mathbb{N}}$ of the target class $\mathcal{C}$ is given. Let $(L_k')_{k\in\mathbb{N}}$ be an indexing comprising $\mathcal{C}$, such that $L_k' = L_{j_1} \cap \cdots \cap L_{j_z}$, if k is the canonical index of the finite set $\{j_1, \ldots, j_z\}$. The proposed learning method uses the family $(L_k')_{k\in\mathbb{N}}$ – which is itself a uniformly recursive family of languages – as a hypothesis space.

On input $t[n]$, compute the set $D = \{j \mid j \le n,\ content(t[n]) \subseteq L_j\}$. Return the canonical index k of D.

Informally, the method used here considers a set of possible hypotheses in each learning step and outputs a hybrid hypothesis which is constructed from this set. Since $\mathcal{C}$ is of finite thickness and $(L_j)_{j\in\mathbb{N}}$ is a one-one indexing of $\mathcal{C}$, there is, for any $L \in \mathcal{C}$ and any text t for L, an index n such that even $t(0) \notin L_j$ for every $j > n$. This simple observation is crucial for showing that this method stabilizes on a correct hypothesis for the target language.

If it is desired to work in an *incremental*, memory-efficient manner, one additional property in the context of finite thickness is needed. Here one requires that it is possible to compute – for any $w \in \Sigma^*$ – a finite set of indices which includes indices for all languages in the target class containing w.

Definition 14 (*Koshiba [49], Lange and Zeugmann [56]*). An indexable class $\mathcal{C}$ is of *recursive finite thickness* iff $\mathcal{C}$ is of finite thickness and there is an indexing $(L_j)_{j\in\mathbb{N}}$ of $\mathcal{C}$ and an algorithm which, on any input $w \in \Sigma^*$, returns indices $j_1, \ldots j_z$, such that $\{L_{j_1}, \ldots, L_{j_z}\} = \{L \in \mathcal{C} \mid w \in L\}$.

Incremental learning means that the memory of the learner is bounded in advance. The study of such learning methods is motivated by the somewhat unrealistic feature of the model of *LimTxt* which demands that an IIM has the memory capacities to process text segments of unbounded length.

An incremental method for learning a class $\mathcal{C}$ which is of recursive finite thickness, witnessed by an indexing $(L_j)_{j\in\mathbb{N}}$ of $\mathcal{C}$, uses an indexing $(L_k')_{k\in\mathbb{N}}$ comprising $\mathcal{C}$, such that $L_k' = L_{j_1} \cap \cdots \cap L_{j_z}$, if k is the canonical index of the finite set $\{j_1, \ldots, j_z\}$.

On input $t[0]$, compute the set $D = \{j \mid j \in \mathbb{N},\ t(0) \in L_j\}$. (* This is possible because of the recursive finite thickness property. *) Return the canonical index of D.
On input $t[n + 1]$ for some $n \in \mathbb{N}$, let k be the hypothesis returned on input $t[n]$. Compute the set D for which k is the canonical index. Compute the set $D' = \{j \mid j \in D,\ t(n + 1) \in L_j\}$. Return the canonical index of D'.

This method uses a recursive indexing comprising the target class and is *iterative*, i.e., in each step of the learning process, it uses only its previous hypothesis and the latest positive example presented in the text. None of the formerly presented examples are required. Such a method could be considered as very memory-efficient, since none of the strings acquired during learning need to be stored; thus it constitutes a special case of incremental learning. Iterative learning was studied by Wiehagen [92] in the context of inferring recursive functions. Lange and Zeugmann [54]

have transferred Wiehagen's model to the case of learning formal languages. We refer the reader to Lange and Zeugmann [56] and Lange [50] for a detailed study concerning the incremental learnability of indexed families.

It is worth noting that recursive finite thickness can even be exploited for learning bounded unions of languages from an indexable class $\mathcal{C}$. Below, for any $k \in \mathbb{N}^+$, we use $\mathcal{C}^k$ to denote the class of all unions of up to k languages from $\mathcal{C}$, i.e.,

$$\mathcal{C}^k = \{L_1 \cup \cdots \cup L_k \mid L_1, \ldots, L_k \in \mathcal{C}\}.$$

Theorem 15 (*Case et al. [20], Lange [50]*). *Let $\mathcal{C}$ be an indexable class and $k \in \mathbb{N}^+$. If $\mathcal{C}$ has recursive finite thickness, then $\mathcal{C}^k \in LimTxt$.*

If $(L_j)_{j \in \mathbb{N}}$ is an indexing as required in the definition of recursive finite thickness, the idea for a corresponding learner can be sketched as follows. We construct a hypothesis space different than $(L_j)_{j \in \mathbb{N}}$. We ensure that each language L in the hypothesis space is the intersection of finitely many languages, each of which is the union of up to k languages in $\mathcal{C}$. Formally, let $(D_j)_{j \in \mathbb{N}}$ be the canonical indexing of all finite subsets of $\mathbb{N}$. The required hypothesis space $(L'_j)_{j \in \mathbb{N}}$ is defined as follows: for all $j \in \mathbb{N}$ we let $L'_j = \bigcap_{z \in D_j} (\bigcup_{r \in D_z} L_r)$.

> On input $t[0]$, compute the set $D = \{j \mid j \in \mathbb{N}, \, t(0) \in L_j\}$. (* This is possible because of the recursive finite thickness property. *) Return the canonical index z of the singleton set $D' = \{r\}$, where r is the canonical index of D.
>
> On input $t[n + 1]$ for some $n \in \mathbb{N}$, let z be the hypothesis returned on input $t[n]$. Compute the set D with the canonical index z. For each $m \in D$ distinguish the following cases.
>
> (a) If $t(n + 1) \in \bigcup_{r \in D_m} L_r$, then mark the index m "alive".
> (b) If $t(n + 1) \notin \bigcup_{r \in D_m} L_r$ and $card(D_m) = k$, kill the index m.
> (c) If $t(n + 1) \notin \bigcup_{r \in D_m} L_r$ and $card(D_m) < k$, kill the index m, as well. In addition compute the set $D' = \{j \mid j \in \mathbb{N}, \, t(n + 1) \in L_j\}$. (* This is possible because of the recursive finite thickness property. *) For every $j \in D'$, compute the canonical index m_j of the set $D_m \cup \{j\}$ and mark the index m_j as "alive".
>
> Return the canonical index z' of the set of all indices that are marked "alive".

However, this result cannot be generalized to the case of learning arbitrary finite unions of languages in $\mathcal{C}$. Consider for instance an indexing given by $L_0 = \Sigma^*$ and $L_{j+1} = \{\omega_j\}$ for $j \in \mathbb{N}$, where $(\omega_j)_{j \in \mathbb{N}}$ is a fixed enumeration of all strings in Σ^*. Obviously, this indexing satisfies the requirements of recursive finite thickness. However, the class of all finite unions of languages therein contains all finite languages as well as one infinite language, namely Σ^*. By Gold [33], no such class is in *LimTxt*.

Another sufficient criterion for learnability in the limit from text is *finite elasticity*.

Definition 16 (*Wright [97], Motoki et al. [68]*). A class $\mathcal{C}$ is of *infinite elasticity* iff there is an infinite sequence $w_0, w_1, \ldots$ of strings and an infinite sequence $L_1, L_2, \ldots$ of languages in $\mathcal{C}$ such that the following conditions are fulfilled for all $n \in \mathbb{N}^+$.

(1) $\{w_0, \ldots, w_{n-1}\} \subseteq L_n$.
(2) $w_n \notin L_n$.

$\mathcal{C}$ is of *finite elasticity* iff $\mathcal{C}$ is not of infinite elasticity.[3]

Theorem 17 (*Wright [97]*). *Let $\mathcal{C}$ be an indexable class. If $\mathcal{C}$ is of finite elasticity, then $\mathcal{C} \in LimTxt$.*

Again, this criterion is sufficient, but not necessary for learnability in the limit from text. It is easily seen that the class of all finite languages is in *LimTxt* but is of infinite elasticity. Note that each class possessing the finite thickness property is also of finite elasticity. The converse is not valid in general; for instance, the class of all languages containing exactly two strings is of finite elasticity but not of finite thickness.

Finite elasticity of a class $\mathcal{C}$ additionally allows for statements concerning the learnability of the classes $\mathcal{C}^k$ for $k \in \mathbb{N}^+$. To see this, the following key property can be used.

---

[3] The concept of finite elasticity was introduced by Wright [97], but the original definition was corrected later on by Motoki, Shinohara, and Wright [68].

Theorem 18 (*Wright [97]*). *Let $\mathcal{C}$ be an indexable class and $k \in \mathbb{N}^+$. If $\mathcal{C}$ is of finite elasticity, then $\mathcal{C}^k$ is of finite elasticity.*

This immediately yields the following corollary.

Corollary 19 (*Wright [97]*). *Let $\mathcal{C}$ be an indexable class and $k \in \mathbb{N}^+$. If $\mathcal{C}$ is of finite elasticity, then $\mathcal{C}^k \in LimTxt$.*

In particular, since each class of finite thickness is also of finite elasticity, we obtain a stronger result than Theorem 15.

Corollary 20. *Let $\mathcal{C}$ be an indexable class and $k \in \mathbb{N}^+$. If $\mathcal{C}$ is of finite thickness, then $\mathcal{C}^k \in LimTxt$.*

Note that finite elasticity was also used to show the learnability of several concept classes defined by using elementary formal systems (see for instance Shinohara [86], Moriyama and Sato [67], and the references therein).

At the end of this subsection, we discuss another sufficient criterion for learnability in the limit from text. This criterion has found several interesting applications in analyzing the learnability of natural language classes (see for instance Kobayashi and Yokomori [48], Sato et al. [84] and Ng and Shinohara [71]), as well as in investigating their polynomial-time learnability (see for example de la Higuera [27]).

Definition 21 (*Angluin [4]*). Let $(L_j)_{j \in \mathbb{N}}$ be any indexing. A family of non-empty finite sets $(S_j)_{j \in \mathbb{N}}$ is called a *family of characteristic sets for* $(L_j)_{j \in \mathbb{N}}$ iff the following conditions are fulfilled for all $j, k \in \mathbb{N}$.

(1) $S_j \subseteq L_j$.
(2) If $S_j \subseteq L_k$, then $L_j \subseteq L_k$.

For all j, S_j is called a *characteristic set* for L_j.

Theorem 22 (*Kobayashi [47]*). *Let $\mathcal{C}$ be an indexable class. If there is an indexing $(L_j)_{j \in \mathbb{N}}$ of $\mathcal{C}$ which possesses a family of characteristic sets, then $\mathcal{C} \in LimTxt$.*

Sketch of proof. Let $(S_j)_{j \in \mathbb{N}}$ be a family of characteristic sets for an indexing $(L_j)_{j \in \mathbb{N}}$ of $\mathcal{C}$ and $(w_j)_{j \in \mathbb{N}}$ an effective enumeration of all strings in Σ^*. An IIM M learning $\mathcal{C}$ in the limit from text with respect to $(L_j)_{j \in \mathbb{N}}$ works according to the following instructions.

> Input: $t[n]$ for some text t of a language $L \in \mathcal{C}$ and some $n \in \mathbb{N}$
> Output: hypothesis $M(t[n]) \in \mathbb{N}$
>
> (1) Determine $\mathcal{C}' = \{j \mid j \le n,\ content(t[n]) \subseteq L_j\}$.
> (2) If there is an index $j \in \mathcal{C}'$ such that, for all $k < j$, Condition (i) below is fulfilled, then fix the least such j and return $M(t[n]) = j$. Otherwise, return $M(t[n]) = 0$.
>   (i) $\{w_z \mid z \le n,\ w_z \in L_k\} \setminus \{w_z \mid z \le n,\ w_z \in L_j\} \ne \emptyset$.

Let L be the target language and j the least index with $L_j = L$. Since t is a text for L and S_j is a characteristic set for L_j, there is some n such that $j \le n$, $S_j \subseteq content(t[n])$, and $\{w_z \mid z \le n,\ w_z \in L_k\} \setminus \{w_z \mid z \le n,\ w_z \in L_j\} \ne \emptyset$ for all $k < j$ with $S_j \subseteq L_k$. Consequently, $M(t[m]) = j$ for all $m \ge n$, and therefore M learns L as required. $\quad\square$

The criterion given in Theorem 22 is sufficient, but not necessary for learnability in the limit from text, as the following example demonstrates. Let $\Sigma = \{a\}$, $L_j = \{a\}^+ \setminus \{a^{j+1}\}$, and $\mathcal{C} = \{L_j \mid j \in \mathbb{N}\}$. It is well-known that $\mathcal{C} \in LimTxt$. However, all languages in $\mathcal{C}$ are pairwise incomparable, and therefore there does not exist an indexing $(L_j)_{j \in \mathbb{N}}$ of $\mathcal{C}$ which possesses a family of characteristic sets.

Theorem 23. *Let $\mathcal{C}$ be an indexable class. If $\mathcal{C}$ is of finite elasticity, then there is an indexing of $\mathcal{C}$ which possesses a family of characteristic sets.*

Sketch of proof. Let $\mathcal{C}$ be an indexable class of languages that is of finite elasticity, $(L_j)_{j \in \mathbb{N}}$ any indexing of $\mathcal{C}$, and $(w_j)_{j \in \mathbb{N}}$ an effective enumeration of all strings in Σ^*.

Let $j \in \mathbb{N}$ and w the string in L_j which appears first in $(w_j)_{j \in \mathbb{N}}$. Consider the following definition of a sequence $w_0', w_1', \ldots$ of strings in L_j and a sequence $L_1', L_2', \ldots$ of languages in $\mathcal{C}$.

Stage 0: Set $w'_0 = w$.

Stage n, $n > 0$: Let $w'_0, \ldots, w'_{n-1}$ denote the strings which have been defined in the previous stages. For all $k \in \mathbb{N}$ with $\{w'_0, \ldots, w'_{n-1}\} \subseteq L_k$ determine whether or not there is a string w_k with $w_k \in L_j \setminus L_k$. If such an index k exists, set $w'_n = w_k$, $L'_n = L_k$, and goto Stage $n + 1$.

Suppose that, for every $n > 0$, a string $w'_n \in L_j$ and a language $L'_n \in \mathcal{C}$ has been defined. Then there is an infinite sequence $w'_0, w'_1, \ldots$ of strings and an infinite sequence of languages $L'_1, L'_2, \ldots$ such that, for all $n > 0$, $\{w'_0, \ldots, w'_{n-1}\} \subseteq L'_n$ and $w'_n \notin L'_n$. Hence $\mathcal{C}$ would be of infinite elasticity, a contradiction. Consequently, there has to be some n, such that Stage n will never be finished. Let $S_j = \{w'_0, \ldots, w'_{n-1}\}$ and let k be an index with $S_j \subseteq L_k$. If $L_j \not\subseteq L_k$, there has to be a string $w_k \in L_j \setminus L_k$. But this would imply that Stage n will be finished. Consequently, it must be the case that $L_j \subseteq L_k$, and therefore S_j is a characteristic set of L_j. $\square$

Next we ask whether or not the converse of Theorem 23 is true as well, which would mean that an indexable class is of finite elasticity, if it has an indexing possessing a family of characteristic sets. However, in general this is not the case, as our next result shows.

Theorem 24. *There is an indexable class $\mathcal{C}$ such that the following requirements are fulfilled.*

(1) *There is an indexing of $\mathcal{C}$ which possesses a family of characteristic sets.*
(2) *$\mathcal{C}$ is of infinite elasticity.*

Sketch of proof. Let $\Sigma = \{a\}$, $L_j = \{a, \ldots, a^{j+1}\}$, and $\mathcal{C} = \{L_j \mid j \in \mathbb{N}\}$.

ad (1) For all $j \in \mathbb{N}$ set $S_j = L_j$. Obviously, $S_j \subseteq L_k$ implies $L_j \subseteq L_k$ for all $j, k \in \mathbb{N}$.

ad (2) For all $j \in \mathbb{N}$ set $w'_j = a^{j+1}$ and $L'_{j+1} = L_j$. Obviously, for all $n > 0$, $L'_n = \{w'_0, \ldots, w'_{n-1}\}$, and therefore $\{w'_0, \ldots, w'_{n-1}\} \subseteq L'_n$ and $w'_n \notin L'_n$. Hence $\mathcal{C}$ is of infinite elasticity. $\square$

3.3.2. Telltales

A further criterion sufficient for learnability in the limit is based on an analysis of prototypical learning processes. The idea is to consider IIMs in limit learning processes in general. If an IIM M learns a target language L in the limit from text, then of course each text t for L must start with a *stabilizing sequence* for M and t, i.e., an initial segment of t after which M will never change its mind on t again. A closer inspection reveals: there must be a text segment σ of some text for L, which is a stabilizing sequence for *every* text for L starting with σ. Otherwise it would be possible to construct a text for L on which M fails to converge; any text segment for L could be gradually extended in such a way that infinitely many mind changes of M are enforced and that each string in L eventually occurs in the text. The existence of such special segments σ – called *stabilizing sequences for M and L* – was verified by Blum and Blum [17].

Definition 25 (*Blum and Blum [17]*). Let L be a recursive language, $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ some hypothesis space comprising $\{L\}$, and M an IIM. Let $\sigma \in SEG$ with $content(\sigma) \subseteq L$.

(1) σ is called a *stabilizing sequence* for M and L iff $M(\sigma\tau) = M(\sigma)$ for all $\tau \in SEG$ with $content(\tau) \subseteq L$.
(2) σ is called a *locking sequence* for M, L, and $\mathcal{H}$ iff $L_{M(\sigma)} = L$ and σ is a stabilizing sequence for M and L.

Note that, if M learns L in the limit with respect to $\mathcal{H}$, then each stabilizing sequence for M and L must be a locking sequence for M, L, and $\mathcal{H}$.

The significance of stabilizing sequences is in the fact that they seem to bear sufficient information for a learner to identify the given language. It is quite natural to assume that it is the content and not the order of the strings in a stabilizing sequence which forms the relevant information. In other words, perhaps, for each language L learned by IIM M, there is a finite subset $T \subseteq L$ such that, when presented to M in the form of a text segment, T bears sufficient information for M to identify L. In particular, this subset T must not be contained in any proper subset L' of L which is learned by M, as well. Otherwise it would not contain enough information to separate L' from L: if M conjectured L after having seen all strings in T and M stuck to this hypothesis as long as only strings in L are presented, then M would fail to learn L', because each string in L' is also an element of L. That means, there would be a text for L' on which M converged to a hypothesis representing L.

Indeed, the concept of such finite sets T, called telltales, was found and analyzed by Angluin [3].

Definition 26 (*Angluin [3]*). Let $(L_j)_{j \in \mathbb{N}}$ be any indexing. A family $(T_j)_{j \in \mathbb{N}}$ of finite non-empty sets is called a *family of telltales for* $(L_j)_{j \in \mathbb{N}}$ iff the following conditions are fulfilled for all $j, k \in \mathbb{N}$.

(1) $T_j \subseteq L_j$.
(2) If $T_j \subseteq L_k \subseteq L_j$, then $L_k = L_j$.

T_j is then called a *telltale* for L_j with respect to $(L_j)_{j \in \mathbb{N}}$.

Note that Angluin's [3] original definition, in what she called *Condition* 1 on indexed families of recursive languages, used an alternative formulation of (2), equivalent to the one given here; hers says: if $T_j \subseteq L_k$, then L_k is not a proper subset of L_j.

One part of Angluin's analysis yields: if $\mathcal{C} \in \textit{LimTxt}$ is an indexable class of languages, then there is an indexing of $\mathcal{C}$ which possesses a family of telltales.

This is a very important result, though in this form it states only a necessary, but not a sufficient condition for learnability in the limit from text. In order to have a telltale structure an IIM can utilize, these telltales must be "accessible" algorithmically, i.e., the IIM must "know" a procedure for computing the telltales in order to check whether some relevant information for hypothesizing a certain language L has already appeared in the text. As it turns out, it is sufficient that there exists a procedure enumerating the telltales for all languages in an indexing used as a hypothesis space, even if there is no criterion for deciding how many strings will be enumerated into a telltale!

Proposition 27 (*Angluin [3]*). *Let $\mathcal{C}$ be an indexable class. If there is an indexing comprising $\mathcal{C}$, which possesses a uniformly r.e. family of telltales, then $\mathcal{C} \in \textit{LimTxt}$.*

Sketch of proof. Angluin's proof can be sketched as follows. Choose an indexing $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ comprising $\mathcal{C}$, which possesses a uniformly r. e. family $(T_j)_{j \in \mathbb{N}}$ of telltales. Fix a partial recursive function f such that $T_j = \{w \in \Sigma^* \mid f(j, w) = 1\}$ for all $j \in \mathbb{N}$. An IIM M learning $\mathcal{C}$ in the limit from text with respect to $\mathcal{H}$ works according to the following instructions.

Input: $t[n]$ for some text t of a language $L \in \mathcal{C}$ and some $n \in \mathbb{N}$
Output: hypothesis $M(t[n]) \in \mathbb{N}$

(1) For all $s \leq n$, let $T_s[n]$ be the set of all strings ω_z such that $z \leq n$, $f(s, \omega_z)$ is defined within n steps of computation, and $f(s, \omega_z) = 1$; (* $(\omega_z)_{z \in \mathbb{N}}$ is a fixed effective enumeration of Σ^* *)
(2) If there is no $s \leq n$ such that $T_s[n] \subseteq content(t[n]) \subseteq L_s$, then return $M(t[n]) = 0$ and stop;
(3) If there is some $s \leq n$ such that $T_s[n] \subseteq content(t[n]) \subseteq L_s$, then return $M(t[n]) = s$ for the least such s and stop.

Informally, M looks for the minimal consistant hypothesis whose telltale is consistent with, and is comprised by, the current text segment.

Suppose L is the target language. Since the telltale sets are finite, the conjectures of M will eventually converge to an index j for a language L_j which

- contains all strings in the presented text (* thus $L \subseteq L_j$ *) and
- has a telltale T_j completely contained in the text (* thus $T_j \subseteq L$ *).

This yields $T_j \subseteq L \subseteq L_j$ for the final conjecture j. The telltale property then implies $L = L_j$, i.e., the conjecture returned by M in the limit is correct. Since this holds for arbitrary $L \in \mathcal{C}$, we obtain $\mathcal{C} \in \textit{LimTxt}$. $\quad \square$

For some classes of languages, there is a procedure for enumerating (uniformly in j) all strings in the telltale T_j, and then stopping afterwards. In such a case, the telltale family is called *recursively generable*.

Definition 28. A family $(T_j)_{j \in \mathbb{N}}$ of finite languages is *recursively generable* iff there is an algorithm that, given $j \in \mathbb{N}$, enumerates all elements of T_j and stops.

A reasonable question would be whether telltales with such an algorithm structure would have any consequence for learning methods. As was shown by Lange and Zeugmann [55], recursively generable telltales allow for learning a target class *conservatively*, i.e., with only justified mind changes in the sense that a learner does not change its conjecture as long as that conjecture is *consistent* with the data seen in the text, where consistency is defined as follows.

Definition 29 (*Barzdin [10], Blum and Blum [17]*). Let t be a text for some language, let $n \in \mathbb{N}$, and let L be a language. The segment $t[n]$ is said to be *consistent* with L iff $content(t[n]) \subseteq L$. Otherwise $t[n]$ is said to be *inconsistent* with L.

Conservative learning essentially demands that a learner stick to consistent hypotheses. Note that this requirement is not explicated in the definition of *LimTxt*.

Definition 30 (*Angluin [3], Lange and Zeugmann [55]*). Let $\mathcal{C}$ be an indexable class and $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ a hypothesis space for $\mathcal{C}$. An IIM M is *conservative for $\mathcal{C}$ with respect to $\mathcal{H}$* iff for any $L \in \mathcal{C}$, any text t for L, and any $n \in \mathbb{N}$: if $j = M(t[n]) \neq M(t[n + 1])$, then $t[n + 1]$ is inconsistent with L_j.

The collection of all indexable classes identifiable in the limit from text by a conservative IIM, with respect to some adequate hypothesis space, is denoted by *ConsvTxt*.

Demanding a learner to be conservative brings us to another important problem, i.e., to avoid or to detect *overgeneralizations* (see for instance Berwick [15], Wexler [90] for a discussion). By overgeneralization we mean that the learner may output a hypothesis that is a proper superset of the target language. Clearly, the incorrectness of an overgeneralized hypothesis cannot be detected while learning from text. Thus one may be tempted to design only IIMs that avoid overgeneralized hypotheses.

An IIM M that learns an indexable class $\mathcal{C}$ conservatively with respect to a hypothesis space $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ has the advantage that it *never* overgeneralizes. That is for every $L \in \mathcal{C}$, every text t for L, and every $n \in \mathbb{N}$, we always have $L_{M(t[n])} \not\supseteq L$. In contrast, the uniform learning method proposed by Angluin (see the proof of Proposition 27) may output overgeneralized hypotheses. This learning method always chooses the minimal consistant hypothesis such that the already enumerated part of the telltale is in the set of strings seen so far and this set is in the hypothesized language. Such a possibly overgeneralized hypothesis may be abandoned later, if the corresponding telltale has not yet been completely enumerated.

The relation of conservative inference to recursively generable telltale families can be stated as follows.

Proposition 31 (*Lange and Zeugmann [55]*). *Let $\mathcal{C}$ be an indexable class. If there is an indexing comprising $\mathcal{C}$ which possesses a recursively generable family of telltales, then $\mathcal{C} \in ConsvTxt$.*

Though it seems quite natural to demand conservativeness, this requirement significantly restricts the capabilities of IIMs. As Lange and Zeugmann [55] have shown, there is an indexable class in *LimTxt* which cannot be learned conservatively. In particular, this result shows that overgeneralized hypotheses are *inevitable*, in general, if one considers language learning from positive examples. For a more detailed discussion we refer the reader to Zeugmann and Lange [102].

Theorem 32 (*Lange and Zeugmann [55]*). *ConsvTxt $\subset$ LimTxt.*

Sketch of proof. Let φ be our fixed Gödel numbering and let Φ be the associated Blum complexity measure (see Section 2.1). For any $i, x \in \mathbb{N}$, we write $\Phi_i(x) \downarrow$ if $\Phi_i(x)$ is defined and $\Phi_i(x) \uparrow$ otherwise. Let $\mathcal{C} = \{L_k \mid k \in \mathbb{N}\} \cup \{L_{k,j} \mid k, j \in \mathbb{N}\}$, where $L_k = \{a^k b^z \mid z \in \mathbb{N}\}$ for all k and

$$L_{k,j} = \begin{cases} \{a^k b^z \mid z \leq \Phi_k(k) - j\} & \text{if } j < \Phi_k(k) \downarrow, \\ L_k & \text{if } j \geq \Phi_k(k) \downarrow \text{ or } \Phi_k(k) \uparrow, \end{cases}$$

for all $k, j \in \mathbb{N}$.

It is not hard to see that $\mathcal{C}$ is an indexable class and that $\mathcal{C} \in LimTxt$. Showing $\mathcal{C} \notin ConsvTxt$ is done indirectly. Suppose $\mathcal{C} \in ConsvTxt$, then the halting set K defined in Section 2.1 would be recursive, a contradiction. For details the reader is referred to Lange and Zeugmann [55]. □

Note that the class $\mathcal{C}$ used in the proof of Theorem 32 is even of finite thickness, so finite thickness alone is not sufficient for conservative learnability. Yet recursive finite thickness is a sufficient condition.

Theorem 33 (*Case et al. [20], Lange [50]*). *Let $\mathcal{C}$ be an indexable class. If $\mathcal{C}$ has recursive finite thickness, then $\mathcal{C}^k \in ConsvTxt$ for any $k \in \mathbb{N}^+$.*

Comparing Proposition 27 to Proposition 31, we see that the difference between *LimTxt* and *ConsvTxt* stems from the algorithmic complexity of the corresponding telltale families, i.e., uniformly r.e. versus recursively generable. With this in mind, it is reasonable to consider indexable classes possessing telltale families with no associated algorithmicity, whatsoever. Interestingly, this yields a connection to yet another model of inductive inference called behaviorally correct learning. Here the sequence of hypotheses conjectured by the learner is no longer required to converge syntactically, but only to converge semantically. This means, after some point in the learning process, all hypotheses returned by the inference machine must correctly describe the target language, yet the learner may alternate its conjectures between different correct ones.

Definition 34 (*Feldman [30], Barzdin [11,14], Case and Lynes [22]*). Let $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ be any hypothesis space, M an IIM, L a recursive language, and t a text for L.

(1) *M identifies L behaviorally correctly from t with respect to $\mathcal{H}$ iff $L_{M(t[n])} = L$ for all but finitely many $n \in \mathbb{N}$.*
(2) *M identifies L behaviorally correctly from text with respect to $\mathcal{H}$ iff it identifies L behaviorally correctly from t for every text t for L.*

Furthermore, *M behaviorally correctly* identifies an indexable class $\mathcal{C}$ from text with respect to $\mathcal{H}$ iff it identifies every $L \in \mathcal{C}$ behaviorally correctly from text with respect to $\mathcal{H}$.

In the following, we use the notion *BcTxt* for the collection of all indexable classes $\mathcal{C}$ for which there is an IIM M and a hypothesis space $\mathcal{H}$ such that M behaviorally correctly identifies $\mathcal{C}$ from text with respect to $\mathcal{H}$.

Next, we present the relation between *BcTxt*-learning and the mere existence of telltales.

Proposition 35 (*Baliga et al. [8]*). *Let $\mathcal{C}$ be an indexable class. If there is an indexing comprising $\mathcal{C}$ which possesses a family of telltales, then $\mathcal{C} \in BcTxt$.*

Giving up the requirement of syntactic convergence of the sequence of hypotheses as in *LimTxt*, and weakening the constraints to semantic convergence, yields an increase in learning power, i.e., there are indexable classes learnable behaviorally correctly from text, but not learnable in the limit from text. This result, formally stated in the next theorem, can actually be obtained from Proposition 35 in combination with a result proved by Angluin [3] (see Baliga et al. [8] for a discussion).

Theorem 36. *LimTxt $\subset$ BcTxt.*

3.4. The impact of hypothesis spaces

Below Definition 3, three different types of hypothesis spaces were proposed for learning an indexable class $\mathcal{C}$ in the limit. Obviously, if $\mathcal{C}$ is learnable in an indexing of $\mathcal{C}$, then $\mathcal{C}$ is learnable in an indexing comprising $\mathcal{C}$, because each indexing of $\mathcal{C}$ in particular comprises $\mathcal{C}$. Not much harder to prove is the fact that, if $\mathcal{C}$ is learnable in an indexing comprising $\mathcal{C}$, then $\mathcal{C}$ is also learnable in the family $(W_j)_{j \in \mathbb{N}}$ induced by the Gödel numbering φ. However, it is not that obvious whether or not the converse relations hold as well. This section deals with an analysis of the impact of hypothesis spaces for learning in the limit, conservative learning in the limit, and behaviorally correct learning.

3.4.1. Learning in the limit

Concerning the *LimTxt*-model, the essential observation is that the type of hypothesis space does not matter at all as pointed out in Gold [33] (see the proof of Theorem 10.1. on pp. 464).

Theorem 37. *Let $\mathcal{C}$ be an indexable class. Then the following statements are equivalent.*

(1) $\mathcal{C} \in LimTxt$.
(2) $\mathcal{C}$ is learnable in the limit with respect to an indexing of $\mathcal{C}$.

(3) $\mathcal{C}$ is learnable in the limit with respect to any indexing of $\mathcal{C}$.

Thus in particular the expressiveness of Gödel numberings as hypothesis spaces does not offer any advantage over exact indexings for learning a class in the limit from positive examples. Moreover, each of the infinitely many exact indexings of the class is suitable. As it turns out, this result also holds, if the notion of hypothesis space is generalized to, for instance, uniformly K-r. e. families as defined in Definition 1.

Moreover, the proof of Theorem 37 is constructive in the sense that it provides a uniform method which, given an indexing $(L_j)_{j \in \mathbb{N}}$, a hypothesis space $\mathcal{H}$, and a learner M identifying $\{L_j \mid j \in \mathbb{N}\}$ with respect to $\mathcal{H}$, computes a program for an IIM M' identifying $\{L_j \mid j \in \mathbb{N}\}$ with respect to $(L_j)_{j \in \mathbb{N}}$.

3.4.2. Behaviorally correct learning

In contrast to the case of learning in the limit, the choice of the hypothesis space *does* have an effect on what is learnable when behaviorally correct learning is considered. The full potential of *BcTxt*-learners can only be exploited, if a numbering like $(W_j)_{j \in \mathbb{N}}$ – induced by a Gödel numbering – may be used as a hypothesis space.

Theorem 38 (*Baliga et al. [8]*). *There is an indexable class $\mathcal{C}$ such that the following two conditions are fulfilled.*

(1) $\mathcal{C} \in BcTxt$.
(2) $\mathcal{C}$ is not BcTxt-learnable with respect to any indexed hypothesis space.

Interestingly, we can immediately characterize the classes learnable with respect to indexed hypothesis spaces as those which are learnable in the limit.

Theorem 39. *Let $\mathcal{C}$ be an indexable class. Then the following two conditions are equivalent.*

(1) $\mathcal{C}$ is BcTxt-learnable with respect to some indexed hypothesis space.
(2) $\mathcal{C} \in LimTxt$.

Proof. Clearly, if $\mathcal{C} \in LimTxt_{\mathcal{H}}$ with respect to some indexed hypothesis space $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ then $\mathcal{C} \in BcTxt_{\mathcal{H}}$, too.

For the opposite direction, let $\mathcal{C}$ be an indexable class, $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ an indexing comprising $\mathcal{C}$, M an IIM that $BcTxt_{\mathcal{H}}$-identifies $\mathcal{C}$, and let $(w_j)_{j \in \mathbb{N}}$ be any effective enumeration of all strings in Σ^*. An IIM M' learning $\mathcal{C}$ in the limit from text with respect to $\mathcal{H}$ works according to the following instructions.

> Input: $t[n]$ for some text t of a language $L \in \mathcal{C}$ and some $n \in \mathbb{N}$
> Output: hypothesis $M'(t[n]) \in \mathbb{N}$
>
> (1) If $n = 0$, then return $M'(t[0]) = M(t[0])$.
> (2) If $n > 0$, test whether or not (i) and (ii) are fulfilled.
>   (i) $content(t[n]) \subseteq L_{M'(t[n-1])}$.
>   (ii) $\{w_z \mid z \le n, \; w_z \in L_{M'(t[n-1])}\} = \{w_z \mid z \le n, \; w_z \in L_{M(t[n])}\}$.
>   If yes, return $M'(t[n]) = M'(t[n-1])$. If no, return $M'(t[n]) = M(t[n])$.

Since M identifies L behaviorally correct from t, there is a least index m such that $L_{M(t[m])} = L_{M(t[n])} = L$ for all $n > m$. Consider the hypothesis $M'(t[m])$. First, if $L_{M'(t[m])} = L$, then $M'(t[n]) = M'(t[m])$ for all $n > m$. Second, if $L_{M'(t[m])} \ne L$, there has to be a least index $z > m$ such that (i) or (ii) will be violated. By definition, $M'(t[z]) = M(t[z])$. Since $L_{M'(t[z])} = L$, $M'(t[n]) = M'(t[z])$ for all $n > z$. Thus in both cases, M converges to a correct hypothesis for the target language L. □

On the other hand, extending the notion of hypothesis spaces by admitting more general families, such as, for instance, uniformly K-r. e. families, does not have any additional effect concerning *BcTxt*-learnability. If a class is *BcTxt*-learnable in any kind of conceivable hypothesis space, then it is also learnable with respect to $(W_j)_{j \in \mathbb{N}}$.

3.4.3. Conservative learning in the limit

Similar to the case of *LimTxt*-learning, to analyze whether or not an indexable class $\mathcal{C}$ is identifiable by a conservative IIM, it does not make a difference whether Gödel numberings or indexed families are considered as hypothesis spaces (see Lange and Zilles [62]) as communicated by Sanjay Jain.[4] Since the proof suggested by Sanjay Jain has not been published yet, we provide the details here. Note that this result is stated only for the case of indexings comprising the target class $\mathcal{C}$ as hypothesis spaces, not for indexings of $\mathcal{C}$.

Theorem 40. *Let $\mathcal{C}$ be an indexable class. Then the following statements are equivalent.*

(1) $\mathcal{C} \in$ *ConsvTxt.*
(2) $\mathcal{C}$ *is conservatively learnable in the limit with respect to an indexed hypothesis space comprising $\mathcal{C}$.*

Proof. It suffices to show that (1) implies (2). For that purpose, fix an indexable class $\mathcal{C} \in$ *ConsvTxt*. Moreover, choose an indexing $(L_j)_{j \in \mathbb{N}}$ of $\mathcal{C}$, such that for every language $L \in \mathcal{C}$ there are infinitely many $j \in \mathbb{N}$ with $L_j = L$.

Suppose M is an IIM which identifies $\mathcal{C}$ conservatively in the limit from text, using the numbering $(W_j)_{j \in \mathbb{N}}$ as a hypothesis space.

The proof proceeds as follows. First, we construct an indexed hypothesis space $\mathcal{H}' = (L'_k)_{k \in \mathbb{N}}$ comprising $\mathcal{C}$; second, we define an IIM M' which learns $\mathcal{C}$ conservatively with respect to $\mathcal{H}'$.

Definition of $\mathcal{H}' = (L'_k)_{k \in \mathbb{N}}$.
Recall that $(\sigma_z)_{z \in \mathbb{N}}$ is an effective one-one enumeration of all finite sequences of strings, such that σ_0 is the empty sequence. Define an indexing of possibly empty languages $(L'_k)_{k \in \mathbb{N}}$ by the following decision procedure:

> Input: $k \in \mathbb{N}$ with $k = \langle j, z \rangle$, $w \in \Sigma^*$
> Output: 'yes', if $w \in L'_k$; 'no', if $w \notin L'_k$

> (1) If $z = 0$, then return 'no' and stop; (* $L'_{\langle j, 0 \rangle} = \emptyset$ for all $j \in \mathbb{N}$. *)
> (2) (* Now σ_z is not empty. *)
>   If $content(\sigma_z) \not\subseteq L_j$ or $w \notin L_j$, then return 'no' and stop; (* $L'_k \subseteq L_j$. *)
> (3) (* Now $content(\sigma_z) \subseteq L_j$ and $w \in L_j$. *)
>   Dovetail the enumerations (A) and (B), until at least one of them terminates.
>
> > (A) Try to enumerate $W_{M(\sigma_z)}$, until $w \in W_{M(\sigma_z)}$ has been verified.
> > (B) Enumerate all $\tau \in SEG$, until some τ is found with $content(\tau) \subseteq L_j$ and
> > $M(\sigma_z) \neq M(\sigma_z \tau)$.
>
> If (A) terminates first, then return 'yes' and stop;
> If (B) terminates first, then return 'no' and stop. (* $L'_k \subseteq W_{M(\sigma_z)}$. *)

Now we prove three central assertions summarized in the following claim.

Claim 41. (1) $(L'_k)_{k \in \mathbb{N}}$ *is a family of recursive languages.*
(2) $L'_{\langle j, z \rangle} \subseteq L_j \cap W_{M(\sigma_z)}$ *for all $j, z \in \mathbb{N}$.*
(3) $(L'_k)_{k \in \mathbb{N}}$ *comprises $\mathcal{C}$.*

Proof of claim. *ad* (1). Note that all instructions in the decision procedure defining $(L'_k)_{k \in \mathbb{N}}$ are algorithmic. However, it remains to be verified that the procedure always terminates, i.e., that for each $k \in \mathbb{N}$ with $k = \langle j, z \rangle$ and $content(\sigma_z) \subseteq L_j$, and for all $w \in L_j$ at least one of the enumerations (A) and (B) stops.

So let $k \in \mathbb{N}$, $k = \langle j, z \rangle$, $content(\sigma_z) \subseteq L_j$, and $w \in L_j$. Note that σ_z is an initial segment of a text for L_j. Assume enumeration (B) does not terminate, i.e., there is no segment τ with $content(\tau) \subseteq L_j$ and $M(\sigma_z) \neq M(\sigma\tau)$. Therefore σ_z is a stabilizing sequence for M and L_j. Since L_j is identified by M with respect to $(W_j)_{j \in \mathbb{N}}$, this implies $W_{M(\sigma_z)} = L_j$ and thus $w \in W_{M(\sigma_z)}$. Then enumeration (A) must eventually stop. Hence at least one of the enumerations (A) and (B) stops.

---

[4] Personal Communication, 2004.

Thus L'_k is defined by a recursive decision procedure, uniformly in k, and therefore $(L'_k)_{k\in\mathbb{N}}$ is indeed a family of recursive languages.

ad (2). This assertion follows immediately from the definition of the decision procedure.

ad (3). It remains to show that $(L'_k)_{k\in\mathbb{N}}$ comprises $\mathcal{C}$. So let $L \in \mathcal{C}$. Since L is identified by M with respect to $(W_j)_{j\in\mathbb{N}}$, there is a locking sequence σ_z for M and L respecting $(W_j)_{j\in\mathbb{N}}$. Now choose some j with $L_j = L$ and fix $k \in \mathbb{N}$ such that $k = \langle j, z\rangle$. Note that $W_{M(\sigma_z)} = L_j$. By Assertion (2), $L'_k \subseteq L_j$. But the construction of L'_k also yields $L_j \subseteq L'_k$: if $w \in L_j \,(= W_{M(\sigma_z)})$, then the inclusion of w in L'_k depends on the outcome of the dove-tailing process of the enumerations (A) and (B). However, since σ_z is a stabilizing sequence for M and L_j, enumeration (B) will never terminate. So enumeration (A) will terminate first and thus $w \in L'_k$.

This implies $L'_k = L_j$ and hence $(L'_k)_{k\in\mathbb{N}}$ comprises $\mathcal{C}$, and thus Claim 41 is proved. $\quad\square$

By Claim 41, Assertions (1) through (3), $\mathcal{H}'$ is an indexed hypothesis space comprising $\mathcal{C}$.

Definition of M'.

Define an IIM M' by the following procedure.

> Input: $t[n]$ for some text t of a language $L \in \mathcal{C}$ and some $n \in \mathbb{N}$
> Output: hypothesis $M'(t[n]) \in \mathbb{N}$

> (1) If $n = 0$, then return $\langle 0, 0\rangle$ and stop; (* $L'_{M'(t[n])} = \emptyset$. *)
> (2) (* Now $n \geq 1$. *)
>   If *content*$(t[n]) \subseteq L'_{M'(t[n-1])}$, then return $M'(t[n-1])$ and stop;
>   (* M' is conservative. *)
> (3) (* Now *content*$(t[n]) \nsubseteq L'_{M'(t[n-1])}$. *)
>   If there exists some $j \leq n$, such that *content*$(t[n]) \subseteq L_j$ and
>
> $$M'(t[m]) \notin \{\langle j, z\rangle \mid \sigma_z \in \{t[0], \ldots, t[m]\}\} \text{ for all } m < n,$$
>
>   then return the value $k = \langle j, z\rangle$ with $\sigma_z = t[n]$ for the least such j and stop;
> (4) (* No new candidate is found. *)
>   Return $M'(t[n-1])$ and stop.

We show that M' learns $\mathcal{C}$ conservatively with respect to $\mathcal{H}' = (L'_k)_{k\in\mathbb{N}}$. By definition, M' is conservative with respect to $\mathcal{H}'$. Hence it suffices to prove that M' learns $\mathcal{C}$ in the limit with respect to $\mathcal{H}'$. So let $L \in \mathcal{C}$ and t a text for L.

First, we show that $L'_{M'(t[n])} \nsupseteq L$ for all $n \in \mathbb{N}$, i.e., M' avoids to output overgeneralized hypotheses when processing t. By definition, $L'_{M'(t[0])} = \emptyset$, and thus $L'_{M'(t[0])} \nsupseteq L$. Next let $n \geq 1$. Then $M'(t[n]) = k$ with $k = \langle j, z\rangle$ for some j, z, m with $m \leq n$, $\sigma_z = t[m]$, and *content*$(t[m]) \subseteq L_j$. Claim 41.(2) implies that $L'_{M'(t[n])} = L'_k \subseteq L_j \cap W_{M(\sigma_z)} \subseteq W_{M(\sigma_z)}$. Since M learns L conservatively with respect to $(W_j)_{j\in\mathbb{N}}$, we know that $W_{M(\sigma_z)} \nsupseteq L$ (see the corresponding discussion below Definition 30), and therefore $L'_{M'(t[n])} \nsupseteq L$.

Second, we show that M' converges to a correct hypothesis for L. Let n be large enough such that $t[n]$ is a stabilizing sequence for M on t, i.e., for all $m > n$, $M(t[m]) = M(t[n])$. Note that $W_{M(t[n])} = L$. Let $j \geq n$ be minimal such that $L_j = L$. (Note that j exists by choice of $(L_j)_{j\in\mathbb{N}}$, since $L_j = L$ for infinitely many $j \in \mathbb{N}$.)

We distinguish two cases.

Case (i). *There is some $m \geq n$ such that $M'(t[m]) = \langle j, z\rangle$ for $\sigma_z = t[m]$.*

Since $W_{M(\sigma_z)} = W_{M(t[m])} = W_{M(t[n])} = L_j$ and M is conservative on L_j with respect to $(W_j)_{j\in\mathbb{N}}$, we know that $M(\sigma_z) = M(\sigma_z\tau)$ for all $\tau \in SEG$ with *content*$(\tau) \subseteq L_j$. Next *content*$(t[m]) \subseteq L_j$ implies that $L'_{\langle j, z\rangle}$ is defined via Instruction (3) in the definition of $\mathcal{H}'$. Enumeration (A) will always stop first in the dove-tailing process. Thus $L'_{M'(t[m])} = L'_{\langle j, z\rangle} = L_j = L$. Since M' is conservative with respect to $(L'_k)_{k\in\mathbb{N}}$, $M'(t[m']) = M'(t[m])$ for all $m' \geq m$.

Case (ii). *There is no $m \geq n$ such that $M'(t[m]) = \langle j, z\rangle$ for $\sigma_z = t[m]$.*

This implies that the sequence of hypotheses returned by M' on t converges to $M'(t[n-1])$. The fact that the premises for Instruction (3) in the definition of $M'(t[m])$ are never fulfilled implies that *content*$(t[m]) \subseteq L'_{M'(t[n-1])}$ for all $m > n$. (Otherwise with $m \geq j$, the premises for Instruction (3) would eventually be fulfilled when computing $M'(t[m])$.) So $L \subseteq L'_{M'(t[n-1])}$. Together with $L'_{M'(t[n-1])} \nsupseteq L$ this yields $L'_{M'(t[n-1])} = L$. $\quad\square$

The indexing $(L'_k)_{k \in \mathbb{N}}$ constructed in the proof of Theorem 40 is, in general, a proper superset of the target class $\mathcal{C}$. In fact, exact indexings are not generally appropriate for conservative learning.[5]

Theorem 42 (*Lange et al. [58]*). *There is an indexable class $\mathcal{C}$ such that the following conditions are fulfilled.*

(1) $\mathcal{C} \in ConsvTxt$.
(2) $\mathcal{C}$ *is not ConsvTxt-learnable with respect to any indexing of $\mathcal{C}$.*[6]

Sketch of proof. The idea is to modify the class defined in the proof of Theorem 32. Consider the class $\mathcal{C} = \{L_k \mid k \in \mathbb{N}\} \cup \{L_{k,j} \mid k, j \in \mathbb{N}\}$, where $L_k = \{a^k b^z \mid z \in \mathbb{N}\}$ for all k and

$$L_{k,j} = \begin{cases} \{a^k b^z \mid z \leq \Phi_k(k) - j \text{ or } z \geq \Phi_k(k)\} & \text{if } j < \Phi_k(k)\downarrow, \\ L_k & \text{if } j \geq \Phi_k(k)\downarrow \text{ or } \Phi_k(k)\uparrow, \end{cases}$$

for all $k, j \in \mathbb{N}$.

One can show that $\mathcal{C}$ is an indexable class in *ConsvTxt*. But $\mathcal{C}$ is not learnable conservatively with respect to any indexing of $\mathcal{C}$. The proof is done indirectly. If $\mathcal{C}$ was *ConsvTxt*-learnable with respect to an indexing of $\mathcal{C}$, then the halting set K defined in Section 2.1 would be recursive, a contradiction. For details the reader is referred to Lange et al. [58]. $\square$

This result contrasts the cases of *LimTxt* and *BcTxt*, where class-comprising indexings, i.e., indexings strictly comprising the target class, do not yield any benefit in learning when compared to exact indexings of the target class. However, the following embedding result shows that each class in *ConsvTxt* can be embedded into a superclass in *ConsvTxt* such that the superclass is conservatively learnable with respect to some exact indexing (we call this learnable in a *class-preserving* manner). Whenever we require class-preserving conservative learning, we allow the IIM to return '?' initially on any text.

Theorem 43 (*Lange and Zeugmann [57]*). *Let $\mathcal{C}$ be an indexable class. Then the following two conditions are equivalent.*

(1) $\mathcal{C} \in ConsvTxt$.
(2) *There is some indexable class $\mathcal{C}'$, such that $\mathcal{C} \subseteq \mathcal{C}'$ and $\mathcal{C}'$ is ConsvTxt-learnable with respect to an indexing of $\mathcal{C}'$.*

This suggests that, if a class is not learnable in a class-preserving manner, then the learning problem may be posed too specifically. A more general formulation of the learning problem, represented by the superclass, may then form a well-posed learning problem in the context of class-preserving identification.

3.5. Characterizations

The relevance of the concept of telltales is finally revealed in conditions that are both necessary and sufficient for learnability—even in all of the three learning models considered so far.

Theorem 44 (*Lange and Zeugmann [55], Angluin [3], Baliga et al. [8]*). *Let $\mathcal{C}$ be an indexable class.*

(1) $\mathcal{C} \in ConsvTxt$ *iff there is an indexing $(L_j)_{j \in \mathbb{N}}$ comprising $\mathcal{C}$ which possesses a recursively generable family of telltales.*
(2) $\mathcal{C} \in LimTxt$ *iff there is an indexing $(L_j)_{j \in \mathbb{N}}$ of $\mathcal{C}$ which possesses a uniformly r.e. family of telltales.*
(3) $\mathcal{C} \in BcTxt$ *iff there is an indexing $(L_j)_{j \in \mathbb{N}}$ of $\mathcal{C}$ which possesses a family of telltales.*

Sketch of proof. We consider the second assertion, only. The proof that an indexing of $\mathcal{C}$ with a uniformly r.e. family of telltales is sufficient for *LimTxt*-learnability has already been sketched with Proposition 27. To prove necessity, assume $\mathcal{C} \in LimTxt$. Then, by Theorem 37, there is an indexing $(L_j)_{j \in \mathbb{N}}$ of $\mathcal{C}$ and an IIM M, such that M learns $\mathcal{C}$ in the limit with respect to $(L_j)_{j \in \mathbb{N}}$.

---

[5] Note that for conservative learning in exact indexings one has to allow the IIM to return a reserved symbol, e. g., '?', as initial hypotheses, because otherwise it would fail on even very simple classes.

[6] This holds even if IIMs are allowed to return '?' as initial hypotheses.

First we define an algorithm A which, given $j, n \in \mathbb{N}$, returns a sequence $\sigma_A(j, n) \subseteq L_j$, such that, for all but finitely many n, $\sigma_A(j, n)$ is a stabilizing sequence for M and L_j. Note that such a sequence must exist, because M identifies L_j in the limit from text. Algorithm A works according to the following instructions:

> Input: $\quad j, n \in \mathbb{N}$
> Output: $\quad \sigma_A(j, n) \in SEG$ with $content(\sigma_A(j, n)) \subseteq L_j$

> (1) If $n = 0$, let $\sigma_A(j, n) = \sigma_m$ and stop, where m is minimal such that $\emptyset \neq content(\sigma_m) \subseteq L_j$;
> (2) If $n > 0$, test whether there is some $z \leq n$, such that $content(\sigma_z) \subseteq L_j$ and $M(\sigma_A(j, n-1)\sigma_z) \neq M(\sigma_A(j, n-1))$;
> > (a) If yes, then let $\sigma_A(j, n) = \sigma_A(j, n-1)\sigma_z$ for the minimal such number z and stop;
> > (b) If no, then let $\sigma_A(j, n) = \sigma_A(j, n-1)$ and stop.

Informally, in step n for an index j, the candidate for a stabilizing sequence is $\sigma_A(j, n-1)$ and A tests, whether within n steps some evidence can be found which proves that $\sigma_A(j, n-1)$ is not a stabilizing sequence. If yes, a new candidate is returned; if no, $\sigma_A(j, n-1)$ is maintained as the current candidate. So, in the limit, A returns a stabilizing sequence σ for M and L_j.

Since M learns L_j in the limit, the content T_j of this stabilizing sequence σ must then be a telltale for L_j with respect to $(L_j)_{j \in \mathbb{N}}$: First, obviously T_j is a finite subset of L_j. Second, if $T_j \subseteq L_k \subseteq L_j$ for some $k \in \mathbb{N}$, then σ is a text segment for L_k and all text segments for L_k are also text segments for L_j. So if σt is a text for L_k and $\sigma t'$ is a text for L_j, then M will converge to the same hypothesis for both texts, because σ is a stabilizing sequence for M and L_j. However, as M learns L_k and L_j in the limit, we obtain $L_k = L_j$. $\quad \square$

Note that an indexing possessing a telltale family can be used as a hypothesis space in the case of *LimTxt* and *ConsvTxt*. However, in the case of *BcTxt*, the existence of an indexing possessing a telltale family is characteristic, but in general, this indexing cannot be used as a hypothesis space (see Theorem 38).

Telltales are also relevant for learning recursive functions from examples. By analyzing this framework, Wiehagen [94,95] found characterizations of learnable classes of recursive functions, which are strongly related to those in Theorem 44 (see Zilles [104] for a discussion of the relationship of both approaches).

The importance of telltales suggests that, for learning from text, the set of strings presented to the learner is of relevance rather than the order in which they appear in the text. Indeed, when considering the uniform learning strategy proposed in the proof of Proposition 27, one observes that the output on a text segment $t[n]$ depends only on the content of that segment and its *length*. Thus rearranging the strings in $t[n]$ to a different sequence $t'[n]$ with the same length and the same content forces the uniform learner to return the same hypothesis on input $t'[n]$ as it did for input $t[n]$. Therefore, *LimTxt*-learners can be normalized to *rearrangement-independent* learners, i.e., learners which, for any text segment $(t(0), \ldots, t(n))$ return the same hypothesis as for any text segment $(t'(0), \ldots, t'(n))$, where $(t'(0), \ldots, t'(n))$ is a permutation (rearrangement) of $(t(0), \ldots, t(n))$. Similarly, *ConsvTxt*-learners can be normalized to rearrangement-independent conservative learners, and *BcTxt*-learners can be normalized to rearrangement-independent *BcTxt*-learners.

Surprisingly, *LimTxt*-learners cannot, in general, avoid returning different hypotheses on text segments with equal content but different length. In other words, such learners cannot be normalized to *set-driven* learners.

Definition 45 (*Wexler and Culicover [91]*). An IIM M is said to be *set-driven* iff $M(\sigma) = M(\tau)$ for any $\sigma, \tau \in SEG$ with $content(\sigma) = content(\tau)$.

Theorem 46 (*Lange and Zeugmann [57]*). (1) *There is an indexable class $\mathcal{C} \in LimTxt$, such that $\mathcal{C}$ is not identifiable in the limit by any set-driven IIM.*
(2) *There is an indexable class $\mathcal{C} \in BcTxt$, such that $\mathcal{C}$ is not identifiable behaviorally correctly by any set-driven IIM.*

Note that the indexable class $\mathcal{C}$ defined in the proof of Theorem 32 witnesses Assertions (1) and (2) of Theorem 46. The proof given by Lange and Zeugmann considers the case of indexings as hypothesis spaces only, but can easily be adapted to the general case.

In contrast to that, set-drivenness does not restrict the learning capabilities of conservative learners.

Theorem 47 (*Lange and Zeugmann [57]*). *Let C be an indexable class. If $C \in ConsvTxt$, then there is a set-driven IIM which identifies C conservatively in the limit from text.*

Interestingly, the following counterpart to Theorem 47 is also true: for any indexable class C and any indexing $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ comprising C, if there is a set-driven IIM witnessing $C \in LimTxt$ using the hypothesis space $\mathcal{H}$, then $C \in ConsvTxt$ (see Lange and Zeugmann [57]).

Considering the characterizations in Theorem 44, the reader may have noticed, that there is a difference between the characterization of *ConsvTxt* and those of *LimTxt* and *BcTxt*. The second and third use an indexing of the target class, whereas the first one uses an indexing comprising the target class. Indeed, the characterization of *ConsvTxt* would not hold, if exact indexings of the target class were used. This is consistent with results presented in the previous subsection concerning the impact of the type of hypothesis space used in learning.

4. A case study: Learning classes of regular languages

Concerning the different aspects of learning discussed so far, i.e., the structure of suitable target classes (sufficient and characteristic conditions), different variants of learning in the limit, the impact of hypothesis spaces, etc., the previous sections are rather general in their statements. The purpose of this section is to illustrate these aspects with the example of learning languages defined by restricted regular expressions.

4.1. The non-erasing case

Let us first consider the class $RREG_+^{\Sigma}$ of all non-erasing languages defined by restricted regular expressions, where Σ is some finite alphabet.

As Proposition 11 states, this class is of finite thickness (thus also of finite elasticity), and hence belongs to *LimTxt*. Moreover, it is not hard to verify that $RREG_+^{\Sigma}$ also has recursive finite thickness. Thus we can conclude, using Theorem 15, that the class $RREG_+^{\Sigma,k}$ of all unions of up to k languages from $RREG_+^{\Sigma}$ is in *LimTxt* for any $k \in \mathbb{N}$. In addition, Theorem 33 implies the following corollary.

Corollary 48. $RREG_+^{\Sigma,k} \in ConsvTxt$ for any $k \in \mathbb{N}^+$ and any finite alphabet Σ.

4.2. The erasing case

Focusing on the class $RREG_*^{\Sigma}$ of all erasing languages defined by restricted regular expressions over some finite alphabet Σ, we observe results quite different from those in the non-erasing case. We have stated in Section 3.2, that $RREG_*^{\Sigma}$ is not learnable in the limit from text, if and only if the underlying alphabet contains at least two symbols. This illustrates an important phenomenon: the size of the alphabet Σ may be of great impact for the feasibility of learning problems! This phenomenon was also observed for the *erasing pattern languages* (see Shinohara [85] for definitions and Reidenbach [77] and Mitchell [64] for results on the impact of the alphabet size).

If Σ contains at least two symbols, we can conclude from Theorems 13 and 17, that $RREG_*^{\Sigma}$ is neither of finite thickness nor of finite elasticity. The finite thickness condition is obviously violated by any string $w \in \Sigma^*$, since it is contained in infinitely many languages in $RREG_*^{\Sigma}$ of the form $L_*(w(v)^{\times})$ for some $v \in \Sigma^*$. However, this also follows from infinite elasticity of $RREG_*^{\Sigma}$. Let $\Sigma \supseteq \{a, b\}$. To verify infinite elasticity, note that with

$$L_j = L_*((b)^{\times}((a)^{\times}(b)^{\times})^j) \text{ and } w_j = (ab)^{j+1},$$

for any $j \in \mathbb{N}$, we have defined an infinite sequence $L_0, L_1, L_2, \ldots$ of languages in $RREG_*^{\Sigma}$ and an infinite sequence $w_0, w_1, w_2, \ldots$ of strings, such that $\{w_0, \ldots, w_{n-1}\} \subseteq L_n$ and $w_n \notin L_n$ for all $n \in \mathbb{N}^+$.

This raises the question whether $RREG_*^{\Sigma}$ is of finite thickness or finite elasticity in the case that Σ is a singleton alphabet. So assume $\Sigma = \{a\}$. Obviously, finite thickness does again not hold, since any string $w \in \{a\}^*$ is contained in any of the infinitely many languages $L_*(w(a^r)^{\times})$ for prime numbers r. In contrast to that, $RREG_*^{\Sigma}$ is of finite elasticity, if the underlying alphabet contains only one symbol. Since the proof is not hard, but rather lengthy, it is omitted.

Finally, consider the telltale aspect. Since we have verified $RREG_*^{\Sigma} \in LimTxt$ for the case that Σ is a singleton alphabet, Assertion (2) of Theorem 44 implies that there must be an indexing of $RREG_*^{\Sigma}$ which possesses a uniformly

r.e. family of telltales. Angluin's [3] proof of Theorem 44, Assertion (2) provides a uniform method for deducing such a family of telltales from any arbitrary IIM learning $RREG_*^{\Sigma}$. However, we can even show that for $\Sigma = \{a\}$ there is an indexing of $RREG_*^{\Sigma}$ which possesses a recursively generable family of telltales.

Define an indexing of $RREG_*^{\Sigma}$ canonically by enumerating all restricted regular expressions r of the form

$$r = a^m (a^{k_1})^{\times} (a^{k_2})^{\times} \cdots (a^{k_s})^{\times}$$

with $s, k_1, \ldots, k_s \in \mathbb{N}$. For each such r define the set T_r by

$$T_r = \{a^m, a^{m+k_1}, a^{m+k_2}, \ldots, a^{m+k_s}\}.$$

It is not hard to verify that T_r serves as a telltale for $L_*(r)$ with respect to $RREG_*^{\Sigma}$. Assume $T_r \subseteq L_*(r') \subseteq L_*(r)$. Since a^m is the shortest string in $L_*(r)$ and $L_*(r') \subseteq L_*(r)$, a^m is the shortest string in $L_*(r')$. Thus $r' = a^m (a^{k_1'})^{\times} (a^{k_2'})^{\times} \cdots (a^{k_{s'}'})^{\times}$ for some $s', k_1', \ldots k_{s'}' \in \mathbb{N}$. Since $a^{m+k_z} \in L_*(r')$ for $1 \leq z \leq s$, each value k_z can be represented as a linear combination of the values $k_1', \ldots k_{s'}'$. Choose a string $w \in L_*(r)$, say $w = a^{m+x_1 k_1 + \cdots + x_s k_s}$ for some $x_1, \ldots, x_s \in \mathbb{N}$. Obviously, $x_1 k_1 + \cdots + x_s k_s$ can be represented as a linear combination of the values $k_1', \ldots k_{s'}'$. This implies $w \in L_*(r')$ and thus $L_*(r') = L_*(r)$. Hence T_r is as a telltale for $L_*(r)$ with respect to $RREG_*^{\Sigma}$.

In contrast, when Σ consists of at least two symbols, then $RREG_*^{\Sigma} \notin LimTxt$ and hence, with Theorem 44, Assertion (2), no indexing of $RREG_*^{\Sigma}$ possesses a uniformly r. e. family of telltales. But the literature tells us even more: Angluin [3] has shown that, for an alphabet of at least two symbols, the class $RREG_*^{\Sigma}$ cannot have any telltale family, no matter which indexing is considered. Applying Assertion (3) of Theorem 44, this yields $RREG_*^{\Sigma} \notin BcTxt$ for any alphabet of cardinality greater than 1.

Corollary 49. *Let Σ be a finite alphabet.*
(1) *If $card(\Sigma) = 1$, then $RREG_*^{\Sigma} \in ConsvTxt$.*
(2) *If $card(\Sigma) > 1$, then $RREG_*^{\Sigma} \notin BcTxt$.*

5. Other approaches to learning

5.1. Learning from good examples

When learning in the limit from text, it is obvious that for some target classes certain examples are more relevant for successful learning than others. For instance, when learning the class of erasing languages defined by restricted regular expressions of the form $a((b)^{\times}(a)^{\times})^j$ for all $j \in \mathbb{N}$, the example string a is of no use for a learner, because it is contained in all target languages. A string $a(ba)^{j+1}$ for $j \in \mathbb{N}$ bears some more information, because it allows for excluding the languages $L_*(a((b)^{\times}(a)^{\times})^z)$ for all $z \leq j$.

In general, considering the concept of telltales for a target class $\mathcal{C}$, it is reasonable to say that the collection of all strings contained in a telltale of a language L with respect to $\mathcal{C}$ somehow forms a set of *good* examples for learning. The existence of such good examples suggests an alternative scenario in which to study learning. Think of a classroom. In this setting, learning involves two parties, namely a teacher and a learner. Since the teacher's aim should be to help the other party in learning something, i.e., in identifying a target language, the teacher should not just present arbitrary examples, but good examples. The scenario intended can be sketched as follows.

- The teacher presents a set of positive examples concerning a target language L. The requirement is that this set contains a subset which forms a set of good examples for L.
  (* Note that the set of examples presented should be finite. We shall define a notion of good examples below. *).
- The learner receives the set of examples provided by the teacher. Since this set contains a set of good examples for L, the learner will return a hypothesis correctly describing L.
  (* Note that (i) the output of the learner depends only on the set of examples and not on the order in which they are presented and (ii) the learner must identify the target language even if additional, possibly irrelevant examples are presented by the teacher. *)

Note that this scenario is a scheme for a *finite* and not a limiting learning process, i.e., the learner processes all available information, returns a single hypothesis, and then stops.

This raised the question for a formal definition of a reasonable concept of "good examples" for learning. Intuitively and roughly formulated, a set of examples should be considered good for a target language, if the above scenario can be successful. Obviously, this depends on several factors, such as

- the class of target languages,
  (* Note that examples are good, if they help in distinguishing the current target language from other possible target languages. Thus, a set of examples cannot be considered good for a target language *L* simply by considering *L* alone. Rather, all members of the target class must be considered. In particular, each language in the target class must have a set of good examples. *)
- the hypothesis space,
  (* Obviously, it is not sufficient to consider the class of target languages only, when defining what good examples are. The examples should also help in distinguishing the target language from any language contained in the hypothesis space, but not in the target class. Note that, for languages outside the target class, good examples do not have to exist necessarily. *)
- the learner itself.
  (* In general, for some class $\mathcal{C}$ of target languages, it is conceivable that there are different criteria for distinguishing two languages in $\mathcal{C}$ from one another. Thus each language in $\mathcal{C}$ might have different sets of good examples, depending upon which criterion a learner focuses on. Which sets of examples are considered good hence depends on the particular learning method. *)

Based on these considerations, a formal definition of learning from good text examples was proposed by Lange *et al.* [52]. A set of positive examples for a language *L* is considered good for *L*, if a learner, having seen at least these examples will always hypothesize *L*. This idea requires a different understanding of learners: instead of IIMs considered up to now, we model the learner as a machine which processes finite sets of strings instead of text segments. To model the classroom scenario adequately, the teachers must in addition be able to compute the corresponding sets of good examples uniformly.

Definition 50 (*Jain et al. [38], Lange et al. [52]*). Let $\mathcal{C}$ be an indexable class of languages and $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ a hypothesis space for $\mathcal{C}$. The class $\mathcal{C}$ is *finitely learnable from good examples with respect to* $\mathcal{H}$ iff there exists a recursively generable family $(ex_j)_{j \in \mathbb{N}}$ and a partial-recursive learner *M* such that the following conditions are fulfilled for all $j \in \mathbb{N}$.

(1) $ex_j \subseteq L_j$,
(2) if $L_j \in \mathcal{C}$ and *A* is a finite subset of L_j with $ex_j \subseteq A$, then $M(A)$ is defined and $L_{M(A)} = L_j$.

GexFinTxt denotes the collection of all indexable classes which are finitely learnable from good examples with respect to some appropriate hypothesis space.[7]

The set ex_j for some *j* with $L_j \in \mathcal{C}$ serves as a set of good examples for L_j. Note that the requirement that *M* learns L_j also in the case a proper superset of the set of good examples is presented avoids coding tricks. Without this requirement, it might for instance be conceivable, that the set ex_j of good examples for L_j is of cardinality *j* and thus the teacher codes a proper hypothesis just in the number of examples presented.

For instance, the class $\mathcal{C}$ of all finite languages belongs to *GexFinTxt*, since a set ex_L of good examples for a finite language *L* can be defined by *L* itself. Then a standard learner always returning a hypothesis representing the content of ex_L witnesses $\mathcal{C} \in$ *GexFinTxt* with respect to a standard indexing of all finite languages.

It is not hard to prove that *GexFinTxt* $\subseteq$ *LimTxt*, however, a more detailed analysis shows that this inclusion is proper, i.e., there are indexable classes in *LimTxt*, which cannot be learned finitely from good examples.

Theorem 51 (*Jain et al. [38]*). *GexFinTxt* $\subset$ *LimTxt*.

Sketch of proof. A separating class in *LimTxt* \ *GexFinTxt* is for instance the class in *LimTxt* \ *ConsvTxt* given in the proof of Theorem 32. For further details, we refer the reader to Jain et al. [38]. □

---

[7] Note that *M* is allowed not to terminate on an input which does not contain a set of good examples for some $L \in \mathcal{C}$ or which is not a subset of some $L \in \mathcal{C}$.

In fact, if a class is learnable finitely from good examples, then the set of good examples for a language L in the target class serves as a telltale for L. Since the family of good example sets is recursively generable, there is apparently a relation to conservative inference (see Theorem 44, Assertion (1)). Moreover, for each class $\mathcal{C}$ in *ConsvTxt* there is some indexing comprising $\mathcal{C}$ which possesses a recursively generable family of telltales. All languages in the class-comprising indexing have a telltale set, whereas the definition of *GexFinTxt* does not require the existence of a set of good examples for $L \notin \mathcal{C}$. But if class-preserving learning is considered, we obtain the following nice characterization.

Theorem 52 (*Lange et al. [52]*). *Let $\mathcal{C}$ be an indexable class. Then the following two conditions are equivalent.*

(1) *There is an indexing $(L_j)_{j \in \mathbb{N}}$ of $\mathcal{C}$ such that $\mathcal{C}$ is finitely learnable from good examples with respect to $(L_j)_{j \in \mathbb{N}}$.*
(2) *There is an indexing $(L'_j)_{j \in \mathbb{N}}$ of $\mathcal{C}$ such that $\mathcal{C}$ is conservatively identifiable in the limit from text with respect to $(L'_j)_{j \in \mathbb{N}}$.*

The proof mainly uses the following characterization of class-preserving conservative identification.

Lemma 53 (*Lange et al. [52]*). *Let $\mathcal{C}$ be an indexable class of languages. $\mathcal{C}$ is identifiable conservatively in the limit from text with respect to an indexing of $\mathcal{C}$ iff there exists an indexing $(L_j)_{j \in \mathbb{N}}$ of $\mathcal{C}$ and a recursively generable family $(T_j)_{j \in \mathbb{N}}$, such that the following conditions are fulfilled for all $j, k \in \mathbb{N}$:*

(1) $T_j \subseteq L_j$.
(2) *if $T_j \subseteq L_k$ and $T_k \subseteq L_j$, then $L_j = L_k$.*

We omit the proof of this lemma. It helps to prove Theorem 52, since the telltale-like sets fulfilling the demands in the lemma can be shown to consist of good examples in the sense of Definition 50. In particular, a class-preserving *GexFinTxt*-learner as well as its suitable hypothesis space can be obtained via a uniform procedure from a corresponding class-preserving *ConsvTxt*-learner and vice versa. We omit the details.

Since not all classes in *LimTxt* are finitely learnable from good examples, one may be tempted to think that the concept of good examples as such is not appropriate for characterizing uniform methods of learning in the limit. However, the telltale characterization in Theorem 44 suggests the opposite. In fact, a modification of Definition 50 is useful for verifying that *LimTxt*-learning can definitely be interpreted as a way of learning from good examples. Here a learner again processes a finite set A of strings, but additionally receives a counter $n \in \mathbb{N}$ which allows for returning different hypotheses on the same set A, if different counters are input. The idea is to interpret n as a step counter and to require that the hypotheses returned by the learner on some set A stabilize with increasing step counters.

Definition 54 (*Jain et al. [38], Lange et al. [52]*). Let $\mathcal{C}$ be an indexable class of languages and $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ a hypothesis space for $\mathcal{C}$. The class $\mathcal{C}$ is *learnable in the limit from good examples with respect to $\mathcal{H}$* iff there exists a recursively generable family $(ex_j)_{j \in \mathbb{N}}$ and a partial-recursive learner M such that the following conditions are fulfilled for all $j \in \mathbb{N}$.

(1) $ex_j \subseteq L_j$,
(2) if $L_j \in \mathcal{C}$ and A is a finite subset of L_j with $ex_j \subseteq A$, then $M(A, n)$ is defined for all $n \in \mathbb{N}$ and there is some $k \in \mathbb{N}$, such that $L_k = L_j$ and $M(A, n) = k$ for all but finitely many n.

GexLimTxt denotes the collection of all indexable classes which are learnable in the limit from good examples with respect to some appropriate hypothesis space.

This finally yields the desired characterization of *LimTxt* in terms of learning from good examples.

Theorem 55 (*Jain et al. [38]*). *GexLimTxt = LimTxt.*

In particular, each *LimTxt*-learner can be algorithmically translated into a learner identifying the same class of languages from good examples. Thus the concept of good examples is shown to be crucial for learning.

It is worth noting that requiring class-preserving behavior severely restricts the capabilities of a learner. In fact, if a class-preserving learner witnesses that a class of languages is learnable in the limit from good examples, then that class of languages is finitely identifiable from good examples.

Theorem 56 (*Lange et al. [52]*). *Let $\mathcal{C}$ be an indexable class. Then the following conditions are equivalent.*

(1) *There is an indexing $(L_j)_{j \in \mathbb{N}}$ of $\mathcal{C}$ such that $\mathcal{C}$ is finitely learnable from good examples with respect to $(L_j)_{j \in \mathbb{N}}$.*
(2) *There is an indexing $(L'_j)_{j \in \mathbb{N}}$ of $\mathcal{C}$ such that $\mathcal{C}$ is learnable in the limit from good examples with respect to $(L'_j)_{j \in \mathbb{N}}$.*

One final note: prior to the analysis of good examples in language learning, a similar concept was introduced and studied by Freivalds, Kinber, and Wiehagen [31] for the case of learning recursive functions.

5.2. Learning from queries

One negative aspect of the model of learning in the limit is the fact that, during the learning process, one never knows whether or not the sequence of conjectures output by the learner has already converged. So one never knows whether the current hypothesis is already a correct one. Since such knowledge would be required for many applications, learning in the limit was compared to approaches of so-called finite learning. Here the learner is required to stop the learning process deliberately and then to guarantee that its final conjecture correctly describes the target concept. However, when learning from text, such a requirement is very strong and restricts the capabilities of the corresponding learners to special classes in which no pair L, L' with $L \subset L'$ exists. Thus, for finite learning, different models, such as Angluin's [6] *query learning* model, were investigated.

In the query learning model, a learner has access to a teacher that truthfully answers queries of a specified kind. In order to learn a target language L, a query learner M may ask queries of the following kind.[8]

Membership queries. Query: a string w. The answer is 'yes' if $w \in L$ and 'no' if $w \notin L$.
Restricted superset queries. Query: an index j. The answer is 'yes' if $L_j \supseteq L$ and 'no' if $L_j \not\supseteq L$.
Restricted disjointness queries. Query: an index j. The answer is 'yes' if $L_j \cap L = \emptyset$ and 'no' if $L_j \cap L \neq \emptyset$.

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 stops (see Angluin [6]). Its queries and hypotheses are interpreted with respect to an underlying hypothesis space according to Definition 3.[9] Thus, when learning $\mathcal{C}$, M is only allowed to query languages belonging to $\mathcal{H}$. More formally:

Definition 57 (*Angluin [6]*). Let $\mathcal{C}$ be an indexable class, let $L \in \mathcal{C}$, let $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ be a hypothesis space for $\mathcal{C}$, and let M be a membership query learner *[restricted superset query learner, restricted disjointness query learner]*. M *learns* L *with respect to* $\mathcal{H}$ *using membership queries [restricted superset queries, restricted disjointness queries]* iff it eventually stops and its only hypothesis, say j, represents L, i.e., $L_j = L$.

So M returns its unique and correct guess j after only finitely many queries.[10] Moreover, M learns $\mathcal{C}$ with respect to $\mathcal{H}$ using some type of queries, iff it learns every $L' \in \mathcal{C}$ with respect to $\mathcal{H}$ using queries of the specified type.

Note that this model in general requires a teacher answering undecidable questions. The term "restricted" was defined by Angluin [6], who considered also a non-restricted variant of superset queries (of disjointness queries), where with each negative reply to a query j, the learner is presented a counterexample, i.e., a string $w \in L \setminus L_j$ (a string $w \in L \cap L_j$, respectively). However, if efficiency issues are neglected and class-comprising hypothesis spaces are allowed, as in the above definition, then counterexamples for a negative reply to a restricted superset query j can easily be achieved by posing queries representing the languages $\Sigma^* \setminus \{w\}$ for all strings w in the complement of L_j, until the reply 'no' is received for the first time. Similarly, if a restricted disjointness query j was answered with 'no', then a query learner may find a corresponding counterexample by posing queries representing the languages $\{w\}$ for strings in L_j, until the answer 'no' is received for the first time.

In the query learning model one has to be aware of the fact that the hypothesis space does not only provide a scheme for representing the hypotheses, but also for representing the queries (at least in the case of restricted superset and restricted disjointness queries). Thus what we call "hypothesis space" here is in fact something like a "query and

---

[8] In the literature (see Angluin [6,7]), more types of queries such as restricted subset queries and restricted equivalence queries were analyzed, but in what follows we concentrate on the three types explained below.

[9] The use of proper class comprising hypothesis spaces was not allowed in the original definition of query learning (see Angluin [6]), but has since been studied extensively (see e.g., in Lange and Zilles [60–62]).

[10] Originally, the main focus in the study of query learning had been the efficiency of query learners in terms of the number of queries they pose. However, as in Lange and Zilles [60–62], we do not consider efficiency issues here.

hypothesis space." As in the context of learning from examples the learnability of a target class $\mathcal{C}$ may depend on the hypothesis space used. In other words, there are classes learnable with respect to any Gödel numbering, but not learnable using an indexed family as a hypothesis space. Therefore we have to use different notions, depending on which type of hypothesis space is assumed.

*MemQ*_{r.e.}, *rSupQ*_{r.e.}, and *rDisQ*_{r.e.} denote the collections of all indexable classes $\mathcal{C}$ for which there is a query learner M which learns $\mathcal{C}$ with respect to $(W_j)_{j \in \mathbb{N}}$ using membership, restricted superset, and restricted disjointness queries, respectively. Analogously, we use the subscript _{rec} instead of _{r.e.} to denote the case in which indexable classes are used as hypothesis spaces. However, for membership queries this differentiation does not have any effect, so *MemQ*_{r.e.} = *MemQ*_{rec}. Obviously, queries are in general not decidable, i.e., the teacher may be non-computable.

If a concept class is not learnable, then there may be an algorithmic barrier (caused by the phenomenon of non-computability) or an information theoretic barrier. In the latter case, non-learnability is caused by the fact that the information available to the learner is not sufficient for identifying the target. In our setting the learner is always computable but we use the possibly non-computable teacher to gain a better understanding of the information theoretic barrier. On the other hand, for understanding the information theoretic barrier in Gold-style learning non-computable learners were considered (see for instance Osherson et al. [74] and Jain et al. [40] and the references therein).

Note that, in contrast to the models of language learning from examples introduced above, learning via queries involves the aspect of finite or one-shot learning, i.e., it is concerned with learning scenarios in which learning may eventuate without mind changes. Additionally, the role of the learner in this model is an active one, whereas in learning from examples, as defined above, the learner is rather passive; thus, the two models implement quite different scenarios of interaction between a teacher and a learner. So, at first glance, these models seem to focus on very different aspects of learning and do not seem to have much in common.

Thus, questions arise as to whether there are similarities between the two models, and to whether there are aspects of learning that both models capture. A related question is whether IIMs, i.e., limit learners, can be replaced by (at least) equally powerful (one-shot) finite learners. Answering these questions requires a comparison of the capabilities of the two types of learners.

Since, in general, membership queries provide the learner with less information than that of restricted superset or restricted disjointness queries, it is not astonishing that there are classes in *ConsvTxt* which cannot be identified with membership queries, such as, for instance, the class of all finite languages. In contrast, each class in *MemQ*_{r.e.} is conservatively learnable in the limit from text. Though this is not hard to prove, it is one of the first results giving evidence of a relationship between the two models of teacher-learner interaction. Note that this proposition can actually be obtained by combining results from Nessel and Lange [70] and Lange et al. [58].

Proposition 58. *MemQ*_{rec} = *MemQ*_{r.e.} ⊂ *ConsvTxt*.

However, the more challenging question is whether there are interaction scenarios adequate for replacing limit learners with equally capable one-shot query learners. Indeed, the query learning scenarios defined above exhibit this property: Lange and Zilles [59,62] showed that the collection of indexable classes learnable conservatively is exactly the collection of indexable classes learnable with either restricted superset or restricted disjointness queries.

Theorem 59 (*Lange and Zilles [59,62]*). *ConsvTxt = rSupQ*_{rec} = *rDisQ*_{rec}.

Recall that classes in *ConsvTxt* in general cannot be learned conservatively in a class-preserving manner (see Theorem 42). A similar result holds in the context of query learning.

Theorem 60. *There is an indexable class $\mathcal{C}$, such that the following conditions are fulfilled.*

(1) $\mathcal{C} \in rSupQ_{\text{rec}}$ [$\mathcal{C} \in rDisQ_{\text{rec}}$].
(2) $\mathcal{C}$ *is not rSupQ*_{rec}*-learnable* [*rDisQ*_{rec}*-learnable*] *with respect to any indexing of $\mathcal{C}$.*

Sketch of proof. We sketch the proof for *rSupQ*-learning, only. Here the claim is witnessed by the class $\mathcal{C}$ containing the language $L_0 = \{a^z \mid z \in \mathbb{N}\} \cup \{b\}$ and all languages $L_{j+1} = \{a^z \mid z \leq j\}$ for $j \in \mathbb{N}$. This class is learnable with restricted superset queries, but only if the learner is allowed to pose a query for the language $\{a^z \mid z \in \mathbb{N}\}$ which is not contained in $\mathcal{C}$. □

However, the embedding result obtained for conservative learning in Theorem 43, stating that classes in *ConsvTxt* can always be embedded into superclasses conservatively identifiable in a class-preserving manner, cannot be transferred to the case of query learning.

Theorem 61. *There is an indexable class* $\mathcal{C} \in rSupQ_{\text{rec}}$ $[\mathcal{C} \in rDisQ_{\text{rec}}]$ *which cannot be learned with restricted superset queries* [*restricted disjointness queries*] *in any indexing of* $\mathcal{C}$.

Sketch of proof. Again we sketch the proof for *rSupQ*-learning, only. Consider the class $\mathcal{C} \in rSupQ$ defined in the proof of Theorem 60. This class is learnable with restricted superset queries. However, one can show that there is no superclass $\mathcal{C}'$ of $\mathcal{C}$, which is learnable with restricted superset queries with respect to some indexing of $\mathcal{C}'$. $\quad\square$

Theorem 59 concerns only indexed families as hypothesis spaces for query learners. As we have already mentioned, it is conceivable to permit more general hypothesis spaces in the query model, i.e., to demand an even more capable teacher. Interestingly, this relaxation helps to characterize learning in the limit in terms of query learning.

Theorem 62 (*Lange and Zilles [62]*). *LimTxt* = $rDisQ_{\text{r.e.}}$.

Comparing this result to Theorem 59, it is evident that a characterization of *LimTxt* in terms of learning with restricted superset queries is missing. Thus there remains the question of whether or not $rDisQ_{\text{r.e.}}$ equals $rSupQ_{\text{r.e.}}$. The following result answers this question to the negative.

Theorem 63 (*Lange and Zilles [62]*). $rDisQ_{\text{r.e.}} \subset rSupQ_{\text{r.e.}}$.

Together with Theorem 62, this implies *LimTxt* $\subset$ $rSupQ_{\text{r.e.}}$. Since *BcTxt* strictly comprises *LimTxt*, this immediately raises the question of how *BcTxt* and $rSupQ_{\text{r.e.}}$ are related. It has turned out that $rSupQ_{\text{r.e.}}$ is an inference type which represents a collection of language classes strictly between *LimTxt* and *BcTxt*.

Theorem 64 (*Lange and Zilles [62]*). *LimTxt* $\subset$ $rSupQ_{\text{r.e.}}$ $\subset$ *BcTxt*.

Sketch of proof. We only show how to separate *LimTxt* from $rSupQ_{\text{r.e.}}$. An indexable class in $rSupQ_{\text{r.e.}} \setminus LimTxt$ can be defined as follows.

For all $k, j \in \mathbb{N}$, let $\mathcal{C}$ contain the languages $L_k = \{a^k b^z \mid z \geq 0\}$ and

$$L_{k,j} = \begin{cases} \{a^k b^z \mid z \leq \ell\}, & \text{if } \ell \leq j \text{ is minimal such that } \varphi_k(\ell)\uparrow, \\ \{a^k b^z \mid z \leq j\} \cup \{b^{j+1} a^{y+1}\}, & \text{if } \varphi_k(x)\downarrow \text{ for all } x \leq j \text{ and } y = \max\{\Phi_k(x) \mid x \leq j\}. \end{cases}$$

It is easily seen that $\mathcal{C} \in rSupQ_{\text{r.e.}}$. The proof of $\mathcal{C} \notin LimTxt$ uses a recursion-theoretic argument. If $\mathcal{C}$ was learnable in the limit from text, then the set *Tot* would be K-recursive (see Section 2.1 for the corresponding definitions) — a contradiction. A similar approach can be used for defining a class $\mathcal{C}' \in BcTxt \setminus rSupQ_{\text{r.e.}}$. $\quad\square$

A remark on the technique used for proving Theorem 64 is worth noting. Former proofs separating *BcTxt* from *LimTxt* use diagonal constructions and thus do not yield separating language classes in a compact definition (see above).

There also exist characterizations of *BcTxt* in terms of query learning (see Lange and Zilles [62]). These characterizations use the concept of K-r. e. language families (see Section 2.1).

All in all, this discussion illustrates a trade-off concerning learning capabilities, given by a balance between the type of information given to the learner and the requirements on the number of guesses a learner may propose.[11]

---

[11] Further relations were found, when additionally K-recursive IIMs were considered as learners; the reader is referred to Lange and Zilles [62] for more information.

6. Efficiency issues

One important aspect not covered yet is the efficiency of learning. Different notions of efficiency are conceivable. For example, one could define efficiency in terms of the run-time of an IIM, or in terms of the number of examples needed before an IIM converges to its final hypothesis. However, defining an appropriate measure of efficiency for learning in the limit is a difficult problem (see Pitt [75]). Various authors have studied the efficiency of learning in terms of the *update time* needed for computing a new *single* hypothesis. However, processing all initial segments quickly is by no means a guarantee to learning efficiently (see Theorem 67).

Intuitively, when looking at applications of learning one would be interested in learners possessing the property that the overall time needed until convergence is "reasonably" bounded. The overall time needed until convergence is called the *total learning time*. Daley and Smith [25] developed general definitions for the complexity of inductive inference that essentially correspond to the total amount of computation time taken by a learner until successfully inferring the target. But if one allows the total learning time to depend on the length of all examples seen until convergence, then even a polynomially bounded total learning time does not guarantee efficient learning, since one may delay convergence until sufficiently long examples have been seen. On the other hand, if one allows the total learning time to depend on the length of the target, only, and requires that the learner has to be successful on all possible input sequences, the total learning time cannot be recursively bounded (see Pitt [75] for a more detailed discussion).

Considering query learning, in particular the number of queries posed before returning a hypothesis may be used for measuring the efficiency of a query learner. When learning from good examples, the minimum cardinality of the sets of good examples is a lower bound for the number of examples needed for learning.

Some of these aspects will be briefly addressed below, just to give an idea of the models and results typically discussed in the field of algorithmic learning theory in this context.

For illustration, we use classes of languages defined by *patterns*.

Definition 65 (*Angluin [2], Shinohara [85]*). Let Σ be a finite alphabet and $X = \{x_1, x_2, x_3, \ldots\}$ an infinite but countable set of variables such that $\Sigma \cap X = \emptyset$. Any non-empty string $\pi \in (\Sigma \cup X)^*$ is called a *pattern* over Σ. Any homomorphism $\sigma : (\Sigma \cup X)^* \to \Sigma^*$ with $\sigma(a) = a$ for all $a \in \Sigma$ is called a *pattern substitution* over Σ.[12]

(i) The non-erasing pattern language $L^{\Sigma}(\pi)$ of a pattern π is defined by
   $L^{\Sigma}(\pi) = \{\sigma(\pi) \mid \sigma$ is a pattern substitution over Σ with $\sigma(x) \neq \varepsilon$ for all $x \in X\}$.
(ii) The erasing pattern language $L_{\varepsilon}^{\Sigma}(\pi)$ of a pattern π is defined by
   $L_{\varepsilon}^{\Sigma}(\pi) = \{\sigma(\pi) \mid \sigma$ is a pattern substitution over $\Sigma\}$.

We use PAT^{Σ} and $EPAT^{\Sigma}$ to refer to the set of all non-erasing pattern languages and of all erasing pattern languages over the alphabet Σ, respectively.

Although the definitions of erasing and non-erasing pattern languages seem to differ only marginally, the telltale criterion helps to demonstrate how much the differences in their structures affect learnability: $PAT^{\Sigma} \in ConsvTxt$ holds for all alphabets Σ (see Angluin [2,3]), whereas, under certain constraints on the size of the alphabet Σ, we have $EPAT^{\Sigma} \notin BcTxt$ (see Reidenbach [77,78]). Note that Reidenbach has demonstrated these non-learnability results by showing that, in each case, there is a particular language in $EPAT^{\Sigma}$ for which there is *no* telltale with respect to $EPAT^{\Sigma}$.

The learnability of different classes of both non-erasing and erasing pattern languages with different types of queries was analyzed systematically, e.g., by Angluin [6], Erlebach et al. [29], Lange and Wiehagen [53], Nessel and Lange [70], and Lange and Zilles [59]. For instance, the class of all non-erasing pattern languages is identifiable in each of the query learning models defined above, whereas the class of all erasing pattern languages is *not* identifiable in any of them.

Concerning Gold-style identification of non-erasing pattern languages, efficient learners can be found, at least if run-time is taken as a measure for efficiency. In the following, for any pattern π and string w we use $|\pi|$ and $|w|$ to denote the length of π and w, respectively.

---

[12] Homomorphism here means that $\sigma(vw) = \sigma(v)\sigma(w)$ for all $v, w \in \Sigma^*$.

Theorem 66 (*Lange and Wiehagen [53]*). *Let Σ be a finite alphabet. There is an IIM M and a polynomial p, such that the following two conditions are fulfilled.*

(1) *M identifies PAT^Σ in the limit from text.*
(2) *For any text segment σ and any string w of length n, the number of steps required by M on input $w\sigma$ is at most $p(n)$.*

Sketch of proof. Lange and Wiehagen [53] have proposed an iterative method for identifying PAT^Σ. The desired IIM *M* is defined as follows. On input $t[0]$ the IIM *M* returns the pattern $t(0)$, i.e., a string, as its first hypothesis. Then, for any $n \in \mathbb{N}$ and on input of any text segment $t[n+1]$, *M* acts as follows.

Let $\pi_n = M(t[n])$.

(1) If $|t(n+1)| > |\pi_n|$, then set $\pi_{n+1} = \pi_n$, return π_{n+1} and stop.
(2) If $|t(n+1)| < |\pi_n|$, then set $\pi_{n+1} = t(n+1)$, return π_{n+1} and stop.
(3) If $|t(n+1)| = |\pi_n|$, then set $\pi_{n+1} = \beta_1 \ldots \beta_s$, where $s = |t(n+1)|$, return π_{n+1} and stop.
   β_j is defined as follows. Let $\pi_n = \alpha_1 \ldots \alpha_s$, let $t(n+1) = a_1 \ldots a_s$, and for $j = 1, \ldots, s$, we set

$$\beta_j = \begin{cases} \alpha_j, & \text{if } \alpha_j = a_j, \\ x_k, & \text{if } \alpha_j \neq a_j \text{ and there is some } r < j \\ & \text{with } \beta_r = x_k \text{ and } \alpha_r = \alpha_j \text{ and } a_r = a_j, \\ x_z, & \text{otherwise, where } z = \min\{k \mid x_k \notin \{\beta_1, \ldots, \beta_{j-1}\}\}. \end{cases}$$

It is not hard to verify that *M* runs in polynomial time and identifies PAT^Σ in the limit from text. As stated above, *M* is iterative, i.e., in each step only the most current string in the text as well as the previous hypothesis are needed for computing the next hypothesis. Details are omitted. $\square$

A naïve implementation of the operation used in (3) of the proof above for computing $\beta_1 \ldots \beta_s$ would require time quadratic in $|\pi_n|$. However, Rossmanith and Zeugmann [81] provided an algorithm computing $\beta_1 \ldots \beta_s$ in linear time.

The algorithm given in the proof of Theorem 66 can also be used to learn PAT^Σ finitely from good examples. Zeugmann [100] showed that $\lfloor \log_{|\Sigma|}(|\Sigma| + k - 1) \rfloor + 1$ is both an upper and lower bound for the size of a minimal set of good examples for a pattern of k variables, where $|\Sigma|$ denotes the size of the underlying alphabet. Note that this number *decreases* if the alphabet size *increases*. Thus, we have found a nice non-trivial example showing that a larger alphabet size does *facilitate* learning.

Note that, in contrast to Theorem 66, Angluin's [2] algorithm for learning PAT^Σ in the limit from text is not run-time efficient, unless P = NP. The main difference between her algorithm and the one presented above is that the latter may also output *inconsistent* hypotheses, i.e., hypotheses that contradict the known data.

Theorem 66 is a special case of a very general result stating that each class in *LimTxt* is identifiable by an IIM with a run-time polynomial in the length of the given input. Still, Theorem 66 is worth noting because of the iterative and quite intuitive strategy proposed in the proof. Moreover, the IIM can be implemented in a way that its run-time is even *linear* in the length of the first string in the given input segment, which does not hold true in the general case.

Theorem 67. *Let $\mathcal{C}$ be any indexable class and $\mathcal{H}$ a hypothesis space such that $\mathcal{C} \in LimTxt_{\mathcal{H}}$. Then there is an IIM M and a polynomial q, such that the following two conditions are fulfilled.*

(1) *M identifies $\mathcal{C}$ in the limit from text with respect to $\mathcal{H}$.*
(2) *For any text segment σ of length n, the number of steps required by M on input σ is at most $q(n)$.*

Sketch of proof. Let M' be any IIM witnessing $\mathcal{C} \in LimTxt_{\mathcal{H}}$. Now we construct the desired IIM *M* which uses M' as a subroutine.

Input: $t[n]$ for some text t of a language $L \in \mathcal{C}$ and some $n \in \mathbb{N}$
Output: hypothesis $M(t[n]) = j \in \mathbb{N}$

(1) Run M' on inputs $t[0], \ldots, t[n]$ for n steps each;
(2) If there is some $z \leq n$ such that $M'(t[z])$ is defined within n steps, then set $M(t[n]) = M'(t[z])$ for the maximal such z, output $M(t[n])$ and stop;

(3) If there is no $z \leq n$ such that $M'(t[z])$ is defined within n steps, then set $M(t[n]) = 0$, output 0 and stop.

It is easy to see that $q(n) = n^2$ and M defined above fulfill the assertions of the theorem. $\square$

An obvious disadvantage of the IIM M constructed in the proof of Theorem 67 is that it reflects the output behavior of M' on a text with delay, and thus its sequence of hypotheses in general converges later than that of M'.

6.1. Efficiency and learning from positive and negative data

In this survey, learning in the limit from both positive and negative data has been neglected. The reason for this is that learnability results for indexable classes become trivial as long as efficiency issues are neglected. To see this, first we define the notion of an *informant*.

Definition 68 (*Gold [33]*). Any total function $i : \mathbb{N} \to \Sigma^* \times \{-, +\}$ with $L = \{w \mid i(j) = (w, +) \text{ for some } j \in \mathbb{N}\}$ and $\overline{L} = \{w \mid i(j) = (w, -) \text{ for some } j \in \mathbb{N}\}$ is called an *informant* for L.

For convenience, given an informant i and some $n \in \mathbb{N}$, $i[n]$ denotes the initial segment $i(0), \ldots, i(n)$ and $content(i[n])$ denotes the set $\{i(0), \ldots, i(n)\}$. Furthermore, we set $content^+(i[n]) = \{w \mid i(j) = (w, +),\ 0 \leq j \leq n\}$ and $content^-(i[n]) = \{w \mid i(j) = (w, -),\ 0 \leq j \leq n\}$.

Definition 69 (*Barzdin [10], Blum and Blum [17]*). Let i be an informant for some language, $n \in \mathbb{N}$, and L a language. The segment $i[n]$ is called *consistent with L* iff $content^+(i[n]) \subseteq L$ and $content^-(i[n]) \subseteq \overline{L}$.

By abuse of notation we sometimes call a hypothesis $k \in \mathbb{N}$ consistent with $i[n]$, if $i[n]$ is consistent with L_k, where $(L_j)_{j \in \mathbb{N}}$ is currently considered as a hypothesis space.

Identification in the limit from informant is defined analogously to identification in the limit from text — an IIM receives successively longer finite initial segments $i[0], i[1], i[2], \ldots$ of any informant i of a target language L and is supposed to return a sequence of hypotheses converging to a correct representation for L. The resulting learning type is denoted by *LimInf*.

A fundamental result, due to Gold [33], is that $\mathcal{C} \in LimInf$ for every indexable class $\mathcal{C}$.

Theorem 70 (*Gold [33]*). *Let $\mathcal{C}$ be an indexable class and $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ an indexing of $\mathcal{C}$. Then $\mathcal{C} \in LimInf_{\mathcal{H}}$.*

Proof. The theorem is shown using the *identification by enumeration* method. Let $\mathcal{C}$ be an indexable class and $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ an indexing of $\mathcal{C}$. The desired IIM M witnessing $\mathcal{C} \in LimInf_{\mathcal{H}}$ is defined as follows.

Input: $i[n]$ for an informant i of a language $L \in \mathcal{C}$ and some $n \in \mathbb{N}$
Output: hypothesis $M(i[n]) \in \mathbb{N}$

     Compute the least j such that $i[n]$ is consistent with L_j and return $M(i[n]) = j$.

By definition, it is easy to see that M witnesses $\mathcal{C} \in LimInf_{\mathcal{H}}$. $\square$

The identification by enumeration method is in some sense universal. In the context of learning recursive functions, Wiehagen [92] has stated the thesis that a learner can be implemented using this technique for any natural Gold-style criterion, and any class of languages learnable under that criterion.

Since, for all alphabets Σ, the classes PAT^{Σ} and $EPAT^{\Sigma}$ are indexable, both are identifiable in the limit from informant. However, identification by enumeration is presumably a non-efficient method, since the problem of checking whether or not an informant segment is consistent with a (non-)erasing pattern language is NP-hard (see Ko et al. [46]).

Nevertheless, Theorem 67 has the following counterpart for *LimInf*, which can be shown by using essentially the same idea as in the proof of Theorem 67.

Theorem 71. *Let $\mathcal{C}$ be an indexable class and $\mathcal{H}$ a hypothesis space such that $\mathcal{C} \in LimInf_{\mathcal{H}}$. Then there is an IIM M and a polynomial q, such that the following two conditions are fulfilled.*

(1) *M identifies $\mathcal{C}$ in the limit from informant with respect to $\mathcal{H}$.*
(2) *For any informant segment σ of length n, the number of steps required by M on input σ is at most $q(n)$.*

Again, it is obvious that the simulation technique used does not yield any advantage. It neither increases the efficiency of the overall learning algorithm nor does it increase the learning power.

One essential property of the identification by enumeration method is missing in the method used to prove Theorem 71, namely *consistency* of the IIM. Identification by enumeration always returns hypotheses that are consistent with the informant segment seen so far, but this does not necessarily hold for the IIM in Theorem 71.

Consistency is a very intuitive requirement to impose on learning algorithms, but apparently it may affect their efficiency. We will address this aspect in more detail in the following subsection.

6.2. Efficiency and consistency

The intuitive consistency requirement suggests the following modification of the learning models originally defined by Gold.

Definition 72 (*Barzdin [10], Blum and Blum [17]*). Let $\mathcal{C}$ be an indexable class, $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ a hypothesis space for $\mathcal{C}$, and M an IIM that learns from text (from informant, respectively).

(1) M is *consistent for $\mathcal{C}$ with respect to $\mathcal{H}$* iff $t[n]$ $[i[n]]$ is consistent with $L_{M(t[n])}$ $[L_{M(i[n])}]$ for every text t for any $L \in \mathcal{C}$ [every informant i for any $L \in \mathcal{C}$] and any $n \in \mathbb{N}$.
(2) The class $\mathcal{C}$ is said to be *consistently learnable in the limit* from text [from informant] with respect to $\mathcal{H}$ iff there is an IIM M' which is consistent for $\mathcal{C}$ with respect to $\mathcal{H}$ witnessing $\mathcal{C} \in LimTxt_\mathcal{H}$ [witnessing $\mathcal{C} \in LimInf_\mathcal{H}$].

We denote the resulting learning types by *ConsTxt* and *ConsInf*. These learning models have not been addressed in the previous sections, because without focussing on efficiency, consistency is a demand that can be trivially fulfilled, as the next theorem states.

Theorem 73. *Let $\mathcal{C}$ be an indexable class and $\mathcal{H}$ any indexing comprising $\mathcal{C}$.*

(1) *If $\mathcal{C} \in LimTxt$, then $\mathcal{C} \in ConsTxt_\mathcal{H}$.*
(2) *$\mathcal{C} \in ConsInf_\mathcal{H}$.*

Sketch of proof. *ad (1)* Let $\mathcal{H}$ be any indexing comprising a given indexable class $\mathcal{C} \in LimTxt$. By Theorem 37, there is an IIM M witnessing $\mathcal{C} \in LimTxt_\mathcal{H}$. It is easily seen that, since membership is uniformly decidable in $\mathcal{H}$, M can be transformed into an IIM M' witnessing $\mathcal{C} \in ConsTxt_\mathcal{H}$. We omit the corresponding technicalities.

ad (2). It is easily seen that the identification by enumeration method can be used to show that $\mathcal{C} \in ConsInf_\mathcal{H}$ for any indexing $\mathcal{H}$ comprising $\mathcal{C}$. $\square$

In particular, each indexable class is identifiable in the limit consistently from informant, such as, for instance, the class of all non-erasing pattern languages and the class of all erasing pattern languages.

In order to study the effects of consistency demands on runtime efficiency, we consider classes of languages identifiable by learners that compute hypotheses in polynomial time. The resulting classes are denoted *Poly-ConsTxt*, *Poly-LimTxt*, *Poly-ConsInf*, and *Poly-LimInf*, respectively. That is, an indexable class $\mathcal{C}$ is in *Poly-ConsTxt* (*Poly-LimTxt*, *Poly-ConsInf*, and *Poly-LimInf*, respectively) iff there is an IIM M witnessing $\mathcal{C} \in ConsTxt$ ($\mathcal{C} \in LimTxt$, $\mathcal{C} \in ConsInf$, and $\mathcal{C} \in LimInf$, respectively) and a polynomial p such that, for every $L \in \mathcal{C}$, for every text t for L (informant i for L), and for every $n \in \mathbb{N}$, the time to compute $M(t[n])$ ($M(i[n])$) is less than or equal to $p(length(t[n]))$ ($p(length(i[n]))$).[13]

For instance, given any finite alphabet Σ, Theorems 66 and 73 state that $PAT^\Sigma \in Poly\text{-}LimTxt$ and $PAT^\Sigma \in ConsTxt$, but whether or not $PAT^\Sigma \in Poly\text{-}ConsTxt$ is still an open question. In the informant case, assuming $P \neq NP$, a negative result was shown.

Theorem 74 (*Wiehagen and Zeugmann [96]*). *Let Σ be a finite alphabet.*

(1) *$PAT^\Sigma \in Poly\text{-}LimInf_{PAT^\Sigma}$,*

---

[13] As usual, if $t[n] = (w_0, \ldots, w_n)$ ($i[n] = ((w_0, b_0), \ldots, (w_n, b_n))$), then $length(t[n])$ ($length(i[n])$) denotes the sum of the lengths of the strings $w_0, \ldots, w_n$.

(2) $PAT^{\Sigma} \notin Poly\text{-}ConsInf_{PAT^{\Sigma}}$, provided $P \neq NP$.

Note that Assertion (1) can be proved by using the algorithm of Theorem 66, thus showing that there are "intelligent" inconsistent techniques.

We finish this subsection with a quite general result showing that polynomial time restricted consistent IIMs are less powerful than inconsistent ones. This result was communicated to us by Martin Kummer.[14]

In the following, an indexable class is said to be *dense* iff for all finite sets S^+, $S^- \subseteq \Sigma^*$ with $S^+ \cap S^- = \emptyset$ there is a language $L \in \mathcal{C}$ such that $S^+ \subseteq L$ and $L \cap S^- = \emptyset$. Then we have the following result.

Theorem 75. *Let $\mathcal{C}$ be any dense indexable class. $\mathcal{C} \in Poly\text{-}ConsInf$ iff there is a $k \in \mathbb{N}$ such that $\mathcal{C} \subseteq \text{DTIME}(n^k)$.*

Sketch of proof. *Necessity*. Let $\mathcal{C}$ be any dense indexable class, $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ an indexing comprising $\mathcal{C}$, M an IIM witnessing $\mathcal{C} \in Poly\text{-}ConsInf$ with respect to $\mathcal{H}$, and $q(n) \in O(n^k)$ the corresponding polynomial.

Let $L \in \mathcal{C}$. Furthermore, let $SEG(L)$ denote the set of all finite sequences of elements from $\Sigma^* \times \{+, -\}$ that may form an initial segment of an informant for L, i.e., $content^+(\sigma) \cap content^-(\sigma) = \emptyset$, $content^+(\sigma) \subseteq L$, and $content^-(\sigma) \subseteq \overline{L}$ for all $\sigma \in SEG(L)$.

Then there is a sequence $\sigma \in SEG(L)$ such that $L_{M(\sigma)} = L$ and $M(\sigma) = M(\sigma\tau)$ for all $\tau \in SEG(L)$ (see Blum and Blum [17]). Note that σ is called a locking sequence for L, M, and $\mathcal{H}$ (see Definition 25 and above, where the corresponding notion for learning from text is considered).

Let the locking sequence σ be given. Consider the following procedure to decide membership in L.

Input: $w \in \Sigma^*$
Output: 'yes', if $w \in L$; 'no', if $w \notin L$

(1) If $w \in content^+(\sigma)$, then return 'yes'.
(2) If $w \in content^-(\sigma)$, then return 'no'.
(3) Otherwise, set $\tau = (w, +)$ and compute $M(\sigma\tau)$. (* Note that $M(\sigma\tau)$ has to be defined, since $\mathcal{C}$ is dense. *)
    If $M(\sigma\tau) = M(\sigma)$, then return 'yes'. Otherwise, return 'no'.

Let $w \notin content^+(\sigma) \cup content^-(\sigma)$. First, let $M(\sigma\tau) = M(\sigma)$. Since M is consistent for $\mathcal{C}$, $w \in L_{M(\sigma\tau)}$. By the choice of σ, we know that $L_{M(\sigma\tau)} = L_{M(\sigma)} = L$, and thus $w \in L$. Second, let $M(\sigma\tau) \neq M(\sigma)$, and suppose to the contrary that $w \in L$. Then $\tau \in SEG(L)$ and, again by the choice of σ, $M(\sigma\tau) = M(\sigma)$, contradicting $M(\sigma\tau) \neq M(\sigma)$. Hence $w \notin L$, and therefore the given procedure decides membership in L. Finally, because of the run-time bounds for M, the run-time of this procedure is bounded by $q(length(\sigma\tau))$, and therefore $L \in \text{DTIME}(n^k)$.

Sufficiency. Let $\mathcal{C} \subseteq \text{DTIME}(n^k)$ for some fixed $k \in \mathbb{N}$.

First note that there is an indexing $(L_j)_{j \in \mathbb{N}}$ comprising $\text{DTIME}(n^k)$ and thus comprising $\mathcal{C}$ such that, for any $j \in \mathbb{N}$ and any $w \in \Sigma^*$, it can be tested in polynomial time whether or not $w \in L_j$.

It is not hard to verify that the following IIM M implementing the identification by enumeration principle witnesses $\mathcal{C} \in Poly\text{-}ConsInf$. Let $(D_j)_{j \in \mathbb{N}}$ be the canonical enumeration of all finite subsets of Σ^*. For all $j \in \mathbb{N}$ set $L'_{2j} = L_j$ and $L'_{2j+1} = D_j$. The IIM M uses the hypothesis space $\mathcal{H}' = (L'_j)_{j \in \mathbb{N}}$ and works as follows.

Input: $i[n]$ for an informant i of a language $L \in \mathcal{C}$ and some $n \in \mathbb{N}$
Output: hypothesis $M(i[n]) \in \mathbb{N}$

For all $j \leq n$ test whether or not $i[n]$ is consistent with L_j. If there is an index j passing this test, fix the least one, and return $M(i[n]) = 2j$. Else compute the canonical index k of the set $D = content^+(i[n])$, and return $M(i[n]) = 2k + 1$. □

Theorem 75 allows for the following corollary for which we introduce a few notations. Let CS denote the class of all non-empty context-sensitive languages and let $\mathcal{H}_{cs}$ be any fixed canonical indexing of all context-sensitive grammars. We assume all context-sensitive grammars to be defined over a fixed terminal alphabet Σ.

---

[14] Personal Communication, 1992.

Corollary 76. (1) $P \in Poly\text{-}LimInf \setminus Poly\text{-}ConsInf$.
(2) $CS \in Poly\text{-}ConsInf_{\mathcal{H}_{cs}}$ iff $P = PSPACE$.

Sketch of proof. We show Assertion (2), only. For this purpose, let *CSL* denote the following decision problem.

    Input: $x\#w$
                   (* x is the encoding of a context-sensitive grammar G_x and $w \in \Sigma^*$. *)
    Output: 1, if $w \in L(G_x)$ and 0, otherwise.

It is well-known that *CSL* is PSPACE-complete and thus, in particular, $CSL \in PSPACE$.
Sufficiency. Assuming $P = PSPACE$, we can conclude $CSL \in P$. By the definition of P this implies that there is a $k \in \mathbb{N}$ such that *CSL* is in DTIME(n^k). This is true for all reasonable encodings of context-sensitive grammars. Thus, we can assume that the canonical encoding is used. Theorem 75 yields $CS \in Poly\text{-}ConsInf_{\mathcal{H}_{cs}}$.
Necessity. Assume $CS \in Poly\text{-}ConsInf_{\mathcal{H}_{cs}}$. Using again Theorem 75 we know that $CS \subseteq$ DTIME(n^k) for some $k \in \mathbb{N}$. Since *CS* is directly mapped to *CSL* by using the encoding given by $\mathcal{H}_{cs}$, we get $CSL \in P$. Consequently, the PSPACE-completeness of *CSL* implies $P = PSPACE$. $\square$

6.3. Other approaches to efficient learning

The definition of efficient learnability as given above is of course only one of several possible formalizations.

De la Higuera [27] proposed the following model of efficient learning from positive and negative data, taking into account also the size of a representation of the target language.

Let $\mathcal{C}$ be an indexable class and $(L_j)_{j \in \mathbb{N}}$ an indexing of $\mathcal{C}$. In de la Higuera's model [27] it is assumed that a function $\# : \mathbb{N} \to \mathbb{N}$ measuring the structural complexity of the hypotheses in $(L_j)_{j \in \mathbb{N}}$ is *a priori* fixed. (For instance, one may imagine that, for any $j \in \mathbb{N}$, $\#(j)$ equals the minimal size of a program for computing the characteristic function of the hypothesis L_j.) Note that this model applies to set-driven IIMs only (see Definition 45, which can be easily adapted for the informant case). To simplify notations, we assume in the following definition that a set-driven IIM receives as input a pair (S^+, S^-) of finite sets of strings instead of finite sequences. The strings in S^+ are considered positive examples, i.e., strings in the target language; those in S^- are negative examples, i.e., strings in the complement of the target language.

Definition 77 (*de la Higuera [27]*). Let $\mathcal{C}$ be an indexable class, $(L_j)_{j \in \mathbb{N}}$ an indexing of $\mathcal{C}$, and $\#$ a function to measure the structural complexity of the hypotheses in $(L_j)_{j \in \mathbb{N}}$. $\mathcal{C}$ is *identifiable in the limit with polynomial time and data with respect to* $(L_j)_{j \in \mathbb{N}}$ iff there exist two polynomials p and q and a set-driven IIM M such that for all $L \in \mathcal{C}$ the following conditions are fulfilled.

(1) For every (S^+, S^-) of size m with $S^+ \subseteq L$, $S^- \subseteq \overline{L}$, the hypothesis $M(S^+, S^-)$ can be computed in time $p(m)$.[15]
(2) For every $j \in \mathbb{N}$ with $L_j = L$ and $\#(j) = n$ there exists a characteristic sample (S_j^+, S_j^-) for L_j of size less than $q(n)$ such that $L_{M(S^+, S^-)} = L$ for all finite sets S^+, S^- with $S_j^+ \subseteq S^+ \subseteq L$ and $S_j^- \subseteq S^- \subseteq \overline{L}$.[16]

This definition can be adapted for the case of learning from positive data only, in which case, the above notion of characteristic sets coincides with that in Definition 21.

In this framework of learning from positive data, Clark and Eyraud [24] presented a learning algorithm for a subclass of context-free languages which they called *substitutable* languages. Roughly speaking, substitutable languages are those context-free languages L which satisfy the condition that $lur \in L$ if and only $lvr \in L$ for pairs of strings u, v. Intuitively, if u and v appear in the same context, there should be a non-terminal generating both of them.

Furthermore, Yoshinaka [99] used Definition 77, also in the context of learning from positive data, to show that the class of languages defined by very simple grammars is efficiently learnable from positive data. Note that his result for very simple grammars is related to an earlier algorithm presented by Yokomori [98]. He also discusses the problem of formally defining the notion of *efficient learning in the limit* in greater detail.

---

[15] As usual, the size of (S^+, S^-) is the sum of the lengths of all strings in S^+ and S^-.

[16] Similarly to Definition 21, (S_j^+, S_j^-) is a characteristic sample for L_j, if $L_j \subseteq L_k$ for all L_k with $S_j^+ \subseteq L_k$ and $S_j^- \subseteq \overline{L_k}$.

Yet another approach is *stochastic finite learning*. In this setting, one studies learning in the limit from randomly generated texts and analyzes the expected total learning time. For example, Reischuk and Zeugmann [80] studied the learnability of one-variable pattern languages in this setting. They showed that for almost all meaningful distributions governing how the pattern variable is replaced by a string to generate a random example of the target pattern language, their stochastic finite learner converges in an *expected constant number of rounds* and has a *total learning time* that is *linear* in the pattern length. We refer the reader to Zeugmann [101] for more information concerning stochastic finite learning.

In summary, the problem of defining *efficient learning in the limit* requires further research. The subject has recently attracted renewed attention in particular in the grammatical inference community. Besides the unsatisfactory state of the art from a theoretical point of view, this interest is caused by the challenges of high speed grammar induction for large text corpora, an area of high practical relevance (see Adriaans et al. [1]).

7. Summary and conclusions

This article has given a survey of models, methods, and central aspects considered in the context of identification of indexable classes of recursive languages in the limit.

Starting from Gold's [33] model, we have focused on sufficient conditions for learnability and the important notion of telltales, thus naturally deriving variants of Gold's model (conservative and behaviorally correct learning). These models and the corresponding telltale criteria have demonstrated how the choice of hypothesis space can be significant, even when efficiency is not a concern. Note that the discussion of conservative learning includes a previously unpublished proof (see Theorem 40).

In order to relate Gold's initial model to other natural approaches to learning, two interesting models were selected, namely learning from good examples and learning from queries. Both of them were shown to be strongly related to Gold's model of identification in the limit, particularly when an indexable class itself is used as a hypothesis space.

Finally, efficiency issues were addressed, with a particular focus on the effect that negative data have in learning, and the problem of keeping hypotheses consistent with the data during the learning process. Note that the corresponding discussion includes previously unpublished results (see Theorem 75 and Corollary 76).

The arrangement of the survey hopefully helps to pinpoint the links between the different phenomena observed; newly published illustrations using two classes of regular languages suggested by Angluin [3] as well as previously unpublished proofs are additionally chosen to yield some new insights. As was already mentioned in the introduction, this survey covers only parts of the research on learning indexable classes of recursive languages in the limit. We have tried to select topics that are relevant to other areas of computer science such as grammatical inference, machine learning, and, more generally, artificial intelligence.

As claimed in the introduction, there are different ways in which these relations are revealed. We recall them here briefly.

Methods and strategies for solving problems. The analysis of Gold's model and its variants discussed above is concerned with fundamental limitations of learning. For instance, techniques used therein reflect the following aspects.

- *Incremental processing:* Due to restricted memory capacities and costly processing of huge amounts of data in machine learning, methods of incremental learning have always been of high interest for the AI and machine learning community. As we have discussed above in the context of sufficient conditions for learnability, in particular concerning recursive finite thickness, this aspect also plays a role in inductive inference (see also Lange and Wiehagen's efficient incremental algorithm for learning the non-erasing pattern languages in the limit from text). There was considerable research on the limitations of incremental learning in this context, which we unfortunately could not discuss here. For further reading see for instance Wiehagen [92], Lange and Zeugmann [54], Lange and Grieser [51], and Jain et al. [39].
- *Searching:* The characterization theorems in Section 3.5 provide not only necessary and sufficient criteria for learnability, but also uniform learning algorithms. These more or less provide a technique for searching through a hypothesis space. The importance of such search strategies in inductive inference is for instance discussed by Wiehagen [95,94].

Environmental constraints. A major branch of studies in the area discussed above is concerned with the environmental constraints reflected in the learning model (see Mitchell [66] for a more general discussion of this topic).

- *Hypothesis spaces:* Section 3.4 addressed the impact of hypothesis spaces in learning—something which is of high interest also in grammatical inference (where mainly class-preserving learning is focused, an aspect discussed above as well) and PAC learning, but also in application oriented research in machine learning.

Efficiency. Of course, efficiency is of high relevance to any application. Thus, efficiency is a major topic in machine learning research. Section 6 provided a brief insight into the role of efficiency in inductive inference of indexed families of recursive languages. The results shown there are of a quite general character. In addition, several such results concerned target classes of significant interest, such as the class of all non-erasing pattern languages, the class of all non-empty context-sensitive languages, and the class P. At least two notions of efficiency played a role in this context.

- *Run-time efficiency:* "Run-time efficiency was considered in Section 6. Particular attention was paid to learning non-erasing pattern languages from informant, and to learning non-erasing patterns languages iteratively from text.
- *Efficiency in terms of sample size:* A related but different aspect is efficiency in terms of the number of examples required for learning. In concrete learning problems of course the number of examples actually required is also often the focus when analyzing learning algorithms in inductive inference. In this regard, again the algorithm designed by Lange and Wiehagen for learning the non-erasing pattern languages from text must be mentioned. It demonstrates how only a small number of examples can suffice for learning. In fact the learner is successful when it gets only a rather small subset of the set of the shortest strings in the target pattern language.

All in all, though many interesting and influential chapters in the history of inductive inference of recursive languages are not reflected in this survey, we hope to have succeeded in providing an accessible overview for those unfamiliar with this topic. We further hope that we have provided ideas for those already knowledgeable in this field of science.

Acknowledgments

References

[1] P.W. Adriaans, M. Trautwein, M. Vervoort, Towards high speed grammar induction on large text corpora. In: SOFSEM 2000: Theory and Practice of Informatics, 27th Conference on Current Trends in Theory and Practice of Informatics, November 25–December 2, 2000, Milovy, Czech Republic, Proceedings, 2000, pp. 173–186.

[2] D. Angluin, Finding patterns common to a set of strings, J. Comput. Syst. Sci. 21 (1) (1980) 46–62.

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

[4] D. Angluin, Inference of reversible languages, J. ACM 29 (3) (1982) 741–765.

[5] D. Angluin, Learning regular sets from queries and counterexamples, Inform. Comput. 75 (2) (1987) 87–106.

[6] D. Angluin, Queries and concept learning, Machine Learning 2 (4) (1988) 319–342.

[7] D. Angluin, Queries revisited, Theoret. Comput. Sci. 313 (2) (2004) 175–194.

[8] G.R. Baliga, J. Case, S. Jain, The synthesis of language learners, Inform. Comput. 152 (1) (1999) 16–43.

[9] J. Barzdin, On synthesizing programs given by examples, in: International Symposium on Theoretical Programming, in: Lecture Notes in Computer Science, vol. 5, Springer-Verlag, 1974, pp. 53–63.

[10] J. Barzdin, Inductive inference of automata, functions and programs, Amer. Math. Soc. Transl. (1977) 107–122.

[11] J.M. Barzdin, Prognostication of automata and functions, in: C.V. Freiman, J.E. Griffith, J.L. Rosenfeld (Eds.), Information Processing 71, Proceedings of IFIP Congress 71, Volume 1 — Foundations and Systems, August 23–28, 1971, Ljubljana, Yugoslavia, North-Holland, 1972, pp. 81–84.

[12] J.M. Barzdin, R.V. Freivald, On the prediction of general recursive functions, Soviet Math. Dokl. 13 (1972) 1224–1228.

[13] Я.М. Барздинь, Сложность и частотное решение некоторых алгоритмически неразрешимых массовых проблем, Докт. диссертация, Новосибирск, 1971.

[14] Я.М. Барздинь, Две теоремы о предельном синтезе функций, in: J.M. Barzdin (Ed.), Теория Алгоритмов и Программ, vol. I, Latvian State University, 1974, pp. 82–88.

[15] R. Berwick, The Acquisition of Syntatic Knowledge, MIT Press, Cambridge, MA, 1985.

[16] C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[17] L. Blum, M. Blum, Toward a mathematical theory of inductive inference, Inform. Control 28 (2) (1975) 125–155.

[18] M. Blum, A machine independent theory of the complexity of recursive functions, Journal of the ACM 14 (2) (1967) 322–336.

[19] J. Case, S. Jain, Synthesizing learners tolerating computable noisy data, Journal of Computer and System Sciences 62 (3) (2001) 413–441.

[20] J. Case, S. Jain, S. Lange, T. Zeugmann, Incremental concept learning for bounded data mining, Inform. Comput. 152 (1) (1999) 74–110.

[21] J. Case, S. Jain, A. Sharma, Synthesizing noise tolerant language learners, Theoret. Comput. Sci. 261 (1) (2001) 31–56.

[22] J. Case, C. Lynes, Machine inductive inference and language identification, in: Automata, Languages and Programming, 9th Colloquium, Proceedings, vol. 140, Springer-Verlag, 1982, pp. 107–115.

[23] O. Cicchello, S.C. Kremer, Inducing grammars from sparse data sets: A survey of algorithms and results, J. Machine Learning Res. 4 (2003) 603–632.

[24] A. Clark, R. Eyraud, Identification in the limit of substitutable context-free languages, in: Algorithmic Learning Theory, 16th International Conference, ALT 2005, Singapore, October 2005, Proceedings, in: Lecture Notes in Artificial Intelligence, vol. 3734, Springer, 2005, pp. 283–296.

[25] R.P. Daley, C.H. Smith, On the complexity of inductive inference, Inform. Control 69 (1–3) (1986) 12–40.

[26] D. de Jongh, M. Kanazawa, Angluin's theorem for indexed families of r.e. sets and applications, in: Proceedings of the Ninth Annual Conference on Computational Learning Theory, ACM Press, New York, NY, 1996, pp. 193–204.

[27] C. de la Higuera, Characerisitc sets for polynomial grammatical inference, Machine Learning 27 (1997) 125–138.

[28] C. de la Higuera, A bibliographical study of grammatical inference, Pattern Recognition 38 (9) (2005) 1332–1348.

[29] T. Erlebach, P. Rossmanith, H. Stadtherr, A. Steger, T. Zeugmann, Learning one-variable pattern languages very efficiently on average, in parallel, and by asking queries, Theoret. Comput. Sci. 261 (1) (2001) 119–156.

[30] J. Feldman, Some decidability results on grammatical inference and complexity, Inform. Control 20 (3) (1972) 244–262.

[31] R. Freivalds, E. Kinber, R. Wiehagen, On the power of inductive inference from good examples, Theoret. Comput. Sci. 110 (1) (1993) 131–144.

[32] E.M. Gold, Limiting recursion, Journal of Symbolic Logic 30 (1965) 28–48.

[33] E.M. Gold, Language identification in the limit, Inform. Control 10 (5) (1967) 447–474.

[34] D. Haussler, M. Kearns, N. Littlestone, M.K. Warmuth, Equivalence of models for polynomial learnability, Inform. Comput. 95 (2) (1991) 129–161.

[35] J.E. Hopcroft, J.D. Ullman, Formal Languages and their Relation to Automata, Addison-Wesley, Boston, MA, USA, 1969.

[36] S. Jain, E. Kinber, S. Lange, R. Wiehagen, T. Zeugmann, Learning languages and functions by erasing, Theoret. Comput. Sci. 241 (1–2) (2000) 143–189.

[37] S. Jain, E. Kinber, R. Wiehagen, Language learning from texts: Degrees of intrinsic complexity and their characterizations, J. Comput. Syst. Sci. 63 (3) (2001) 305–354.

[38] S. Jain, S. Lange, J. Nessel, On the learnability of recursively enumerable languages from good examples, Theoret. Comput. Sci. 261 (1) (2001) 3–29.

[39] S. Jain, S. Lange, S. Zilles, Towards a better understanding of incremental learning, in: Algorithmic Learning Theory, 17th International Conference, ALT 2006, Barcelona, Spain, October 2006, Proceedings, in: Lecture Notes in Artificial Intelligence, vol. 4264, Springer, 2006, pp. 169–183.

[40] S. Jain, D. Osherson, J.S. Royer, A. Sharma, Systems that Learn: An Introduction to Learning Theory, second edition, MIT Press, Cambridge, MA, 1999.

[41] S. Jain, A. Sharma, Generalization and specialization strategies for learning r.e. languages, Ann. Math. Artificial Intelligence 23 (1–2) (1998) 1–26.

[42] S. Kaufmann, F. Stephan, Resource bounded next value and explanatory identification: Learning automata, patterns and polynomials on-line, in: Proceedings of the Tenth Annual Conference on Computational Learning Theory, July 6th–9th, 1997, Nashville, Tennessee, ACM Press, New York, NY, 1997, pp. 263–274.

[43] M. Kearns, L. Pitt, A polynomial-time algorithm for learning k-variable pattern languages from examples, in: Proceedings of the Second Annual Workshop on Computational Learning Theory, Santa Cruz, CA, Morgan Kaufmann, San Mateo, CA, 1989, pp. 57–71.

[44] M.J. Kearns, U.V. Vazirani, An Introduction to Computational Learning Theory, The MIT Press, Cambridge, MA, 1994.

[45] E. Kinber, F. Stephan, Language learning from texts: Mindchanges, limited memory, and monotonicity, Inform. Comput. 123 (2) (1995) 224–241.

[46] K.-I. Ko, A. Marron, W.-G. Tzeng, Learning string patterns and tree patterns from examples, in: Machine Learning: Proceedings of the Seventh International Conference on Machine Learning, Morgan Kaufmann, 1990, pp. 384–391.

[47] S. Kobayashi, Approximate identification, finite elasticity and lattice structure of hypothesis spaces. Technical Report CSIM 96-04, Dept. of Compt. Sci. and Inform. Math., University of Electro-Communications, Tokyo, Japan, 1996.

[48] S. Kobayashi, T. Yokomori, Identifiability of subspaces and homomorphic images of zero-reversible languages, in: Algorithmic Learning Theory, 8th International Workshop, ALT '97, Sendai, Japan, October 1997, Proceedings, in: Lecture Notes in Artificial Intelligence, vol. 1316, Springer, Berlin, 1997, pp. 48–61.

[49] T. Koshiba, Typed pattern languages and their learnability, in: Computational Learning Theory, Second European Conference, EuroCOLT '95, Barcelona, Spain, March 1995, Proceedings, in: Lecture Notes in Artificial Intelligence, vol. 904, Springer, 1995, pp. 367–379.

[50] S. Lange, Algorithmic Learning of Recursive Languages, Mensch & Buch Verlag, Berlin, 2000.

[51] S. Lange, G. Grieser, Variants of iterative learning, Theoret. Comput. Sci. 292 (2) (2003) 359–376.

[52] S. Lange, J. Nessel, R. Wiehagen, Learning recursive languages from good examples, Ann. Math. Artificial Intelligence 23 (1/2) (1998) 27–52.

[53] S. Lange, R. Wiehagen, Polynomial-time inference of arbitrary pattern languages, New Generation Comput. 8 (4) (1991) 361–370.

[54] S. Lange, T. Zeugmann, Types of monotonic language learning and their characterization, in: Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory, ACM Press, New York, NY, 1992, pp. 377–390.

[55] S. Lange, T. Zeugmann, Language learning in dependence on the space of hypotheses, in: Proceedings of the Sixth Annual ACM Conference on Computational Learning Theory, ACM Press, New York, NY, 1993, pp. 127–136.

[56] S. Lange, T. Zeugmann, Incremental learning from positive data, J. Comput. Syst. Sci. 53 (1) (1996) 88–103.

[57] S. Lange, T. Zeugmann, Set-driven and rearrangement-independent learning of recursive languages, Math. Syst. Theory 29 (6) (1996) 599–634.

[58] S. Lange, T. Zeugmann, S. Kapur, Monotonic and dual monotonic language learning, Theoret. Comput. Sci. 155 (2) (1996) 365–410.

[59] S. Lange, S. Zilles, On the learnability of erasing pattern languages in the query model, in: Algorithmic Learning Theory, 14th International Conference, ALT 2003, Sapporo, Japan, October 2003, Proceedings, in: Lecture Notes in Artificial Intelligence, vol. 2842, Springer, 2003, pp. 129–143.

[60] S. Lange, S. Zilles, Formal language identification: Query learning vs. Gold-style learning, Inform. Process. Lett. 91 (6) (2004) 285–292.

[61] S. Lange, S. Zilles, Replacing limit learners with equally powerful one-shot query learners, in: Learning Theory: 17th Annual Conference on Learning Theory, COLT 2004, Banff, Canada, July 1–4, 2004. Proceedings, in: Lecture Notes in Artificial Intelligence, vol. 3120, Springer, 2004, pp. 155–169.

[62] S. Lange, S. Zilles, Relations between Gold-style learning and query learning, Inform. Comput. 203 (2) (2005) 211–237.

[63] N. Littlestone, Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm, Machine Learning 2 (4) (1988) 285–318.

[64] A.R. Mitchell, Learnability of a subclass of extended pattern languages, in: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, ACM Press, New York, NY, 1998, pp. 64–71.

[65] T. Mitchell, Generalization as search, Art. Int. 18 (1982) 203–226.

[66] T.M. Mitchell, Machine Learning, WCB/McGraw-Hill, Boston, MA, 1997.

[67] T. Moriyama, M. Sato, Properties of language classes with finite elasticity, IEICE Trans. Inform. Syst. E78-D (5) (1995) 532–538.

[68] T. Motoki, T. Shinohara, K. Wright, The correct definition of finite elasticity: Corrigendum to Identification of unions, in: Proceedings of the Fourth Annual Workshop on Computational Learning Theory, Morgan Kaufmann, San Mateo, CA, 1991, p. 375.

[69] B. Natarajan, Machine Learning: A Theoretical Approach, Morgan Kaufmann, San Mateo, CA, 1991.

[70] J. Nessel, S. Lange, Learning erasing pattern languages with queries, Theoret. Comput. Sci. 348 (1) (2005) 41–57.

[71] Y.K. Ng, T. Shinohara, Characteristic sets for inferring the unions of the tree pattern languages by the most fitting hypotheses, in: Grammatical Inference: Algorithms and Applications, 8th International Colloquium, ICGI 2006, Tokyo, Japan, September 20–22, 2006, Proceedings, in: Lecture Notes in Artificial Intelligence, vol. 4201, Springer, Berlin, 2006, pp. 307–319.

[72] Y.K. Ng, T. Shinohara, Developments from enquiries into the learnability of the pattern languages from positive data, Theoret. Comput. Sci. 397 (1–3) (2008) 150–165.

[73] P. Odifreddi, Classical Recursion Theory, North Holland, Amsterdam, 1989.

[74] D.N. Osherson, M. Stob, S. Weinstein, Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists, MIT Press, Cambridge, MA, 1986.

[75] L. Pitt, Inductive inference, DFAs, and computational complexity, in: Analogical and Inductive Inference, International Workshop AII '89, Reinhardsbrunn Castle, GDR, October 1989, Proceedings, in: Lecture Notes in Artificial Intelligence, vol. 397, Springer-Verlag, 1989, pp. 18–44.

[76] D. Reidenbach, A negative result on inductive inference of extended pattern languages, in: Algorithmic Learning Theory, 13th International Conference, ALT 2002, Lübeck, Germany, November 2002, Proceedings, in: Lecture Notes in Artificial Intelligence, vol. 2533, Springer, 2002, pp. 308–320.

[77] D. Reidenbach, On the learnability of E-pattern languages over small alphabets, in: Learning Theory: 17th Annual Conference on Learning Theory, COLT 2004, Banff, Canada, July 1–4, 2004. Proceedings, in: Lecture Notes in Artificial Intelligence, vol. 3120, Springer, 2004, pp. 140–154.

[78] D. Reidenbach, A non-learnable class of E-pattern languages, Theoret. Comput. Sci. 350 (1) (2006) 91–102.

[79] D. Reidenbach, Discontinuities in pattern inference, Theoret. Comput. Sci. 397 (1–3) (2008) 166–193.

[80] R. Reischuk, T. Zeugmann, An average-case optimal one-variable pattern language learner, J. Comput. Syst. Sci. 60 (2) (2000) 302–335.

[81] P. Rossmanith, T. Zeugmann, Stochastic finite learning of the pattern languages, Machine Learning 44 (1/2) (2001) 67–91.

[82] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 2002.

[83] Y. Sakakibara, Recent advances of grammatical inference, Theoret. Comput. Sci. 185 (1) (1997) 15–45.

[84] M. Sato, Y. Mukouchi, D. Zheng, Characteristic sets for unions of regular pattern languages and compactness, in: Algorithmic Learning Theory, 9th International Conference, ALT '98, Otzenhausen, Germany, October 1998, Proceedings, in: Lecture Notes in Artificial Intelligence, vol. 1501, Springer, Berlin, 1998, pp. 220–233.

[85] T. Shinohara, Polynomial time inference of extended regular pattern languages, in: RIMS Symposium on Software Science and Engineering, Kyoto, 1982, Proceedings, in: Lecture Notes in Computer Science, vol. 147, Springer-Verlag, 1982, pp. 115–127.

[86] T. Shinohara, Rich classes inferable from positive data: Length-bounded elementary formal systems, Inform. Comput. 108 (2) (1994) 175–186.

[87] F. Stephan, Noisy inference and oracles, Theoret. Comput. Sci. 185 (1) (1997) 129–157.

[88] B.A. Trakhtenbrot, Y.M. Barzdin, Finite Automata, Behavior and Synthesis, North Holland, Amsterdam, 1973.

[89] L. Valiant, A theory of the learnable, Commun. ACM 27 (11) (1984) 1134–1142.

[90] K. Wexler, The subset principle is an intensional principle, in: E. Reuland, W. Abraham (Eds.), Knowledge and Language, vol. 1, Kluwer Academic Publishers, 1993, pp. 217–239.

[91] K. Wexler, P. Culicover, Formal Principles of Language Acquisition, MIT Press, Cambridge, MA, 1980.

[92] R. Wiehagen, Limes-Erkennung rekursiver Funktionen durch spezielle Strategien, Elektronische Informationsverarbeitung und Kybernetik 12 (1–2) (1976) 93–99.

[93] R. Wiehagen, Identification of formal languages, in: Mathematical Foundations of Computer Science 1977, Proceedings, 6th Symposium, Tatranská Lomnica, September 5–9, 1977, in: Lecture Notes in Computer Science, vol. 53, Springer-Verlag, 1977, pp. 571–579.

[94] R. Wiehagen, Characterization problems in the theory of inductive inference, in: Automata, Languages and Programming, Fifth Colloquium, July 17–21, 1978, Udine, Italy, in: Lecture Notes in Compute Science, vol. 62, Springer-Verlag, 1978, pp. 494–508.

[95] R. Wiehagen, A thesis in inductive inference, in: Nonmonotonic and Inductive Logic, 1st International Workshop, Karlsruhe, Germany, December 1990, Proceedings, in: Lecture Notes in Artificial Intelligence, vol. 543, Springer-Verlag, 1990, pp. 184–207.

[96] R. Wiehagen, T. Zeugmann, Ignoring data may be the only way to learn efficiently, J. Experimental Theoret. Artif. Intell. 6 (1) (1994) 131–144.

[97] K. Wright, Identification of unions of languages drawn from an identifiable class, in: Proceedings of the Second Annual Workshop on Computational Learning Theory, Santa Cruz, CA, Morgan Kaufmann, 1989, pp. 328–333.

[98] T. Yokomori, Polynomial-time identification of very simple grammars from positive data, Theoret. Comput. Sci. 298 (1) (2003) 179–206.

[99] R. Yoshinaka, Learning efficiency of very simple grammars from positive data, in: Algorithmic Learning Theory, 18th International Conference, ALT 2007, Sendai, Japan, October 2007, Proceedings, in: Lecture Notes in Artificial Intelligence, vol. 4754, Springer, Berlin, 2007, pp. 227–241.

[100] T. Zeugmann, Lange and Wiehagen's pattern language learning algorithm: An average-case analysis with respect to its total learning time, Ann. Math. Artificial Intelligence 23 (1998) 117–145.

[101] T. Zeugmann, From learning in the limit to stochastic finite learning, Theoret. Comput. Sci. 364 (1) (2006) 77–97.

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

[103] T. Zeugmann, S. Zilles, Learning recursive functions — a survey, Theoret. Comput. Sci. 397 (1–3) (2008) 4–56.

[104] S. Zilles, Uniform Learning of Recursive Functions, in: Dissertationen zur Künstlichen Intelligenz, vol. 278, Akademische Verlagsgesellschaft, Aka GmbH, Berlin, 2003.