Extensional Set Learning*

Sebastiaan A. Terwijn†
Mathematisches Institut
Ludwig-Maximilians-Universität München
Theresienstraße 39
D-80333 München, Germany
terwijn@rz.mathematik.uni-muenchen.de

January 5, 2000

Abstract

We investigate the model recBC of learning of r.e. sets, where changes in hypotheses only count when there is an extensional difference. We study the learnability of collections that are uniformly r.e. We prove that, in contrast with the case of uniformly recursive collections, identifiability does not imply recursive BC-identifiability. This answers a question of D. de Jongh. In contrast to the model of recursive identifiability, we prove that the BC-model separates the notions of finite thickness and finite elasticity.

1 Introduction

In this paper we consider a model of learning where two hypotheses about the data under consideration are considered equal when they denote the same object, i.e. when they are extensionally the same. This model was first defined for identification of functions in Feldman [6], Barzdin [3]. The first reference for this model in the context of set learning (learning from text) seems to be Osherson and Weinstein [14]. The model, and similar ones, have appeared in the literature under various names (e.g. in Osherson et al. it is

*An extended abstract of this work appeared in Proceedings of the Twelfth Annual Conference on Computational Learning Theory (COLT '99), University of California at Santa Cruz, July 7-9, 1999.

†The author holds a TMR Marie Curie fellowship of the European Union under grant no. ERB-FMBI-CT98-3248.

called *extensional identification*), but the term dominantly used seems to be *behaviorally correct learning.* We will follow this use of terminology, and speak of BC-learning in the sequel.

The classes of recursive functions that can be recursively BC-identified were characterized by Wiehagen (see Odifreddi [13]). In the context of set learning, Baliga, Case, and Jain [2, Corollary 2] proved that every identifiable uniformly recursive class is already recursively BC-identifiable. De Jongh and Kanazawa [9] gave a characterization of the uniformly r.e. classes that are identifiable by a recursive function. This result is completely general in the sense that every class that is recursively identifiable is included in a recursively identifiable uniformly r.e. class (Fulk [8], this is called "r.e.-boundedness").

The notion of finite thickness was introduced by Angluin [1]. She proved that for uniformly recursive collections, finite thickness implies recursive identifiability. Wright [20], corrected in Motoki et al. [12], proved that the same holds with the weaker hypothesis of finite elasticity.

In this paper we concentrate on the case of uniformly r.e. classes. We prove that, in contrast to the uniformly recursive case, there are identifiable uniformly r.e. classes that are not recursively BC-identifiable. This answers a question of Dick de Jongh. Furthermore, we prove that for finite-to-1 enumerable classes finite thickness implies recursive BC-identifiability, whereas finite elasticity does not. De Jongh and Kanazawa [9] proved that for 1-1-enumerable classes, finite thickness implies recursive identifiability. At the end of Section 5 we give an example of a 2-1-enumerable class with finite thickness that is not recursively identifiable.

It should be pointed out that no complete characterization of the recursively BC-identifiable classes is known. The proof of Fulks result quoted above yields that every recursively BC-identifiable class is Σ^0_2-bounded, so perhaps this is the natural area to look for a characterization. On the other hand, Fulks result may also hold for recursively BC-identifiable classes.[1] (This might serve as a motivation for concentrating on uniformly r.e. classes.)

[1]Kurtz and Royer [11] outlined a proof, which is currently still incomplete, that every recursive BC-learner φ can be transformed into a *prudent* recursive φ' that identifies at least as much as φ does. (A learner is prudent if it only outputs codes of sets that it can identify.) Since the set of hypotheses of a prudent recursive learner is a learnable and uniformly r.e. class, this would imply that every recursively BC-learnable class is included in a uniformly r.e. class that is recursively BC-learnable.

2 Basic definitions

Our recursion-theoretic notation follows Soare [18]. ω is the set of natural numbers. $\langle .,. \rangle$ is some standard recursive bijection from ω^2 to ω. $\omega^{<\omega}$ and ω^ω are the sets of finite and infinite **strings** of natural numbers. We will mostly identify a string σ with the set $\mathrm{rng}(\sigma) = \{n : \exists m(\sigma(m) = n)\}$. This will never cause confusion. Let λ be the empty string. For a string $\sigma \in \omega^{<\omega}$ and a set $X \in \omega^\omega$ we write $\sigma\hat{\ }X$ for the infinite string defined by

$$\sigma\hat{\ }X(n) = \begin{cases} \sigma(n) & \text{if } n < |\sigma| \\ X(n - |\sigma|) & \text{otherwise.} \end{cases}$$

For a string σ, $\sigma\restriction k$ denotes initial segment of σ of length k. By $\omega^{[n]}$ we denote the n-th section $\{\langle i, n \rangle : i \in \omega\}$ of ω. We use the following standard notation from computability theory. W_e is the e-th recursively enumerable (r.e.) set, φ_e is the e-th partial recursive function, and $W_{e,s}$ and $\varphi_{e,s}$ are their s-step finite approximations. A class $\mathcal{L}$ of r.e. sets is **uniformly recursive [uniformly r.e.]**[2] if there is a recursive [r.e.] set $A \subseteq \omega$ such that $\mathcal{L} = \{\{n : \langle n, i \rangle \in A\} : i \in \omega\}$.

Definition 2.1 A **learner** is a (possibly partial) function $\varphi : \omega^{<\omega} \to \omega$. Let L be an r.e. set. A **text** t for L is a (not necessarily recursive) element of ω^ω such that $\mathrm{rng}(t) = L$. The initial segment of length n of t is denoted by t_n. The learner φ **identifies** L if for every text t for L, $\lim_{n\to\infty} \varphi(t_n) = e$ exists and $W_e = L$. A function φ identifies a class $\mathcal{L}$ of r.e. sets if it identifies every $L \in \mathcal{L}$. A class $\mathcal{L}$ of r.e. sets is **identifiable** if there is some (not necessarily recursive) function φ that identifies $\mathcal{L}$.

Definition 2.2 A learning function φ **BC-identifies** L if for every text t for L we have that for almost every n, $\varphi(t_n)$ is defined and is a code for L. Again, a class $\mathcal{L}$ is BC-identifiable if there is a learner that identifies every element of it. We define **recBC** $= \{\mathcal{L} : \mathcal{L}$ is recursively BC-identifiable$\}$.

That is, in the BC-model of learning we do not require that the learner converges to a single code, but instead we require only semantical convergence, i.e. that the outputs of the learner all code the same r.e. set from some point onwards. Note that Definition 2.2 is only interesting when we restrict the class of learning functions, since for the unbounded case BC-identification collapses to ordinary identification. Also note that although in this paper

[2] In the computational learning theory literature the less precise term *indexed family* is sometimes used for this.

our main interest is in classes that are uniformly r.e., the above definitions do not mention enumerations.

3 Extensional learning of r.e. sets

In this section we will prove that there are identifiable uniformly r.e. classes (in fact, classes identifiable with oracle K) that cannot be recursively BC-identified. That there are identifiable classes that are not in recBC can be proved by a cardinality argument: By Osherson et al. [15, Prop. 4.1A] there are continuously many identifiable classes whereas recBC is of course countable.

Theorem 3.1 *There exists a uniformly r.e. class of recursive sets that is identifiable but not recursively BC-identifiable.*[3]

Proof. We define such a class $\mathcal{L}$ as follows. For every potential recursive learner φ_e we define a class of recursive sets $\mathcal{L}_e$ that φ_e cannot BC-identify. We then define $\mathcal{L}$ to be $\bigcup_e \mathcal{L}_e$. To ensure that $\mathcal{L}$ is identifiable we separate the strategies by letting all the elements of $\mathcal{L}_e$ be subsets of $\omega^{[e]}$. This will guarantee that there is a learner (as it happens, of Turing-degree $\mathbf{0}'$) that identifies $\mathcal{L}$.

The classes $\mathcal{L}_e$ are uniformly enumerated in stages as follows. (We simultaneously construct an r.e.-indexing of the classes $\mathcal{L}_e$ along with an enumeration of their elements.) At every stage $s+1$ we have defined an initial segment $\sigma_{e,s}$ (not necessarily of length s) of a particular 'diagonal set' V_e in $\mathcal{L}_e$. Elements enumerated in V_e at some later stage must be larger than s. This makes V_e recursive. While enumerating V_e we may decide to put additional sets of the form $\sigma_{e,s}\hat{}\omega^{[e]}$ into $\mathcal{L}_e$.

Stage $s = 0$. Set $\sigma_{e,0} = 0^{\langle 0,e\rangle -1}1$. (Hence, seen as a finite set, $\sigma_{e,0}$ equals $\{\langle 0, e\rangle\}$.

Stage $s+1$. At this stage $\sigma_{e,s}$ is defined. Put $\sigma_{e,s}\hat{}\omega^{[e]}$ into $\mathcal{L}_e$ by starting an enumeration of it in this and all the next stages. See whether there exists a triple $\langle \tau, t, x\rangle \leq s+1$ such that

(1) $\sigma_{e,s} \sqsubseteq \tau \sqsubset \sigma_{e,s}\hat{}\omega^{[e]}$

[3] Makoto Kanazawa and Carl Smith noted (June 1996) that the theorem has an analog for function learning which can be proved in the same way. A referee pointed out that one can strengthen the theorem by considering the notions of mind-change and learning with anomalies. (We will consider anomalies in Section 6.) Another referee pointed out that the class from the proof of Theorem 3.1 is also not learnable from recursive text. This also contrasts the uniformly recursive case.

(2) $\varphi_{e,t}(\tau)\downarrow$

(3) $x \geq |\tau|$ and $x \in W_{\varphi_e(\tau),t} - \tau$.

Note that the notation $\sigma \sqsubseteq \tau$ implies that τ, interpreted as a (characteristic sequence of a) set, contains no new elements smaller than the largest element of σ. If such τ, t, and x exist define

$$\sigma_{e,s+1} = ((\sigma_{e,s}\hat{\ }\omega^{[e]})\restriction x)0.$$

Otherwise, define $\sigma_{e,s+1} = \sigma_{e,s}$. This ends the construction of $\mathcal{L}_e$.

The point of (2) and (3) above is that if φ_e is to identify $\sigma_{e,s}\hat{\ }\omega^{[e]}$, it has to conjecture this set on the basis of some finite $\tau \sqsupset \sigma_{e,s}$, and in particular it should conjecture that some $x \in \omega^{[e]}$ is in the set *without having seen this evidence*, i.e. $x \geq |\tau|$. When we see a point where such a conjecture is made we diagonalize subsequently by extending τ in another way, namely by avoiding x.

So $\mathcal{L}_e$ looks like this: Either it contains the finite set V_e and a finite number of sets of the form $\sigma_{e,s}\hat{\ }\omega^{[e]}$, or it contains the (possibly infinite) set V_e and infinitely many sets of the form $\sigma_{e,s}\hat{\ }\omega^{[e]}$. These two cases are analyzed below. From the construction it is clear that the $\mathcal{L}_e$ are r.e.-indexable in a uniform way, so $\mathcal{L} = \bigcup_e \mathcal{L}_e$ is uniformly r.e. It is also easy to see that every $\mathcal{L}_e$ is identifiable (for example by using Theorem 5.1). Since we have separated the strategies by putting every element of $\mathcal{L}_e$ in $\omega^{[e]}$ we also have that $\mathcal{L}$ is identifiable too. Finally, to show that $\mathcal{L}$ cannot be BC-identified by any recursive function it suffices to show that φ_e does not BC-identify its subcollection $\mathcal{L}_e$.

Case 1. The collection $\mathcal{L}_e$ is infinite. Then $V_e = \bigcup_s \sigma_{e,s}$ is an infinite set, and φ_e cannot identify it: At infinitely many stages triples $\langle \tau, t, x\rangle$ are found, and these force φ_e to extensionally change its hypothesis when fed the enumeration of V_e. After all, if φ_e extensionally sticks to its hypothesis after such a triple is found, the construction guarantees that x is not in V_e whereas x is in $W_{\varphi_e(\tau)}$. Hence, in that case φ_e does not BC-identify V_e. But if φ_e infinitely often extensionally changes its mind it does not BC-identify V_e either.

Case 2. The collection $\mathcal{L}_e$ is finite. Then there is a least stage $s+1$ such that at no later stage a triple $\langle \tau, t, x\rangle$ is found. But this means that φ_e does not BC-identify $\sigma_{e,s}\hat{\ }\omega^{[e]}$: If it did, then for sufficiently large initial segments $\tau \sqsubset \sigma_{e,s}\hat{\ }\omega^{[e]}$, $\varphi_e(\tau)$ would be defined and the sets $W_{\varphi_e(\tau)}$ would all be equal to $\sigma_{e,s}\hat{\ }\omega^{[e]}$. Taking one such τ and choosing t and x large enough would give us a triple $\langle \tau, t, x \rangle$ satisfying (1), (2), and (3) above, contradicting our assumption. □

The previous result contrasts with the following theorem of Baliga, Case, and Jain, that implies that in Theorem 3.1 we cannot have 'uniformly recursive' instead of 'uniformly r.e.'

Theorem 3.2 (Baliga, Case, and Jain [2, Corollary 2]) *Every identifiable uniformly recursive class is already recursively BC-identifiable.*

Finally, we note that the class $\mathcal{L}$ constructed in the proof of Theorem 3.1 is 1-1-enumerable (see Definition 5.2). Also, from the proof it can be seen that $\mathcal{L}$ can be 1-1-enumerated in such a way that for any two codes from the 1-1-enumeration it is decidable whether for the sets coded by them one is included in the other.

4 Oracles

In this section we make some remarks on learning with oracles.

If $\mathcal{L}$ is recursively identifiable then it is trivially in recBC. The reverse implication does not hold by the standard counterexample

$$\{K \cup \{x\} : x \in \omega\}. \tag{1}$$

This class is known to be not recursively identifiable (Osherson et al. [15, Lemma 4.2.1C]), and it is easy to see that it is in recBC (after seeing σ, conjecture $K \cup \sigma$. This is sometimes called 'hard wiring' σ into the hypothesis.)

Proposition 4.1 *If $\mathcal{L}$ is in recBC then $\mathcal{L}$ is identifiable by a function that is K-computable.*

Proof. Suppose that φ recBC-identifies $\mathcal{L}$. Without loss of generality φ is total. Define the learning function f inductively as follows. Let $f(\lambda)$ be arbitrary, and if $f(\sigma)$ is defined then let $f(\sigma\hat{\ }n)$ equal $f(\sigma)$ unless there exists $k \leq |\sigma|$ such that $W_{\varphi(\sigma\hat{\ }n)}(k) \neq W_{f(\sigma)}(k)$. In that case let $f(\sigma\hat{\ }n) = \varphi(\sigma\hat{\ }n)$. That is, we let f follow the hypotheses of φ, but we do not change to a new hypothesis unless we have seen that it is really a new one. For this last purpose we use K. It is clear that f is K-computable. If f changes an hypothesis this means that at some point φ made an extensional change. Since φ makes only finitely many extensional changes on every $\mathcal{L}$-text we have that f converges on every $\mathcal{L}$-text. The limit of f has to be correct, for otherwise we would find a difference with φ's hypotheses at some point (note that in the definition of f we really look at extensional differences). □

Since it is easy to see that the class from the proof of Theorem 3.1 is identifiable recursively in K, we have the following strict implications:

$$\text{recursively identifiable}$$
$$\Downarrow$$
$$\text{recursively BC-identifiable}$$
$$\Downarrow$$
$$K\text{-recursively identifiable}$$

Note that the second implication is optimal, because the example (1) above is in recBC, and if A is a set such that (1) is A-recursively identifiable then $K \leq_T A$ ([15]). The first implication is not optimal since there are nonrecursive oracles A such that A-recursive identifiability is equivalent to recursive identifiability (Pleszkoch et al. [16], Slaman and Solovay [17] proved this for function learning. The results obtained there also hold for set learning, Kummer and Stephan [10, Theorem 10.5]). We do not know a characterization of those oracles A for which A-recursive identifiability implies recursive BC-identifiability.

5 Positive examples

We will make use of the following fundamental characterization of identifiable classes:

Theorem 5.1 (Angluin [1])[4] *A collection of r.e. sets $\mathcal{L}$ is identifiable if and only if for all $L \in \mathcal{L}$ there is a finite $D \subseteq L$ such that for every $L' \in \mathcal{L}$ with $D \subseteq L'$ it holds that $L' \not\subset L$.*

Definition 5.2 Let $\mathcal{L}$ be a collection of r.e. sets.

(i) (Angluin [1]) $\mathcal{L}$ has **finite thickness** if for every finite $D \neq \emptyset$ the collection $\{L \in \mathcal{L} : D \subseteq L\}$ is finite.

[4] The quoted theorem is actually an easier version of Angluins theorem, with the computability aspects stripped off.

(ii) (Motoki et al. [12]) $\mathcal{L}$ has **infinite elasticity** if there is an infinite sequence $s_0, s_1, \ldots$ of numbers and an infinite sequence $L_0, L_1, \ldots$ of languages in $\mathcal{L}$ such that for every $n \in \omega$ it holds that $\{s_0, \ldots, s_n\} \subseteq L_n$ and $s_{n+1} \notin L_n$. $\mathcal{L}$ has **finite elasticity** if it does not have infinite elasticity.

(iii) $\mathcal{L}$ is **finite-to-1 enumerable** if there is a recursive function f such that $\mathcal{L} = \{W_{f(i)} : i \in \omega\}$ and for every member $L \in \mathcal{L}$ there are at most finitely many i such that $L = W_{f(i)}$. (Note that this finite number may depend on L.) Similarly, $\mathcal{L}$ is **1-1-enumerable** if it has an enumeration in which every set has only one code.

Angluin [1] proved that for uniformly recursive collections, finite thickness implies recursive identifiability. Wright [20], corrected in [12], proved that the same holds with the weaker hypothesis of finite elasticity. We now proceed by proving that for finite-to-1 enumerable classes the BC-model of learning separates these two notions.

Theorem 5.3 *If $\mathcal{L}$ is finite-to-1 enumerable and has finite thickness then it is recursively BC-identifiable.*[5]

Proof. Suppose $\mathcal{L} = \{W_{f(i)} : i \in \omega\}$ has finite thickness and that f is a finite-to-1 enumeration of $\mathcal{L}$. Write L_i for $W_{f(i)}$ and $L_{i,s}$ for $W_{f(i),s}$. We define a computable BC-learner φ for $\mathcal{L}$. We first describe our strategy from the viewpoint of a single L_i. In the informal description we assume that we are seeing finite parts t_n of a text t and that we are defining $\varphi(t_n)$, where this definition depends on $\varphi(t_m)$, $m < n$. We will also refer to t_n as the **stage** of the construction. The idea is that, at stage t_n, we look whether $t_n \subseteq L_{i,n}$. If not, we do not choose i as an hypothesis. Now if later in the construction, that is when we are defining $\varphi(t_m)$ for some $m > n$, we find out that at stage t_n we were wrong, i.e. $t_n \subseteq L_i$ but $t_n \nsubseteq L_i, n$, we correct this by choosing i as hypothesis at stage t_m. To avoid too much changes between good and bad hypotheses, if we conjecture i at stage t_n we will *block* all L_j, $j \neq i$ by not allowing j as an hypothesis until we have seen that $L_{j,n} \subseteq t$. So we do not go back to an abandoned hypothesis i unless we have seen that $t_n \subseteq L_i$ and $L_{i,n} \subseteq t$, where t_n is the last stage at which L_i was blocked. If a blocked L_i satisfies these requirements we *unblock* it

[5] Alternative proofs of this theorem using results from the literature were suggested to us by Makoto Kanazawa and two referees. Also, the proof below actually yields a stronger conclusion, namely what is called in [15] *bounded extensional* identifiability (BEXT), where the learners hypotheses end in a finite set of indices.

and say that it is *free*. To give every set a chance of coming back into the game we do not block a set if it is already blocked. Now if t is a text for L_i then in the construction we will infinitely often conjecture i, hence any $L_j \neq L_i$ will be blocked infinitely often. After some point L_j will then be blocked forever, because either $t \not\subseteq L_j$ or $L_j \not\subseteq t$ and hence no large n are found such that both $t_n \subseteq L_j$ and $L_{j,n} \subseteq t$. Hence, since by finite thickness of $\mathcal{L}$ there are only finitely many L_j's in the game, after some point the only j's that will be conjectured are those with $L_j = L_i$. To make sure that we hit the set of right hypotheses we let every one of a finite (but beforehand unknown) number of L_j's have its turn, thereby allowing them to block other hypotheses. This we do by switching hypothesis as often as possible. After some time, the only hypotheses that are unblocked are the correct ones, and we end up switching only between right hypotheses.

We now proceed by giving the formal description of the construction. First we define some useful terminology. Write $\boldsymbol{\sigma - 1}$ for $\sigma\restriction(|\sigma|-1)$. At stage λ all L_j are **free**. L_j is **blocked** at stage σ if L_j is free at $\sigma - 1$ and $\varphi(\sigma) \neq j$, or if L_j is blocked at $\sigma - 1$ and is not unblocked at σ. L_j is **unblocked** at σ if if it is blocked at $\sigma - 1$ and for the unique $\tau \sqsubset \sigma - 1$ of maximal length such that L_j is free at τ we have

$$\tau \subseteq L_{j,|\sigma|} \qquad \text{and} \qquad L_{j,|\tau|} \subseteq \sigma.$$

L_j is **free** at σ if it is free at $\sigma - 1$ and it is not blocked at σ, or if L_j is blocked at $\sigma - 1$ and unblocked at σ. Note that L_j is always either blocked or free at a stage σ, and not both.

We now define $\varphi(\sigma)$ inductively. Let $\varphi(\lambda)$ be arbitrary, say 0. Given σ with $n = |\sigma| > 0$, suppose that $\varphi(\sigma\restriction k)$ has been defined for all $k < n$. Let $\varphi(\sigma)$ be the smallest $j \leq n$ for which

(1) $\sigma(0) \in L_{j,n}$,

(2) L_j is free at σ,

(3) $j \neq \varphi(\sigma - 1)$,

and let $\varphi(\sigma) = \varphi(\sigma - 1)$ if such j does not exist. This ends the construction of φ.

Clearly φ is recursive. To verify that φ BC-identifies $\mathcal{L}$, suppose that t is a text for $L \in \mathcal{L}$. We have to prove that for almost every n, $\varphi(t_n)$ is a code for L. By finite thickness of $\mathcal{L}$ there are only finitely many $L' \in \mathcal{L}$ such that $t_0 \in L'$. Because we only choose $\varphi(t_n) = j$ if $t_0 \in L_j$, and because $\mathcal{L}$ is finite-to-1 enumerable, there are only finitely many codes that can possibly be a value of φ on t. We will argue that the wrong ones among these are all eliminated after some point.

Lemma. Every wrongly selected hypothesis is blocked later.

Proof of Lemma. Suppose $L_i = L$, $\varphi(t_n) = j$, and $W_j \neq L$. Claim: There exists $t_k \sqsupset t_n$ such that L_i is free at t_k. Namely, suppose that L_i is blocked at t_{n+1}. Let $\tau \sqsubseteq t_n$ be of maximal length such that L_i is free at τ. If k is minimal such that $\tau \subseteq L_{i,k}$ and $L_{i,|\tau|} \subseteq t_k$ then L_i is free at t_k. This proves the claim. By the claim and (2) and (3) in the definition of φ it cannot be the case that $\varphi(\sigma) = j$ for all σ with $t_n \sqsubseteq \sigma \sqsubseteq t_k$ because there is at least one other free hypothesis (namely i). Hence L_j is blocked at some point after t_n. This proves the lemma.

We now argue that after some point such a wrong L_j cannot recover from being blocked. Suppose for a contradiction that some $L_j \neq L$ is selected infinitely often, that is, $\varphi(t_n) = j$ for infinitely many n. Then L_j is free at infinitely many stages t_n, and by the above lemma, L_j is also blocked at infinitely many stages. By the definition of "unblocked" we then have that $t_s \subseteq L_j$ and $L_{j,s} \subseteq t$ for infinitely many s, hence $L_j = L$, contradicting our assumption.

Recapitulating, we have that every wrong L_j is selected only finitely often as an hypothesis. Since by (1) and finite thickness only finitely many L_j's are considered during the whole construction, we have that after finitely many steps $\varphi(t_n)$ is always a code for L. □

In contrast to the previous theorem, we have

Theorem 5.4 *There is a 1-1 enumerable identifiable collection $\mathcal{L}$ that has finite elasticity and that is not recursively BC-identifiable.*

Proof. It suffices to check that the collection $\mathcal{L}$ of Theorem 3.1 has finite elasticity. This holds because elements from different subcollections $\mathcal{L}_e$ are disjoint, and the $\mathcal{L}_e$ themselves have finite elasticity: V_e is included in all other elements of $\mathcal{L}_e$, and every set in $\mathcal{L}_e$ different from V_e includes all but a finite number of the sets in $\mathcal{L}_e$. □

We give an example showing that in general, in Theorem 5.3 we cannot strengthen the conclusion to recursive identifiability. The collection consisting of

$$L_{2e} = \{\langle e, x\rangle : x \in W_{\pi_0(e)}\}$$

$$L_{2e+1} = \{\langle e, x\rangle : x \in W_{\pi_1(e)}\}$$

($e \in \omega$, π_0 and π_1 projection functions) has finite thickness and is 2-1-enumerable, hence is in recBC. We see that there are classes in recBC for which the inclusion problem is Π^0_2-complete. This $\mathcal{L}$ is not recursively identifiable because then the Π^0_2-complete set $\{e : W_e = \omega\}$ would be K-computable: Using the recursive learner for $\mathcal{L}$ first K-compute a locking sequence for ω and then K-decide whether this locking sequence is a subset of W_e. Then $W_e = \omega$ if and only if this is the case.

The converse of Theorem 5.3 fails because it is relatively easy to construct a uniformly recursive identifiable collection which has infinite elasticity. Note that the example constructed in the proof of Theorem 3.1 is 1-1-enumerable, and that it does indeed not have finite thickness, in accordance with Theorem 5.3. We do not know whether the hypothesis of finite-to-1 enumerability in Theorem 5.3 is necessary[6]. It was proved in de Jongh and Kanazawa [9] that if we strengthen the hypothesis in Theorem 5.3 to 1-1-enumerability we may conclude recursive identifiability rather than recursive BC-identifiability.

6 Extensional learning with anomalies

Osherson and Weinstein [14] also introduced the model of BC-identification with finitely many errors. In this section we make some comments on this model. For sets V and W we write $V =^n W$ to denote that V equals W with the possible exception of at most n points. $V =^* W$ denotes that there is an n such that $V =^n W$.

Definition 6.1 A function φ **$\mathbf{BC}^n$-identifies** a set L if φ on any text for L outputs in the limit only codes e that satisfy $W_e =^n L$. The model $\mathbf{BC}^*$ is defined similarly, with $=^n$ replaced by $=^*$.

Note that in general (no computability restraints on the learner) BC^n-identifiability is the same as identifiability with finitely many errors, so that the definition only becomes interesting if we restrict our attention to smaller classes of learners, such as the computable ones. The next theorem shows that we have a strict hierarchy.

[6] Note added in proof (July 20, 1999): Frank Stephan proved that the hypothesis is necessary. Namely he proved that there is a uniformly r.e. class $\mathcal{L}$ that has finite thickness and that is not recBC-identifiable.

Theorem 6.2 (Osherson et al. [15], Case and Lynes [4]) *There exists a uniformly recursive class $\mathcal{L}$ that is recursively* BC**-identifiable but not* BC^n*-identifiable for any n. Also, for every $n > 0$ there is a uniformly recursive class $\mathcal{L}_n$ that is recursively* BC^n*-identifiable but not* BC^{n-1}*-identifiable.*

Proof. In Osherson et al. [15, Prop. 6.3.3A] it is proved that the collection $\mathcal{L} = \{\omega - D : D \text{ finite}\}$ is recursively BC*-identifiable (always output a code for ω) but not identifiable ($\mathcal{L}$ does not satisfy the condition of Theorem 5.1). One can show that $\mathcal{L}$ is also not identifiable with a fixed number of errors. Case and Lynes [4] showed that $\mathcal{L}_n = \{\omega - D : \|D\| \leq 2n\}$ is recursively BC^n-identifiable but not BC^{n-1}-identifiable. □

Note that the $\mathcal{L}$ from Theorem 6.2 is not identifiable. The proof of Theorem 3.1 generalizes to show that

Theorem 6.3 *There exists a uniformly r.e. class of recursive sets that is identifiable but not recursively* BC^n*-identifiable for any n.*

Proof. In the proof of Theorem 3.1, instead of looking in *Stage* $s+1$ for triples $\langle \tau, t, x \rangle$, look for triples $\langle \tau, t, \vec{x} \rangle$ where $\vec{x}$ is a *vector* of ever increasing length rather than a single number. □

Let K be the Halting Problem. The next fact is easily proved using Shoenfields Limit Lemma [18].

Fact 6.4 *If a class is K-recursively* BC**-identifiable then it is recursively* BC**-identifiable.*

The next result shows that Theorem 6.3 is the best possible, namely that we cannot improve it from $\bigcup \text{BC}^n$ to BC*. The proof is a slight extension of the proof of [19, Theorem 3.1]. For the sake of completeness we include it.

Theorem 6.5 *Let $\mathcal{L}$ be an identifiable class that is uniformly r.e. Then $\mathcal{L}$ is recursively* BC**-identifiable.*

Proof. Let f be a recursive function indexing $\mathcal{L}$, i.e. $\mathcal{L} = \{W_{f(i)} : i \in \omega\}$. We define a K-recursive learner M that K-recursively BC-identifies $\mathcal{L}$. Given a finite piece of text σ, we define the program $M(\sigma)$ as follows. Define I_0 to be the set of all $i \leq |\sigma|$ such that $i \in \text{rng}(f)$, and $\sigma \subseteq W_i$. Note that I_0 is K-computable. $M(\sigma)$ will enumerate all numbers that are enumerated by all sets in I_0, throwing out some indices of I_0 every now and then. Define $I = \bigcap_t I_t$, where the sets I_t are inductively defined by

$$i \in I_{t+1} \Leftrightarrow i \in I_t \wedge (\forall j < i)[j \notin I_t \vee W_i \restriction t \subseteq W_j \restriction t].$$

Now the program $M(\sigma)$ is defined by

$$W_{M(\sigma)} = \{x : (\exists t)(\forall i \in I_t)[x \in W_i]\}.$$

To verify that M BC-identifies $\mathcal{L}$ note that by Angluins theorem 5.1 for every $i \in \mathrm{rng}(f)$ there is a finite set D such that for all $j \in \mathrm{rng}(f)$ with $D \subseteq W_j$ the following holds:

- if $j < i$ then $W_j \supseteq W_i$,
- if $j > i$ then $W_j \not\subset W_i$.

Therefore whenever $D \subseteq \sigma \subseteq W_i$ and $|\sigma| > i$ then $i \in I$ and for every $j \in I$ it holds that $W_j \supseteq W_i$. So $W_{M(\sigma)} = W_i$ for almost all prefixes σ of a text for W_i. Hence M BC-identifies $\mathcal{L}$. Since M is K-recursive, it follows with Fact 6.4 that $\mathcal{L}$ is recursively BC-identifiable. □

Acknowledgments. The question whether every identifiable uniformly r.e. collection is recursively BC-identifiable was posed to me by Dick de Jongh in Amsterdam, May 1996. I solved it (in the negative) and soon thereafter learned of the positive answer for uniformly recursive classes when John Case was visiting Amsterdam. I thank Dick de Jongh for introducing me to the subject. I thank John Case, Dick de Jongh, Bill Gasarch, Makoto Kanazawa, Carl Smith, and two anonymous referees for discussions, comments, and corrections.

References

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

[2] G. R. Baliga, J. Case, and S. Jain, *Synthesizing enumeration techniques for language learning*, 9th Conference on Computational Learning Theory, 1996

[3] J. M. Barzdin, *Two theorems on the limiting synthesis of functions*, Latv. Gos. Univ. Uče. Zap. 210 (1974) 82–88.

[4] J. Case and C. Lynes, *Machine inductive inference and language identification*, 9th International Colloquium on Automata, Languages and Programming, Lecture Notes in Comp. Sci. 140, 107–115, Springer-Verlag, 1982.

[5] J. Case and C. Smith, *Comparison of identification criteria for machine inductive inference*, Theoretical Computer Science 25 (1983) 193–220.

[6] J. Feldman, *Some decidability results on grammatical inference and complexity*, Information and Control 20 (1972) 244–262.

[7] L. Fortnow, W. Gasarch, S. Jain, E. Kinber, M. Kummer, S. Kurtz, M. Pleszkoch, T. Slaman, R. Solovay, F. Stephan, *Extremes in the degrees of inferability*, Annals of Pure and Applied Logic 66 (1994) 231–276.

[8] M. Fulk, *Prudence and other conditions on formal language learning*, Information and Computation 85 (1990) 1–11.

[9] D. de Jongh and M. Kanazawa, *Angluin's theorem for indexed families of r.e. sets and applications*, 9th Annual Conference on Computational Learning Theory, 1996.

[10] M. Kummer and F. Stephan, *On the structure of degrees of inferability*, Journal of Computer and System Sciences 52 (Special Issue COLT'93) 214–238, 1996.

[11] S. A. Kurtz and J. S. Royer, *Prudence in language learning*, 2nd Annual Workshop on Computational Learning Theory, 206–219, University of Calif., Santa Cruz, 1989.

[12] T. Motoki, T. Shinohara, and K. Wright, *The correct definition of finite elasticity: Corrigendum to identification of unions*, In the 4th Annual Workshop on Computational Learning Theory, p. 375, San Mateo, Calif., Morgan Kauffmann, 1991.

[13] P. Odifreddi, *Inductive inference of total functions*, in: *Computability, Enumerability, Unsolvability. Directions in Recursion Theory*, S. B. Cooper, T. A. Slaman, S. S. Wainer (eds.), London Math. Soc. Lecture Note Series 224, (1996) 259–288.

[14] Osherson and Weinstein, *Criteria of language learning*, Information and Control 52 (1982) 123–138.

[15] Osherson, Stob, and Weinstein, *Systems that learn*, Bradfort– The MIT Press, Cambridge, Massachusetts, 1986.

[16] M. G. Pleszkoch, W. Gasarch, S. Jain, and R. Solovay, *Learning via queries to an oracle*, manuscript 1990. Shorter version, by Gasarch and Pleszkoch, appeared in 2nd Annual Workshop on Computational Learning Theory, 214-229, 1989.

[17] T. A. Slaman and R. Solovay, *When Oracles do not help*, 4th Annual Conference on Computational Learning Theory, 379–383, 1991.

[18] R. I. Soare, *Recursively enumerable sets and degrees*, Springer-Verlag, 1987.

[19] F. Stephan and S. A. Terwijn. *The complexity of universal text-learners*, to appear in Information and Computation. Preliminary version in: B. S. Chlebus and L. Czaja (eds.), Proceedings of the Eleventh International Symposium on Fundamentals of Computation Theory (FCT '97), Lecture Notes in Computer Science 1279, 441–451, Springer-Verlag, 1997.

[20] K. Wright, *Identification of unions of languages drawn from an identifiable class*, 2nd Annual Workshop on Computational Learning Theory, 328–333, University of Calif., Santa Cruz, 1989.