The Complexity of Universal Text-Learners*

Frank Stephan†
University of Heidelberg

Sebastiaan A. Terwijn‡
University of München

Abstract The present work deals with language learning from text. It considers universal learners for classes of languages in models of additional information and analyzes their complexity in terms of Turing-degrees. The following is shown: If the additional information is given by a set containing at least one index for each language from the class to be learned but no index for any language outside the class then there is a universal learner having the same Turing degree as the inclusion problem for recursively enumerable sets. This result is optimal in the sense that any other successful learner has the same or higher Turing degree. If the additional information is given by the index set of the class of languages to be learned then there is a computable universal learner. Furthermore, if the additional information is presented as an upper bound on the size of some grammar that generates the language then a high oracle is necessary and sufficient. Finally, it is shown that for the concepts of finite learning and learning from good examples, the index set of the class to be learned gives insufficient information: due to the restrictive convergence constraints, these criteria need the jump of the index set instead of the index set itself. So they have infinite access to the information of the index set in finite time.

1 Introduction

Gold [10] introduced the notion of learning languages from text: A learner reads an infinite sequence, called *text*, of data which contains every element of the language to be learned but no element outside the language. The task of the learner is to output after each data

*This research has been presented on the Eleventh International Symposium on Fundamentals of Computation Theory, Krakow, 1997.

†Mathematisches Institut, Universität Heidelberg, Im Neuenheimer Feld 294, 69120 Heidelberg, Germany, EU, fstephan@math.uni-heidelberg.de, supported by the Deutsche Forschungsgemeinschaft (DFG) grant no. Am 60/9-2.

‡Mathematisches Institut, Ludwig-Maximilians-Universität München, Theresienstrasse 39, 80333 München, Germany, EU, terwijn@rz.mathematik.uni-muenchen.de, this author holds a TMR Marie Curie fellowship of the European Union under grant no. ERB-FMBI-CT98-3248.

item a guess for the grammar of the language such that the sequence of these guesses converges to a single correct grammar. A collection of languages is called learnable if there is a single computable learner for this collection.

There exist classes which cannot be learned by a computable learner. For example the class REC of all computable functions cannot be learned by a computable machine which receives as input the sequence $f(0)f(1)\ldots$ of the values of f. Adleman and Blum [1] as well as Gasarch and Pleszkoch [9] considered nonrecursive learners and measured their complexity in terms of Turing degrees. Adleman and Blum showed, for example, that a learner for REC exists in a Turing degree $\mathbf{a}$ if and only if $\mathbf{a}$ is high ($\mathbf{a}' \geq \mathbf{0}''$).

A learner which can learn every object in the target concept class is called omniscient. Adleman and Blum [1] constructed nonrecursive omniscient learners for function learning. But such omniscient learners do not exist for the model of learning from text: The class $\{\mathbb{N}\} \cup \{D \subseteq \mathbb{N} : D \text{ is finite}\}$ cannot be learned from text relative to any oracle [10]. Let Λ denote the collection of all classes $\mathcal{L}$ of languages which are learnable by some (possibly nonrecursive) oracle machine. The languages are represented in an abstract way as recursively enumerable subsets of the natural numbers so that a grammar is just an algorithm which enumerates all elements of the language but no nonelements. Jain and Sharma [13] showed that no Turing degree suffices to learn all classes in Λ: for any oracle A, there exists a class $\mathcal{L} \in \Lambda$ which is not learnable relative to A. An alternative proof for this fact is given by Osherson, Stob and Weinstein [21, Proposition 4.1A] who showed that no denumerable set of learners can learn every class from Λ — this implies the just mentioned fact directly since there are only countably many learners computable relative to a given Turing degree.

It is a natural question to ask what resources are needed to learn all the learnable in a uniform way. Although no fixed resource allows to learn all classes in Λ, one can inquire whether there exists a uniform learning procedure that is given as a parameter extra information about the class of which the current target language is an element. Following a model presented by Kaufmann and Stephan [16] the question is asked whether there is a learner M which succeeds for every class $\mathcal{L} \in \Lambda$ when M receives as additional information an oracle B which describes $\mathcal{L}$ in some specified way.

It is shown that such a learner exists if B contains an index for all languages in $\mathcal{L}$ but for no language outside of $\mathcal{L}$. The Turing degrees of such learners are exactly the degrees above $\mathbf{0}''$, that is, every universal learner must be able to solve the inclusion problem for recursively enumerable sets. While in this general case the learner is inherently nonrecursive, it is shown that for the more restricted case where $B = \{e : W_e \in \mathcal{L}\}$ there exists already a computable universal learner. After presenting these results in Section 2, they are adapted to the world of learning recursive languages in Section 3. Section 4 deals with the case where an upper bound on the size of some grammar for each language L is provided to the learner instead of information on the whole class $\mathcal{L}$. In this setting, which was introduced by Freivalds and Wiehagen [8] and explored by Jain and Sharma [12], there is a single learner for the whole class of all recursively enumerable languages. Such a learner exists in a Turing degree $\mathbf{a}$ if and only if $\mathbf{a}$ is high. In Section 5 it is investigated to which extent it is possible to transfer the results of the previous sections to the concept of finite learning.

Even if B is an index set of $\mathcal{L}$, it might in some cases be necessary to work with B' instead of B since a finite learner cannot investigate the whole set B in finite time.

Osherson, Stob and Weinstein [23] proved a result similar to those in Sections 2 and 3 in a model-theoretic context. They constructed a universal inductive inference machine that learns in the limit from data about a model of a set of sentences T whether a given sentence θ holds in this model, provided that T is given as an oracle and that both θ and $\neg\theta$ are equivalent under T to an existential-universal sentence. Further related work considers the case where uniformly recursively enumerable classes are given by a single index e where the class to be learned is of the form $\mathcal{L} = \{W_{e'} : e' \in W_e\}$. Osherson, Stob and Weinstein [22] introduced this concept and Baliga, Case and Jain [3] extended the study. One fundamental result is that on the one hand there is a computable learner which identifies every finite class $\mathcal{L}$ provided that $\mathcal{L}$ is given via a set W_e which contains for every $L \in \mathcal{L}$ exactly one index but that on the other hand this fails if W_e may contain up to 2 indices per set in $\mathcal{L}$. Kapur and Bilardi [15] considered the case where the family to be learned is uniformly recursive. They showed that it is impossible to learn these families universally if the only information supplied is an index of a uniformly recursive enumeration of the family. Nevertheless they give some natural subcollections of Λ which have universal learners using an index of the families to be learned as the only additional information.

For further background information on inductive inference and recursion theory see [5, 10, 25, 19, 20, 21, 26]. The notation mainly follows the given references. So W_e is the e-th recursively enumerable set given as the domain of the e-th partial computable function φ_e computed by the e-th program. K is the halting problem $\{e : e \in W_e\}$. For any finite or infinite set A, let A^* denote the set of the finite strings of elements in A. $range(\sigma)$ is the set of all elements occurring in the string σ. The concatenation of strings σ and τ is denoted by $\sigma\tau$. Finally, $\sigma \preceq \tau$ denotes that σ is a prefix of τ, that is, $\tau = \sigma\eta$ for some string η. Sets and languages are identified with their characteristic function, so $L(x) = 1$ for $x \in L$ and $L(x) = 0$ for $x \notin L$.

2 Universal Learning From Text With Index Sets

This section contains the two main results: in the general scenario the Turing degrees of the universal learners (as defined below) are just the cone above $\mathbf{0}''$, that is, the Turing degree $\mathbf{0}''$ is necessary and sufficient. Furthermore, in the special case where the oracle B contains exactly those indices e where $W_e \in \mathcal{L}$ there is already a computable universal learner, that is a universal learner of Turing degree $\mathbf{0}$.

Definition 2.1 A set B is *for* $\mathcal{L}$ if $\mathcal{L} = \{W_e : e \in B\}$. Let Λ be the collection of all classes $\mathcal{L}$ of languages which are learnable by some (possibly nonrecursive) learner without any additional information. A *universal* (*text*) *learner* is a (not necessarily computable) machine M which learns every class $\mathcal{L} \in \Lambda$ using a set B for $\mathcal{L}$ as additional information.

Note that a set B for $\mathcal{L}$ is not fully determined by $\mathcal{L}$, that is, B can be a proper subset of the index set $\{e : W_e \in \mathcal{L}\}$ of $\mathcal{L}$. The next two theorems taken together form one of the main results of this paper.

Theorem 2.2 *There is a universal learner M of Turing degree $\mathbf{0}''$.*

Proof The learner M works as follows:

> On input σ, M looks for the first $i \leq |\sigma|$ such that
>
> $$range(\sigma) \subseteq W_i; \quad (1)$$
> $$range(\sigma) \subseteq W_j \subset W_i \text{ for no } j \in B \text{ with } j \leq |\sigma|. \quad (2)$$
>
> If such i are found then M outputs the smallest one of them. Otherwise M abstains from guessing by outputting "?".

For the verification, let i be the minimal index within B of a language $L \in \mathcal{L}$. Angluin [2, Theorem 1] and, for the noneffective case, Osherson, Stob and Weinstein [21, Proposition 2.4A] showed that for the given W_i there is a finite set F such that F is a subset of W_i and no language in $\mathcal{L}$ is between F and W_i:

$$F \subseteq W_i \wedge (\forall W_j \in \mathcal{L})\,[F \not\subseteq W_j \vee W_j \not\subset W_i] \quad (3)$$

and so no $j \in B$ satisfies $F \subseteq W_j \subset W_i$. Any text T for L has a sufficiently long prefix σ such that $F \subseteq range(\sigma)$ and $i \leq |\sigma|$. It is easy to see that the algorithm for this and any longer prefix outputs i.

So it remains to show that the algorithm can be executed by a machine having Turing degree $\mathbf{0}''$. The only two uncomputable operations required in the algorithm, besides the access to the oracle B, are to compare whether one language is a subset of another and to check whether the range of a finite string is contained in some language. Both operations can be performed by a machine whose Turing degree is $\mathbf{0}''$. ∎

The next theorem shows that it is necessary to have the ability to check subset-conditions. It shows that every universal learner has the computational power to decide whether $W_i \subseteq W_j$ or not.

Theorem 2.3 *The Turing degree of every universal learner is at least $\mathbf{0}''$.*

Proof Let M be a universal learner and define $\mathcal{L} = \{L : L = K \vee (L \not\subseteq K \wedge L \text{ is finite})\}$, that is, $\mathcal{L}$ contains K and all finite sets which are not subsets of K. This class is in Λ since it is learnable with oracle K: the learner guesses K if $range(\sigma) \subseteq K$ and guesses $range(\sigma)$ otherwise.

The expression $use(M, B, \sigma)$ denotes all oracle queries made by the learner M^B to compute $M^B(\tau)$ for some $\tau \preceq \sigma$. This use is always a finite set. Now the following statement holds by adapting the construction of a locking sequence to the general case of universal learners:

> There exists a finite set C of indices of languages in $\mathcal{L}$ including some index of K and there is a string $\sigma \in K^*$ such that $M^B(\sigma\tau) = M^C(\sigma)$ for all $\tau \in K^*$ whenever B contains only indices of languages in $\mathcal{L}$ and $B(x) = C(x)$ for all $x \in use(M, B, \sigma)$. (4)

If (4) fails, then one can construct inductively a sequence of finite sets C_i and strings $\sigma_i \in K^*$ such that $M^{C_{i+1}}(\sigma_{i+1}) \neq M^{C_i}(\sigma_i)$, each C_i contains the minimal index of K and perhaps finitely many other indices of languages in $\mathcal{L}$, $C_{i+1}(x) = C_i(x)$ for all $x \in use(M, C_i, \sigma_i)$ and $\sigma_{i+1} \succeq \sigma_i a_i$ where $a_0 a_1 \ldots$ is an enumeration of K. It follows that $T = \lim_i \sigma_i$ is a text for K and the set $B = \bigcup_i C_i$ contains only indices of languages in $\mathcal{L}$. Furthermore, $M^B(\sigma_i) = M^{C_i}(\sigma_i)$ for all i and so M^B diverges on the text T. But since B is a set for a class in Λ containing K, M^B has to learn K from the text T and from this contradiction it follows that (4) holds.

Now the condition (4) is used in order to decide the inclusion problem. Let C and σ be as in (4) and define a total recursive function g such that

$$\varphi_{g(e,e',t)}(x) = \begin{cases} 0 & \text{if } W_{e,t} \subseteq W_{e'}; \\ \uparrow & \text{otherwise.} \end{cases}$$

So g produces a sequence of indices such that all indices belong to K in the case that $W_e \subseteq W_{e'}$ (since the functions $\varphi_{g(e,e',t)}$ are total and therefore defined at their indices $g(e,e',t)$) and some index does not belong to K in the case $W_e \not\subseteq W_{e'}$, namely $g(e,e',t)$ for the first t such that $W_{e,t} \not\subseteq W_{e'}$. Now let

$$W_{f(e,e')} = range(\sigma) \cup \{g(e,e',t) : (\forall s < t)\,[g(e,e',s) \in K]\}$$

and use padding in order to obtain that $range(f)$ is disjoint from $use(M, \sigma, C)$. If $W_e \not\subseteq W_{e'}$ then $W_{f(e,e')}$ is a finite set not contained in K. If $W_e \subseteq W_{e'}$ then $W_{f(e,e')}$ is a subset of K. The function mapping every x to 1 has infinitely many indices which are all in K but only finitely many of them can be in $W_{f(e,e')}$ since these are not of the form $g(e,e',t)$ and have to be members of $range(\sigma)$. So $W_{f(e,e')}$ is either a finite set not contained in K or a proper subset of K.

Let $B = C \cup \{f(e,e')\}$. Since B is finite, the class of all languages with indices in B is learnable and M^B has to identify $W_{f(e,e')}$. Furthermore, if $W_{f(e,e')}$ is a proper subset of K, then $M^B(\sigma\tau) \neq M^C(\sigma)$ for some $\tau \in K^*$. Otherwise $W_{f(e,e')}$ is in $\mathcal{L}$ and $M^B(\sigma\tau) = M^C(\sigma)$ for all $\tau \in K^*$. So one obtains

$$W_e \subseteq W_{e'} \;\Leftrightarrow\; (\exists \tau \in K^*)\,[M^B(\sigma\tau) \neq M^C(\sigma)]$$

where the characteristic function of B is uniformly recursive with parameters e and e'. The inclusion problem $\{(e,e') : W_e \subseteq W_{e'}\}$ is recursively enumerable relative to the learner M as an oracle.

Since $\overline{K}$ is many-one-reducible to the inclusion problem, $\overline{K}$ is recursively enumerable relative to M. Hence K is computable relative to M. Since the complement $\{(e,e') : W_e \not\subseteq W_{e'}\}$ of the inclusion problem is recursively enumerable relative to K, it is also recursively enumerable relative to M. So the inclusion problem is computable relative to M. Hence, the Turing degree of M is $\mathbf{0}''$ or above. ▮

If B contains all indices of the languages to be learned and not only some then there exist computable universal learners. This is chiefly due to the fact that index sets have a high complexity in terms of Turing degree. Rice [24] showed that every nontrivial index set (and those of learnable classes are always nontrivial) has at least the Turing degree $\mathbf{0}'$. The construction exploits that whenever a learnable class contains an infinite language then its index set B has even degree $\mathbf{0}''$, in particular, one can find in the limit an algorithm which computes relative to B the inclusion problem $\{(e,e') : W_e \subseteq W_{e'}\}$.

Theorem 2.4 *There is a computable universal learner M such that M^B learns every class $\mathcal{L} \in \Lambda$ provided that B is the index set of $\mathcal{L}$.*

Proof A canonical encoding D_i of a finite set allows to compute the cardinality and a list of all elements of D_i from the index i. Since canonical indices can be translated effectively into recursively enumerable indices, not only the question whether $W_i \in \mathcal{L}$ but also whether $D_i \in \mathcal{L}$ can be answered effectively by inspecting the index set B of $\mathcal{L}$.

A pair (D_i, W_j) (or better said, its indices) are a potential candidate to compute the inclusion problem if

$$D_i \subseteq W_j,\ D_i \notin \mathcal{L} \text{ and } W_j \in \mathcal{L}; \tag{5}$$

$$D_k \notin \mathcal{L} \text{ for all } k \text{ with } D_i \subseteq D_k \subseteq W_j. \tag{6}$$

The set A of (the indices of) these candidates is the difference of two sets which are recursively enumerable relative to B: the first one enumerates the pairs (D_i, W_j) which satisfy the condition (5) and the second one all pairs which fail to satisfy (6). So A has a B-recursive approximation: $A = \lim_s A_s$.

Recall that whenever $\mathcal{L}$ contains an infinite language W_j then this W_j has a finite subset $D_i \notin \mathcal{L}$ such that no set in $\mathcal{L}$, in particular no finite set $D_k \in \mathcal{L}$, is between D_i and W_j [2, 21]. So whenever $\mathcal{L}$ contains an infinite language then there is also a pair (D_i, W_j) satisfying (5) and (6). Furthermore, whenever a pair (D_i, W_j) belongs to A, then W_j is infinite: If W_j is finite then it has a canonical index k. By $D_i \subseteq D_k = W_j \wedge D_k \in \mathcal{L}$ it follows that no subset D_i satisfies (6) together with W_j. The inclusion problem whether $W_e \subseteq W_{e'}$ can be computed relative to B under the assumption that a given pair (D_i, W_j) is in A.

> From the index e let $W_{e,x}$ denote the finitely many elements enumerated into W_e within x steps and define
>
> $$W_{f(e,e')} = D_i \cup \{x \in W_j : [W_{e,x} \subseteq W_{e'}]\}.$$
>
> Now $W_e \subseteq W_{e'} \Leftrightarrow f(e,e') \in B$.

Assuming that (D_i, W_j) is in A, the verification is based on the fact that W_j is infinite and on the two implications $W_e \subseteq W_{e'} \Rightarrow (\forall x)\,[W_{e,x} \subseteq W_{e'}] \Rightarrow (\forall x \in W_j)\,[W_{e,x} \subseteq W_{e'}] \Rightarrow W_{f(e,e')} = W_j \Rightarrow W_{f(e,e')} \in \mathcal{L} \Rightarrow f(e,e') \in B$ and $W_e \not\subseteq W_{e'} \Rightarrow (\forall^\infty x)\,[W_{e,x} \not\subseteq W_{e'}] \Rightarrow (\exists x)\,[W_{f(e,e')} = \{y \in W_j : y < x \vee y \in D_i\}] \Rightarrow (\exists k)\,[D_k = W_{f(e,e')} \wedge D_i \subseteq D_k \subseteq W_j] \Rightarrow W_{f(e,e')} \notin \mathcal{L} \Rightarrow f(e,e') \notin B$.

All pairs (D_i, W_j) can be put into an ordering equivalent to that of the natural numbers, so that it is possible to speak of a first pair, second pair and so on. Recall that it is computable relative to B in the limit as to which of these pairs belong to A — $A_0, A_1, \ldots$

denotes this B-recursive approximation of A. Having this approximation, it is possible to describe a computable universal learner which learns every class $\mathcal{L}$ from text using the index set B for $\mathcal{L}$ provided that $\mathcal{L} \in \Lambda$.

> On input σ check whether there is a pair in $A_{|\sigma|}$ among the first $|range(\sigma)|$ pairs.
>
> If so, take the least such pair $(D_i, W_j) \in A_{|\sigma|}$ and emulate the algorithm from Theorem 2.2 using this pair to answer the inclusion queries at positions (1) and (2).
>
> If not, just output the canonical index for $range(\sigma)$.

For the verification of the algorithm it is necessary to consider two cases where L denotes the actual language $L \in \mathcal{L}$ whose text is fed into the learner. Recall that $|L|$ is the cardinality of L and note that $n \leq |L|$ for all n if L is infinite.

First, the case that, for all $n \leq |L|$, the n-th pair does not belong to A. Then L has to be finite since otherwise such a pair must exist and must have an index below the cardinality ∞ of L. So for sufficiently long prefixes σ of a given text, $range(\sigma) = L$ and none of the first $|L|$ pairs is in the current approximation $A_{|\sigma|}$. So for all these sufficiently long prefixes of the text, M outputs the canonical index of L and so converges to a correct index.

Second, there is a least n-th pair $(D_i, W_j) \in A$ and $n \leq |L|$. Then for all sufficiently long prefixes σ of a given text of L, $|range(\sigma)| > n$, (D_i, W_j) belongs to $A_{|\sigma|}$ but none of the pairs before (D_i, W_j). So M goes into the first case and uses the pair (D_i, W_j) for deciding the subset queries of type (1) and (2). Therefore the algorithm produces for almost all prefixes σ of the given text the same output as the algorithm in Theorem 2.2 and converges to an index of L. ∎

3 Learning From Recursive Indices

In the case of learning classes of recursive languages, one may consider the situation where one or more *programs* are given for each $L \in \mathcal{L}$, rather than just one or more grammars generating it. An index e is a program for L if φ_e computes the characteristic function of L: φ_e is total and $L(x) = \varphi_e(x)$ for all x. In the following every total function φ_e is identified with the set $\{x : \varphi_e(x) > 0\}$. Furthermore, B is *a set of programs for* $\mathcal{L}$ if and only if B contains only total programs (which converge on every input) and $\mathcal{L} = \{\varphi_e : e \in B\}$. A machine M BC-learns a class $\mathcal{L}$ from text if and only if on every text for some $L \in \mathcal{L}$, M outputs an infinite sequence $e_0, e_1, \ldots$ of hypotheses such that almost all e_n are grammars for L.

The next result is quite parallel to the a result of Baliga, Case and Jain [3, Theorem 11], who showed that there is an algorithm which translates every index of a uniform decision procedure of a class $\mathcal{L} \in \Lambda$ into a program of a BC-learner for $\mathcal{L}$. The essential ideas of the proof of that result and the one below are the same.

Theorem 3.1 *There is a computable universal* BC*-learner M such that M^B learns a class $\mathcal{L} \in \Lambda$ whenever $\mathcal{L}$ consists of recursive languages and B is a set of programs for $\mathcal{L}$.*

Proof The BC-learner M is constructed as follows: M computes on input σ first the set I_0 of all indices $i \leq |\sigma|$ such that $i \in B$ and $range(\sigma) \subseteq \varphi_i$. The guess $f(I_0)$ then does two processes in parallel: First it throws out all indices from I_0 which are believed to be incorrect. Second it enumerates all elements which are in all remaining sets φ_i with $i \in I$ into $W_{f(I_0)}$. Formally the set of the "correct indices" is given as an intersection $I = I_0 \cap I_1 \cap I_2 \cap \ldots$ where the sets I_t are defined inductively by

$$i \in I_{t+1} \Leftrightarrow i \in I_t \wedge (\forall j < i)\,[j \notin I_t \vee \varphi_i(t) \leq \varphi_j(t)].$$

Now $W_{f(I_0)} = \bigcap_{i \in I} \varphi_i$ is recursively enumerable by the following formula:

$$W_{f(I_0)} = \{x : (\exists t)\,(\forall i \in I_t)\,[x \in \varphi_i]\}.$$

For the verification note that, for every $i \in B$, there is a finite subset F of φ_i such that, for all $j \in B$ with $F \subseteq \varphi_j$, the following holds:

- if $j < i$ then $\varphi_j \supseteq \varphi_i$;
- if $j > i$ then $\varphi_j \not\subseteq \varphi_i$.

Therefore whenever $F \subseteq range(\sigma) \subseteq \varphi_i$ and $|\sigma| > i$ then $i \in I$ and every $j \in I$ is a program of a superset of φ_i. So $W_{f(I_0)} = \varphi_i$ for almost all prefixes σ of a text for L. ∎

If a learner having Turing degree $\mathbf{a} \geq \mathbf{0}'$ is chosen then even Ex-convergence is possible, that is, the learner makes on every text of a language in $\mathcal{L}$ only finitely many mind changes. In addition to that, such a learner can take characteristic indices. So given a BC-learner M, the new Ex-learner N is obtained by

$$N(\sigma) = \min\{e : (\forall x \leq |\sigma|)\,[\varphi_e(x){\downarrow} = W_{M(\sigma)}(x)\,]\,\}.$$

On the other hand, a universal Ex-learner which succeeds on every family $\mathcal{L} \in \Lambda$ of recursive languages given as a set of programs for $\mathcal{L}$ needs at least Turing degree $\mathbf{0}'$.

Theorem 3.2 *There is a universal* Ex*-learner M such that M^B learns a class $\mathcal{L} \in \Lambda$ whenever $\mathcal{L}$ consists of recursive languages and B is a set of programs for $\mathcal{L}$. The Turing degrees of such learners are just those above $\mathbf{0}'$.*

Proof As just mentioned, any universal computable BC-learner can be transferred into a $\mathbf{0}'$-recursive Ex-learner. So the converse direction is the interesting one. Its proof is obtained by adapting the one of Theorem 2.3.

The role of K is replaced by the role of the set E of even numbers, any other infinite and coinfinite recursive set could also be used. Now $\mathcal{L}$ just consists of the set E and every finite set containing an odd number, that is, a nonelement of E. One can again show that for every universal learner M there is a finite set C of characteristic indices for sets in $\mathcal{L}$ including one for E, a constant c and $\sigma \in E^*$ such that, for all sets B containing only characteristic indices for languages in $\mathcal{L}$ and agreeing with C below c and for all $\tau \in E^*$, the equation $M^B(\sigma\tau) = M^C(\sigma)$ holds. $M^C(\sigma)$ is then a characteristic index for E. Having this, one can enumerate $\overline{K}$ relative to C:

There is a recursive function f with range $\{c+1, c+2, \ldots\}$ which assigns to any x the characteristic index of the language $range(\sigma)$ in the case $x \notin K$ and $range(\sigma) \cup \{2s+1\}$ in the case that x is enumerated into K exactly at stage s. Then

$$x \notin K \;\Leftrightarrow\; (\exists \tau \in E^*)\,[M^{C \cup \{f(x)\}}(\sigma\tau) \neq M^C(\sigma)]$$

is an existentially quantified formula for $\overline{K}$ relative to the Turing degree of M. It follows that K is computable relative to M, that is, the Turing degree of M is at least $\mathbf{0}'$. ∎

The above result uses the fact that the learner M must also succeed with nonrecursive sets B for some languages. If one requires that B is recursive, one obtains a further restriction. These restricted classes are then exactly the uniformly recursive classes. The next result shows that the Turing degrees of universal learners for these classes are exactly the high ones.

Theorem 3.3 *There are universal learners who learn those classes $\mathcal{L} \in \Lambda$ from every recursive set B of characteristic indices for the languages in $\mathcal{L}$ which have such a set B. The Turing degrees of these universal learners are just the high degrees.*

Proof In a high Turing degree $\mathbf{a}$, the learner can first identify an index for the set B in the limit. Whenever the learner makes a mind change concerning B, the learning algorithm for the language learner is restarted. So it is sufficient to formulate the algorithm for the case that a program for B is known to the universal learner M. The learner M uses a list of pairs (i, τ) such that all pairs of indices and strings occur exactly once in the list. The set

$$A = \{(i, \tau) : (\forall j \in B)\,(\forall x)\,(\exists y)\,[range(\tau) \subseteq \varphi_j \wedge x \in \varphi_i - \varphi_j \Rightarrow y \in \varphi_j - \varphi_i]\}$$

contains all (i, τ) such that $W_i \in \mathcal{L}$ and $(range(\tau), W_i)$ satisfy Angluin's condition (3) for the class given by B. The set A is in Π_2 and thus membership in A can be computed relative to $\mathbf{a}$ in the limit, let A_s be the corresponding $\mathbf{a}$-recursive approximation. Now the learning algorithm is the following:

> $M(\sigma)$ searches the first pair (i, τ) such that
>
> - $i \in B$, $i \leq |\sigma|$ and $\tau \preceq \sigma$,
> - $\varphi_i(x) = 1$ for all $x \in range(\sigma)$,
> - $(i, \tau) \in A_{|\sigma|}$.
>
> If the pair (i, τ) is found then $M(\sigma) = i$ else $M(\sigma) = ?$.

For any text T for a language $L \in \mathcal{L}$ there is a first pair $(i, \tau) \in A$ satisfying $\tau \preceq T$, $i \in B$ and $\varphi_i = L$. The learner converges either to this pair or to some previous one, thus also the output of the learner converges to some i'.

Assume now by way of contradiction, that i' is not an index for L. Then i' must be an index of a proper superset and there must be a τ' such that the leaner converges to (i', τ').

$range(\tau')$ is a subset of L and therefore $(i', \tau') \notin A$ since the relation $range(\tau') \subseteq \varphi_i \subset \varphi_{i'}$ holds. This contradiction gives the correctness of M.

For the converse direction let $\mathbf{a}$ be the Turing degree of some learner M for the class $\mathcal{L}$ containing all languages $\{2x, 2x+2, \ldots\}$, all finite sets with at least one odd element and all sets $\{2x, 2x+2, \ldots, 2x+2y\}$ whenever W_x has at least y elements but is finite. There is a recursive set B for $\mathcal{L}$: Let B contain standard indices for the languages $\{2x, 2x+2, 2x+4, \ldots\}$ and the finite languages with odd elements. Furthermore, let $f(x, y, t)$ be an index of the set having the elements $2x, 2x+2, \ldots, 2x+2y$ plus the element $2s+1$ in the case that $s > t$ and s is the first stage where a new element not yet in $W_{x,t}$ is enumerated into W_x. Let B contain those $f(x, y, t)$ where $W_{x,t}$ has at least y elements.

Now one enumerates relative to $\mathbf{a}'$ the locking sequences σ_x for the sets $\{2x, 2x+2, \ldots\}$ and defines that $g(x) = \max(range(\sigma_x))$. Since each set $\{2x, 2x+2, \ldots\}$ has at least one locking sequence, $g(x)$ is defined for all x. One has that $g(x) > 2x + 2|W_x|$ whenever W_x is finite and it follows that W_x is finite if and only if W_x has at most $g(x)$ elements. The condition $|W_x| > g(x)$ can be checked relative to $\mathbf{a}'$ and one can compute relative to $\mathbf{a}'$ which sets W_x are finite. Therefore $\mathbf{a}$ is high. ▮

Kapur and Bilardi [15, Theorem 3] showed that there is no computable learner which is universal for recursively enumerable families which are learnable from text by a computable learner. Indeed they showed that the Turing degree $\mathbf{a}$ of such a learner satisfies $\mathbf{a}'' \geq \mathbf{0}'''$, that is, $\mathbf{a}$ is high_2. The above proof uses a single family such that this family is not learnable without a high oracle. So the previous theorem does not imply the result of Kapur and Bilardi, but one can adapt the above proof by considering the parameterized classes

$$\mathcal{L}_x = \{L \in \mathcal{L} : \min(L) \in \{2x, 2x+1\}\}$$

which have uniformly recursive decision procedures whose index can be computed from x. These classes contain the set $\{2x, 2x+2, \ldots\}$, some finite sets with at least one odd element and perhaps finitely many subsets of $\{2x, 2x+2, \ldots\}$, so they are all learnable by a recursive learner. But a universal learner for all of them can be translated into a learner for $\mathcal{L}$ by waiting until some first data-item appears in the text and then emulating always the learning procedure for that $\mathcal{L}_x$ where x is the minimal number such that some number $y \leq 2x+1$ has occurred in the input text so far.

So any learner which is universal for those recursively enumerable families that are learnable by a computable learner must have high Turing degree. This improves the result of Kapur and Bilardi quoted above.

4 Bounds on the Grammar Size

Freivalds and Wiehagen [8] introduced the model where the learner receives in addition to the data $f(0), f(1), \ldots$ of the function to be learned some upper bound b on the size of some program e of f, that is, some number b such that $b > e$ for at least one of the programs for f. They showed that in this case there exists a computable universal learner which is able to learn all computable functions. Jain and Sharma [12] transferred this model to the scenario of language learning from text and showed that the result does not hold in this setting. This kind of nonlearnability is not a principal one but is only caused by the limited computational abilities of a recursive machine. Using more complex machines it is possible to learn the class of all recursively enumerable languages with one machine whose input is a text for a language L to be learned and an upper bound b on the size of some grammar for L. The Turing degrees of these machines are exactly the high degrees and so the result is very similar to those of Adleman and Blum [1] and Fortnow et al. [6] for many other learning criteria.

Theorem 4.1 *Let M be a learner which can infer every recursively enumerable language L from a text for L and from an upper bound b on some grammar for L. Then M has a high Turing degree. Furthermore, there exists such a learner in every high Turing degree.*

Proof Let $\mathbf{a}$ be a high Turing degree. Now a learner M as specified in the theorem is constructed which is computable relative to $\mathbf{a}$. Consider the following function f:

$$f(i,j) = \begin{cases} x & \text{for the smallest number } x \text{ with } W_i(x) \neq W_j(x); \\ 0 & \text{if there is no such } x\text{, that is, if } W_i = W_j. \end{cases}$$

This function is computable relative to $\mathbf{0}''$. The high degrees are those which can compute in the limit every $\mathbf{0}''$-recursive function. So it follows that for each pair i, j, $f(i,j)$ can be computed in the limit by some machine of Turing degree $\mathbf{a}$. In particular, for every b, the value $c(b) = \max\{f(i,j) : i < j \leq b\}$ can be approximated by a sequence $c_s(b)$ which is $\mathbf{a}$-recursive in both parameters s and b. The learner M uses the approximation $c_s(b)$:

$$M(\sigma, b) = i \text{ for the smallest } i \text{ with } W_{i,|\sigma|}(x) = range(\sigma)(x) \text{ for all } x \leq c_{|\sigma|}(b).$$

Note that $M(\sigma, b)$ is always defined since every finite set $range(\sigma)$ has a canonical index and one might define that, for canonical indices, every element appears already at stage 0.

Now, for the verification, let T be a text and s be so large that $c_s(b)$ already has converged to $c(b)$, that every element x of L which is smaller than $c(b)$ has already appeared in the text and that every $y \in W_j$ with $j \leq b$ and $y \leq c(b)$ has already been enumerated to W_j. Then $M(\sigma, b)$ outputs the least index i of L for every prefix $\sigma \preceq T$ of length at least s: W_i and $range(\sigma)$ coincide below $c_{|\sigma|}(b)$ since $W_i = L$ and the conditions above are satisfied. For $j < i$ the values of W_j and W_i differ below $c(b)$ and $W_{j,|\sigma|}$ and W_j coincide below $c(b)$. Thus $W_{j,|\sigma|}$ and $range(\sigma)$ disagree below $c_{|\sigma|}(b)$. The algorithm outputs i and therefore converges to the minimal index of L.

It remains to show that such a learner has high Turing degree. Let M learn every recursively enumerable language from text with an upper bound b on a grammar of this language and let $\mathbf{a}$ be the Turing degree of M. Now it is shown that the problem whether a set equals $\mathbb{N}$ can be computed relative to $\mathbf{a}'$ and thus $\mathbf{a}$ is high.

Let a be an index for the language $\mathbb{N}$ and consider the behaviour of M with the additional information $e + a$, which is an upper bound for the indices of both languages, W_e and $\mathbb{N}$. So M must learn them both using this upper bound. The language $\mathbb{N}$ has a locking sequence σ in the sense that $M(e + a, \sigma\tau) = M(e + a, \sigma)$ for all strings τ. Such a

σ can be found by a suitable algorithm of Turing degree $\mathbf{a}'$. If $W_e \neq \mathbb{N}$ then the difference must occur in $range(\sigma)$ since otherwise M would fail to learn W_e. So

$$W_e = \mathbb{N} \;\Leftrightarrow\; range(\sigma) \subseteq W_e.$$

This test whether $range(\sigma) \subseteq W_e$ is recursive in $\mathbf{0}'$ and in particular recursive in $\mathbf{a}'$. So it can be computed within Turing degree $\mathbf{a}'$ whether $W_e = \mathbb{N}$. This problem has the complexity $\mathbf{0}''$ and thus the Turing degree $\mathbf{a}$ is high. ▮

The algorithm to learn all languages from an upper bound needs nonrecursive information for exactly one part: the computation of $c(b)$. Taking now b as the minimal index of a language L, the Ex-learner can be made recursive by supplying an upper bound $c \geq c(b)$ instead of b itself. So one obtains an (unpublished) result of Jain, that an Ex-learner identifies all recursively enumerable languages with a sufficiently large upper bound as additional information.

Corollary 4.2 *There is a computable learner M which infers every recursively enumerable language L from text, given a bound c such that for the minimal index i of L and every $j < i$ there is some $x < c$ such that W_i and W_j differ on x.*

The difficulty for learning with upper bounds on the size of a grammar is due to the fact that it is impossible to know whether two languages are equal or not. To overcome this problem, Bārzdins and Podnieks [4] have introduced the slightly weaker criterion called FEx: Here the learner is not required to converge syntactically but is allowed to alternate between finitely many correct indices infinitely often. Jain and Sharma [12, Proposition 16] showed that there is a universal FEx-learner which succeeds on every recursively enumerable language L provided that an upper bound b on some grammar for L is given to the learner.

The algorithm is quite easy: For every string σ the learner takes just that index e below the given bound b for which the value

$$x(e,\sigma) = \max\{y \leq |\sigma| : (\forall z \leq y)\,[range(\sigma)(z) = W_{e,|\sigma|}(z)]\}$$

is maximal. On a text for the language L, $x(e,\sigma)$ is bounded uniformly for all prefixes σ of the text if $W_e \neq L$ and converges to ∞ if $W_e = L$. Thus the learner outputs from some certain stage only correct indices.

So weakening from Ex to FEx brings down the complexities of universal learners from the high Turing degrees to computable.

5 Finite Learning With Additional Information

Smith proposed to study topics related to those in the previous sections also for finite learning. Gold [10] introduced the notion of finite learning where the learner outputs exactly one guess which has to be correct. So a finite learner differs from an Ex-learner in that the first hypothesis output has to be the correct one. The classes which are finitely learnable, even relative to oracles, are more restricted than for the other learning criteria. Therefore, besides Λ, also the collections Λ_{fin} and Λ_{if} are considered, where Λ_{fin} contains all classes of languages learnable by some finite learner with access to an oracle and Λ_{if} all inclusion-free classes. A class $\mathcal{L}$ is inclusion-free if and only if any two distinct languages $L, H \in \mathcal{L}$ satisfy $L \not\subseteq H$ and $H \not\subseteq L$. Osherson, Stob and Weinstein [21, Exercise 1.5.2C] characterized Λ_{fin}; Mukouchi [18, Theorem 7] did the same for uniformly recursive families.

Fact 5.1 [21] *A class $\mathcal{L}$ is in Λ_{fin} if and only if every $L \in \mathcal{L}$ has a finite subset $F \subseteq L$ such that $F \not\subseteq H$ for every $H \in \mathcal{L}$ different from L.*

The proper inclusions $\Lambda_{\mathsf{fin}} \subset \Lambda_{\mathsf{if}} \subset \Lambda$ hold: Clearly every finitely learnable class is inclusion-free and the class $\mathcal{L}$ containing the set E of all even numbers and every set $\{0, 2, 4, \ldots, 2x, 2x+1\}$ witnesses the properness of the first inclusion since $\mathcal{L}$ is inclusion-free but E has no finite subset F such that E is the only superset of F within $\mathcal{L}$. The algorithm "Learning by Enumeration" which outputs an index of the first L_i to satisfy $range(\sigma) \subseteq L_i$ witnesses that every class $\{L_0, L_1, \ldots\} \in \Lambda_{\mathsf{if}}$ is also in Λ. The class $\{\emptyset, \mathbb{N}\}$ witnesses the properness of this second inclusion $\Lambda_{\mathsf{if}} \subset \Lambda$.

Behaviourally correct and explanatory learning can take into account the whole set B since they have the right to withdraw or update a hypothesis if some assumption on B turns out to be false. This is no longer true for finite learning, therefore a finite universal learner cannot succeed if it has access only to the index set. Some method to obtain infinite information on B is necessary.

Theorem 5.2 *There is no universal learner M (of arbitrarily high Turing degree) which learns every $\mathcal{L} \in \Lambda_{\mathsf{fin}}$ finitely from text with the index set $B = \{e : W_e \in \mathcal{L}\}$ of $\mathcal{L}$ as the only additional information about $\mathcal{L}$.*

Proof Let $\mathcal{L}$ contain the set $E = \{0, 2, 4, \ldots\}$ of all even numbers and perhaps also a finite set L which contains an odd number. The learner M^B has to output after reading some part $0\,2\,4\,\ldots\,2x$ of the canonical text for E a guess, this guess must compute E. Since M^B does not know whether there is any language in $\mathcal{L}$ besides E, M^B outputs this hypothesis after having queried only elements $B(b)$ with $b \leq a$ and each such b is in B if and only if b is an index for E.

Now there is another set $L = \{0, 2, 4, \ldots, 2y, 2y+1\}$ with $y \geq x$ which has no index below a — this can be obtained by taking a sufficiently large y. So L can be added to the class $\mathcal{L}$ without changing B at the queried places. The language L has a text which starts with $0\,2\,4\,\ldots\,2x$ and therefore M^B fails to identify it. So M is not a finite universal learner for all languages in Λ_{fin}. ∎

A direct consequence of the proof is that there is no finite learner — with an arbitrarily high oracle — which learns the class containing E and all sets of the form $\{0, 2, 4, \ldots, 2x, 2x+1\}$ from text. Thus the inclusion $\Lambda_{\mathsf{fin}} \subset \Lambda_{\mathsf{if}}$ is proper.

It is possible to find an uniform learner for $\mathcal{L}$ if more information than an index set is supplied to M. This information is the halting problem relative to the index set which then also allows to derive some facts of the structure of the whole set by one query. The learner can be taken to be recursive. This fact is not very surprising, since $K \leq_m B'$ in a uniform way and by Rice's Theorem [24] even $K' \leq_m B'$.

Theorem 5.3 *There is a machine M such that $M^{B'}$ finitely learns every class $\mathcal{L} \in \Lambda_{\mathrm{fin}}$ where the oracle B' is the halting problem relativized to the index set B of $\mathcal{L}$.*

Proof If $i, j \in B$ and $W_i \neq W_j$, then the union $W_i \cup W_j$ does not belong to $\mathcal{L}$ since for no proper superset of a set in $\mathcal{L}$ is also in $\mathcal{L}$; but if $W_i = W_j$ then the union equals to W_i and is still in $\mathcal{L}$. Thus the index $f(i,j)$ of the union is in B if and only if $W_i = W_j$. The finite learner has to read new data-items until there is a unique superset of the data seen so far in $\mathcal{L}$ and so the learner can check this condition by asking to B' whether

$$(\exists i, j \in B)\, [range(\sigma) \subseteq W_i \cap W_j \wedge f(i,j) \notin B] \tag{7}$$

and outputs the symbol "?" for no guess as long as (7) is satisfied. Then the learner checks using B' whether there is an $i \in B$ with $range(\sigma) \subseteq W_i$. If so, the learner outputs the smallest such i, otherwise the data is incorrect and the learner continues to output the special symbol "?". ▮

Instead of going from B to B' one might ask whether there are other ways to improve learnability. Indeed one can use the concept of using an upper bound on the size of the smallest grammar to generate a concept. This still does not work for the class containing $\{0\}$ and $\{0,1\}$ since an upper bound on the size of both programs does not help to decide whether a given text starting with a lot of 0's will eventually have also a 1. But it works for all inclusion-free classes, so this learning criterion is more powerful than finite learning alone. Progress is made in two directions: the collection of learnable classes is increased from Λ_{fin} to Λ_{if} and the complexity of the additional input for the universal learner is decreased from B' to B.

Theorem 5.4 *There is a computable learner M which learns every class $\mathcal{L} \in \Lambda_{\mathrm{if}}$ from the additional information consisting of the index set B for $\mathcal{L}$ and an upper bound a on the size of a grammar generating the language L to be learned.*

Proof M executes the following algorithm.

> $M^B(a, \sigma)$ computes the finite sets
> $B_0 = \{i \in B : \text{the size of } i \text{ is below } a\}$ and
> $B_1 = \{j \in B_0 : (\forall i \in B_0)\, [i < j \Rightarrow W_i \neq W_j]\}$.
> If there is a unique $i \in B_1$ with $range(\sigma) \subseteq W_i$
> Then $M^B(a, \sigma)$ outputs this i else $M^B(a, \sigma)$ outputs the symbol "?".

First, it is shown that all steps of the algorithm can be computed using the data given. Since $\mathcal{L}$ is inclusion-free one knows that, for $i, j \in B$ with $W_i \neq W_j$, the union $W_i \cup W_j$ is a proper superset of W_i and W_j and thus not in $\mathcal{L}$. As in the previous proof, $f(i,j)$ computes an index of $W_i \cup W_j$ and, for $i, j \in B$, one has that $W_i = W_j \Leftrightarrow f(i,j) \in B$. Similarly it can be checked using B whether $range(\sigma) \subseteq W_i$ for any $i \in B$ and σ: If so, then $W_i \cup range(\sigma)$ is in $\mathcal{L}$, otherwise $W_i \cup range(\sigma)$ is a proper superset of W_i and not in $\mathcal{L}$.

Second, one verifies that the algorithm never outputs a wrong index. Whenever T is a text for some $L \in \mathcal{L}$ and a is an upper bound on the size of some grammar generating L then the minimal index i of L is in B_1. For any $\sigma \preceq T$ it holds that $range(\sigma) \subseteq W_i$ and therefore the output can either be this i or the symbol "?". So the output cannot be an incorrect index.

Third, one verifies that the correct index is output eventually. Each two indices in B_1 are minimal indices of different languages in $\mathcal{L}$. They enumerate sets which are incomparable according to the choice of $\mathcal{L}$. Thus for every $j \in B_1$ different from i there is an x_j in $W_i - W_j$. After sufficient long time, for the finitely many $j \in B_1 - \{i\}$, the corresponding x_j have shown up in the text and thus $range(\sigma) \not\subseteq W_j$. It follows that from this time on, the index i is unique. The learner M outputs this index i. ∎

The nonlearnability of the class containing $\{0\}$ and $\{0,1\}$ cannot be overcome by using powerful oracles combined with upper bounds on programs. Freivalds, Kinber and Wiehagen [7] introduced the concept of learning from good examples. Lange, Nessel and Wiehagen [17] transferred the concepts to learning from text and showed that the learning power can be increased to that of conservative learning [2] from text for uniformly recursive families [17, Theorem 2]. Using sufficiently powerful oracles, one can turn every text learner into a conservative learner, thus one knows that every class from Λ is learnable from good examples with a sufficiently powerful learner. So good examples are a variant of finite learning having the advantage of covering all classes in Λ. Goldman and Mathias [11] defined the same notion and addressed the role of a teacher (that is, the algorithm to compute F from e in the definition below) in learning concrete classes like Horn formulas and decision lists.

Definition 5.5 [11, 17] A class $\mathcal{L}$ is *learnable from good examples* if and only if there are a partial learner M and a partial function ψ such that, for every e with $W_e \in \mathcal{L}$, $\psi(e)$ is the canonical index of a finite subset $D_{\psi(e)}$ of W_e such that, for all finite sets D_d with $D_{\psi(e)} \subseteq D_d \subseteq W_e$, $M(d)$ is defined and an index for W_e.

Again it is not possible to generate the learner M and the partial function ψ from the index set B alone. The proof of Theorem 5.2 can be adapted by taking $\mathbb{N}$ instead of E and any finite superset L of some prefix $0\,1\,\ldots\,x$ of the canonical text for $\mathbb{N}$. Therefore the next result uses B' instead of B.

Definition 5.5 omitted any constraints on the computability of the mappings related to the learning process. Therefore it was possible to define the process without caring about the enumeration on which the concept is based — Lange, Nessel and Wiehagen [17] used uniformly recursive families which makes it easier to compute $\psi(e)$ than in the case where e is taken from some acceptable numbering of all recursively enumerable sets — but the next result shows that besides the set B' no extra source of nonrecursive computation power is necessary for computing M and ψ.

Theorem 5.6 *There is a universal learner which learns every $\mathcal{L}$ in Λ from good examples using the oracle B' where B is the index set of $\mathcal{L}$. Furthermore, no fixed $\mathcal{L} \notin \Lambda$ can be learned from good examples, even when no bound in terms of Turing degrees is placed on the complexity of the learner.*

Proof Let M be the universal learner from Theorem 2.4. For every $W_e \in \mathcal{L}$, M has a locking sequence σ satisfying $M^B(\sigma\tau) = M^B(\sigma)$ for all $\tau \in W_e^*$. This definition can be checked using oracle B' and so one can define the following B'-recursive algorithm to compute $\psi(e)$:

> Enumerate W_e^* until a $\sigma \in W_e^*$ with $(\forall \tau \in W_e^*)\,[M^B(\sigma\tau) = M^B(\sigma)]$ is found. Then let $\psi(e)$ be the canonical index for $range(\sigma)$. (8)

Since M^B learns all sets in $\mathcal{L}$ and B is the index set for $\mathcal{L}$, $\psi(e)$ is defined for all $e \in B$.

Given M^B, the following machine $N^{B'}$ is a universal learner for the criterion of learning from good examples.

> $N^{B'}(d)$ enumerates all strings in D_d^* until a $\eta \in D_d^*$ is found such that $D_d \subseteq W_{M^B(\eta)}$ and $(\forall \tau \in W^*_{M^B(\eta)})\,[M^B(\eta\tau) = M^B(\eta)]$. Then $N^{B'}(d)$ outputs the index $M^B(\eta)$. (9)

For the verification of $N^{B'}$, assume that $W_e \in \mathcal{L}$ and $D_{\psi(e)} \subseteq D_d \subseteq W_e$. Let $\sigma \in D^*_{\psi(e)}$ be the string from (8).

The algorithm for $N^{B'}(d)$ is defined since it could take σ for the value η in its definition (9). For the case that the η taken there is different from σ, consider the strings $\sigma\eta$ and $\eta\sigma$. Recall that M^B was constructed in Theorem 2.4 such that M^B outputs for strings of the same length and range the same index, thus $M^B(\sigma\eta) = M^B(\eta\sigma)$. Furthermore, η is in $W^*_{M^B(\sigma)}$ since the set $W_{M^B(\sigma)}$ equals to W_e and contains therefore D_d and $range(\eta)$. It follows that $M^B(\sigma\eta) = M^B(\eta)$. Since σ is in $D^*_{\psi(e)}$, $W_{M^B(\eta)}$ contains D_d and D_d contains $D_{\psi(e)}$, the equality $M^B(\eta\sigma) = M^B(\eta)$ holds as well. Putting these observations together, one obtains that $N^{B'}(d)$ outputs also in the case $\eta \neq \sigma$ the hypothesis $M^B(\sigma)$: $N^{B'}(d) = M^B(\eta) = M^B(\eta\sigma) = M^B(\sigma\eta) = M^B(\sigma)$.

Since there is a text for W_e starting with σ and since M^B does not withdraw the hypothesis $M^B(\sigma)$ on this text, it follows that $M^B(\sigma)$ is a recursively enumerable index for W_e and the output $N^{B'}(d)$ is correct. So $N^{B'}$ witnesses together with the auxiliary function $\psi^{B'}$ that the languages in Λ are universally learnable from good examples.

The second result can be obtained by just transferring the learnability result [17, Section 3] from the world of uniformly recursive classes to arbitrary classes: the usage of learners of higher Turing degrees compensates the loss by giving up the uniform decision procedure. The direct proof is nevertheless shorter and therefore included here.

Let M and ψ witness that $\mathcal{L}$ is learnable from good examples. Since constraints on computability are absent, M and ψ are without loss of generality total. A new learner N which infers $\mathcal{L}$ from text in the limit, is defined as follows:

> $N(\sigma)$ is the minimal index e of the language generated by $M(d)$ where d is the canonical index for $range(\sigma)$.

For every $W_e \in \mathcal{L}$, $D_{\psi(e)}$ is a finite subset of W_e and whenever $D_{\psi(e)} \subseteq range(\sigma) \subseteq W_e$, then $N(\sigma)$ outputs the minimal index of W_e. All the finitely many elements of $D_{\psi(e)}$ show up eventually on every text for W_e, thus the learner N converges on every text of W_e to an index for W_e. So N learns $\mathcal{L}$ in the limit from text and $\mathcal{L} \in \Lambda$. ∎

Acknowledgments The authors would like to thank Dick de Jongh, Wolfgang Merkle, André Nies, Theodore Slaman and Carl Smith for helpful discussions.

References

[1] Lenny Adleman and Manuel Blum. Inductive inference and unsolvability. *Journal of Symbolic Logic*, 56:891–900, 1991.

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

[3] Ganesh Baliga, John Case and Sanjay Jain. Synthesizing enumeration techniques for language learning. *Proceedings of the Ninth Conference on Computational Learning Theory* (COLT), 169–180, 1996.

[4] Janis Bārzdins and Karlis Podnieks. The theory of inductive inference. *Proceedings of the Conference on Mathematical Foundations of Computer Science* (MFCS), 9–15, 1973.

[5] Leonard Blum and Manuel Blum. Towards a mathematical theory of inductive inference. *Information and Control*, 28:125–155, 1975.

[6] Lance Fortnow, William Gasarch, Sanjay Jain, Efim Kinber, Martin Kummer, Steven Kurtz, Mark Pleszkoch, Theodore Slaman, Robert Solovay and Frank Stephan. Extremes in the degrees of inferability. *Annals of Pure and Applied Logic*, 66:231–276, 1994.

[7] Rūsiņš Freivalds, Efim Kinber and Rolf Wiehagen. On the power of inductive inference from good examples. *Theoretical Computer Science*, 110:131–144, 1993.

[8] Rūsiņš Freivalds and Rolf Wiehagen. Inductive inference with additional information. *Elektronische Informationsverarbeitung und Kybernetik* 15:179–185, 1979.

[9] William Gasarch and Mark Pleszkoch. Learning via queries to an oracle. *Proceedings of the Second Annual Conference on Computational Learning Theory* (COLT), 214–229, 1989.

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

[11] Sally Goldman and David Mathias. Teaching a smarter learner. *Journal of Computer and System Sciences*, 52:255–267, 1996.

[12] Sanjay Jain and Arun Sharma. Learning with the knowledge of an upper bound on program size. *Information and Computation*, 102:118–166, 1993.

[13] Sanjay Jain and Arun Sharma. On the non-existence of maximal inference degrees for language identification. *Information Processing Letters*, 47:81–88, 1993.

[14] Klaus-Peter Jantke. Natural properties of strategies identifying recursive functions. *Elektronische Informationsverarbeitung und Kybernetik* 15:487-496, 1979.

[15] Shyam Kapur and Gianfranco Bilardi. On uniform learnability of language families. *Information Processing Letters*, 44:35–38, 1992.

[16] Susanne Kaufmann and Frank Stephan (1997): Robust learning with infinite additional information. *Proceedings of the Third European Conference on Computational Learning Theory* (EuroCOLT), 316–330, 1997. *Forschungsberichte Mathematische Logik* 23 / 1996, Mathematisches Institut, Universität Heidelberg, 1996.

[17] Steffen Lange, Jochen Nessel and Rolf Wiehagen. Language learning from good examples. *Joint Proceedings of the Fourth International Workshop on Analogical and Inductive Inference* (AII) *and of the Fifth Workshop on Algorithmic Learning Theory* (ALT) *Springer Lecture Notes in Artificial Intelligence* 872:423–437, 1994.

[18] Yasuhito Mukouchi. Characterization of finite identification. *Proceedings of the Third International Workshop on Analogical and Inductive Inference* (AII), *Springer Lecture Notes in Artificial Intelligence* 642:260–267, 1992.

[19] Piergiorgio Odifreddi. *Classical Recursion Theory.* North-Holland, Amsterdam, 1989.

[20] Daniel Osherson, Dick de Jongh, Eric Martin and Scott Weinstein. Formal learning theory. *Handbook of Logic and Language*, edited by J. van Benthem and A. ter Meulen. Elsevier, 737–775, 1997.

[21] Daniel Osherson, Michael Stob and Scott Weinstein. *Systems That Learn, An Introduction to Learning Theory for Cognitive and Computer Scientists.* Bradford — The MIT Press, Cambridge, Massachusetts, 1986.

[22] Daniel Osherson, Michael Stob and Scott Weinstein. Synthesizing inductive expertise. *Information and Computation*, 77(2):138–161, 1988.

[23] Daniel Osherson, Michael Stob and Scott Weinstein. A universal inductive inference machine. *Journal of Symbolic Logic*, 56:661–672, 1991.

[24] H. Gordon Rice. Classes of enumerable sets and their decision problems. *Transactions of the American Mathematical Society* 74:358–366, 1953.

[25] Hartley Rogers. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, New York, 1967.

[26] Robert Soare. *Recursively Enumerable Sets and Degrees. A Study of Computable Functions and Computably Generated Sets.* Springer-Verlag, Heidelberg, 1987.