TCS Technical Report

Learning Indexed Families of Recursive Languages from Positive Data

by

STEFFEN LANGE, THOMAS ZEUGMANN AND SANDRA ZILLES

Division of Computer Science

Report Series A

October 30, 2007

Hokkaido University
Graduate School of
Information Science and Technology

Email:	thomas@ist.hokudai.ac.jp	Phone:	+81-011-706-7684
		Fax:	+81-011-706-7684

Contents

Learning Indexed Families of Recursive Languages from Positive Data

STEFFEN LANGE

Fachbereich Informatik

Hochschule Darmstadt

Haardtring 100

64295 Darmstadt, Germany

s.lange@fbi.h-da.de

THOMAS ZEUGMANN

Division of Computer Science

Hokkaido University

N-14, W-9

Sapporo 060-0814, Japan

thomas@ist.hokudai.ac.jp

SANDRA ZILLES

Department of Computing Science

University of Alberta

Edmonton

Alberta, Canada T6G 2E8

zilles@cs.ualberta.ca

October 30, 2007

Abstract

In the past 40 years, research on inductive inference has developed along different lines, concerning different formalizations of learning models and in particular of target concepts for learning. One common root of many of these 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 in this context focuses on so-called 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 high relevance, since their naturalness makes them interesting for many applications. 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, or artificial intelligence in general.

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

1. Introduction

Forty years ago, at a time when computer scientists had already discovered many challenges in the emerging field of artificial intelligence, an upcoming focus of research was on the aspect of learning. For instance, the question of how humans learn to speak a language bothered the AI community as well as researchers in the fields of computational linguistics and psycholinguistics.

In this context, E. Mark Gold [31] 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. [31]

1.1. *A First Model of Language Learning*

In Gold's [31] 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 in general be learned from some kind of implicit information, namely examples. Here 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 as whether or not they belong to the target language) are available.

Since human language learning is a process in which the learner revises hypothetical assumptions about the target language from time to time—never knowing at a certain state whether or not the current assumptions 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, such that at each step of the process the learner may have an intermediate assumption about the target language, which can be revised later on. In particular, the process is modeled such that it never ends, i.e., in each step an example is presented (where no assumptions are made concerning the order of examples and the multiplicity with which single examples appear in that process).

Formally, 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 assumptions 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 [31] considered representation schemes assumed to 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 correct hypothesis describing L. Similarly, grammars generating all and only the strings in L might be considered as sensible descriptions.

Now 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. Then Gold considers a learner as some algorithmic device which is given examples for the target language, step by step, and in each of the infinitely many steps it is supposed to return a hypothesis. Gold's model of *identification in the limit* declares a learner successful, if its sequence of hypotheses thus generated fulfills two requirements:

(1) it has to converge to some single hypothesis, i.e., after some step the learner returns the same hypothesis over and over again for all upcoming steps;

(2) the hypothesis it converges to must be 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 case some requirements are made concerning the sequence of examples presented. A *text* of a language L is an infinite sequence of strings that eventually contains all strings of L. Alternatively, Gold [31] considers learning from *informant*. An informant of a language L is an infinite sequence of all strings over the underlying alphabet that are labeled as whether or not they belong 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 here is to study classes of possible target languages in the sense that one learner is supposed to be able to identify all the languages in the given class from examples. If such a learner exists for a class C of languages, then C is said to be identifiable (or learnable) in the limit.

1.2. *Gold Set the Ball Rolling . . .*

Gold [31] 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 has really set a ball rolling, or one should better say several balls.

First, Trakhtenbrot and Barzdin [85] studied the learnability of finite automata intensively and obtained, in particular, positive results for the case that a complete set of labeled examples is given.

Second, his model allows not only for studying language learning in general, but also learning of recursive functions. Here, following Gold's [30, 31] papers, research on inductive inference of classes of recursive functions has started (cf., e.g., Barzdin [13, 11, 9, 10], Barzdin and Freivald [12], and Blum and Blum [17]). The reader is referred to Zeugmann and Zilles [101] for a survey on this branch of research.

Third, by focusing on general phenomena of learning in Gold's model, nice characterizations of classes of recursively enumerable languages learnable in the limit have been discussed by Wiehagen [91], thus opening the way for 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 also to relevant questions in the area of machine learning, such as the role of the information offered, noisy information, mind change complexity of learners, see for instance Case and Lynes [20], Kinber and Stephan [43], de Jongh and Kanazawa [24], Stephan [84], Jain and Sharma [39], and Jain *et al.* [34, 35]. Still today this branch of research is very active.

The fourth "ball" that has been set rolling by the studies on inductive inference as initiated by Gold [31] is the one this survey now is mainly concerned with, namely learning indexable classes of recursive languages. Here the key basics have been laid by Dana Angluin [2, 3], who suggested focusing on this type of language classes. 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. Such classes might be considered more natural than general classes of recursively enumerable languages. For instance, the classes of all regular or of all context-free languages are of that type; many computer systems deal with information represented as instances of some language contained in an indexed family.

Now Angluin [3] has studied learning of such indexed families in general, discussing a nice and 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.

On the one hand, they encouraged people to study learnability of special indexable classes of recursive languages, such as, for instance, the pattern languages, or the so-called k-reversible languages and generalizations thereof. On the other hand, 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, a third 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 approaches.

1.2.1. Learning in the Limit of Special Indexed Families

Angluin [2] initiated the study of special indexed families of recursive languages first by defining the so-called pattern languages and analyzing them in Gold's model. Roughly speaking, a pattern language is a set of strings that are matching patterns consisting of terminal symbols over some given alphabet and variable symbols. Matching 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 corresponding class of languages has been of interest, especially in the learning theory community, until today.

First of all, Shinohara [82] has extended Angluin's [2] definition of pattern languages: in her notion, the replacement of variables by strings in order to generate strings from patterns was constrained in the sense that replacing variables by empty strings was forbidden. By dropping this constraint, Shinohara defined the so-called extended pattern languages (nowadays also called 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 has bothered scientists for many many years, until Daniel Reidenbach finally was able to prove their non-learnability for alphabets of different sizes; here the reader is referred to [73, 75] as well as to [76] contained in this issue. Studying erasing pattern languages has additionally involved the analysis of interesting subclasses, such as the so-called regular erasing pattern languages as studied by Shinohara [82], the quasi-regular erasing pattern languages analyzed by Mitchell [61], or erasing pattern languages with restrictions on the number of occurring variables, see Wright [95].

Furthermore, the learnability of the pattern languages has been studied with a focus on efficient learning algorithms. For example, Kearns and Pitt [41] studied the learnability of k-variable pattern languages in the PAC model. Lange and Wiehagen [50] designed an iterative polynomial-time algorithm identifying the class of all pattern languages in the limit. Their algorithm has been analyzed with respect to its average-case behavior by Zeugmann [98]. Rossmanith and Zeugmann [78] converted this learning algorithm into a stochastic finite learner (see also [99]). We refer the reader to Ng and Shinohara [69] 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] has been of impact especially for the grammatical inference community, see for instance Pitt [72], Sakakibara [80], and Ciccello and Kremer [21].

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 Subsection 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 have started to investigate more general questions in the context of learning indexed families of recursive languages, see Zeugmann and Lange [100] and Lange [48] and the references therein. Natural constraints on learning, reflecting demands for a so-called conservative, monotonous, or incremental behavior (to name just a few), have been 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 Subsection 1.3 below.

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 [31] 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 comprising and is not meant to consider other approaches not worth discussing either.

1.2.3. Relations to PAC, On-Line, and Query Learning of Indexed Families

Neglecting work in related fields would make this survey only fragmentary—just as research would risk being fragmentary if related studies were neglected. Therefore identification in the limit, being just one of several formal models of concept learning examined in the field of algorithmic learning theory, has been compared to other models of inference.

Three prominent approaches in this context are

- Probably Approximately Correct (PAC) learning introduced by Valiant [86], see also Natarajan [66] and Kearns and Vazirani [42],

- on-line learning, see Littlestone [60], and

- learning from queries as in the model defined by Angluin [5, 6, 7].

The most explicit relations to Gold-style learning have been demonstrated for learning from queries, showing that 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 [59] for a detailed discussion; however, this aspect will be addressed later in Section 5.2.

Apparently, learning a representation of a language from examples can also be seen as a special case of learning to predict the strings in a language and thus of Littlestone's approach [60]. Hence it is not astonishing that corresponding relations between Gold-style learning and on-line prediction have been analyzed also in the

setting of learning recursive functions, see Barzdin [13] for the crucial definitions of on-line prediction in this setting as well as Zeugmann and Zilles [101] for a brief discussion. The strong relations between the two models are also immanent 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 [40].

Finally, Gold-style learning of indexed families is also related to PAC learning, albeit treated in a more subtle and implicit way. Though Gold-style and PAC learning are hardly compared directly in the literature (cf. [78, 99]), 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, as well as to Haussler *et al.* [32] and Littlestone [60] for PAC learning compared to on-line learning.

1.3. *The Relevance and Impact of the Rolling Balls*

Some of what you can read above already indicates what we would like to further illustrate with this survey, namely that the analysis of learning indexed families of recursive languages in the limit is of high relevance not only in algorithmic learning theory, but also in a broader sense.

The corresponding theory is not only concerned with many conceptualizations and phenomena known in fields such as artificial intelligence, machine learning, or grammatical inference, but has also been of impact for these fields in several ways.

The relevance for artificial intelligence (see Russell and Norvig [79]) is evident in the analysis of natural properties of learners, such as monotonicity, incremental functioning, efficiency. A further essential aspect in AI, namely that of problem solving in terms of search (and thus the use of different methods of searching a hypothesis space), see Mitchell [62], also plays an important role in learning in the limit. This will become apparent in the characterization theorems discussed below in Section 3.5.

Focusing especially on machine learning issues, algorithmic learning theory is concerned with conceptions like biasing, additional information in learning, and hypothesis spaces, like noisy data, or like 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 [63] 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

of a particular form for instance might be considered as possible hypotheses when learning regular languages of the corresponding particular form. For some background on grammatical inference, see for instance the literature survey by de la Higuera [26].

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 [31] 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 [70] and Hopcroft and Ullman [33]. From now on, a fixed finite alphabet Σ with $\{a, b\} \subseteq \Sigma$ (or $\Sigma = \{a\}$ in the special case of a singleton alphabet) is given. Then Σ^* denotes the free monoid over Σ. We refer to the elements of Σ^* as to 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 $|Z|$ 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 the 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 (cf. Odifreddi [70]). 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 $\langle\cdot,\cdot\rangle\colon \mathbb{N} \times \mathbb{N} \mapsto \mathbb{N}$ 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, *if there is a recursive function* $f\colon \mathbb{N} \times \Sigma^* \mapsto \{0, 1\}$ *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., *if there is a partial recursive function* $f\colon \mathbb{N} \times \Sigma^* \mapsto \{0, 1\}$ *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., *if there is a recursive function* $g\colon \mathbb{N} \times \Sigma^* \times \mathbb{N} \mapsto \{0, 1\}$ *such that* $L_j = \{w \in \Sigma^* \mid g(j, w, n) = 1 \text{ for all but finitely many } n\}$ *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 $f\colon \mathbb{N} \times \Sigma^* \mapsto \{0, 1\}$ 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, *if 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 to 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. [49] 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 by 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 so-called hypothesis spaces below.

Note that many interesting and surprising results have been obtained also in the setting of learning classes of recursively enumerable languages (cf., e.g., Osherson *et al.* [71], Jain *et al.* [38] 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 has been initiated by Gold [31] 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*

For defining a formal learning model one has to specify at least 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 to use them also 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 *if* $(L_j)_{j \in \mathbb{N}}$ *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 is always indexed hypothesis space.

We shall distinguish between the following types of hypothesis spaces for an indexable class $\mathcal{C}$: (i) a suitably chosen indexing for $\mathcal{C}$, (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 answer the question from what information the learner should perform its task. Gold [31] has mainly proposed two different approaches, namely 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 augmenting initial segments of such sequences. Both approaches are of significant relevance for learning theory, however, due to the space constraints, this survey mainly focuses on learning from positive examples.

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

Definition 5 (Gold [31]). *Let* L *be any non-empty language. Every total function* $t : \mathbb{N} \mapsto \Sigma^*$ *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. So, a text for a language L may enumerate the elements in any order with any number of repetitions.

---

[1] Note that, in the literature, type (i) hypothesis spaces have been 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 do no longer use the latter term exclusively for type (ii) hypothesis spaces.

We continue with the specification of the learners. Since we are dealing with algorithmic learning, our learners must be computable, henceforth called inductive inference machines, and formally defined as follows.

Definition 6 (Gold [31]). *An* inductive inference machine *(IIM)* M *is an algorithmic device that reads longer and longer initial segments* σ *of a text and outputs numbers* $\mathsf{M}(\sigma) \in \mathbb{N}$ *as its hypotheses.*

Note that, given the hypothesis space $\mathcal{H} = (\mathsf{L}_\mathsf{j})_{\mathsf{j} \in \mathbb{N}}$, 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* and requires that the sequence of hypotheses has to converge to a hypothesis correctly describing the target to be learned. Formally, a sequence $(\mathsf{j}_\mathsf{n})_{\mathsf{n} \in \mathbb{N}}$ is said to *converge* to a number j if there is a number n_0 such that $\mathsf{j}_\mathsf{n} = \mathsf{j}$ for all $\mathsf{n} \geqslant \mathsf{n}_0$.

Definition 7 (Gold [31]). *Let* $\mathcal{C}$ *be any indexable class,* $\mathcal{H} = (\mathsf{L}_\mathsf{j})_{\mathsf{j} \in \mathbb{N}}$ *a hypothesis space, and* $\mathsf{L} \in \mathcal{C}$. *An IIM* M *$Lim\,Txt_{\mathcal{H}}$-identifies* L, *if*

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

(2) $\mathsf{L} = \mathsf{L}_\mathsf{j}$.

Furthermore, M *$Lim\,Txt_{\mathcal{H}}$-identifies* $\mathcal{C}$, *if, for each* $\mathsf{L} \in \mathcal{C}$, M *$Lim\,Txt_{\mathcal{H}}$-identifies* L.

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

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 *$Lim\,Txt_{\mathcal{H}}$*-identifies L.

Since, by the definition of convergence, only finitely many strings in L have been seen by the IIM upto the (unknown) point of convergence, whenever an IIM identifies the possibly infinite language L, some form of learning must have taken place, that is the ability to generalize. 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 for a class learnable in the limit is the class of all finite languages. Given any initial segment $\mathsf{t}[\mathsf{n}]$ of a text t for a finite language L, an IIM just has to conjecture the language *content*$(\mathsf{t}[\mathsf{n}])$. As soon as n is

large enough such that $content(\mathsf{t}[\mathsf{n}]) = \mathrm{L}$, this hypothesis is correct and will never be revised by the IIM. In contrast to this, Gold [31] 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 *Lim Txt*, such as, e. g., the class of all regular languages or the class of all 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 has been done by Angluin [3], who has considered restrictions of regular expressions and thus the learnability of subclasses of regular languages. To state her results, first recall that for any $\mathsf{X}, \mathsf{Y} \subseteq \Sigma^*$ the *product* of X and Y is defined as $\mathsf{XY} = \{\mathsf{xy} \mid \mathsf{x} \in \mathsf{X}, \mathsf{y} \in \mathsf{Y}\}$. Furthermore, we define $\mathsf{X}^0 = \{\varepsilon\}$ and for all $\mathsf{i} \geqslant 0$ we set $\mathsf{X}^{\mathsf{i}+1} = \mathsf{X}^{\mathsf{i}}\mathsf{X}$. Then the *Kleene closure* of X is defined as $\mathsf{X}^* = \bigcup_{\mathsf{i} \geqslant 0} \mathsf{X}^{\mathsf{i}}$ and the *semi-group closure* of X is $\mathsf{X}^+ = \bigcup_{\mathsf{i} \geqslant 1} \mathsf{X}^{\mathsf{i}}$. We refer to the * and $^+$ operator as *Kleene star* and *Kleene plus*, respectively.

Now, the restriction of regular expressions can be defined as follows, where the special reserved symbol $^\times$ can be interpreted as Kleene plus or Kleene star, depending on which type of regular languages is considered.

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

- *For all $\mathsf{a} \in \Sigma$, a is a restricted regular expression.*

- *If p, q are restricted regular expressions over Σ, then pq and $(\mathsf{p})^\times$ 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 definition of the corresponding languages:

Definition 9. *Let Σ be a finite alphabet. The* non-erasing language $\mathrm{L}_+(\mathsf{r})$ *of a restricted regular expression r over Σ is defined inductively as follows:*

- *For all $\mathsf{a} \in \Sigma$, $\mathrm{L}_+(\mathsf{a}) = \{\mathsf{a}\}$.*

- *For all restricted regular expressions p and q, $\mathrm{L}_+(\mathsf{pq}) = \mathrm{L}_+(\mathsf{p})\mathrm{L}_+(\mathsf{q})$.*

- *For all restricted regular expressions p, $\mathrm{L}_+((\mathsf{p})^\times) = \mathrm{L}_+(\mathsf{p})^+$.*

Let RREG_+ *denote the class of all non-erasing languages of restricted regular expressions.*

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, $L_*(pq) = L_*(p)L_*(q)$.*

- *For all restricted regular expressions p, $L_*((p)^\times) = L_*(p)^*$.*

Let RREG_* denote the class of all erasing languages of restricted regular expressions.

Since one can effectively enumerate all restricted regular expressions, both RREG_+ and RREG_* are indexable classes and both are subclasses of the class of all regular languages. Next, we ask whether or not RREG_+ and RREG_*, respectively, are in *LimTxt*.

Angluin [3] has shown that the class RREG_+ is learnable in the limit from text—in contrast to its superclass of all regular languages. One method to prove this is based on the following fact.

Proposition 1 (Angluin [3]). *Let Σ be a finite alphabet and $w \in \Sigma^*$ any string. Then there are only finitely many languages $L \in \mathrm{RREG}_+$ such that $w \in L$.*

Now an IIM learning RREG_+ in the limit from text can use any fixed one-one indexing $(L_j)_{j \in \mathbb{N}}$ of RREG_+ 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 $j \leqslant n$ 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 $j \leqslant n$ and $content(t[n]) \subseteq L_j$. Since, by Proposition 1, 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_+ containing the target language. Obviously, this intersection must equal the target language. Note that this proof works independently of the size of the underlying alphabet Σ. Moreover, this is not the only successful method, as we shall see in Section 3.3.1.

In contrast to that, Angluin has 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.

As has been verified by Angluin [3], the class RREG_* 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_*. We will discuss this criterion later in Theorem 24.

However, if Σ is a singleton set, then $\mathrm{RREG}_* \in LimTxt$ can be verified as follows:

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

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

with $s, k_1, \ldots, k_s \in \mathbb{N}$, such that $k_1 < k_2 < \ldots < k_s$ 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) $L_*(((p)^\times)^\times) = L_*((p)^\times)$. Basing on this property, an IIM for RREG_* may work according to the following instructions.

Input: $t[n]$ for some text t of a language over $\{a\}$ 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 $w \in \textit{content}(t[n]) \setminus L_*(p)$ do {
   (a) Let $w \in \textit{content}(t[n])$ 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 characteristic 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 11 (Angluin [3]). *A class $\mathcal{C}$ is of* finite thickness, *if 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 $RREG_+$. The finite thickness property of $RREG_+$ is essentially what Proposition 1 states.

Theorem 2 (Angluin [3]). *Let $\mathcal{C}$ be an indexable class. If $\mathcal{C}$ is of finite thickness, then $\mathcal{C} \in Lim\,Txt$.*

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

There are several methods for IIMs exploiting the finite thickness property of the target class. The one used for the proof of $RREG_+ \in Lim\,Txt$ 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 $L'_k = L_{j_1} \cap \ldots \cap L_{j_z}$, 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 $D = \{j \mid j \leqslant n,\ content(t[n]) \subseteq L_j\}$.
> Return the canonical index k of D.

Informally, the method is used here to consider a set of possible hypotheses in each learning step and to output a hybrid hypothesis which is constructed from all of these. Since $\mathcal{C}$ has finite thickness and $\mathcal{C}$ 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 covers indices for all languages in the target class containing w.

Definition 12 (Koshiba [47], Lange and Zeugmann [53]). *An indexable class $\mathcal{C}$ has* recursive finite thickness, *if $\mathcal{C}$ is of finite thickness and there is an indexing $(L_j)_{j \in \mathbb{N}}$ of $\mathcal{C}$ and an algorithm which, on 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 $Lim\,Txt$ which demands that an IIM has enough memory capacities to process text segments of unbounded length.

Now an incremental method for learning a class $\mathcal{C}$ which has 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 $L'_k = L_{j_1} \cap \ldots \cap L_{j_z}$, 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 has been studied by Wiehagen [90] in the context of inferring recursive functions. Lange and Zeugmann [51] have transferred Wiehagen's model to the case of learning formal languages. We refer the reader to Lange and Zeugmann [53] and Lange [48] 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 $\mathcal{C}$. Here, for any $k \in \mathbb{N}^+$, we use the notion $\mathcal{C}^k$ for the class of all unions of up to k languages from $\mathcal{C}$, i.e.,

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

Theorem 3 (Case *et al.* [19], Lange [48]). *Let $\mathcal{C}$ be an indexable class and $k \in \mathbb{N}^+$. If $\mathcal{C}$ has recursive finite thickness, then $\mathcal{C}^k \in \mathit{Lim\,Txt}$.*

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. Note that in general a hypothesis space different from $(L_j)_{j \in \mathbb{N}}$ is used. We require 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 $\mathcal{C}$. More 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 $L'_j = \cap_{z \in D_j}(\cup_{r \in D_z} L_r)$.

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. Now, for each $m \in D$, distinguish the following cases:

(a) If $t(n+1) \in \cup_{r \in D_m} L_r$, then mark the index m "alive."

(b) If $|D_m| = k$, kill the index m.

(c) If $|D_m| < k$, kill the index m, too. 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 [31], no such class is in *Lim Txt*.

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

Definition 13 (Wright [95], Motoki *et al.* [65]). *A class $\mathcal{C}$ is of* infinite elasticity, *if there is an infinite sequence w_0, w_1, ... of strings and an infinite sequence L_1, L_2, ... of languages in $\mathcal{C}$ such that 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, *if $\mathcal{C}$ is not of infinite elasticity.*[2]

Theorem 4 (Wright [95]). *Let $\mathcal{C}$ be an indexable class. If $\mathcal{C}$ is of finite elasticity, then $\mathcal{C} \in Lim\,Txt$.*

Again, this criterion is sufficient, but not necessary for learnability in the limit from text. The class of all finite languages is in *Lim Txt* but is of infinite elasticity; this property is straightforward. 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 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.

Theorem 5 (Wright [95]). *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.

---

[2]The concept of finite elasticity was introduced by Wright [95], but the original definition was corrected later on by Motoki, Shinohara, and Wright [65].

Corollary 6 (Wright [95]). *Let $\mathcal{C}$ be an indexable class and $k \in \mathbb{N}^+$. If $\mathcal{C}$ is of finite elasticity, then $\mathcal{C}^k \in \text{Lim Txt}$.*

In particular, since each class of finite thickness is also of finite elasticity, we obtain a stronger result than Theorem 3.

Corollary 7. *Let $\mathcal{C}$ be an indexable class and $k \in \mathbb{N}^+$. If $\mathcal{C}$ is of finite thickness, then $\mathcal{C}^k \in \text{Lim Txt}$.*

Note that finite elasticity has also been used to show the learnability of several concept classes defined by using elementary formal systems (cf., e.g., Shinohara [83], Moriyama and Sato [64], 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 [46], Sato *et al.* [81] and Ng and Shinohara [68], as well as when investigating their polynomial-time learnability, see for example de la Higuera [25].

Definition 14 (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 $(L_j)_{j \in \mathbb{N}}$, if for all $j, k \in \mathbb{N}$:*

(1) $S_j \subseteq L_j$.

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

S_j is then called a characteristic set *for L_j.*

Theorem 8 (Kobayashi [45]). *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 $\mathcal{C} \in \text{Lim Txt}$.*

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^*. Now 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 $\mathcal{C}' = \{j \mid j \leqslant n,\ content(t[n]) \subseteq L_j\}$.
> 2. If there is an index $j \in \mathcal{C}'$ such that, for all $k < j$, Condition (i) is fulfilled, then fix the least such j and return $M(t[n]) = j$. Otherwise, return $M(t[n]) = 0$.
>   (i) $\{w_z \mid z \leqslant n,\ w_z \in L_k\} \setminus \{w_z \mid z \leqslant n,\ w_z \in L_j\} \neq \emptyset$.

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 $j \leqslant n$, $S_j \subseteq content(t[n])$, and $\{w_z \mid z \leqslant n,\ w_z \in L_k\} \setminus \{w_z \mid z \leqslant n,\ w_z \in L_j\} \neq \emptyset$ for all $k < j$ with $S_j \subseteq L_k$. Consequently, $M(t[m]) = j$ for all $m \geqslant n$, and therefore M learns L as required. ∎

The criterion given in Theorem 8 is sufficient, but *not* necessary for learnability in the limit from text, as the following example demonstrates. Let $\Sigma = \{a\}$, let $L_j = \{a\}^+ \setminus \{a^{j+1}\}$, and let $\mathcal{C} = \{L_j \mid j \in \mathbb{N}\}$. It is well-known that $\mathcal{C} \in \mathit{Lim\,Txt}$. 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 9. *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 $\{w_0', \ldots, w_{n-1}'\} \subseteq L_k$ verify 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 $n+1$.

Now 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. Now 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. ∎

Next we ask whether or not the converse of Theorem 9 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 10. *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. ∎

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. Paying some more attention to this fact leads to a more subtle observation: 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*—has been verified by Blum and Blum [17].

Definition 15 (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, if $M(\sigma\tau) = M(\sigma)$ for all $\tau \in SEG$ with content$(\tau) \subseteq L$.*

(2) *σ is called a* locking sequence *for M, L, and $\mathcal{H}$, if $L_{M(\sigma)} = L$ 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 $\mathcal{H}$.

Now these stabilizing sequences are important, since they seem to bear enough information for the learner to stick to the right conjecture. 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. That means, for each language L learned by an IIM M there must be special subset $T \subseteq L$ which—presented to the M in the form of some text segment—bears the relevant information. In particular, this subset T must be finite and it must not be contained in any proper subset L' of L which is learned by M, too. 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, has been found and analyzed by Angluin [3].

Definition 16 (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 $(L_j)_{j \in \mathbb{N}}$, if 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.

Now one part of Angluin's analysis yields: if $\mathcal{C} \in \mathit{Lim\,Txt}$ 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 has turned out, it is sufficient if there is 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 11 (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 $\mathcal{C} \in \mathit{Lim\,Txt}$.*

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 $j \in \mathbb{N}$. Now 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 and some $n \in \mathbb{N}$
Output: hypothesis $M(t[n]) \in \mathbb{N}$

(1) For all $s \leqslant n$, let $T_s[n]$ be the set of all strings ω_z such that $z \leqslant n$, $f(s, \omega_z)$ 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 $s \leqslant n$ such that $T_s[n] \subseteq \mathit{content}(t[n]) \subseteq L_s$, then return $M(t[n]) = 0$ and stop;

(3) If there is some $s \leqslant n$ such that $T_s[n] \subseteq \mathit{content}(t[n]) \subseteq L_s$, then return $M(t[n]) = s$ for the least such s and stop;

Informally, M looks for the minimal hypothesis for which the current text segment is consistent and comprises the currently known part of its telltale.

Now 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 $\mathcal{C} \in LimTxt$. ∎

However, what happened if the telltales had a simpler algorithmic complexity, such that it was really decidable how many strings belonged to a telltale? This would imply there was a procedure enumerating (uniformly in j) all strings in the telltale T_j and *stopping* afterwards. In this case, the telltale family would be called *recursively generable*.

Definition 17 (Angluin [3]). *A family $(T_j)_{j \in \mathbb{N}}$ of finite languages is* recursively generable*, if there is a recursive function that, given $j \in \mathbb{N}$, enumerates all elements of T_j and stops.*

Now a justified question would be whether such an algorithmic structure of telltales would have any consequence for learning methods. As has been shown by Lange and Zeugmann [52], recursively generable telltales allow for learning a target class *conservatively*, i.e., with only justified mind changes in the sense that a learner is not allowed to change its conjecture as long as its current hypothesis is consistent with the data seen in the text. Here consistency is defined as follows.

Definition 18 (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, if $content(t[n]) \subseteq L$. Otherwise $t[n]$ is said to be* inconsistent *with L.*

Conservative learning essentially demands always to stick to consistent hypotheses. Note that this quite natural demand has not been explicated in the definition of *LimTxt*. Demanding a learner to be conservative also brings us to another important problem, i.e., to avoid or to detect *overgeneralizations*. Here by overgeneralization we mean that the learner may output a hypothesis that is a proper superset of the target language. Clearly, an overgeneralized hypothesis cannot be detected from text. Thus one may be tempted to avoid overgeneralized hypotheses at all.

Several authors proposed the so-called *subset principle* to handle the so-called sub-set problem (cf., e.g., Berwick [15], Wexler [88]). Informally, the subset principle requires the learner to guess the "least" language from the hypothesis space with respect to set inclusion that fits with the data the learner has seen so far. But this is easier said than done as the proof of Proposition 11 shows. Here the learner always chooses the "least" 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. Note that such an hypothesis may be abandoned later if the corresponding telltale has not yet been completely enumerated.

Clearly, if the telltale family is recursively generable, then this option does not exist and a mind change must be forced due to a detected inconsistency. Conservative learners formalize this requirement.

Definition 19 (Angluin [3], Lange and Zeugmann [52]). *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 $\mathcal{H}$, if 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 *Consv Txt*.

Now, if a conservative IIM M learns an indexable class $\mathcal{C}$ with respect to $\mathcal{H}$, then it is easy to see that M can never overgeneralize. That is for every $L \in \mathcal{C}$ and every text t for L and any $n \in \mathbb{N}$, we always have $L_{M(t[n])} \not\supset L$.

The relation to recursively generable telltale families can then be stated as follows.

Proposition 12 (Lange and Zeugmann [52]). *Let $\mathcal{C}$ be an indexable class. If there is an indexing comprising $\mathcal{C}$, which possesses a recursively generable family of telltales, then $\mathcal{C} \in Consv\,Txt$.*

Though it seems quite natural to demand conservativeness, this really restricts the capabilities of IIMs. As Lange and Zeugmann [52] have shown, there is an indexable class in *Lim Txt*, 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 [100].

Theorem 13 (Lange and Zeugmann [52]). *Consv Txt $\subset$ Lim Txt.*

Sketch of proof. Let φ be our fixed Gödel numbering and let Φ be the associated Blum complexity measure (cf. 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

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

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

It is not hard to see that $\mathcal{C}$ is an indexable class and that $\mathcal{C} \in Lim\,Txt$. Showing $\mathcal{C} \notin Consv\,Txt$ is done indirectly. Suppose $\mathcal{C} \in Consv\,Txt$, 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 [52]. ∎

Note that the class $\mathcal{C}$ used in the proof of Theorem 13 is even of finite thickness, so finite thickness alone is not sufficient for conservative learnability. Yet recursive finite thickness is a sufficient condition:

Theorem 14 (Case *et al.* [19], Lange [48]). *Let $\mathcal{C}$ be an indexable class. If $\mathcal{C}$ has recursive finite thickness, then*

(1) $\mathcal{C} \in Consv\,Txt$, and

(2) $\mathcal{C}^{k} \in Consv\,Txt$ *for any* $k \in \mathbb{N}^{+}$.

Comparing Proposition 11 to Proposition 12, we see that the difference between *Lim Txt* and *Consv Txt* is completely expressed by the algorithmic complexity of the corresponding telltale families, i.e., uniformly r.e. versus recursively generable. In contrast to that, it is reasonable to analyze the learnability properties of indexable classes which possess telltale families without any additional requirements concerning the algorithmic complexity of these telltale families. Interestingly, this yields a connection to a further model of inductive inference, so-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 20 (Feldman [28], Barzdin [11, 14], Case and Lynes [20]). *Let* $\mathcal{H} = (L_j)_{j\in\mathbb{N}}$ *be any hypothesis space,* M *an IIM,* L *a recursive language, and let* t *be a text for* L.

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

(2) M *identifies* L *behaviorally correctly from text with respect to* $\mathcal{H}$, *if it identifies* L *behaviorally correctly from* t *for every text* t *for* L.

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

In the following, we use the notion *Bc Txt* 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 *Bc*-identifies $\mathcal{C}$ from text with respect to $\mathcal{H}$.

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

Proposition 15 (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* $\mathcal{C} \in Bc\,Txt$.

Giving up the requirement of syntactic convergence of the sequence of hypotheses as in *Lim Txt*, and weakening the constraints to semantic convergence, yields an add-on 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 15 in combination with a result proved by Angluin [3] (see Baliga *et al.* [8] for a discussion).

Theorem 16. *Lim Txt* $\subset$ *Bc Txt*.

3.4. *The Impact of Hypothesis Spaces*

Below Definition 3, three different types of hypothesis spaces have been 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, the converse relations do not trivially hold. 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 *Lim Txt*-model, the essential observation is that the type of hypothesis space does not matter at all as pointed out by Gold [31].

Theorem 17. *Let $\mathcal{C}$ be an indexable class. Then the following statements are equivalent.*

(1) $\mathcal{C} \in Lim\,Txt$.

(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 have any effect on the suitability of exact indexings for learning a class in the limit from positive examples. Moreover, each of the infinitely many exact indexings for the class is suitable. 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 17 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, in the case of behaviorally correct learning, the choice of hypothesis spaces in fact influences the learnability results. The full potential of $Bc\,Txt$-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 18 (Baliga *et al.* [8]). *There is an indexable class $\mathcal{C}$ such that the following two conditions are fulfilled.*

(1) $\mathcal{C} \in Bc\,Txt$.

(2) $\mathcal{C}$ *is not $Bc\,Txt$-learnable with respect to any indexed hypothesis space.*

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

Theorem 19. *Let $\mathcal{C}$ be an indexable class. Then the following two conditions are equivalent.*

(1) $\mathcal{C}$ *is Bc Txt-learnable with respect to some indexed hypothesis space.*

(2) $\mathcal{C} \in \mathit{Lim\,Txt}$.

Proof. Clearly, if $\mathcal{C} \in \mathit{Lim\,Txt}_{\mathcal{H}}$ with respect to some indexing $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ then $\mathcal{C} \in \mathit{Bc\,Txt}_{\mathcal{H}}$, 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 $\mathit{Bc\,Txt}_{\mathcal{H}}$-identifies $\mathcal{C}$, and let $(w_j)_{j \in \mathbb{N}}$ be any effective enumeration of all strings in Σ^*. Now 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) $\mathit{content}(t[n]) \subseteq L_{M'(t[n-1])}$.

     (ii) $\{w_z \mid z \leqslant n,\, w_z \in L_{M'(t[n-1])}\} = \{w_z \mid z \leqslant n,\, w_z \in L_{M(t[n])}\}$.

     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 $L_{M'(t[m])} \neq L$, 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 *Bc Txt*-learnability. If a class is *Bc Txt*-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 *Lim Txt*-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 [59], as communicated by Sanjay Jain. 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 indexed families comprising the target class $\mathcal{C}$ as hypothesis spaces, not for indexings of $\mathcal{C}$.

Theorem 20. *Let $\mathcal{C}$ be an indexable class. Then the following statements are equivalent.*

(1) $\mathcal{C} \in Consv\,Txt$.

(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 $\mathcal{C} \in Consv\,Txt$. 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 21.

(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 computable. 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. Now 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.

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 directly 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 (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 Claim 21 is proved. ∎

By Claim 21, 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 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 $n \geqslant 1$. *)
>   If *content*$(t[n]) \subseteq L'_{M'(t[n-1])}$, then return $M'(t[n-1])$ and stop;
>   (* M' is conservative. *)

(3) (* Now $content(t[n]) \not\subseteq L'_{M'(t[n-1])}$. *)

　　If there exists some $j \leqslant n$, such that $content(t[n]) \subseteq L_j$ and
　　$M'(t[m]) \notin \{\langle j, z \rangle \mid \sigma_z \in \{t[0], \ldots, t[m]\}\}$ 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.

Finally, 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 prove that $L'_{M'(t[n])} \not\supseteq L$ for all $n \in \mathbb{N}$, i.e., M' avoids to output over-generalized hypotheses when processing t. By definition, $L'_{M'(t[0])} = \emptyset$, and thus $L'_{M'(t[0])} \not\supseteq L$. Next let $n \geqslant 1$. Then $M'(t[n]) = k$ with $k = \langle j, z \rangle$ for some j, z, m with $m \leqslant n$, $\sigma_z = t[m]$, and $content(t[m]) \subseteq L_j$. Claim 21.(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 $W_{M(\sigma_z)} \not\supseteq L$ (see the corresponding discussion below Definition 19), and therefore $L'_{M'(t[n])} \not\supseteq L$.

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., $M(t[m]) = M(t[n])$ for all $m > n$. Note that $W_{M(t[n])} = L$. Let $j \geqslant n$ 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 $j \in \mathbb{N}$.)

Now we distinguish two cases.

Case (i). *There is some $m \geqslant n$ 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). Here 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 $m' \geqslant m$.

Case (ii). *There is no $m \geqslant n$ 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 $m \geqslant j$, the premises for instruction (3) would eventually be fulfilled when computing $M'(t[m])$.) So $L \subseteq L'_{M'(t[n-1])}$. Together with $L'_{M'(t[n-1])} \not\supseteq L$ this yields $L'_{M'(t[n-1])} = L$. ∎

The indexing $(L'_k)_{k \in \mathbb{N}}$ constructed in the proof of Theorem 20 in general strictly comprises the target class $\mathcal{C}$. In fact, exact indexings are not generally appropriate for conservative learning.[3]

---

[3]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.

Theorem 22 (Lange *et al.* [55]). *There is an indexable class* $\mathcal{C}$ *such that the following conditions are fulfilled.*

(1) $\mathcal{C} \in \mathit{Consv\,Txt}$.

(2) $\mathcal{C}$ *is not Consv Txt-learnable with respect to any indexing of* $\mathcal{C}$.[4]

Sketch of Proof. The idea is to modify the class defined in the proof of Theorem 13. 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

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

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

Now, one can show that $\mathcal{C}$ is an indexable class in *Consv Txt*. But $\mathcal{C}$ is not learnable conservatively with respect to any indexing of $\mathcal{C}$. The proof is done indirectly. If $\mathcal{C}$ was *Consv Txt*-learnable with respect to an indexing of $\mathcal{C}$, then the halting set K (see, Section 2.1) would be recursive, a contradiction. For details we refer the reader to Lange *et al.* [55]. ∎

This result contrasts the cases of *Lim Txt* and *Bc Txt*, 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 classes in *Consv Txt* can always be embedded into super-classes in *Consv Txt*, such that these superclasses are 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 23 (Lange and Zeugmann [54]). *Let* $\mathcal{C}$ *be an indexable class. Then the following two conditions are equivalent.*

(1) $\mathcal{C} \in \mathit{Consv\,Txt}$.

(2) *There is some indexable class* $\mathcal{C}'$, *such that* $\mathcal{C} \subseteq \mathcal{C}'$ *and* $\mathcal{C}'$ *is Consv Txt-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.

---

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

Theorem 24 (Angluin [3], Lange and Zeugmann [52], Baliga *et al.* [8]).
Let $\mathcal{C}$ be an indexable class.

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

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

(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 first 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 11. To prove necessity, assume $\mathcal{C} \in LimTxt$. Then, by Theorem 17, 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 $(L_j)_{j \in \mathbb{N}}$.

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. A works according to the following instructions:

Input: $j, n \in \mathbb{N}$
Output: $\sigma_A(j, n) \in SEG$ 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 $z \leqslant n$, 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 locking 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 $L_k = L_j$. ∎

Note that the indexings possessing telltale families can be used as a hypothesis space in the case of *Lim Txt* and *Consv Txt*. However, in the case of *Bc Txt*, the existence of an indexing possessing a telltale family is characteristic, but in general, this indexing cannot be used as a hypothesis space (cf. Theorem 18).

Telltales are also relevant for learning recursive functions from examples. Analyzing this framework, Wiehagen [92, 93] has found characterizations of learnable classes of recursive functions, which are strongly related to those in Theorem 24. See Zilles [102] 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 11, 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, *Lim Txt*-learners can be normalized to so-called *rearrangement-independent* learners, i.e., learners which, for any text segment $t(0), \dots, t(n)$ return the same hypothesis as for any text segment $t'(0), \dots, t'(n)$, where $(t'(0), \dots, t'(n))$ is a permutation (rearrangement) of $(t(0), \dots, t(n))$. Similarly, also *Consv Txt*-learners can be normalized to rearrangement-independent conservative learners, and *Bc Txt*-learners can be normalized to rearrangement-independent *Bc Txt*-learners.

Astonishingly, rearrangement-independence does *not* allow the learner for making its hypotheses depend *only on the content* of the given text segment; in general *Lim Txt*-learners cannot avoid returning different hypotheses on text segments with equal content but different length. In other words, they cannot be normalized to so-called *set-driven* learners.

Definition 21 (Wexler and Culicover [89]). *An IIM* M *is said to be* set-driven, *if* $M(\sigma) = M(\tau)$ *for any* $\sigma, \tau \in SEG$ *with content*$(\sigma) = $ *content*(τ).

Theorem 25 (Lange and Zeugmann [54]).

(1) *There is an indexable class* $\mathcal{C} \in Lim\,Txt$, *such that* $\mathcal{C}$ *is not identifiable in the limit by any set-driven IIM.*

(2) *There is an indexable class* $\mathcal{C} \in Bc\,Txt$, *such that* $\mathcal{C}$ *is not identifiable behaviorally correctly by any set-driven IIM.*

In contrast to that, when conservative learning is focused, set-drivenness is a demand which does not restrict the capabilities of learners.

Theorem 26 (Lange and Zeugmann [54]). *Let* $\mathcal{C}$ *be an indexable class. If* $\mathcal{C} \in Consv\,Txt$, *then there is a set-driven IIM which identifies* $\mathcal{C}$ *conservatively in the limit from text.*

Considering the characterizations in Theorem 24, the reader may have noticed, that there is a difference between the characterization of *Consv Txt* and those of *Lim Txt*

and $Bc\,Txt$. 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 $Consv\,Txt$ would not hold, if exact indexings of the target class were used. This fits into the picture obtained in the previous subsection concerned with the impact of the type of hypothesis spaces 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 have been rather general in their statements. The scope 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_+ of all non-erasing languages defined by restricted regular expressions.

As Proposition 1 states, this class is of finite thickness (thus also of finite elasticity), and hence belongs to $Lim\,Txt$. Moreover, it is not hard to verify that RREG_+ also has recursive finite thickness. Thus we can conclude, using Theorem 3, that the class $\mathrm{RREG}_+^{\mathrm{k}}$ of all unions of up to k languages from RREG_+ is in $Lim\,Txt$ for any $\mathrm{k} \in \mathbb{N}$. In addition, Theorem 14 implies the following corollary.

Corollary 27.

(1) $\mathrm{RREG}_+ \in Consv\,Txt$.

(2) $\mathrm{RREG}_+^{\mathrm{k}} \in Consv\,Txt$ *for any* $\mathrm{k} \in \mathbb{N}^+$.

4.2. *The Erasing Case*

Focusing on the class RREG_* of all erasing languages defined by restricted regular expressions, we observe results quite different from those in the non-erasing case. We have stated in Section 3.2, that RREG_* 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 has also been observed for the so-called *erasing pattern languages*, see Shinohara [82] for definitions and Reidenbach [74] and Mitchell [61] for results on the impact of the alphabet size.

If Σ contains at least two symbols, we can conclude from Theorems 2 and 4, that RREG_* 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 $RREG_*$ of the form $L_*(w(v)^\times)$ for some $v \in \Sigma^*$. However, this also follows from infinite elasticity of $RREG_*$. 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 $RREG_*$ and an infinite sequence $w_0, w_1, w_2, \ldots$ of strings, such that $w_j \in L_k \setminus L_j$ for all $k > j$.

This raises the question whether $RREG_*$ is of finite thickness or finite elasticity in case 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, $RREG_*$ 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, for the case that Σ is a singleton alphabet, we have verified $RREG_* \in LimTxt$, Theorem 24 implies, that if Σ is a singleton alphabet, then there must be an indexing of $RREG_*$ which possesses a uniformly r.e. family of telltales. Angluin's [3] proof of her characteristic condition now provides a uniform method for deducing such a family of telltales from any arbitrary IIM learning $RREG_*$ in the limit from text using an indexed family as a hypothesis space. However, we can even show that for $\Sigma = \{a\}$ there is an indexing of $RREG_*$ which possesses a recursively generable family of telltales:

Define an indexing of $RREG_*$ canonically by enumerating all restricted regular expressions r of the form

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

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 $RREG_*$: 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 $r' = a^m(a^{k_1'})^\times(a^{k_2'})^\times \ldots (a^{k_{s'}'})^\times$ for some $s', k_1', \ldots k_{s'}' \in \mathbb{N}$. Since $a^{m+k_z} \in L_*(r')$ for $1 \leqslant z \leqslant s$, each value k_z can be represented as a linear combination of the values $k_1', \ldots k_{s'}'$. Now choose a string $w \in L_*(r)$, say $w = a^{m+x_1 k_1 + \ldots + x_s k_s}$ for some $x_1, \ldots, x_s \in \mathbb{N}$. Obviously, $x_1 k_1 + \ldots + x_s k_s$ 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 $RREG_*$.

In contrast to that, when Σ consists of at least two symbols, then $RREG_* \notin LimTxt$ and hence, with Theorem 24, no indexing of $RREG_*$ possesses a uniformly r. e. family

of telltales. But literature tells us even more: Angluin [3] has shown that, for an alphabet of at least two symbols, the class RREG_* cannot have any telltale family, no matter which indexing is considered and no matter which algorithmic structure of the telltale family is assumed. Again applying Theorem 24, this yields $\mathrm{RREG}_* \notin Bc\,Txt$ for any alphabet of cardinality greater than 1.

Corollary 28. *Let Σ be a finite alphabet.*

(1) *If $|\Sigma| = 1$, then $\mathrm{RREG}_* \in Consv\,Txt$.*

(2) *If $|\Sigma| > 1$, then $\mathrm{RREG}_* \notin Bc\,Txt$.*

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 $\mathsf{a}((\mathsf{b})^\times(\mathsf{a})^\times)^{\mathsf{j}}$ for all $\mathsf{j} \in \mathbb{N}$, then the example string a is of no use for a learner, because it is contained in all target languages and thus bears no special information. A string $\mathsf{a}(\mathsf{b}\mathsf{a})^{\mathsf{j}+1}$ for $\mathsf{j} \in \mathbb{N}$ bears some more information, because it allows for excluding the languages $\mathsf{L}_*(\mathsf{a}((\mathsf{b})^\times(\mathsf{a})^\times)^{\mathsf{z}})$ for all $\mathsf{z} \leqslant \mathsf{j}$.

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 quite good examples for learning. The existence of such good examples could be an essential condition for the success of a learning scenario resulting from a modification of Gold's approach. Think of a classroom scenario, where 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 at can be sketched as follows:

(1) 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. *).

(2) 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 they are presented in and (ii) the learner also has to identify the target language, if additional, maybe 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 and afterwards returns a single hypothesis and stops.

This had 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:

- the class of target languages,
  (* Note that examples are good, if they help distinguishing the current target language from other possible target languages. Thus whether or not a set of examples is good for a language L cannot be defined from a local point of view, i.e., considering L only, but from a more global point of view, i.e., taking into account the whole class of target languages. 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 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 on which criterion a learner focuses on. Hence, which sets of examples are considered good, depends on the particular learning method. *)

Basing on these considerations, a formal definition of learning from good text examples has been proposed by Lange *et al.* [49]. Here 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.

Definition 22 (Jain *et al.* [36], Lange *et al.* [49]). *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 $\mathcal{H}$, *if 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.[5]

Here 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 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.

Prior to the analysis of good examples in language learning, a similar concept has been introduced and studied by Freivalds, Kinber, and Wiehagen [29] for the case of learning recursive functions.

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 29 (Jain *et al.* [36]). *GexFinTxt $\subset$ LimTxt.*

Sketch of proof. A separating class in $LimTxt \setminus GexFinTxt$ is for instance the class in $LimTxt \setminus ConsvTxt$ given in the proof of Theorem 13. For further details, we refer the reader to [36].   ∎

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. Now, since the family of good example sets is recursively generable, there is apparently a relation to conservative inference (cf. Theorem 24, Assertion (2)). Moreover, for each class $\mathcal{C}$ in *ConsvTxt* there is some indexing comprising $\mathcal{C}$ which possesses a recursively generable family of telltales. Here all languages in the class-comprising indexing have a telltale set, whereas the definition of *GexFinTxt* does not require that ex_j is a set of good examples, if $L_j \notin \mathcal{C}$. But if class-preserving learning is considered, we obtain the following nice characterization.

Theorem 30 (Lange *et al.* [49]). *Let $\mathcal{C}$ be an indexable class. Then the following two conditions are equivalent.*

---

[5]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 $L \in \mathcal{C}$.

(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 is mainly based on the following characterization of class-preserving conservative identification.

Lemma 31 (Lange *et al.* [49]). *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 30, since the telltales fulfilling the demands in the lemma can be shown to form sets of good examples in the sense of Definition 22. 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 24 suggests the opposite. In fact, a modification of Definition 22 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 A stabilize with increasing step counters.

Definition 23 (Jain *et al.* [36], Lange *et al.* [49]). *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 $\mathcal{H}$, *if 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.

Gex Lim Txt 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 *Lim Txt* in terms of learning from good examples.

Theorem 32. *Gex Lim Txt = Lim Txt.*

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

As an add-on to this discussion, in particular concerning the impact of hypothesis spaces, it is worth noting that requiring a class-preserving behavior in learning in the limit from good examples severely restricts the potential of learners. In fact, then one falls back to the capabilities of class-preserving learners identifying finitely from good examples.

Theorem 33 (Lange *et al.* [49]). *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 $(L'_j)_{j \in \mathbb{N}}$.*

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 a knowledge would be required for some application scenarios, learning in the limit has been 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, for instance, Angluin's [6] *query learning* model have been investigated.

In the query learning model, a learner has access to a teacher that truthfully answers queries of a specified kind. A *query learner* M is an algorithmic device that, depending on the reply on the previous queries, either computes a new query or returns a hypothesis and halts, see Angluin [6]. Its queries and hypotheses are interpreted with respect to an underlying hypothesis space according to Definition 3.[6]

---

[6]The use of proper class comprising hypothesis spaces has not been allowed in the original definition of query learning, see Angluin [6], but has meanwhile been studied extensively, e.g., in [57, 58, 59].

Thus, when learning $\mathcal{C}$, M is only allowed to query languages belonging to $\mathcal{H}$. More formally:

Definition 24 (Angluin [6]). *Let $\mathcal{C}$ be an indexable class, let $\mathsf{L} \in \mathcal{C}$, let $\mathcal{H} = (\mathsf{L}_j)_{j \in \mathbb{N}}$ be a hypothesis space for $\mathcal{C}$, and let M be a query learner. M learns L with respect to $\mathcal{H}$ using some type of queries if it eventually halts and its only hypothesis, say j, represents L, i.e., $\mathsf{L}_j = \mathsf{L}$.*

So M returns its unique and correct guess j after only finitely many queries.[7] Moreover, M learns $\mathcal{C}$ with respect to $\mathcal{H}$ using some type of queries, if it learns every $\mathsf{L}' \in \mathcal{C}$ with respect to $\mathcal{H}$ using queries of the specified type. In order to learn a target language L, a query learner M may ask:

Membership queries. Query: a string w. The answer is 'yes' if $w \in \mathsf{L}$ and 'no' if $w \notin \mathsf{L}$.

Restricted superset queries. Query: an index j. The answer is 'yes' if $\mathsf{L}_j \supseteq \mathsf{L}$ and 'no' if $\mathsf{L}_j \not\supseteq \mathsf{L}$.

Restricted disjointness queries. Query: an index j. The answer is 'yes' if $\mathsf{L}_j \cap \mathsf{L} = \emptyset$ and 'no' if $\mathsf{L}_j \cap \mathsf{L} \neq \emptyset$.

Note that this model in general requires a teacher answering undecidable questions. The term "restricted" has been 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 $w \in \mathsf{L} \setminus \mathsf{L}_j$ (a string $w \in \mathsf{L} \cap \mathsf{L}_j$, respectively). However, if efficiency issues are neglected and class-comprising hypothesis spaces may be used, 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 has been 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, when using the notion of hypothesis spaces, 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 hypothesis space." Now, as in the models of learning from examples, the learnability of a target class $\mathcal{C}$ may depend on the type of hypothesis spaces used. In other words, there are classes learnable with respect to any Gödel numbering, but not learnable using an indexed

---

[7]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 [57, 58, 59], we neglect efficiency issues here.

family as a hypothesis space. Therefore, we have to use different notions, depending on which type of hypothesis space is assumed:

$MemQ_{\mathrm{r.e.}}$, $rSupQ_{\mathrm{r.e.}}$, and $rDisQ_{\mathrm{r.e.}}$ denote the collections of all indexable classes $\mathcal{C}$ for which there is a query learner M which learns $\mathcal{C}$ with respect to $(W_{\mathsf{j}})_{\mathsf{j}\in\mathbb{N}}$ 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.[8] However, for membership queries this differentiation does not have any effect, so $MemQ_{\mathrm{r.e.}} = MemQ_{\mathrm{rec}}$. 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 have been considered (cf., e.g., Osherson *et al.* [71] and Jain *et al.* [38] and the references therein).

Note that, in contrast to the models of language learning from examples introduced above, learning via queries focuses the aspect of 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; that means, 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 the question arises, whether there are any similarities in these models at all and whether there are aspects of learning both models capture. Answering this question requires a comparison of both models concerning the capabilities of the corresponding learners. In particular, one central question in this context is whether IIMs, i.e., limit learners can be replaced by at least equally powerful (one-shot) query learners.

Since, in general, membership queries provide the learner with less information than restricted superset or restricted disjointness queries do, 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 $MemQ_{\mathrm{r.e.}}$ 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:

---

[8]In the literature, see Angluin [6, 7], more types of queries such as subset queries and equivalence queries have been analyzed, but in what follows we concentrate on the three types explained above.

Proposition 34. $MemQ_{\mathrm{rec}} = MemQ_{\mathrm{r.e.}} \subset Consv\,Txt$.

However, the more challenging question is whether there are interaction scenarios adequate for replacing limit learners with equally capable one-shot learners. Indeed, the query learning scenarios defined above exactly yield this property: Lange and Zilles [56, 59] have shown that the collection of indexable classes learnable with restricted superset queries coincides with $Consv\,Txt$. And, moreover, this also holds for the collection of indexable classes learnable with restricted disjointness queries.

Theorem 35 (Lange and Zilles [56, 59]). $Consv\,Txt = rSupQ_{\mathrm{rec}} = rDisQ_{\mathrm{rec}}$.

Recall that classes in $Consv\,Txt$ in general cannot be learned conservatively in a class-preserving manner (cf. Theorem 22). A similar result holds in the context of query learning:

Theorem 36. *There is an indexable class* $\mathcal{C}$, *such that the following conditions are fulfilled.*

(1) $\mathcal{C} \in rSupQ_{\mathrm{rec}}$ $[\mathcal{C} \in rDisQ_{\mathrm{rec}}]$.

(2) $\mathcal{C}$ *is not* $rSupQ_{\mathrm{rec}}$-*learnable* $[rDisQ_{\mathrm{rec}}$-*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 $\mathcal{C}_{\mathrm{sup}}$ containing the language $L_0 = \{a^z \mid z \in \mathbb{N}\} \cup \{b\}$ and all languages $L_{j+1} = \{a^z \mid z \leqslant j\}$ 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 $\mathcal{C}_{\mathrm{sup}}$. ∎

However, the embedding result obtained for conservative learning in Theorem 23, stating that classes in $Consv\,Txt$ can always be embedded into superclasses conservatively identifiable in a class-preserving manner, cannot be transferred to the case of query learning.

Theorem 37. *There is an indexable class* $\mathcal{C} \in rSupQ_{\mathrm{rec}}$ $[\mathcal{C} \in rDisQ_{\mathrm{rec}}]$ *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 $\mathcal{C}_{\mathrm{sup}} \in rSupQ$ defined in the proof of Theorem 36. This class is learnable with restricted superset queries. However, one can show that there is no superclass $\mathcal{C}$ of $\mathcal{C}_{\mathrm{sup}}$, which is learnable with restricted superset queries with respect to some indexing of $\mathcal{C}$. ∎

Theorem 35 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 38 (Lange and Zilles [59]). $Lim\,Txt = rDisQ_{\mathrm{r.e.}}$.

Comparing this result to Theorem 35, it is evident that a characterization of $Lim\,Txt$ in terms of learning with restricted superset queries is missing. Thus there remains the question whether or not $rDisQ_{\mathrm{r.e.}}$ equals $rSupQ_{\mathrm{r.e.}}$. The following result answers this question to the negative.

Theorem 39 (Lange and Zilles [59]). $rDisQ_{\mathrm{r.e.}} \subset rSupQ_{\mathrm{r.e.}}$.

Together with Theorem 38, this implies $Lim\,Txt \subset rSupQ_{\mathrm{r.e.}}$. Since $Bc\,Txt$ strictly comprises $Lim\,Txt$, this immediately raises the question of how $Bc\,Txt$ and $rSupQ_{\mathrm{r.e.}}$ are related. It has turned out that $rSupQ_{\mathrm{r.e.}}$ is an inference type which represents a collection of language classes strictly between $Lim\,Txt$ and $Bc\,Txt$.

Theorem 40 (Lange and Zilles [59]). $Lim\,Txt \subset rSupQ_{\mathrm{r.e.}} \subset Bc\,Txt$.

Here a remark on the technique used for proving this result is worth noting. Former proofs separating $Bc\,Txt$ from $Lim\,Txt$ use diagonal constructions and thus do not yield separating language classes in a compact definition. In contrast, for the proof of Theorem 40, two classes have been defined compactly—for instance the class $\mathcal{C}_1$ contained in $rSupQ_{\mathrm{r.e.}} \setminus Lim\,Txt$:

Let, for all $k, j \in \mathbb{N}$, $\mathcal{C}_1$ contain the languages $L_k = \{a^k b^z \mid z \geqslant 0\}$ and

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

The idea is to use a recursion-theoretic argument. If $\mathcal{C}_1$ 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 $\mathcal{C}_2 \in Bc\,Txt \setminus rSupQ_{\mathrm{r.e.}}$.

Finally, there are also characterizations of $Bc\,Txt$ in terms of query learning. Now, the concept of uniformly K-r.e. language families is used, see Section 2.1.

All in all, this 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.[9]

6. Efficiency Issues

One important aspect not covered yet is the efficiency of learning. Here different notions of efficiency are conceivable, such as run-time efficiency of the IIMs or efficiency in terms of the number of examples needed before convergence to the final hypothesis. However, defining an appropriate measure of efficiency for learning in the

---

[9]Further relations have been found, when additionally K-recursive IIMs are considered as learners; the reader is referred to Lange and Zilles [59] for more information.

limit is a difficult problem (cf. Pitt [72]). 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 learn efficiently (cf. Theorem 42).

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 [23] 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, the total learning time cannot be recursively bounded if it shall exclusively depend on the length of the target, but one allows arbitrarily adverse input sequences (cf. Pitt [72] 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 models and results typically discussed in algorithmic learning in this context.

6.1. *Efficiency and Learning from Positive Data*

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

Definition 25 (Angluin [2], Shinohara [82]). *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 $\sigma \colon (\Sigma \cup X)^* \mapsto \Sigma^*$ with $\sigma(a) = a$ for all $a \in \Sigma$ is called a* pattern substitution.[10]

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

(ii) *The erasing pattern language $L_\varepsilon(\pi)$ of a pattern π is defined by*
     $L_\varepsilon(\pi) = \{\sigma(\pi) \mid \sigma \text{ is a pattern substitution}\}$.

We use *PAT* and *EPAT* to refer to the set of all non-erasing pattern languages and of all erasing pattern language, 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

---

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

in their structures affect learnability: $PAT \in Consv\,Txt$, see Angluin [2, 3], whereas, under certain constraints on the size of the alphabet Σ, we have $EPAT \notin Lim\,Txt$, see Reidenbach [74, 75].

The learnability of different classes of both non-erasing and erasing pattern languages with different types of queries has been analyzed systematically, e.g., by Angluin [6], Erlebach *et al.* [27], Lange and Wiehagen [50], Nessel and Lange [67], Lange and Zilles [56]. 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. Efficiency issues are the main concern in the cited literature; however, this survey restricts the focus on run-time efficiency of IIMs in Gold-style models.

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.

Theorem 41 (Lange and Wiehagen [50]). *There is an IIM* M *and a polynomial* p*, such that the following two conditions are fulfilled:*

(1) M *identifies* PAT *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 [50] have proposed an iterative method for identifying PAT. 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 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.
   Here, 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

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

It is not hard to verify that M runs in polynomial time and identifies *PAT* 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. ∎

A naïve implementation of the operation used in (3) of the proof above for computing $\beta_1 \dots \beta_s$ would require time quadratic in $|\pi_n|$. However, Rossmanith and Zeugmann [78] provided an algorithm computing $\beta_1 \dots \beta_s$ in linear time.

The algorithm given in the proof of Theorem 41 can also be used to learn *PAT* finitely from good examples. As far as the minimal size of sets of good examples is concerned, Zeugmann [98] showed the matching upper and lower bound of

$$\lfloor \log_{|\Sigma|}(|\Sigma| + k - 1) \rfloor + 1$$

for every pattern π possessing k different variables. 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 41, Angluin's [2] algorithm for learning *PAT* in the limit from text is not run-time efficient, unless $\mathcal{P} = \mathcal{NP}$. The main difference of her algorithm and the one presented above is that the latter may also output *inconsistent* hypotheses.

Theorem 41 is a special case of a very general result stating that each class in *Lim Txt* is identifiable by an IIM with a run-time polynomial in the length of the given input. Still, Theorem 41 is worth noting because of the iterative and quite intuitive strategy proposed in the proof. Moreover, it states that the run-time of the achieved IIM 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 42. *Let $\mathcal{C}$ be any indexable class and $\mathcal{H}$ a hypothesis space such that $\mathcal{C} \in Lim\,Txt_{\mathcal{H}}$. 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 $\mathcal{C} \in Lim\,Txt_{\mathcal{H}}$. Now, we construct the desired IIM M which uses M' as a subroutine.

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

(1) Run M' on inputs $t[0], \dots, t[n]$ for n steps each;

(2) If there is some $z \leqslant n$ 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 $z \leqslant n$ such that $M'(t[z])$ is defined within n steps, then set $M(t[n]) = 0$, output 0 and stop.

Now, it easy to see that $q(n) = n^2$ and M defined above fulfill the assertions of the theorem. ∎

An obvious disadvantage of the IIM M constructed in the proof of Theorem 42 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'.

In general, if efficiency is considered, this sheds a different light on some of the aspects discussed above, such as, for instance,

- the influence of the kind of information obtained in learning in the limit,

- the influence of natural constraints in learning such as consistency,

- the influence of query and hypothesis spaces as well as counterexamples in query learning.

For avoiding tricks as used in the proof of Theorem 42, one has to require the learner to be consistent and conservative. A learner is said to be *consistent* if it only outputs consistent hypotheses. Consistency alone is not sufficient if membership is in $\mathcal{P}$ (with respect to the chosen hypothesis space $\mathcal{H}$), since one could output Σ^* as a trivial hypothesis provided it belongs to $\mathcal{H}$. Therefore, de la Higuera [25] proposed a new model for efficient learning and Clark and Eyraud [22] adapted it to learning from positive data.

Let $\mathcal{C}$ be an indexable class and $(L_j)_{j \in \mathbb{N}}$ an indexing of $\mathcal{C}$. In the model of de la Higuera [25] it is assumed that a function $\# : \mathbb{N} \to \mathbb{N}$ is *a priori* fixed that allows for measuring the structural complexity of the hypotheses in $(L_j)_{j \in \mathbb{N}}$. (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.) Moreover, note that the model of de la Higuera [25] focusses on set-driven IIMs, only (see Definition 21). To ease notations, we assume in the following definition that a set-driven IIM receives as input finite sets of strings instead of finite sequences thereof.

Definition 26 (de la Higuera [25]). *Let $\mathcal{C}$ be an indexable class, $(L_j)_{j \in \mathbb{N}}$ an indexing of $\mathcal{C}$, and $\#$ the specified function to measure the structural complexity of the hypotheses in $(L_j)_{j \in \mathbb{N}}$. Now $\mathcal{C}$ is* identifiable in the limit from text with polynomial time and data with respect to $(L_j)_{j \in \mathbb{N}}$, *if there exist two polynomials p, q and a set-driven IIM M such that for all $L \in \mathcal{C}$:*

(1) *For every $S \subseteq L$ of size m, $M(S)$ can be computed in time $p(m)$.*

(2) *For every* $j \in \mathbb{N}$ *with* $L_j = L$ *and* $\#(j) = n$ *there exists a characteristic set* S_j *for* L_j *of size less than* $q(n)$ *such that* $L_{M(S)} = L$ *for all finite sets* $S \subseteq L$ *with* $S_j \subseteq S$.[11]

Within this framework, Clark and Eyraud [22] 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 if $lvr \in L$ 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 [97] used Definition 26 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 [96]. He also discusses the problem to formally define the notion of of *efficient learning in the limit* in greater detail.

Yet another different approach is *stochastic finite learning*. Within this setting, one studies learning in the limit from randomly generated texts and analyzes the expected total learning time. For example, Reischuk and Zeugmann [77] studied the learnability of one-variable pattern languages in this setting and showed that for almost all meaningful distributions defining how the pattern variable is replaced by a string to generate random examples 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 [99] for more information concerning stochastic finite learning.

Summarizing, the problem to define *efficient learning in the limit* is still requiring more research and 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 (cf. Adriaans *et al.* [1]).

6.2. *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 is, that learnability results for indexable classes otherwise become trivial as long as efficiency issues are neglected. To see this, first we define the notion of *informant*.

Definition 27 (Gold [31]). *Any total function* $i \colon \mathbb{N} \mapsto \Sigma^* \times \{-, +\}$ *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.

---

[11]Recall that S_j is said to be a characteristic set for L_j, if $L_j \subseteq L_k$ for all $k \in \mathbb{N}$ with $S_j \subseteq L_k$ (see Definition 14).

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 $content^+(i[n]) = \{w \mid i(j) = (w, +), \ 0 \leqslant j \leqslant n\}$ and $content^-(i[n]) = \{w \mid i(j) = (w, -), \ 0 \leqslant j \leqslant n\}$.

Definition 28 (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, if $content^+(i[n]) \subseteq L$ and $content^-(i[n]) \cap L = \emptyset$.*

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.

Learning in the limit from informant is defined analogously to identification in the limit from text—an IIM receives gradually growing 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*.

Furthermore, we define consistent learners as follows.

Definition 29 (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.*

(1) *M is* consistent *for $\mathcal{C}$ with respect to $\mathcal{H}$, if $i[n]$ is consistent with $L_{M(i[n])}$ for 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 informant *with respect to $\mathcal{H}$ if there is an IIM M which is consistent for $\mathcal{C}$ with respect to $\mathcal{H}$ witnessing $\mathcal{C} \in LimInf_{\mathcal{H}}$.*

We denote the resulting learning type by *ConsInf*.

Next, we recall Gold's [31] result stating that $\mathcal{C} \in ConsInf$ for every indexable class $\mathcal{C}$.

Theorem 43 (Gold [31]). *Let $\mathcal{C}$ be an indexable class. Then there is an indexing $\mathcal{H} = (L_j)_{j \in \mathbb{N}}$ and a general recursive IIM M such that M is consistent on every input and $\mathcal{C} \in ConsInf_{\mathcal{H}}$.*

Proof. The theorem is shown by using the *identification by enumeration* method. Let $\mathcal{C}$ be an indexable class and $(L'_j)_{j \in \mathbb{N}}$ an indexing for $\mathcal{C}$. Furthermore, let $(F_j)_{j \in \mathbb{N}}$ be any fixed canonical indexing of the class of all finite languages (over the same alphabet Σ used for defining $\mathcal{C}$), and let *aux* be a recursive mapping from all finite subsets of Σ^* to $\mathbb{N}$ such that $F = F_{aux(F)}$ for all finite $F \subseteq \Sigma^*$. We set $L_{2j} = L'_j$ and $L_{2j+1} = F_j$ for every $j \in \mathbb{N}$. The desired IIM M is defined as follows.

> Input: $i[n]$ for some informant i and some $n \in \mathbb{N}$
> Output: hypothesis $j \in \mathbb{N}$
>
> Search the least j such that $i[n]$ is consistent with L_j, return j and stop.

By construction, it is easy to see that M is general recursive and that M witnesses $\mathcal{C} \in ConsInf_{\mathcal{H}}$. ∎

A closer look at that proof above also shows the following. If learners were allowed not to terminate on inputs not corresponding to any target language, then $\mathcal{C} \in ConsInf_{\mathcal{H}}$ for every indexing $\mathcal{H}$ of $\mathcal{C}$.

The identification by enumeration method is in some sense universal. In the context of learning recursive functions, Wiehagen [90] has stated the thesis that given a natural inference type in Gold-style terms, successful learners can always be normalized to learners implementing the method of identification by enumeration with respect to some appropriate hypothesis space.

Since the classes *PAT* and *EPAT* are indexable, both are consistently 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 $\mathcal{NP}$-hard (cf. Ko *et al.* [44]).

Nevertheless, Theorem 42 directly translates to *LimInf*. That is, if *inconsistent* learners are admissible, then again one can always construct IIM having polynomial update time.

Theorem 44. *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 $\mathsf{q}(\mathsf{n})$.*

Again, it is obvious that the simulation technique used does not yield any advantage. It does neither increase the efficiency of the overall learning algorithm, if one adds all steps of computation until the learning task is successfully finished, nor does it increase the learning power.

Next, we include the requirement to compute hypotheses in polynomial time directly into the definition of our learning types. That is, we require the existence of a polynomial p such that the time to compute $\mathsf{M}(\mathfrak{i}[\mathsf{n}])$ is less than or equal to $\mathsf{p}(length(\mathfrak{i}[\mathsf{n}]))^{12}$ for all informants $\mathfrak{i}$ corresponding to some language in the target class and all $\mathsf{n} \in \mathbb{N}$. The resulting learning types are denoted by *Poly-ConsInf* and *Poly-LimInf*, respectively.

Of course, the identification by enumeration method is just one technique to achieve consistent learning by directly using the uniform decidability of membership problem. Many different techniques are imaginable. Therefore, we ask whether or not inconsistent IIMs do yield more learning power than consistent ones when restricted to

---

[12] As usual, if $z[\mathsf{n}] = (w_0, \ldots, w_{\mathsf{n}})$ or $z[\mathsf{n}] = ((w_0, b_0), \ldots, (w_{\mathsf{n}}, b_{\mathsf{n}}))$, then $length(z[\mathsf{n}])$ denotes the sum of the length of the strings $w_0, \ldots, w_{\mathsf{n}}$.

polynomial time learnability. As first result in this regard has been found by Wiehagen and Zeugmann [94].

Theorem 45.

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

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

Note that Assertion (1) can be proved by using the algorithm of Theorem 41, thus showing that there are "intelligent" inconsistent techniques.

We finish this section by presenting a rather general result showing that polynomial time restricted consistent IIM are less powerful than inconsistent ones. This result has been communicated to us by Martin Kummer[13].

An indexable class is said to be *dense* if for all disjunct sets E^+ and E^- there is a language $L \in \mathcal{C}$ such that $E^+ \subseteq L$ and $L \cap E^- = \emptyset$. Then, we have the following Theorem.

Theorem 46. *Let $\mathcal{C}$ be any dense indexable class. Then $\mathcal{C} \in Poly\text{-}ConsInf$ if and only if there is a $k \in \mathbb{N}$ such that $\mathcal{C} \subseteq \mathrm{DTIME}(\mathfrak{n}^k)$.*

Proof. Necessity. Let M be any IIM witnessing $\mathcal{C} \in Poly\text{-}ConsInf$ and let $\mathfrak{n}^i$ be the corresponding polynomial. Furthermore, let S be the set of all finite sequences $(w_0, b_0), \ldots, (w_k, b_k)$ with pairwise different $w_j \in \Sigma^*$ and $b_j \in \{+, -\}$. Since $\mathcal{C}$ is dense, $M(s)$ is defined for every $s \in S$ and $M(s)$ is consistent with s.

Let $L \in \mathcal{C}$ be arbitrarily fixed. Then there is a sequence $s_0 \in S$ such that $content^+(s_0) \in L$, $content^-(s_0) \cap L = \emptyset$ and $M(s) = M(s_0)$ for all sequences s that are consistent with L and extending s_0 (cf. Blum and Blum [17]). Thus, $M(s_0)$ is an index for L. Now, we claim the following.

Claim 1. For all $w \in \Sigma^*$ such that $w \notin content^+(s_0) \cup content^-(s_0)$ we have: $w \in L$ if and only if $M(s_0, (w, +)) = M(s_0)$.

If $w \in L$ then $M(s_0, (w, 1)) = M(s_0)$ by the choice of s_0. For the opposite direction, if $M(s_0, (w, +)) = M(s_0)$, then the hypothesis $M(s_0)$ is consistent with $(w, +)$. Since $M(s_0)$ is an index for L we get $w \in L$. This proves Claim 1.

Finally, we have to estimate the time complexity of the algorithm presented. Since the time to compute $M(s_0, (w, +))$ is bounded by $\mathfrak{n}^i$, where $\mathfrak{n} = length(s_0, (w, +))$, we thus have $L \in \mathrm{DTIME}(\mathfrak{n}^i)$ and the necessity is shown.

Sufficiency. Let $\mathcal{C} \subseteq \mathrm{DTIME}(\mathfrak{n}^k)$ for some fixed $k \in \mathbb{N}$. Then it is well-known that there is an indexing $(L_j)_{j \in \mathbb{N}}$ comprising $\mathrm{DTIME}(\mathfrak{n}^k)$ 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 $w \in L_j$.

---

[13]Personal Communication, 1992

Now it is not hard to verify that the following IIM M implementing the identification by enumeration principle witnesses $\mathcal{C} \in \textit{Poly-ConsInf}$. 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 $j \leqslant n$ 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 $D = \textit{content}^+(i[n])$, and return $M(i[n]) = 2k + 1$.

∎

Theorem 46 allows the following corollary. Before we can state it, we need the following notations. Let $\mathcal{CS}$ denote the class of all 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.

Corollary 47.

(1) $\mathcal{CS} \in \textit{Poly-ConsInf}_{\mathcal{H}_{cs}}$ *if and only if* $\mathcal{P} = \mathcal{PSPACE}$.

(2) $\mathcal{P} \notin \textit{Poly-ConsInf}_{\mathcal{H}}$ *for every hypothesis space* $\mathcal{H}$ *for* $\mathcal{P}$.

Proof. For showing Assertion (1), 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.

Then, it is well-known that *CSL* is $\mathcal{PSPACE}$-complete and thus, in particular, $CSL \in \mathcal{PSPACE}$ (cf., e.g., Wegener [87]).

Now, assuming $\mathcal{P} = \mathcal{PSPACE}$, we can conclude $CSL \in \mathcal{P}$. By the definition of $\mathcal{P}$ this implies that there is a $k \in \mathbb{N}$ such that *CSL* is in $\mathrm{DTIME}(n^k)$. This is true for all reasonable encodings of context-sensitive grammars. Thus, we can assume that the canonical encoding is used. Now Theorem 46 yields $\mathcal{CS} \in \textit{Poly-ConsInf}_{\mathcal{H}_{cs}}$.

Next, assume $\mathcal{CS} \in \textit{Poly-ConsInf}_{\mathcal{H}_{cs}}$. Using again Theorem 46 we know that $\mathcal{CS} \subseteq \mathrm{DTIME}(n^k)$ for some $k \in \mathbb{N}$. Since $\mathcal{CS}$ is directly mapped to *CSL* by using the encoding given by $\mathcal{H}_{cs}$ we get $CSL \in \mathcal{P}$. Finally, the $\mathcal{PSPACE}$-completeness of *CSL* implies $\mathcal{P} = \mathcal{PSPACE}$, and Assertion (1) is proved.

Assertion (2) should be obvious. ∎

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 [31] 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 pointed out that the choice of hypothesis spaces may be of huge impact, even when not concerned with efficiency, thus leading us to the discussion of how the algorithmic complexity and the richness of hypothesis spaces may influence learning in different variants of Gold's model. Note that the discussion of conservative learning here included a previously unpublished proof (Theorem 20). Taking up the concept of telltales again, the survey has discussed typification theorems characterizing the classes learnable in Gold's model and its variants.

In order to relate Gold's initial model to other natural approaches to learning, two interesting models have been picked out, namely learning from good examples and learning from queries. Both of them have been shown to be strongly related to Gold's model of identification in the limit, particularly when indexable classes are chosen as target classes for learning.

Finally, efficiency issues have been addressed, with a particular focus on the effect that negative data have in learning as well as with a focus on the problem of keeping hypotheses consistent with the data during the learning process.

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 has already been mentioned in the introduction, this survey covers only parts of the research on learning indexable classes of recursive languages in the limit. However, the selection we have made is very useful for pointing out the relevance of the studies on inductive inference when concerned with other areas of research, such as grammatical inference, machine learning, or, more generally, artificial intelligence.

As claimed in the introduction, these relations show in different aspects; let us just briefly pick them up here again and see how they are reflected in the survey:

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

- *Approximation:* Identification in the limit is in fact a quite natural way of formalizing learning processes. For instance in human language learning as well

as in many other real life learning processes, it is in general not the case that the learner is aware of when it has actually reached its goal. In particular when the learner is instead required to just gradually approximate the target, identification in the limit can be seen as a way of learning by approximation—a very natural technique for describing concepts. For instance, in numerical analysis or in probability theory, approximation and limiting processes are central conceptions.

- *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 has been much more research on the limitations of incremental learning in this context, which we unfortunately could not discuss here. For further reading, see for instance Wiehagen [90], Lange and Zeugmann [51], Jain *et al.* [37].

- *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 implement a technique of search through a hypothesis space. The importance of such search strategies in inductive inference is for instance discussed by Wiehagen [93, 92].

Environmental constraints. A major branch of studies in the area discussed above is concerned with the environmental constraints reflected in the learning model. Especially in this case the reader should be aware of the fact that this survey is by no means comprising concerning the work done in that area. So we can just pick out some examples.

- *Hypothesis spaces:* In this context, Section 3.4 addresses 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.

- *Bias:* In machine learning, the environmental constraints are often regarded in terms of a bias on the learning problem. Throughout the sections above, this is reflected especially in the way the hypothesis spaces are chosen, but in particular in the implicit knowledge about the target class the learner can make use of. This knowledge may for instance concern structural properties of the possible target languages. Also in the model of learning from good examples such a bias

is implicit. If the learner knows that the examples presented are good examples, i.e., they form a sample with special properties, then this information can be used as an additional bias.

Efficiency. Finally, efficiency is of course of high relevance for any kind of application and thus a major topic in machine learning research. Section 6 provides a brief insight into the nature in which efficiency is discussed also in inductive inference of indexed families of recursive languages. The results shown there are partly of a quite general character, but partly also concern concrete target classes, such as the class of all non-erasing pattern languages, the class of all context-sensitive languages, or the class $\mathcal{P}$. At least two notions of efficiency play a role in this context.

- *Run-time efficiency:* First, one might want a learner to be run-time efficient. This aspect is addressed in Section 6 on a general level, but also consistent learning of non-erasing pattern languages from informant, as well as for learning non-erasing pattern languages iteratively from text. Of course, also here the survey can only give examples for the questions addressed in inductive inference of recursive languages.

- *Efficiency in terms of sample size:* Second, a related but different aspect is efficiency in terms of the number of examples required for learning. This aspect is not so much dealt with explicitly above, but having a closer look, one finds it implicitly in several places. For instance, when considering learning from good examples, one implicitly assumes that a certain portion of examples may be sufficient 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 the discussion above, again the algorithm designed by Lange and Wiehagen for learning the non-erasing pattern languages from text must be mentioned. It proves in fact that generally almost all of the examples contained in a pattern language are completely irrelevant 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 easy access and helpful overview for those who are not so familiar with the topic as well as some new insights and ideas for those who have themselves already studied this field of science and contributed to it.

References

[1] P. W. Adriaans, M. Trautwein, and 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, Milovy, Czech Republic, November 25 - December 2, 2000, Proceedings, pages 173–186, 2000.

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

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

[4] D. Angluin. Inference of reversible languages. *Journal of the ACM*, 29(3):741–765, July 1982.

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

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

[7] D. Angluin. Queries revisited. *Theoret. Comput. Sci.*, 313(2):175–194, 2004. Special issue for ALT 2001.

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

[9] J. Barzdin. On synthesizing programs given by examples. In *International Symposium on Theoretical Programming*, volume 5 of *Lecture Notes in Computer Science*, pages 53–63. Springer-Verlag, 1974.

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

[11] J. M. Barzdin. Prognostication of automata and functions. In C. V. Freiman, J. E. Griffith, and J. L. Rosenfeld, editors, *Information Processing 71, Proceedings of IFIP Congress 71, Volume 1 - Foundations and Systems, Ljubljana, Yugoslavia, August 23-28, 1971*, pages 81–84. North-Holland, 1972.

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

[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, editor, Теория Алгоритмов и Программ, volume I, pages 82–88. Latvian State University, 1974.

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

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

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

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

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

[20] J. Case and C. Lynes. Machine inductive inference and language identification. In *Automata, Languages and Programming, 9th Colloquium, Proceedings*, volume 140, pages 107–115. Springer-Verlag, 1982.

[21] O. Cicchello and S. C. Kremer. Inducing grammars from sparse data sets: A survey of algorithms and results. *Journal of Machine Learning Research*, 4:603–632, 2003.

[22] A. Clark and R. Eyraud. Identification in the limit of substitutable context-free languages. In *Algorithmic Learning Theory, 16th International Conference, ALT 2005, Singapore, October 2005, Proceedings*, volume 3734 of *Lecture Notes in Artificial Intelligence*, pages 283–296. Springer, Oct. 2005.

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

[24] D. de Jongh and 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*, pages 193–204, New York, NY, 1996. ACM Press.

[25] C. de la Higuera. Characterisitc sets for polynomial grammatical inference. *Machine Learning*, 27:125–138, 1997.

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

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

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

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

[30] E. M. Gold. Limiting recursion. *Journal of Symbolic Logic*, 30:28–48, 1965.

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

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

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

[34] S. Jain, E. Kinber, S. Lange, R. Wiehagen, and T. Zeugmann. Learning languages and functions by erasing. *Theoret. Comput. Sci.*, 241(1-2):143–189, 2000. Special issue for ALT '96.

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

[36] S. Jain, S. Lange, and J. Nessel. On the learnability of recursively enumerable languages from good examples. *Theoret. Comput. Sci.*, 261(1):3–29, 2001. Special issue for ALT '97.

[37] S. Jain, S. Lange, and S. Zilles. Towards a better understanding of incremental learning. In *Algorithmic Learning Theory, 17th International Conference, ALT 2006, Barcelona, Spain, October 2006, Proceedings*, volume 4264 of *Lecture Notes in Artificial Intelligence*, pages 169–183. Springer, 2006.

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

[39] S. Jain and A. Sharma. Generalization and specialization strategies for learning r.e. languages. *Annals of Mathematics and Artificial Intelligence*, 23(1/2):1–26, 1998. Special issue for AII '94 and ALT '94.

[40] S. Kaufmann and 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*, pages 263–274, New York, NY, 1997. ACM Press.

[41] M. Kearns and 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*, pages 57–71, San Mateo, CA, 1989. Morgan Kaufmann.

[42] M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory.* The MIT Press, Cambridge, Massachusetts, 1994.

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

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

[45] 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.

[46] S. Kobayashi and 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*, volume 1316 of *Lecture Notes in Artificial Intelligence*, pages 48–61, Berlin, 1997. Springer.

[47] T. Koshiba. Typed pattern languages and their learnability. In *Computational Learning Theory, Second European Conference, EuroCOLT '95, Barcelona, Spain, March 1995, Proceedings*, volume 904 of *Lecture Notes in Artificial Intelligence*, pages 367–379. Springer, 1995.

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

[49] S. Lange, J. Nessel, and R. Wiehagen. Learning recursive languages from good examples. *Annals of Mathematics and Artificial Intelligence*, 23(1/2):27–52, 1998. Special issue for AII '94 and ALT '94.

[50] S. Lange and R. Wiehagen. Polynomial-time inference of arbitrary pattern languages. *New Generation Computing*, 8(4):361–370, 1991.

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

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

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

[54] S. Lange and T. Zeugmann. Set-driven and rearrangement-independent learning of recursive languages. *Math. Syst. Theory*, 29(6):599–634, 1996. Earlier version in Fifth ALT conf, 1994, Lecture Notes in AI 872.

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

[56] S. Lange and 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*, volume 2842 of *Lecture Notes in Artificial Intelligence*, pages 129–143. Springer, 2003.

[57] S. Lange and S. Zilles. Formal language identification: query learning vs. Gold-style learning. *Information Processing Letters*, 91(6):285–292, 2004.

[58] S. Lange and 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*, volume 3120 of *Lecture Notes in Artificial Intelligence*, pages 155–169. Springer, 2004.

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

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

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

[62] T. Mitchell. Generalization as search. *Art. Int.*, 18:203–226, 1982.

[63] T. M. Mitchell. *Machine Learning*. WCB/McGraw-Hill, Boston, Massachusetts, 1997.

[64] T. Moriyama and M. Sato. Properties of language classes with finite elasticity. *IEICE Transactions on Information and Systems*, E78-D(5):532–538, 1995.

[65] T. Motoki, T. Shinohara, and K. Wright. The correct definition of finite elasticity: corrigendum to Identification of unions. In *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, page 375, San Mateo, CA, 1991. Morgan Kaufmann.

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

[67] J. Nessel and S. Lange. Learning erasing pattern languages with queries. *Theoret. Comput. Sci.*, 348(1):41–57, 2005. Special issue for ALT 2000.

[68] Y. K. Ng and 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*, volume 4201 of *Lecture Notes in Artificial Intelligence*, pages 307–319, Berlin, 2006. Springer.

[69] Y. K. Ng and T. Shinohara. Developments from enquiries into the learnability of the pattern languages from positive data. *Theoret. Comput. Sci.*, xxx:xxx–xxx, 2008.

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

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

[72] L. Pitt. Inductive inference, DFAs, and computational complexity. In *Analogical and Inductive Inference, International Workshop AII '89, Reinhardsbrunn Castle, GDR, October 1989, Proceedings*, volume 397 of *Lecture Notes in Artificial Intelligence*, pages 18–44. Springer-Verlag, 1989.

[73] 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*, volume 2533 of *Lecture Notes in Artificial Intelligence*, pages 308–320. Springer, 2002.

[74] 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*, volume 3120 of *Lecture Notes in Artificial Intelligence*, pages 140–154. Springer, 2004.

[75] D. Reidenbach. A non-learnable class of E-pattern languages. *Theoret. Comput. Sci.*, 350(1):91–102, 2006. Special issue for ALT 2002.

[76] D. Reidenbach. Discontinuities in pattern inference. *Theoret. Comput. Sci.*, xxx:xxx–xxx, 2008.

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

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

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

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

[81] M. Sato, Y. Mukouchi, and 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*, volume 1501 of *Lecture Notes in Artificial Intelligence*, pages 220–233, Berlin, 1998. Springer.

[82] T. Shinohara. Polynomial time inference of extended regular pattern languages. In *RIMS Symposium on Software Science and Engineering, Kyoto, 1982, Proceedings*, volume 147 of *Lecture Notes in Computer Science*, pages 115–127. Springer-Verlag, 1982.

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

[84] F. Stephan. Noisy inference and oracles. *Theoret. Comput. Sci.*, 185(1):129–157, 1997. Special issue for ALT '95.

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

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

[87] I. Wegener. *Theoretische Informatik — eine algorithmenorientierte Einführung*. Teubner, Stuttgart, 1993.

[88] K. Wexler. The subset principle is an intensional principle. In E. Reuland and W. Abraham, editors, *Knowledge and Language*, volume 1, pages 217–239. Kluwer Academic Publishers, 1993.

[89] K. Wexler and P. Culicover. *Formal Principles of Language Acquisition*. MIT Press, Cambridge, Mass., 1980.

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

[91] R. Wiehagen. Identification of formal languages. In *Mathematical Foundations of Computer Science 1977, Proceedings, 6th Symposium, Tatranská Lomnica, September 5-9, 1977*, volume 53 of *Lecture Notes in Computer Science*, pages 571–579. Springer-Verlag, 1977.

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

[93] R. Wiehagen. A thesis in inductive inference. In *Nonmonotonic and Inductive Logic, 1st International Workshop, Karlsruhe, Germany, December 1990, Proceedings*, volume 543 of *Lecture Notes in Artificial Intelligence*, pages 184–207. Springer-Verlag, 1990.

[94] R. Wiehagen and T. Zeugmann. Ignoring data may be the only way to learn efficiently. *J. of Experimental and Theoret. Artif. Intell.*, 6(1):131–144, 1994. Special issue for AII '92.

[95] 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*, pages 328–333. Morgan Kaufmann, 1989.

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

[97] 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*, volume 4754 of *Lecture Notes in Artificial Intelligence*, pages 227–241, Berlin, oct 2007. Springer.

[98] T. Zeugmann. Lange and Wiehagen's pattern language learning algorithm: An average-case analysis with respect to its total learning time. *Annals of Mathematics and Artificial Intelligence*, 23:117–145, 1998.

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

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

[101] T. Zeugmann and S. Zilles. Learning recursive functions – a survey. *Theoret. Comput. Sci.*, xxx:xxx–xxx, 2008.

[102] S. Zilles. *Uniform Learning of Recursive Functions*, volume 278 of *Dissertationen zur Künstlichen Intelligenz*. Akademische Verlagsgesellschaft, Aka GmbH, Berlin, 2003.