6-2000

Learning Languages and Functions by Erasing

Sanjay Jain
National University of Singapore

Efim Kinber
Sacred Heart University, kinbere@sacredheart.edu

Steffen Lange
Universitat Leipzig

Rolf Wiehagen
Universitat Kaiserslautern

Thomas Zeugmann
Kyushu University

Recommended Citation

Jain, S., Kinber, E., Lange, S., Wiehagen, R., & Zeugmann, T. (2000). Learning languages and functions by erasing. *Theoretical Computer Science*, *241*(1–2), 143-189. doi: 10.1016/S0304-3975(99)00269-8

Theoretical Computer Science 241 (2000) 143–189

Theoretical Computer Science

www.elsevier.com/locate/tcs

Learning languages and functions by erasing

Sanjay Jain[a, *], Efim Kinber[b], Steffen Lange[c], Rolf Wiehagen[d], Thomas Zeugmann[e]

[a]*School of Computing, National University of Singapore, Singapore*
[b]*Department of Computer Science, Sacred Heart University, Fairfield, CT 06432-1000, USA*
[c]*Institut für Informatik, Universität Leipzig, PF 920, D-04009 Leipzig, Germany*
[d]*FB Informatik, Universität Kaiserslautern, PF 3049, D-67653 Kaiserslautern, Germany*
[e]*Department of Informatics, Kyushu University, Fukuoka 812-81, Japan*

Abstract

Learning by erasing means the process of eliminating potential hypotheses from further consideration thereby converging to the least hypothesis never eliminated. This hypothesis must be a solution to the actual learning problem. The capabilities of learning by erasing are investigated in relation to two factors: the choice of the overall hypothesis space itself *and* what sets of hypotheses must or may be erased. These learning capabilities are studied for two fundamental kinds of objects to be learned, namely languages and functions. For learning *languages* by erasing, the case of learning *indexed families* is investigated. A complete picture of all separations and coincidences of the considered models is derived. Learning by erasing is compared with standard models of language learning such as learning in the limit, finite learning and conservative learning. The exact location of these types within the hierarchy of the models of learning by erasing is established. Necessary and sufficient conditions for language learning by erasing are presented. For learning *functions* by erasing, mainly the case of learning *minimal* programs is studied. Various relationships and differences between the considered types of function learning by erasing and also to standard function learning are exhibited. In particular, these types are explored in Kolmogorov numberings that can be viewed as natural Gödel numberings of the partial recursive functions. Necessary and sufficient conditions for function learning by erasing are derived. © 2000 Elsevier Science B.V. All rights reserved.

1. Introduction

Learning by erasing means the process of eliminating potential hypotheses from further consideration thereby converging to a unique hypothesis which will never be eliminated. This hypothesis has to be a correct solution to the actual learning problem.

* Corresponding author.

E-mail addresses: sanjay@comp.nus.edu.sg (S. Jain), kinbere@sacredheart.edu (E. Kinber), slange@informatik.uni-leipzig.de (S. Lange), wiehagen@informatik.uni-kl.de (R. Wiehagen), thomas@i.kyushu-u.ac.jp (T. Zeugmann).

This approach is motivated by similarities to both human learning or, more general, human problem solving as well as automated problem solving. Actually, in solving a problem we mostly find out several "non-solutions" to that problem first, contradicting the data we have or explaining them unsatisfactorily. Of course, we then will exclude these non-solutions from our further consideration and keep only a more or less explicitly given remaining set of potential solutions. Often, at any time of the solving process we have an actual "favored candidate" among all the remaining candidates for a solution which, though, up to now cannot be proved to be really a solution and which also may change from time to time. Then, at least, the following can happen. Eventually we find a solution to the problem, can even prove its correctness and hence successfully stop the solving process. Or, our "favored candidate" will be stable from some point on, it is really a solution, but we are not absolutely sure of that. The latter case is a version of successful learning in the limit, which is what we do in building theories or, even more real world, in writing computer programs. In our approach of learning by erasing we can model both situations of being successful above. However, our main intention is a rigorous study of learning by erasing *in the limit*.

All the types of learning by erasing defined below have in common that at any step of the learning process the "favored candidate" will always be the *least* hypothesis not yet eliminated. This seems to be just the most natural choice. Moreover, in our opinion, this choice is justified by the following observations. First, by the principle of Occam's razor "simple" hypotheses should be favored. Second, in case that even in the limit many hypotheses remain uncanceled, we get a distinguished final hypothesis, just the least uncanceled one, and thus one can decide from outside whether or not the learning process was successful. And third, more formally, in case the learning machine eventually finds a provably correct hypothesis, then it can eliminate all the other hypotheses up to that one (or even all but that one) thereby making that hypothesis the least uncanceled one.

A special case of our approach, so-called co-learning, was introduced in [10], and then further studied in [11] for learning of recursively enumerable classes of recursive functions. In that case the learner has to eliminate all hypotheses but one and this one has to be correct. This approach was then used by Kummer [20] who showed that a recursively enumerable class of recursive functions is co-learnable with respect to all of its numberings iff all of these numberings are equivalent (i.e., intercompilable); thus giving a learning-theoretic solution to a longstanding problem of recursion-theoretic numbering theory. Furthermore, co-learning of indexed families of languages from text was studied in [13].

We relax the all-but-one approach by giving the learner more freedom concerning the sets of hypotheses it is allowed to erase eventually. Then the capabilities of learning by erasing are investigated in relation to two factors: the choice of the overall hypothesis space *and* what sets of hypotheses must or may be erased.

The capabilities of learning by erasing are studied for two fundamental kinds of target objects, namely languages and functions. Learning of languages and functions is usually very different from one another (cf., e.g., [3, 5, 8, 14, 19, 27], also for a general background of learning theory). Hence, it is only natural to ask whether or not there are major differences between language learning by erasing and function learning by erasing, too. In this paper, we provide both similarities and distinctions. However, the overall goal is much more far-reaching. In particular, we are mainly interested in the general capabilities of learners that achieve their learning goal by erasing non-appropriate hypotheses.

For learning *languages* by erasing, the case of learning *indexed families* is investigated. A complete picture of all separations and coincidences of the considered models is derived. Learning by erasing is compared with standard models of language learning such as learning in the limit, finite inference and conservative learning. The exact location of these types within the hierarchy of the learning by erasing models is established. Necessary and sufficient conditions for language learning by erasing are presented.

For learning *functions* by erasing, mainly the case of learning *minimal* programs is studied. Various relationships and differences between the considered types of function learning by erasing and also to standard function learning are exhibited. In particular, these types are explored in Kolmogorov numberings that can be viewed as natural Gödel numberings of the partial recursive functions. Necessary and sufficient conditions for function learning by erasing are derived.

The paper is organized as follows. Section 2 presents notations which are common to the whole paper. Section 3 deals with *language* learning by erasing. Section 3.1 gives the definitions which are specific for language learning by erasing as well as for standard language learning. In Section 3.2 the characterizations of the types of language learning by erasing are exhibited. In Section 3.3 learning from text is studied, whereas in Section 3.4 learning from informant is investigated. Section 4 deals with *function* learning by erasing. Section 4.1 contains the definitions which are specific for function learning by erasing and those for standard function learning. Section 4.2 deals with Kolmogorov numberings as hypothesis spaces. In Section 4.3 the corresponding learning problems are investigated for Gödel numberings as hypothesis spaces. In Section 4.4 the characterizations will be derived. In Section 5 the results are discussed and open problems are outlined. Preliminary versions of these results appeared in [16, 21]

2. Notations

$\mathbb{N}$ is the set of natural numbers. We set $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$. By $\emptyset$, $\in$, $\subset$, $\subseteq$, $\supset$, $\supseteq$, and $\#$ we denote the empty set, element of, proper subset, subset, proper superset, superset, and incomparability of sets, respectively. Furthermore, let $\max S$ and $\min S$ be the maximum and minimum of a set S, respectively, where, by convention, $\max \emptyset = 0$ and $\min \emptyset = \infty$. We write $|S|$ for the cardinality of a set S. By $\langle \cdot, \cdot \rangle$ we denote *Cantor's pairing function*, i.e., $\langle x, y \rangle = ((x+y)^2 + 3x + y)/2$ for all $x, y \in \mathbb{N}$. Note that $\langle \cdot, \cdot \rangle$ is monotonically increasing in both of its arguments. The quantifiers $\overset{\infty}{\forall}$ and $\overset{\infty}{\exists}$ denote 'for all but finitely many' and 'there exist infinitely many', respectively. Let ε be the empty word.

For $n \geqslant 1$, $\mathscr{R}^n$ and $\mathscr{P}^n$ denote the sets of total and partial computable functions of n arguments, respectively. We also write $\mathscr{R} = \mathscr{R}^1$ and $\mathscr{P} = \mathscr{P}^1$ for the (partial) computable functions of one argument. A numbering ψ is a (possibly partial) computable function from $\mathbb{N}^2$ to $\mathbb{N}$, i.e., $\psi \in \mathscr{P}^2$. For a numbering ψ, ψ_i denotes the function $\lambda x.\psi(i,x)$ and $\mathscr{R}_\psi = \{\psi_i \mid i \in \mathbb{N}, \psi_i \in \mathscr{R}\}$. ψ is said to have a *recursive equality problem* iff there is a total recursive predicate p such that, for all $j,k \in \mathbb{N}$, $p(j,k) = 1$ if and only if $\psi_j = \psi_k$. A class $\mathscr{C}$ of recursive functions is said to be *recursively enumerable* iff $\mathscr{C} = \mathscr{R}_\psi$ for some $\psi \in \mathscr{R}^2$, i.e., iff $\mathscr{C}$ can be recursively enumerated by some *total* recursive numbering.

A numbering ψ is called *acceptable* iff for every numbering η, there exists a recursive function h such that for all $i \in \mathbb{N}$, $\eta_i = \psi_{h(i)}$ (cf. [28]). Acceptable numberings are also called *Gödel* numberings. A numbering ψ is a *Kolmogorov* numbering iff for every numbering η, there exist a recursive function h and a constant c such that for all $i \in \mathbb{N}$, $\eta_i = \psi_{h(i)}$ and $h(i) \leqslant \max\{c \cdot i, c\}$. A numbering ψ is called 1–1 iff $\psi_i \neq \psi_j$ for all $i \neq j$.

Let $\mathscr{C} \subseteq \mathscr{R}$ be recursively enumerable. Then there is a numbering $\psi \in \mathscr{R}^2$ such that $\mathscr{C} = \mathscr{R}_\psi$ and ψ has a recursive equality problem. If $\mathscr{C}$ is infinite, then there is a 1–1 numbering $\psi \in \mathscr{R}^2$ such that $\mathscr{C} = \mathscr{R}_\psi$. For a proof we refer the reader to Kummer [20], Fact 1.

By φ we denote a standard acceptable (Gödel) numbering. Φ denotes a Blum complexity measure for φ (cf. [4]). We use $\downarrow$ to denote that a computation converges. Thus, $\varphi_i(x)\downarrow$ denotes that $\varphi_i(x)$ is defined.

3. Language learning by erasing

In this section our objects to be learned are indexed families of languages, i.e., recursively enumerable classes of uniformly recursive languages. Since the paper of Angluin [1] learning of indexed families of languages has attracted much attention (cf., e.g., [33]). Mainly, because most of the established language families such as regular languages, context-free languages, context-sensitive languages, or pattern languages are indexed families.

In the approaches of learning by erasing below we introduce the following possibilities for sets of hypotheses, which must or may be erased during the inference process:
- an arbitrary set of hypotheses may be erased,
- exactly all hypotheses less than the least correct one have to be erased,
- only incorrect hypotheses may be erased,
- exactly all incorrect hypotheses have to be erased,
- all incorrect hypotheses have to be erased and an arbitrary set of correct hypotheses may be erased, too,
- all but one hypothesis have to be erased.

We consider both modes of information presentation established in language learning, text (positive data, only) and informant (positive and negative data). And we study *class preserving learning* (the hypothesis spaces exactly enumerate the language family to be learned), *class comprising learning* (the hypothesis spaces enumerate a possibly proper superset of the family to be learned) and *absolute learning* (the families have to be learned with respect to *all* hypothesis spaces enumerating them exactly).

Our results can be classified along the lines of characterizations, comparisons inside, and comparisons with known types of language learning.

Characterizations: For all types of learning by erasing we present characterizations, i.e., conditions that are both necessary and sufficient for learnability in the corresponding sense. Often these characterizations are stated in terms being independent from learning theory. In several cases, the corresponding condition is a purely structural one, namely that the language family may not contain any language together with a proper sublanguage. In other cases, the characterization achieves the "granularity" of deriving necessary and sufficient learnability conditions for any given pair of a language family and a hypothesis space. Such granularity results were already derived in language learning theory (cf., e.g., [1, 2, 18, 22, 34]). Surprisingly, our characterizations do work without the explicit use of so-called "telltales" which were commonly used in most previous characterizations in language learning. Even more surprisingly, up to now no such granularity results are known in Gold's [14] paradigm of learning recursive functions. There the basic structure of most of the characterizations is the following. Given a class $\mathscr{C}$ of recursive functions and some learning type Lt; then $\mathscr{C}$ is Lt-learnable iff *there is a suitable* hypothesis space such that ... (cf. [32]). Note that also some characterizations in language learning have this "there is" flavor (cf., e.g., [17]).

Comparisons inside: We derive a complete picture containing all separations and coincidences of the types of learning by erasing defined. Fortunately, this picture is of a pretty regular structure and not as sophisticated as sometimes in inductive inference. Several of the separations follow from the characterizations above.

Comparisons with known types of language learning: We compare the types of language learning by erasing with well-known standard types of learning indexed language families such as learning in the limit, finite learning and conservative learning (or, equivalently, learning without overgeneralization, cf. [1, 18, 24, 25, 33]). We present the exact location of these established learning types in the hierarchy of the types of language learning by erasing.

3.1. Definitions

The class of all $\{0,1\}$ valued functions $f \in \mathscr{R}^n$ is denoted by $\mathscr{R}^n_{0,1}$; for $n=1$ we omit the upper index. For $\psi \in \mathscr{R}^2_{0,1}$, let $L(\psi_j)$ denote the language generated or described by ψ_j, i.e., $L(\psi_j) = \{x \mid \psi_j(x) = 1, x \in \mathbb{N}\}$. We call $\mathscr{L} = (L(\psi_j))_{j \in \mathbb{N}}$ an *indexed family* (cf. [1]). Then *range*$(\mathscr{L}) = \{L(\psi_j) \mid j \in \mathbb{N}\}$. We sometimes write $L \in \mathscr{L}$ instead of $L \in$ *range*$(\mathscr{L})$. For the sake of presentation, we restrict ourselves to consider exclusively indexed families of *non-empty* languages. Let $\mathscr{L}$ be an indexed family. Then $\mathscr{L}$ is said to be *inclusion-free* iff $L \not\subset \hat{L}$ for all languages $L, \hat{L} \in$ *range*$(\mathscr{L})$. Every numbering $\psi \in \mathscr{R}^2_{0,1}$ is called *hypothesis space*. A hypothesis space ψ is said to be *class comprising* for $\mathscr{L}$ iff $range(\mathscr{L}) \subseteq \{L(\psi_j) \mid j \in \mathbb{N}\}$. Furthermore, we call a hypothesis space ψ *class preserving* for $\mathscr{L}$ iff $range(\mathscr{L}) = \{L(\psi_j) \mid j \in \mathbb{N}\}$. For a hypothesis space ψ and a language L, we set $min_\psi(L) = \min\{j \mid L(\psi_j) = L\}$.

Let L be a language and let $t = s_0, s_1, s_2, \ldots$ be an infinite sequence of natural numbers such that $content(t) =_{df} \{s_k \mid k \in \mathbb{N}\} = L$. Then t is said to be a *text* for L or, synonymously, a *positive presentation*. Let $text(L)$ denote the set of all positive presentations of L, and let $text(\mathscr{L}) = \bigcup_{L \in \mathscr{L}} text(L)$. Moreover, let t be a text, and let $y \in \mathbb{N}$. Then t_y is the initial segment of t of length $y+1$, i.e., $t_y = s_0, \ldots, s_y$. Finally, t_y^+ denotes the *content* of t_y, i.e., $t_y^+ = \{s_z \mid z \leqslant y\}$.

Next, we recall the notion of the *canonical text* (cf. [22]) that turns out to be helpful in proving some characterizations. Let L be any non-empty recursive language, and let $0, 1, 2, \ldots$ be the ordered text of $\mathbb{N}$. The canonical text of L is obtained as follows. Test sequentially whether $z \in L$ for $z = 0, 1, 2, \ldots$ until the first z is found such that $z \in L$. Since $L \neq \emptyset$, there must be at least one z fulfilling the test. Set $t_0 = z$. We proceed inductively. For all $x \in \mathbb{N}$ we define

$$t_{x+1} = \begin{cases} t_x, z+x+1 & \text{if} \quad z+x+1 \in L, \\ t_x, n & \text{otherwise, where } n \text{ is the last element in } t_x. \end{cases}$$

An *inductive inference machine* (abbr. IIM) is an algorithmic mapping from initial segments of a text to $\mathbb{N} \cup \{?\}$. We interpret the hypotheses output by an IIM with respect to some hypothesis space ψ. When an IIM outputs a number j, we interpret it to mean that the machine is hypothesizing the language $L(\psi_j)$. The output "?" represents the case where the machine outputs "no conjecture".

Furthermore, we define an *erasing learning machine* (ELM) to be an algorithmic device working exactly as an IIM does. However, there is a major difference in the *semantics* of the output of an IIM and an ELM, respectively. Let $\psi \in \mathscr{R}^2_{0,1}$ be any hypothesis space. Suppose an ELM M has been successively fed an initial segment t_y of a text t, and it has output numbers $j_0, \ldots, j_z$. Then we interpret $j = \min(\mathbb{N} \setminus \{j_0, \ldots, j_z\})$ as M's *actual guess*. Intuitively, if an ELM outputs a number j, then it *definitely deletes* j from its list of potential hypotheses. For an ELM M, a text t and $y \in \mathbb{N}$, let $\mathrm{ProgSet}(M, t_y)$ be the set of all numbers output by M when successively fed t_y, and let $\mathrm{ProgSet}(M, t)$ be the overall set of numbers output by M on text t.

We define convergence of IIMs as usual. Let t be a text, and let M be an IIM. The sequence $(M(t_y))_{y \in \mathbb{N}}$ is said to *converge* to a number j iff all but finitely many terms of $(M(t_y))_{y \in \mathbb{N}}$ are equal to j.

An ELM M is said to *stabilize* to a number j on a text t iff its sequence of actual guesses converges to j, i.e., $j = \min(\mathbb{N} \setminus \mathrm{ProgSet}(M, t))$.

Now we are ready to define learning and learning by erasing.

Definition 1 (Gold [14]). Let $\mathscr{L}$ be an indexed family, let L be a language, and let $\psi \in \mathscr{R}^2_{0,1}$ be a hypothesis space. An IIM M *$CExTxt_\psi$-infers* L iff for every $t \in text(L)$, there exists a $j \in \mathbb{N}$ with $L = L(\psi_j)$ such that the sequence $(M(t_y))_{y \in \mathbb{N}}$ converges to j.

M *$CExTxt_\psi$-infers* $\mathscr{L}$ iff *M* *$CExTxt_\psi$-infers* *L* for each $L \in range(\mathscr{L})$.

Let $CExTxt_\psi$ denote the collection of all indexed families $\mathscr{L}$ for which there is an IIM *M* such that *M* *$CExTxt_\psi$-infers* $\mathscr{L}$.

Finally, *CExTxt* denotes the collection of all indexed families $\mathscr{L}$ for which there are an IIM *M* and a hypothesis space ψ such that *M* *$CExTxt_\psi$-infers* $\mathscr{L}$.

Since, by the definition of convergence, an IIM has only seen a finite amount of data about *L* until the (unknown) point of convergence is reached, whenever an IIM infers the language *L*, some form of learning must have taken place. For this reason, hereinafter the terms *infer*, *learn*, and *identify* are used interchangeably.

In Definition 1 the prefix *C* is used to indicate *class comprising* learning, i.e., the fact that $\mathscr{L}$ may be learned with respect to some class comprising hypothesis space ψ for $\mathscr{L}$. The restriction of *CExTxt* to class preserving hypothesis spaces is denoted by *ExTxt* and referred to as *class preserving* inference. Moreover, we use the prefix *A* to express the fact that an indexed family $\mathscr{L}$ must be inferred with respect to *all* class preserving hypothesis spaces for $\mathscr{L}$, and we refer to this learning model as to *absolute* learning. We adopt these conventions in the definitions of the learning types below.

The following proposition states that, if there is a hypothesis space ψ such that an indexed family $\mathscr{L}$ can be $CExTxt_\psi$-learned, then it can be *ExTxt*-inferred with respect to *every* class preserving hypothesis space for $\mathscr{L}$.

Proposition 1 (Lange and Zeugmann [25]). $AExTxt = ExTxt = CExTxt$.

Note that, in general, it is not decidable whether or not an IIM *M* has already converged on a text *t* for the target language *L*. With the next definition, we consider a special case where it has to be decidable whether or not an IIM has successfully finished the learning task.

Definition 2 (Gold [14]; Trakhtenbrot and Barzdin [31]). Let $\mathscr{L}$ be an indexed family, let *L* be a language, and let $\psi \in \mathscr{R}^2_{0,1}$ be a hypothesis space. An IIM *M* *$CFinTxt_\psi$-infers* *L* iff for every $t \in text(L)$, there exist $j, z \in \mathbb{N}$ such that $L = L(\psi_j)$, $M(t_y) = ?$ for all $y < z$, and $M(t_y) = j$ for all $y \geqslant z$.

M *$CFinTxt_\psi$-infers* $\mathscr{L}$ iff *M* *$CFinTxt_\psi$-infers* *L* for each $L \in range(\mathscr{L})$.

Finally, $CFinTxt_\psi$ and *CFinTxt* are defined analogously as above.

The analogue to Proposition 1 also holds for finite learning.

Proposition 2 (Zeugmann et al. [34]). $AFinTxt = FinTxt = CFinTxt$.

Now, we define *conservative* IIMs. Conservative IIMs maintain their actual hypothesis at least as long as they have not received data that "provably misclassify" it. Hence, whenever a conservative IIM performs a mind change it is because it has perceived a clear contradiction between its hypothesis and the actual input.

Definition 3 (Angluin [1]). Let $\mathscr{L}$ be an indexed family, let L be a language, and let $\psi \in \mathscr{R}^2_{0,1}$ be a hypothesis space. An IIM M *$CConsvTxt_\psi$-infers* L iff
(1) M *$CExTxt_\psi$-infers* L,
(2) for all $t \in text(L)$ and all $y, k \in \mathbb{N}$ such that $M(t_y), M(t_{y+k}) \neq ?$, if $M(t_y) \neq M(t_{y+k})$ then $t^+_{y+k} \not\subseteq L(\psi_{M(t_y)})$.
M *$CConsvTxt_\psi$-infers* $\mathscr{L}$ iff M *$CConsvTxt_\psi$-infers* L for all $L \in range(\mathscr{L})$.
$CConsvTxt_\psi$ and *$CConsvTxt$* are defined analogously to Definition 1.

The following proposition shows that conservative learning is sensitive to the particular choice of the hypothesis space.

Proposition 3 (Lange and Zeugmann [24]). *$AConsvTxt \subset ConsvTxt \subset CConsvTxt \subset AExTxt$.*

Next, we define learning by erasing.

Definition 4. Let $\mathscr{L}$ be an indexed family, let L be a language, and let $\psi \in \mathscr{R}^2_{0,1}$ be a hypothesis space. An ELM M *$CArbTxt_\psi$-infers* L iff for every $t \in text(L)$, there exists a $j \in \mathbb{N}$ with $L = L(\psi_j)$ such that M on t stabilizes to j.
M *$CArbTxt_\psi$-infers* $\mathscr{L}$ iff M *$CArbTxt_\psi$-infers* L for each $L \in range(\mathscr{L})$.
$CArbTxt_\psi$ denotes the collection of all indexed families $\mathscr{L}$ for which there is an ELM M such that M *$CArbTxt_\psi$-infers* $\mathscr{L}$.
Finally, let *$CArbTxt$* denote the collection of all indexed families $\mathscr{L}$ for which there are an ELM M and a hypothesis space ψ such that M *$CArbTxt_\psi$-infers* $\mathscr{L}$.

Definition 5. Let $\mathscr{L}$ be an indexed family, let L be a language, and let $\psi \in \mathscr{R}^2_{0,1}$ be a hypothesis space. An ELM M is said to
(A) *$CMinTxt_\psi$-infer* L,
(B) *$CSubTxt_\psi$-infer* L,
(C) *$CEqualTxt_\psi$-infer* L,
(D) *$CSuperTxt_\psi$-infer* L,
(E) *$CAllTxt_\psi$-infer* L,
iff M *$CArbTxt_\psi$-infers* L and for each $t \in text(L)$, the following corresponding condition is satisfied:
(A) $\mathrm{ProgSet}(M,t) = \{j \mid j < min_\psi(L),\ j \in \mathbb{N}\}$, i.e., M has to erase exactly all hypotheses prior to the least correct index for L;
(B) $\mathrm{ProgSet}(M,t) \subseteq \{j \mid L(\psi_j) \neq L,\ j \in \mathbb{N}\}$, i.e., M is only allowed to erase hypotheses that are incorrect for L;
(C) $\mathrm{ProgSet}(M,t) = \{j \mid L(\psi_j) \neq L,\ j \in \mathbb{N}\}$, i.e., M has to erase exactly all hypotheses that are incorrect for L;
(D) $\mathrm{ProgSet}(M,t) \supseteq \{j \mid L(\psi_j) \neq L,\ j \in \mathbb{N}\}$, i.e., M has to erase all hypotheses that are incorrect for L but it may additionally erase correct hypotheses for L;
(E) $|\mathbb{N} \setminus \mathrm{ProgSet}(M,t)| = 1$ i.e., M has to erase all but one hypothesis.

Finally, $CMinTxt_\psi$, $CSubTxt_\psi$, $CEqualTxt_\psi$, $CSuperTxt_\psi$, and $CAllTxt_\psi$ as well as $CMinTxt$, $CSubTxt$, $CEqualTxt$, $CSuperTxt$, and $CAllTxt$ are defined analogously to Definition 4.

In order to study learning by erasing from both positive and negative data we have to introduce some more notations and definitions. Let $L \subseteq \mathbb{N}$ be a language, and let $i = (s_0, b_0), (s_1, b_1), \dots$ be an infinite sequence of elements of $\mathbb{N} \times \{+, -\}$ such that $content(i) =_{df} \{s_k \mid k \in \mathbb{N}\} = \mathbb{N}$, $i^+ =_{df} \{s_k \mid b_k = +,\ k \in \mathbb{N}\} = L$ and $i^- =_{df} \{s_k \mid b_k = -,\ k \in \mathbb{N}\} = \mathbb{N} \setminus L$. Then we refer to i as an *informant* for L. If L is classified via an informant then we also say that L is represented by *positive and negative data*. By $info(L)$ we denote the set of all informants for L. We use i_x to denote the initial segment of i of length $x+1$, and define $i_x^+ = \{s_k \mid b_k = +,\ k \leqslant x\}$ and $i_x^- = \{s_k \mid b_k = -,\ k \leqslant x\}$. ProgSet$(M, i)$ where i is an informant is defined analogously as ProgSet(M, t) where t is a text. Furthermore, $CExInf$ and $CFinInf$ are defined analogously as in Definitions 1 and 2, respectively, by replacing everywhere text by informant. Finally, we extend all the definitions of learning by erasing in the same way, and denote the resulting learning types by $CLtInf$ for all $Lt \in \{Arb, Min, Sub, Equal, Super, All\}$.

Fig. 1 summarizes most of the relations between the learning types studied in Section 3. It may also serve as a kind of map for the reader. Each learning type is represented as a vertex in a directed graph. A directed edge (or path) from vertex A to vertex B indicates that A is a proper subset of B. Finally, Lt stands for Arb, Min, Sub, $Equal$ and $Super$, respectively, and λ stands for A, ε and C, respectively. Note that $FinInf \subset ConsvTxt$ also holds, cf. Proposition 7 below, which is not indicated by an arrow in Fig. 1.

3.2. Characterizations

In this section we present characterizations of all the models of learning by erasing. These characterizations may help to better understand what these models have in common and what their differences are. Note that there will be two kinds of characterizations. On the one hand, we present characterizations of learning types in terms being independent of learning theory (cf. Theorems 1, 4–6 and Proposition 4). On the other hand, we characterize learning types by showing that they coincide with other learning types. The latter approach is also technically useful for the remainder of Section 3 in that it allows to prove results only for one representative of each group of coinciding learning types.

Our first result characterizes $EqualTxt$ in purely structural terms.

Theorem 1. *For any indexed family $\mathcal{L}$, $\mathcal{L} \in EqualTxt$ iff $\mathcal{L}$ is inclusion-free.*

Proof. *Necessity*: Let $\mathcal{L} \in EqualTxt$. Hence $\mathcal{L} \in CEqualTxt$. Consequently, $\mathcal{L}$ is inclusion-free by Claim A below.

Claim A. *For every indexed family $\mathcal{L}$, if $\mathcal{L} \in CEqualTxt$, then $\mathcal{L}$ is inclusion-free.*

$$\begin{tikzcd}
 & \mathit{ExInf} = \mathit{ALtInf} = \mathit{AllInf} & \\
 & \mathit{ExTxt} = \mathit{ArbTxt} = \mathit{SuperTxt} = \mathit{AllTxt} \arrow[u] & \\
 & \mathit{CSubTxt} = \mathit{CMinTxt} \arrow[u] & \mathit{CConsvTxt} \arrow[ul] \\
 & & \mathit{ConsvTxt} \arrow[u] \\
 & \mathit{SubTxt} = \mathit{MinTxt} \arrow[uu] \arrow[ur] & \mathit{AConsvTxt} \arrow[u] \\
\mathit{AAllInf} \arrow[uuuur] & \lambda\mathit{EqualTxt} = \mathit{ALtTxt} \arrow[u] \arrow[ur] & \\
\mathit{FinInf} \arrow[u] & \mathit{AAllTxt} \arrow[u] \arrow[ul] & \\
 & \mathit{FinTxt} \arrow[u] \arrow[ul] &
\end{tikzcd}$$

Fig. 1. Summary.

Let $\mathscr{L}$ be any indexed family, let ψ be any class comprising hypothesis space for $\mathscr{L}$, and let M be any ELM witnessing $\mathscr{L} \in \mathit{CEqualTxt}_{\psi}$. Suppose that there are $L, \hat{L} \in \mathscr{L}$ with $L \subset \hat{L}$. Let $t \in \mathit{text}(L)$. Since M $\mathit{CEqualTxt}_{\psi}$-learns L, on successive input t, M has to delete sometimes a ψ-index for $\hat{L}$, i.e., there is a least $y \in \mathbb{N}$ such that $M(t_y) = j$ with $L(\psi_j) = \hat{L}$. Because of $L \subset \hat{L}$, t_y can be extended to a text $\hat{t} \in \mathit{text}(\hat{L})$. Moreover, $\hat{L} \in \mathscr{L}$ and thus M must $\mathit{CEqualTxt}_{\psi}$-identify $\hat{L}$. However, M, when fed the initial segment $\hat{t}_y$, outputs a correct ψ-index for $\hat{L}$, a contradiction.

Sufficiency: Clearly, it suffices to prove the following Claim B.

Claim B. *For any indexed family $\mathscr{L}$, if $\mathscr{L}$ is inclusion-free, then $\mathscr{L} \in \mathit{AEqualTxt}$.*

Let $\psi \in \mathscr{R}^2_{0,1}$ be any class preserving hypothesis space for any inclusion-free indexed family $\mathscr{L}$. Choose an ELM M that meets $\mathrm{ProgSet}(M,t) = \{j \mid (\exists y)[t_y^+ \not\subseteq L(\psi_j)]\}$ for all $t \in \mathit{text}(\mathscr{L})$. By construction, M, when fed a text t for $L \in \mathscr{L}$, never outputs a correct ψ-index for L. On the other hand, M eventually outputs all incorrect ψ-indices for L. Actually, if $L(\psi_j) \neq L$ for some $j \in \mathbb{N}$, then $L \not\subseteq L(\psi_j)$, since $\mathscr{L}$ is inclusion-free. Hence, $t_y^+ \not\subseteq L(\psi_j)$ for some $y \in \mathbb{N}$, and j is erased by M. Consequently, M $\mathit{EqualTxt}_{\psi}$-identifies L. $\square$

Claims A and B above immediately yield the following corollary.

Corollary 2. *$\mathit{AEqualTxt} = \mathit{EqualTxt} = \mathit{CEqualTxt}$.*

Furthermore, Theorem 1 and Corollary 2 can be exploited to characterize *AArbTxt*, *ASubTxt* and *ASuperTxt* as well.

Theorem 3. *For all $Lt \in \{Arb, Sub, Super\}$, $ALtTxt = AEqualTxt$.*

Proof. By Definitions 4 and 5, we have $AEqualTxt \subseteq ASubTxt \subseteq AArbTxt$ and $AEqualTxt \subseteq ASuperTxt \subseteq AArbTxt$. Hence, it suffices to show that $AArbTxt \subseteq AEqualTxt$. But this follows from the claim below via Claim B from the proof of Theorem 1.

Claim. For any indexed family $\mathscr{L}$, if $\mathscr{L} \in AArbTxt$, then $\mathscr{L}$ is inclusion-free.

Suppose to the contrary that there are $L, \hat{L} \in \mathscr{L}$ with $L \subset \hat{L}$. Now, choose any class preserving hypothesis space ψ for $\mathscr{L}$ such that $L(\psi_0) = \hat{L}$ and $L(\psi_j) \neq \hat{L}$ for all $j > 0$. Clearly, such a hypothesis space always exists. By assumption, $\mathscr{L} \in AArbTxt$, and hence there is an ELM M which $ArbTxt_\psi$-identifies $\mathscr{L}$. Now, let t be any text for L. Then, $M(t_x) = 0$ for some $x \in \mathbb{N}$, since, otherwise, M would stabilize on t to 0, but $L(\psi_0) \neq L$. Choose any text $\hat{t}$ for $\hat{L} \supset L$ with the initial segment t_x. Obviously, on $\hat{t}$, M deletes the one and only ψ-index for $\hat{L}$, a contradiction. □

For characterizing $AAllTxt$ we have to combine the structural approach with the numbering theoretical one used by Kummer [20].

Theorem 4. *For any indexed family $\mathscr{L}$, $\mathscr{L} \in AAllTxt$ iff*
(1) *$\mathscr{L}$ is inclusion-free, and*
(2) *every class preserving hypothesis space for $\mathscr{L}$ has a recursive equality problem.*

Proof. *Necessity*: Let $\mathscr{L} \in AAllTxt$. By definition, $\mathscr{L} \in AArbTxt$, and thus $\mathscr{L}$ is inclusion-free (cf. the claim in the proof of Theorem 3). On the other hand, $AAllTxt \subseteq AAllInf$. Kummer [20] has shown that $\mathscr{L} \in AAllInf$ iff every class preserving hypothesis space for $\mathscr{L}$ has a recursive equality problem (cf. Proposition 4 below). Hence, we are done.

Sufficiency: Let $\mathscr{L}$ be any inclusion-free indexed family, and let ψ be any class preserving hypothesis space for $\mathscr{L}$. Then, by (2), ψ has a recursive equality problem. Define an ELM M such that ProgSet(M, t) contains all and only the ψ-indices j satisfying (i) or (ii), where
(i) $t_y^+ \not\subseteq L(\psi_j)$ for some $y \in \mathbb{N}$,
(ii) $\psi_k = \psi_j$ for some $k < j$.
Consider M when fed any $t \in text(L)$ for some $L \in \mathscr{L}$. By clause (i), M eventually erases all incorrect ψ-indices for L, since $\mathscr{L}$ is inclusion-free, and therefore $L(\psi_j) \neq L$ implies $L \not\subseteq L(\psi_j)$. Moreover, clause (ii) guarantees that there is exactly one correct ψ-index for L that never will be erased, namely the minimal one. Hence, M $AllTxt_\psi$-learns L. □

Next, we characterize $CSubTxt$ and $SubTxt$. Now, we derive necessary and sufficient conditions for any given pair of an indexed family and a hypothesis space for it. Again, the characterization is mainly based on structural properties of the relevant hypothesis spaces. However, we have to add a component of computability to these structural properties. Within the next definition we provide the necessary framework for establishing the desired characterization theorems.

Definition 6. Let $\mathscr{L}$ be any indexed family, and let ψ be any class comprising hypothesis space for $\mathscr{L}$. Then we set:
(1) $Bad(\mathscr{L},\psi)=\{j \mid (\exists L\in range(\mathscr{L}))[L\subset L(\psi_j)\wedge j < min_\psi(L)]\}$,
(2) $Wrong(\mathscr{L},\psi)=\{j \mid L(\psi_j)\notin range(\mathscr{L})\}$.

Theorem 5. *For any indexed family $\mathscr{L}$ and any class comprising hypothesis space ψ for $\mathscr{L}$, $\mathscr{L}\in CSubTxt_\psi$ iff $Bad(\mathscr{L},\psi)\subseteq W\subseteq Wrong(\mathscr{L},\psi)$ for some recursively enumerable set W.*

Proof. *Necessity*: Let $\mathscr{L}$ be any indexed family, let ψ be any class comprising hypothesis space for $\mathscr{L}$, and let M be any ELM which $CSubTxt_\psi$-learns $\mathscr{L}$.

Next, we use M to define $f\in\mathscr{P}$ such that $W=range(f)$. For every $k\in\mathbb{N}$, let t^k denote the canonical text for the language $L(\psi_k)$. For every $k,x\in\mathbb{N}$, we set:

$$f(\langle k,x\rangle)=\begin{cases} M(t^k_x) & \text{if } content(t^k_x)\subseteq L(\psi_{M(t^k_x)}),\\ \text{not defined} & \text{otherwise.}\end{cases}$$

Using the convention that, if $M(t^k_x)=?$ then $f(\langle k,x\rangle)$ is not defined, we obviously have $f\in\mathscr{P}$. It remains to show that $Bad(\mathscr{L},\psi)\subseteq W\subseteq Wrong(\mathscr{L},\psi)$.
Claim A. $W\subseteq Wrong(\mathscr{L},\psi)$.

If $W=\emptyset$, we are done. Now, let $z=f(\langle k,x\rangle)$ for some $k,x\in\mathbb{N}$. By definition of f, we have $M(t^k_x)=z$ and $content(t^k_x)\subseteq L(\psi_z)$. Suppose, $L(\psi_z)\in\mathscr{L}$. Since $content(t^k_x)\subseteq L(\psi_z)$, t^k_x is an initial segment of some text $\hat{t}$ for $L(\psi_z)$. Thus M, when fed the text $\hat{t}$ for $L(\psi_z)\in\mathscr{L}$, outputs the correct ψ-index z for $L(\psi_z)$. This contradicts our assumption that M *$CSubTxt_\psi$-infers* $\mathscr{L}$. Thus, Claim A follows.
Claim B. $Bad(\mathscr{L},\psi)\subseteq W$.

Suppose the converse, i.e., there is a $z\in Bad(\mathscr{L},\psi)\setminus W$. Hence, $z < min_\psi(L)$ for some $L\in\mathscr{L}$ with $L\subset L(\psi_z)$. Let k be any ψ-index for L. Consider M when fed the canonical text t^k for L. Since M $CSubTxt_\psi$-identifies L, M must stabilize on t^k to $min_\psi(L)$. Because of $z < min_\psi(L)$, there has to be an $x\in\mathbb{N}$ with $M(t^k_x)=z$. Thus, $f(\langle k,x\rangle)=z$, and hence $z\in W$, a contradiction. Thus, Claim B follows, and we are done.

Sufficiency: Let $\mathscr{L}$ be any indexed family, let ψ be any class comprising hypothesis space for $\mathscr{L}$, and let W be any recursively enumerable set with $Bad(\mathscr{L},\psi)\subseteq W\subseteq Wrong(\mathscr{L},\psi)$. Define an ELM M such that ProgSet(M,t) contains exactly the ψ-indices j with (i) or (ii), where
(i) $t^+_y\nsubseteq L(\psi_j)$ for some $y\in\mathbb{N}$,
(ii) $j\in W$.
Let $L\in\mathscr{L}$ and $t\in text(L)$. By construction, on input t, M never outputs a correct ψ-index for L. Moreover, M stabilizes to $min_\psi(L)$. Actually, for any $j < min_\psi(L)$, either $L\nsubseteq L(\psi_j)$ or $L\subset L(\psi_j)$. In the former case, j is erased by (i). In the latter case, j is erased by (ii), since $j\in Bad(\mathscr{L},\psi)\subseteq W$. Hence, M $CSubTxt_\psi$-learns $\mathscr{L}$. $\square$

Theorem 6. *For any indexed family $\mathscr{L}$ and any class preserving hypothesis space ψ for $\mathscr{L}$, $\mathscr{L} \in SubTxt_\psi$ iff $Bad(\mathscr{L},\psi) = \emptyset$.*

Proof. Recall that $Wrong(\mathscr{L},\psi) = \emptyset$ for any class preserving hypothesis space ψ for $\mathscr{L}$. Hence, Theorem 6 follows immediately from Theorem 5. □

We now prove the equivalence of all corresponding variants of *Min*-learning by erasing and *Sub*-learning by erasing from text.

Theorem 7. *For all $\lambda \in \{A, \varepsilon, C\}$, $\lambda MinTxt = \lambda SubTxt$.*

Proof. By definition, $\lambda MinTxt \subseteq \lambda SubTxt$ for all $\lambda \in \{A, \varepsilon, C\}$. The converse easily follows by transforming any given ELM M into an ELM $\hat{M}$ such that for every text t, $\mathrm{ProgSet}(\hat{M},t) = \{j \mid (\forall i \leqslant j)[i \in \mathrm{ProgSet}(M,t)]\}$. □

From Theorem 7 it follows immediately that for *CMinTxt* and *MinTxt*, the characterizations from Theorems 5 and 6, respectively, also apply. Moreover, since $AMinTxt = ASubTxt = AEqualTxt = EqualTxt$ by Theorem 7, Theorem 3 and Corollary 2, the characterization of Theorem 1 is also valid for *AMinTxt*.
Next, we characterize the remaining models of learning by erasing from text.

Theorem 8. *For all $Lt \in \{All, Super, Arb\}$, $LtTxt = ExTxt$.*

Proof. By definition, $AllTxt \subseteq SuperTxt \subseteq ArbTxt$. Since $ExTxt \subseteq AllTxt$ (cf. [13, Theorem 3]), it suffices to show that $ArbTxt \subseteq ExTxt$.
Claim. For any indexed family $\mathscr{L}$ and for any class comprising hypothesis space $\psi \in \mathscr{R}^2_{0,1}$ for $\mathscr{L}$, if $\mathscr{L} \in CArbTxt_\psi$ then $\mathscr{L} \in CExTxt_\psi$.

Let M be an ELM witnessing $\mathscr{L} \in CArbTxt_\psi$. Let an IIM $\hat{M}$ always output the least ψ-number not yet definitely deleted by M. Obviously, $\hat{M}$ $CExTxt_\psi$-learns $\mathscr{L}$.
Since the claim above especially holds for any class preserving hypothesis space, we obtain $ArbTxt \subseteq ExTxt$, and the theorem follows. □

Proposition 1 and the claim in the proof of Theorem 8 above directly allow the following corollary.

Corollary 9. *For all $Lt \in \{All, Super, Arb\}$, $CLtTxt = ExTxt$.*

Finally, we derive characterizations for language learning by erasing from informant.

Theorem 10. *For all $Lt \in \{Arb, Min, Sub, Equal, Super\}$, $ALtInf = LtInf = CLtInf = ExInf$.*

Proof. First, we prove that every indexed family belongs to both *AEqualInf* and *AMinInf*.

Claim A. *For any indexed family $\mathscr{L}$, $\mathscr{L} \in AEqualInf$.*

Let $\psi \in \mathscr{R}^2_{0,1}$ be any class preserving hypothesis space for $\mathscr{L}$. Define an ELM M which deletes every $j \in \mathbb{N}$ such that $L(\psi_j)$ is inconsistent with the given informant i, i.e., $\text{ProgSet}(M,i) = \{j \mid (\exists y)[i^+_y \not\subseteq L(\psi_j) \vee i^-_y \not\subseteq \mathbb{N} \setminus L(\psi_j)]\}$. By construction, M never outputs a correct ψ-index for L. Moreover, M eventually deletes all incorrect ψ-indices for L.

Claim B. *For any indexed family $\mathscr{L}, \mathscr{L} \in AMinInf$.*

Let $\psi \in \mathscr{R}^2_{0,1}$ be any class preserving hypothesis space for $\mathscr{L}$. By Claim A above, there is an ELM M that $EqualInf_\psi$-learns $\mathscr{L}$. Clearly, an ELM M' $MinInf_\psi$-learns $\mathscr{L}$ provided that $\text{ProgSet}(M',i) = \{j \mid (\forall k \leqslant j)[k \in \text{ProgSet}(M,i)]\}$.

By definition, $AEqualInf \subseteq ALtInf$ for all $Lt \in \{Arb, Sub, Super\}$, and thus, by Claim A, every indexed family is contained in $ALtInf$, too. On the other hand, every indexed family belongs to $ExInf$ (cf. [14]). Finally, taking into account that $ALtInf \subseteq LtInf \subseteq CLtInf$ for any learning type $Lt \in \{Arb, Sub, Equal, Super, Min\}$, the theorem directly follows. $\square$

The remaining characterizations, namely for $AAllInf$, $AllInf$ and $CAllInf$, can be found already in the literature.

Proposition 4 (Kummer [20]). *For any indexed family $\mathscr{L}$, $\mathscr{L} \in AAllInf$ iff every class preserving hypothesis space for $\mathscr{L}$ has a recursive equality problem.*

Proposition 5 (Freivalds et al. [10]). *$AllInf = ExInf$.*

The latter result immediately yields $CAllInf = ExInf$, since $ExInf$ already contains any indexed family (cf. Gold [14]).

3.3. Learning from text

In this subsection, we compare the capabilities of all the types of learning from positive data by erasing to one another as well as to finite inference, learning in the limit and conservative identification from text. Recall that several coincidences of the corresponding types were already exhibited in the previous subsection (cf. Theorems 3, 7, 8 and Corollaries 2, 9). Hence, we now confine ourselves to derive mainly proper inclusion and incomparability results. Thereby, we also analyze the power of learning by erasing in dependence on the set of admissible hypothesis spaces. In Sections 3.3.1–3.3.3 we are dealing with class preserving, class comprising and absolute learning, respectively.

3.3.1. Class preserving learning

Since $AllTxt = SuperTxt = ArbTxt = ExTxt$ by Theorem 8 and $MinTxt = SubTxt$ by Theorem 7, it remains to clarify the power of $EqualTxt$ and $SubTxt$ as well as to compare these types with both $FinTxt$ and $ConsvTxt$. This will be done by the following result.

Theorem 11. *FinTxt* $\subset$ *EqualTxt* $\subset$ *SubTxt* $\subset$ *ConsvTxt*.

Proof. *Claim* A. *FinTxt* $\subset$ *EqualTxt*.

Let $\mathscr{L} \in FinTxt$ and let $\psi \in \mathscr{R}^2_{0,1}$ be a class preserving hypothesis space for $\mathscr{L}$. By Proposition 2, there exists an IIM M witnessing $\mathscr{L} \in FinTxt_\psi$. To $EqualTxt_\psi$-learn $\mathscr{L}$ simulate M and, as soon as M outputs its first (and correct) guess, say j, erase all i with $\psi_i \neq \psi_j$.

In order to separate *FinTxt* and *EqualTxt* consider the indexed family $\mathscr{L} = (L_j)_{j\in\mathbb{N}}$ with $L_j = \mathbb{N}\setminus\{j\}$ for all $j \in \mathbb{N}$. Note that $\mathscr{L} \notin FinTxt$ (cf. [13]). Since $\mathscr{L}$ is inclusion-free, $\mathscr{L} \in EqualTxt$ (cf. Theorem 1).

Claim B. *EqualTxt* $\subset$ *SubTxt*.

Since, by definition, $EqualTxt \subseteq SubTxt$, it suffices to show that $SubTxt\setminus EqualTxt \neq \emptyset$. For all $j \in \mathbb{N}$, let $L_j = \{0,\dots,j\}$, and set $\mathscr{L} = (L_j)_{j\in\mathbb{N}}$.

First, we verify that $\mathscr{L} \in SubTxt$. Choose the hypothesis space $\psi \in \mathscr{R}^2_{0,1}$ with $L(\psi_j) = L_j$ for all $j \in \mathbb{N}$. Define an ELM M such that $\text{ProgSet}(M,t) = \{j \mid (\exists y)[t^+_y \nsubseteq L(\psi_j)]\}$ for all $t \in text(\mathscr{L})$. Clearly, when fed any $t \in text(\mathscr{L})$, all hypotheses erased by M are incorrect. Moreover, M stabilizes to the only ψ-index for the language to be learned. Thus, M $SubTxt_\psi$-identifies $\mathscr{L}$.

Since $\mathscr{L}$ is not inclusion-free, we have $\mathscr{L} \notin EqualTxt$ by Theorem 1, and Claim B follows.

Claim C. *SubTxt* $\subset$ *ConsvTxt*.

Let $L_0 = \mathbb{N}$ and for all $j \in \mathbb{N}$, let $L_{j+1} = \{j\}$ as well as $\mathscr{L} = (L_j)_{j\in\mathbb{N}}$. We claim that $\mathscr{L} \in ConsvTxt\setminus SubTxt$. Obviously, $\mathscr{L} \in ConsvTxt_{\mathscr{L}}$. Suppose $\mathscr{L} \in SubTxt$. Thus, there are a class preserving hypothesis space ψ and an ELM M witnessing $\mathscr{L} \in SubTxt_\psi$. Let $k = min_\psi(L_0)$. Since $range(\mathscr{L})$ is infinite, there must be an $L \in \mathscr{L}$ such that $min_\psi(L) > k$. Thus, $Bad(\mathscr{L},\psi) \neq \emptyset$, and, by Theorem 6, $\mathscr{L} \notin SubTxt_\psi$, a contradiction.

It remains to prove $SubTxt \subseteq ConsvTxt$. Let ψ be some class preserving hypothesis space ψ such that $\mathscr{L} \in SubTxt_\psi$. Then, for all $L \in \mathscr{L}$, and all $j \in \mathbb{N}$, $j < min_\psi(L)$ implies $L \not\subset L(\psi_j)$, since $Bad(\mathscr{L},\psi) = \emptyset$ by Theorem 6. The desired conservative IIM M uses the hypothesis space ψ and is defined as follows. On a possible input t_x, M outputs the least j with $t^+_x \subseteq L(\psi_j)$. By construction, if $M(t_x), \neq M(t_{x+z})$ for some $z \in \mathbb{N}$, then $t^+_{x+z} \nsubseteq L(\psi_{M(t_x)})$; thus M is conservative. Let $L \in \mathscr{L}$, $t \in text(L)$, and $k = min_\psi(L)$. Since $L \not\subset L(\psi_j)$ for all $j < k$, every such j must be abandoned eventually. Thus, M converges to k. This proves Claim C. $\square$

3.3.2. *Class comprising learning*

Taking into account that $CEqualTxt = EqualTxt$ by Corollary 2, $CLtTxt = ExTxt$ for all $Lt \in \{All,\ Super,\ Arb\}$ by Corollary 9, and $CMinTxt = CSubTxt$ by Theorem 7, it remains to investigate the learning power of the learning type *CSubTxt*, only.

Theorem 12. (1) *SubTxt* $\subset$ *CSubTxt* $\subset$ *ExTxt*.

(2) *CSubTxt* # *CConsvTxt*.

Proof. *Claim* A. $ConsvTxt \setminus CSubTxt \neq \emptyset$.

One easily verifies that the indexed family $\mathscr{L}$ used in the proof of Theorem 11, Claim C, separates $ConsvTxt$ and $CSubTxt$, too, and thus Claim A follows.
Since $ConsvTxt \subset CConsvTxt \subset ExTxt$ (cf. Proposition 3), Claim A yields both $CConsvTxt \setminus CSubTxt \neq \emptyset$ and $ExTxt \setminus CSubTxt \neq \emptyset$. By definition and by Corollary 9, $CSubTxt \subseteq CArbTxt = ExTxt$. Hence $CSubTxt \subset ExTxt$. The following claim provides us the remaining part of Assertions (1) and (2).
Claim B. $CSubTxt \setminus CConsvTxt \neq \emptyset$.

Note that the verification of this claim is based on a technique from Lange and Zeugmann [24]. First, we define the desired indexed family $\mathscr{L}$ witnessing the claimed separation. For the sake of presentation, we describe $\mathscr{L}$ as a family of languages over the alphabet $\Sigma = \{a, b, c\}$. Note that, by convention, $x^0 = \varepsilon$ for $x \in \Sigma$.

Subsequently, we assume that $\Phi_k(k) \geqslant 1$ for all $k \in \mathbb{N}$. For all $k, j \in \mathbb{N}$, we set:

$$L_{\langle k,j\rangle} = \begin{cases} \{a^k b^m \mid m \leqslant j - \Phi_k(k)\} & \text{if} \quad \Phi_k(k) \leqslant j \leqslant 2\Phi_k(k), \\ \{a^k b^m \mid m \in \mathbb{N}\} & \text{otherwise.} \end{cases}$$

Finally, let $\mathscr{L} = (L_{\langle k,j\rangle})_{k,j \in \mathbb{N}}$. Since $\mathscr{L} \notin CConsvTxt$ (cf. Theorem 1 in [24]), it suffices to show that $\mathscr{L} \in CSubTxt$.

The desired ELM M uses the following class comprising hypothesis space $\mathscr{H} = (H_{\langle k,j\rangle})_{k,j \in \mathbb{N}}$. For all j, $k \in \mathbb{N}$, we set:

$$H_{\langle k,j\rangle} = \begin{cases} L_{\langle k,j\rangle} \cup \{a^k c^{\Phi_k(k)}\} & \text{if not } \Phi_k(k) \leqslant j, \\ L_{\langle k,j\rangle} & \text{otherwise.} \end{cases}$$

By definition, $\mathscr{H}$ serves as a class comprising hypothesis space for $\mathscr{L}$. Note that, by convention, $L_{\langle k,j\rangle} \cup \{a^k c^{\Phi_k(k)}\}$ equals $L_{\langle k,j\rangle}$, if $\varphi_k(k)$ is undefined. On the other hand, if $\varphi_k(k)$ is defined, then the languages $H_{\langle k,j\rangle}$ with $j < \Phi_k(k)$ are clearly out of $range(\mathscr{L})$. Therefore this "comprising part"of the space $\mathscr{H}$ can be erased "without risk" by a machine $CSubTxt_{\mathscr{H}}$-learning $\mathscr{L}$. On any $t \in text(\mathscr{L})$, such an ELM M works as follows.

***ELM** M*: "On input text t determine the unique k such that $content(t) \subseteq \{a^k b^n \mid n \in \mathbb{N}\}$. Erase all $i \in \mathbb{N} \setminus \{\langle k,j\rangle \mid j \in \mathbb{N}\}$. If and when $\varphi_k(k)$ turns out to be defined, then erase all $\langle k,j\rangle$ such that
(i) $j < \Phi_k(k)$ or
(ii) $\Phi_k(k) \leqslant j \leqslant 2\Phi_k(k)$ and, for some $y \in \mathbb{N}$, $t_y^+ \not\subseteq H_{\langle k,j\rangle}$".

Let $L \in \mathscr{L}$, and let $k \in \mathbb{N}$ be unique such that $L \subseteq \{a^k b^n \mid n \in \mathbb{N}\}$. In order to verify the correctness of M we distinguish the following cases.

Case 1: $\varphi_k(k)$ is undefined. Hence, $L = L_{\langle k,j\rangle} = H_{\langle k,j\rangle}$ for all $j \in \mathbb{N}$. By construction, M outputs exactly the set $\mathbb{N} \setminus \{\langle k,j\rangle \mid j \in \mathbb{N}\}$. Thus, M erases exclusively incorrect $\mathscr{H}$-indices for L, and M stabilizes to $\langle k,0\rangle$, the minimal $\mathscr{H}$-index for L.

Case 2: $\varphi_k(k)$ is defined. Let $L = H_{\langle k,j\rangle}$. Then $j \geqslant \Phi_k(k)$, since $j < \Phi_k(k)$ would imply $H_{\langle k,j\rangle} \notin \mathscr{L}$. As in Case 1, it is justified to erase all the numbers from $\mathbb{N} \setminus \{\langle k,j\rangle \mid j \in \mathbb{N}\}$. Moreover, erasing all the $\langle k,i\rangle$ with $i < \Phi_k(k)$, as M does by (i), is justified and necessary. Now, if $\Phi_k(k) \leqslant j \leqslant 2 \cdot \Phi_k(k)$, then L is finite and, by (ii), M will stabilize to $\langle k, j\rangle$, the only $\mathscr{H}$-index of L. Finally, if $j > 2 \cdot \Phi_k(k)$, then $L = \{a^k b^n \mid n \in \mathbb{N}\}$ and, again by (ii), M will stabilize to $\langle k, 2 \cdot \Phi_k(k) + 1\rangle$, the minimal $\mathscr{H}$-index of L.

To sum up, M $CSubTxt_{\mathscr{H}}$-learns L, and thus M witnesses $\mathscr{L} \in CSubTxt$.

Clearly, Assertion (2) follows immediately by Claim A and Claim B. Finally, since $SubTxt \subseteq CSubTxt$ and $SubTxt \subset ConsvTxt$ (cf. Theorem 11), we obtain $SubTxt \subset CSubTxt$ via Claim B. Thus Assertions (1) and (2) are proved. □

3.3.3. Absolute learning

We start with studying *AAllTxt*. As we shall see, this type is the least powerful one of learning by erasing from text.

Theorem 13. *FinTxt* $\subset$ *AAllTxt* $\subset$ *AEqualTxt.*

Proof. We know already that *FinTxt* $\subset$ *AAllTxt* (cf. Theorems 11 and 21 in [13]). Thus, it remains to show that *AAllTxt* $\subset$ *AEqualTxt*. By definition and by Theorem 3, *AAllTxt* $\subseteq$ *AArbTxt* $=$ *AEqualTxt*.

Claim. *AEqualTxt*$\setminus$*AAllTxt* $\neq \emptyset$.

We define the desired indexed family $\mathscr{L}$ as follows. For all $k \in \mathbb{N}$, let $L_{2k} = \{2^k, 2^{k+\Phi_k(k)}+1\}$ and $L_{2k+1} = \{2^k, 2^{k+\Phi_k(k)}+3\}$, where, by definition, $L_{2k} = L_{2k+1} = \{2^k\}$ iff $\varphi_k(k)$ is undefined. One easily verifies that $\mathscr{L}$ is indeed an indexed family. Moreover, $\mathscr{L}$ is inclusion-free, and therefore $\mathscr{L} \in$ *AEqualTxt* by Claim B in the proof of Theorem 1. In order to show that $\mathscr{L} \notin$ *AAllTxt* consider the hypothesis space ψ with $L(\psi_k) = L_k$ for all $k \in \mathbb{N}$. Now, one easily verifies that the equality problem for ψ is not recursive. Hence, $\mathscr{L} \notin$ *AAllTxt* follows from Theorem 4. □

Since for all $Lt \in \{Arb, Min, Sub, Equal, Super\}$, the types *ALtTxt* coincide by Theorems 3 and 7, *AAllTxt* indeed turns out to be the least powerful type of learning by erasing from text by Theorem 13. On the other hand, the "*A*-requirement" results in decreasing the power of all the types of learning by erasing from text, as our next result shows (the only exception is *AEqualTxt* $=$ *EqualTxt*, Corollary 2).

Corollary 14. *For all* $Lt \in \{Arb, Min, Sub, Super, All\}$, *ALtTxt* $\subset$ *LtTxt.*

Proof. By Theorems 8, 11, 13 and Corollary 2, *AAllTxt* $\subset$ *AEqualTxt* $=$ *EqualTxt* $\subset$ *ExTxt* $=$ *AllTxt*. Moreover, by Theorems 3, 8, 11 and Corollary 2, *ASuperTxt* $=$ *AArbTxt* $=$ *AEqualTxt* $=$ *EqualTxt* $\subset$ *ExTxt* $=$ *SuperTxt* $=$ *ArbTxt*. Finally, by Theorems 3, 7, 11 and Corollary 2, *AMinTxt* $=$ *ASubTxt* $=$ *AEqualTxt* $=$ *EqualTxt* $\subset$ *SubTxt* $=$ *MinTxt*. □

Finally, we compare *AConsvTxt*-inference with the models of learning by erasing.

Theorem 15. (1) *AConsvTxt*$\setminus$*CSubTxt* $\neq \emptyset$.

(2) *SubTxt*$\setminus$*AConsvTxt* $\neq \emptyset$.

Proof. First, we show (1). Let $L_0 = \mathbb{N}$ and, for all $j \in \mathbb{N}$, let $L_{j+1} = \{j\}$ as well as $\mathscr{L} = (L_j)_{j\in\mathbb{N}}$. We know already that $\mathscr{L} \notin CSubTxt$ (cf. Theorem 12, Claim A) and $\mathscr{L} \in ConsvTxt_{\mathscr{L}}$ (cf. Theorem 11, Claim C). Now, let $\psi \in \mathscr{R}^2_{0,1}$ be any class preserving hypothesis space for $\mathscr{L}$. Then, there is a recursive compiler c from $\mathscr{L}$ to ψ, i.e., $L_j = L(\psi_{c(j)})$ for all $j \in \mathbb{N}$. Given c and an IIM M witnessing $\mathscr{L} \in ConsvTxt_{\mathscr{L}}$, the IIM M' $ConsvTxt_\psi$-learns $\mathscr{L}$ where $M'(t_x) = c(M(t_x))$ for all possible inputs t_x, and, by convention, $c(?) = ?$.

To verify (2), let $\mathscr{L} = (L_{\langle k,j\rangle})_{k,j\in\mathbb{N}}$ with $L_{\langle k,j\rangle} = \{\langle k,0\rangle, \langle k,j\rangle\}$ for all $k,j \in \mathbb{N}$. Clearly, $\mathscr{L} \in SubTxt_{\mathscr{L}}$ by an ELM with $\text{ProgSet}(M,t) = \{j \mid (\exists y)[t_y^+ \not\subseteq L_j]\}$. Next, we define a class preserving hypothesis space $\mathscr{L}'$ for $\mathscr{L}$ such that $\mathscr{L} \notin ConsvTxt_{\mathscr{L}'}$. For all $k,j \in \mathbb{N}$, let $L'_{\langle k,0\rangle} = \{\langle k,0\rangle, \langle k,\Phi_k(k)\rangle\}$, and, in case that $j \geqslant 1$, let $L'_{\langle k,j\rangle} = \{\langle k,0\rangle, \langle k,j\rangle\}$, if $\Phi_k(k) \neq j$, and $L'_{\langle k,j\rangle} = \{\langle k,0\rangle\}$, otherwise. Clearly, $range(\mathscr{L}') = range(\mathscr{L})$. Now, let $I = \{\langle k,j\rangle \mid |L'_{\langle k,j\rangle}| = 1\}$. Furthermore, let $K = \{k \mid \varphi_k(k) \text{ is defined}\}$ denote the halting set of φ. From the definition of $\mathscr{L}'$, it follows immediately that, for any $k \in \mathbb{N}$, there is exactly one $j \in \mathbb{N}$ such that $|L'_{\langle k,j\rangle}| = 1$, namely $j = 0$, if $k \notin K$, and $j = \Phi_k(k)$, if $k \in K$.
Claim A. *$\mathscr{L} \in ConsvTxt_{\mathscr{L}'}$ implies that I is recursively enumerable.*

Let M be an IIM that $ConsvTxt_{\mathscr{L}'}$-identifies $\mathscr{L}$. Then define a function $f \in \mathscr{R}$ as follows. On input k, simulate M when successively fed $t^k = \langle k,0\rangle, \langle k,0\rangle, \ldots$ If and when M outputs its first guess j with $\langle k,0\rangle \in L'_j$, then define $f(k) = j$. Since M, in particular, $ConsvTxt_{\mathscr{L}'}$-identifies the singleton language $L = \{\langle k,0\rangle\}$ on its unique text t^k, L'_j must equal L, and thus $j \in I$. Moreover, j is the only $\mathscr{L}'$-index of $\{\langle k,0\rangle\}$. Hence, $I = range(f)$.
Claim B. *I is not recursively enumerable.*

Assume to the contrary that I is recursively enumerable. Then, given any $k \in \mathbb{N}$, compute the only $j \in \mathbb{N}$ such that $\langle k,j\rangle \in I$. If $j = \Phi_k(k)$, then $k \in K$, otherwise $k \notin K$. Consequently, K is recursive, a contradiction.

From Claims A and B we immediately obtain $\mathscr{L} \notin ConsvTxt_{\mathscr{L}'}$, and therefore $\mathscr{L} \notin AConsvTxt$. □

Theorem 16. *$EqualTxt \subset AConsvTxt$.*

Proof. Since $EqualTxt \subseteq SubTxt$, $AConsvTxt \setminus EqualTxt \neq \emptyset$ follows immediately from Theorem 15, Assertion (1). Now, let $\mathscr{L} \in EqualTxt$. By Theorem 1, Claim A, $\mathscr{L}$ is inclusion-free. Let $\psi \in \mathscr{R}^2_{0,1}$ be any class preserving hypothesis space for $\mathscr{L}$. An IIM M that $ConsvTxt_\psi$-learns $\mathscr{L}$ can be defined as follows. M, when successively fed any $t \in text(L)$ for any $L \in \mathscr{L}$, outputs the least ψ-index j with $t_x^+ \subseteq L(\psi_j)$. By definition, M performs exclusively justified mind changes, and therefore M is conservative. Since $\mathscr{L}$ is inclusion-free, we have $L \not\subseteq L(\psi_j)$ for all $j < min_\psi(L)$, and thus M converges to $min_\psi(L)$. □

3.4. Learning from informant

As we have seen in Theorem 10, most of the types of learning by erasing from informant coincide with *ExInf*. It only remains to clarify the power of *AAllInf* which will be done by Proposition 6 and Theorem 17 below. We then study the interplay between information presentation and learning capabilities, i.e., we compare "*Inf*-types" with "*Txt*-types" and vice versa.

Freivalds et al. [10] introduced the learning types *AllInf* and implicitly also *AAllInf*, and referred to them as to *co-learning*. Furthermore, they considered the co-learnability of recursively enumerable classes of *arbitrary* total recursive functions. This contrasts our scenario, since we exclusively study the learnability of $\{0,1\}$ valued functions. Nevertheless, their results easily translate into our setting.

Proposition 6 (Freivalds et al. [10]). $\mathit{FinInf} \subseteq \mathit{AAllInf} \subset \mathit{AllInf}$.

Using a deep result due to Selivanov [30], Freivalds et al. [11] could exhibit a recursively enumerable function class which is co-learnable with respect to all of its total recursive numberings, but which is not finitely learnable. Note, however, that the used function class is *not* $\{0,1\}$ valued. The same result was independently obtained by Kummer [20]. This result directly raises the question whether or not $\mathit{AAllInf}\setminus\mathit{FinInf} \neq \emptyset$. Our next theorem answers this question. Again, the proof is based on Selivanov's [30] result. For this proof and also in the following we need the notion of discreteness. An indexed family $\mathscr{L}=(L(\psi_j))_{j\in\mathbb{N}}$ is said to be *discrete* iff for every $k\in\mathbb{N}$, there is a finite function $\delta_k\subseteq\psi_k$ such that for all $j\in\mathbb{N}$, if $\delta_k\subseteq\psi_j$ then $\psi_k=\psi_j$. We refer to δ_k as to a *separating function* for ψ_k. An indexed family $\mathscr{L}=(L(\psi_j))_{j\in\mathbb{N}}$ is said to be *effectively discrete* iff there exists an algorithm computing for every $k\in\mathbb{N}$ a separating function δ_k for ψ_k.

Theorem 17. $\mathit{AAllTxt}\setminus\mathit{FinInf} \neq \emptyset$.

Proof. Selivanov [30] showed that there is a recursively enumerable class $\mathscr{U}_{\mathrm{se}}$ of total recursive functions fulfilling the following requirements:
(1) every numbering $\tau\in\mathscr{R}^2$ for $\mathscr{U}_{\mathrm{se}}$ has a recursive equality problem, and
(2) $\mathscr{U}_{\mathrm{se}}$ is not effectively discrete.
Since $\mathscr{U}_{\mathrm{se}}$ is not $\{0,1\}$ valued, some transformation of it is in order. Using $\mathscr{U}_{\mathrm{se}}$ we define an indexed family $\mathscr{L}_{\mathrm{se}}$ that is well suited to separate *AAllTxt* and *FinInf*.

Let $\tau\in\mathscr{R}^2$ be any numbering for $\mathscr{U}_{\mathrm{se}}$. For all $j,x,y\in\mathbb{N}$ we set:

$$\psi_j(\langle x,y\rangle)=\begin{cases}1 & \text{if } \tau_j(x)=y,\\ 0 & \text{otherwise.}\end{cases}$$

Finally, set $\mathscr{L}_{\mathrm{se}}=(L(\psi_j))_{j\in\mathbb{N}}$. Clearly, $\mathscr{L}_{\mathrm{se}}$ is an indexed family.
Claim A. $\mathscr{L}_{\mathrm{se}}\in\mathit{AAllTxt}$.

Applying the characterization of *AAllTxt* (cf. Theorem 4) it suffices to show that $\mathscr{L}_{\mathrm{se}}$ is inclusion-free and, furthermore, every class preserving hypothesis space for $\mathscr{L}_{\mathrm{se}}$ has a recursive equality problem.

Let $j\in\mathbb{N}$. Since τ is a numbering of total recursive functions, we may easily conclude that, for every $x\in\mathbb{N}$, there is exactly one $y\in\mathbb{N}$ with $\langle x,y\rangle\in L(\psi_j)$. Hence $\mathscr{L}_{\mathrm{se}}$ is inclusion-free.

Now, assume any class preserving hypothesis space $\hat{\psi}$ for $\mathscr{L}_{\mathrm{se}}$. For all j, $x \in \mathbb{N}$, set $\hat{\tau}(j,x) = y$, where y is the uniquely determined number with $\langle x, y \rangle \in L(\hat{\psi}_j)$. Obviously, $\hat{\tau} \in \mathscr{R}^2$. Since $\hat{\psi}$ is class preserving for $\mathscr{L}_{\mathrm{se}}$, $\hat{\tau}$ is a numbering for $\mathscr{U}_{\mathrm{se}}$. Clearly, $\hat{\tau}_j = \hat{\tau}_k$ iff $L(\hat{\psi}_j) = L(\hat{\psi}_k)$. By Property (1), $\hat{\tau}$ has a recursive equality problem, and thus we are done.

Claim B. $\mathscr{L}_{\mathrm{se}} \notin \mathit{FinInf}$.

Suppose the converse, i.e., $\mathscr{L}_{\mathrm{se}} \in \mathit{FinInf}$. Since finite inference is invariant with respect to choice of the hypothesis space (cf. [26]), we may assume that there is an IIM M witnessing $\mathscr{L}_{\mathrm{se}} \in \mathit{FinInf}_{\psi}$, where ψ is the hypothesis space defined above.

Given M, we define an algorithm $\mathscr{A}$ that assigns a separating function δ_k to every τ_k. $\mathscr{A}$ is defined as follows. On input $k \in \mathbb{N}$, execute the following instructions:

(A1) For $z = 0, 1, 2, \ldots$ generate successively the lexicographically ordered informant i^k for $L(\psi_k)$ until M outputs a guess, say on input $i^k_{\hat{z}}$.

(A2) Let $m = \max\{x \mid \exists y[\langle x, y \rangle \in \mathit{content}(i^k_{\hat{z}})]\}$. Set $\delta_k = \{(x, \tau_k(x)) \mid x \leqslant m\}$.

Since M FinInf_{ψ}-identifies $\mathscr{L}_{\mathrm{se}}$, we may conclude that Instruction $(A1)$ terminates for every $k \in \mathbb{N}$, and thus $\mathscr{A}$ is recursive. It suffices to show that, for all $j \in \mathbb{N}$, $\delta_k \subseteq \tau_j$ implies $\tau_k = \tau_j$.

Suppose any $j \in \mathbb{N}$ with $\delta_k \subseteq \tau_j$. Clearly, $\tau_k(x) = \tau_j(x)$ for all $x \leqslant m$, and therefore $\psi_j(\langle x, y \rangle) = \psi_k(\langle x, y \rangle)$ for all $y \in \mathbb{N}$ and all $x \leqslant m$. By the choice of m, $i^k_{\hat{z}}$ is an initial segment of the lexicographically ordered informant i^j for $L(\psi_j)$. By Definition 2, M, when successively fed i^k and i^j, respectively, is only allowed to generate a single, but correct hypothesis. Since M, when fed $i^k_{\hat{z}} = i^j_{\hat{z}}$, has output its one and only hypothesis, we obtain $M(i^k_{\hat{z}}) = M(i^j_{\hat{z}})$, and hence $L(\psi_k) = L(\psi_j)$. Consequently, $\tau_k = \tau_j$, too. Thus, $\mathscr{U}_{\mathrm{se}}$ is effectively discrete, a contradiction. □

So far we have studied learning from text and learning from informant separately. Now we focus our attention to another aspect, namely the interplay between information presentation and learnability constraints, i.e., we compare "*Inf*-types" with "*Txt*-types" and vice versa. The first known result along this line relates finite learning from informant to conservative inference from text.

Proposition 7 (Lange and Zeugmann [23]). *FinInf* $\subset$ *ConsvTxt.*

Since *FinInf* $\subseteq$ *AAllInf* by Proposition 6, the question arises whether or not Proposition 7 generalizes to *AAllInf* $\subset$ *ConsvTxt* or at least to *AAllInf* $\subset$ *ExTxt*. While the validity of the former inclusion remains open, we show that the latter one is indeed valid by Corollary 19 below. In order to establish this result we first prove that discreteness implies *ExTxt*-learnability.

Theorem 18. *For any indexed family $\mathscr{L}$, if $\mathscr{L}$ is discrete, then $\mathscr{L} \in \mathit{ExTxt}$.*

Proof. Let $\mathscr{L}$ be any indexed family that is discrete, and let ψ be any class preserving hypothesis space for $\mathscr{L}$. For any $t \in \mathit{text}(\mathscr{L})$ and for any $i, x \in \mathbb{N}$, let $a^x_i = \max\{z \leqslant x \mid$

$(\forall y \leqslant z)[\psi_i(y) = 1$ iff $y \in t_x^+]\}$. Clearly, given t_x and i, a_i^x is computable. Define an IIM M as follows.

$M(t_x) = \min\{i \mid i \leqslant x$ and $a_i^x = \max\{a_j^x \mid j \leqslant x\}\}$.

Let $L \in \mathscr{L}$, and let $k = min_\psi(L)$. Then, by discreteness of $\mathscr{L}$ and since ψ is class preserving for $\mathscr{L}$, there is an $a \in \mathbb{N}$ such that, for all $j \in \mathbb{N}$ with $L(\psi_j) \neq L$, $L(\psi_j)$ differs from L on some $y \leqslant a$. For any $t \in text(L)$, there is an $x' > \max\{a, k\}$ such that $t_{x'}^+ \supseteq \{y \leqslant a \mid y \in L\}$. Now, for all $x \geqslant x'$, we have $a_k^x \geqslant a$, whereas $a_j^x < a$ for all $j \in \mathbb{N}$ with $L(\psi_j) \neq L$. By definition, $M(t_x) = k$ for all $x \geqslant x'$, and thus M converges to the minimal ψ-index for L. Hence, M $ExTxt_\psi$-identifies $\mathscr{L}$. □

Note that Theorem 18 cannot be sharpened to show that discreteness implies conservative learnability, since the indexed family $\mathscr{L}$ defined in the proof of Theorem 12, Claim B is discrete but $\mathscr{L} \notin CConsvTxt$ (cf. Theorem 1 in [24]).

Corollary 19. *$AAllInf \subset ExTxt$.*

Proof. Let $\mathscr{L} \in AAllInf$. Then $\mathscr{L}$ is discrete (cf. Theorem 10 and Fact 5 in [20]). Hence, $\mathscr{L} \in ExTxt$ follows by Theorem 18 above. On the other hand, let $\mathscr{L}_{\mathrm{fin}}$ denote the indexed family canonically enumerating all finite sets of natural numbers. Obviously, $\mathscr{L}_{\mathrm{fin}}$ is not discrete, and thus, $\mathscr{L}_{\mathrm{fin}} \notin AAllInf$. But $\mathscr{L}_{\mathrm{fin}} \in ExTxt$, and the corollary follows. □

By Theorem 8, we may esily conclude:

Corollary 20. *For all $Lt \in \{Arb, Super, All\}$, $AAllInf \subset LtTxt$.*

The following theorem enables us to clarify the relation between the remaining models of learning by erasing from text and informant, respectively.

Theorem 21. (1) *$FinInf \setminus CSubTxt \neq \emptyset$.*
(2) *$EqualTxt \setminus AAllInf \neq \emptyset$.*

Proof. For verifying Assertion (1) recall the definition of the indexed family $\mathscr{L} = (L_j)_{j\in\mathbb{N}}$ used in the proof of Theorem 11, i.e., $L_0 = \mathbb{N}$ and $L_{j+1} = \{j\}$. Obviously, $\mathscr{L}$ is *FinInf*-identifiable, and since $\mathscr{L} \notin CSubTxt$ (cf. Theorem 12, Claim A), Assertion (1) follows.

To verify Assertion (2), we refer the reader to the proof of Theorem 13. There, an indexed family $\mathscr{L} \in EqualTxt$ is presented that possesses a class preserving hypothesis space having no recursive equality problem. Hence, $\mathscr{L} \notin AAllInf$ by Proposition 4. □

Taking into account that $ExTxt \subset ExInf$ (cf. [14]), we directly arrive at the following corollary displaying the consequences of Theorem 21.

Corollary 22. *For all $Lt \in \{Arb, Min, Sub, Equal, Super, All\}$ and for all $\lambda \in \{A, \varepsilon, C\}$, $\lambda LtTxt \subset \lambda LtInf$.*

Putting Theorem 21 together with Corollary 2 we can easily conclude:

Corollary 23. (1) $AEqualTxt \# AAllInf$.
(2) $SubTxt \# AAllInf$.
(3) $CSubTxt \# AAllInf$.
(4) $AAllTxt \# FinInf$.

4. Function learning by erasing

We now turn to the problem of learning (by erasing) of functions instead of languages. For function learning, it is more interesting to study the relationship between criteria for a fixed numbering (in Section 4.4 we consider the case of "identifiable in some numbering"). Freivalds et al. [10] have studied the analogue of *All*-learning, which they called co-learning. For special computable non-Gödel numberings of partial recursive functions, this analogue of *All*-learning, turned out to be significantly more restrictive than learning in the limit. However, in Gödel numberings any learnable family can be learned by an erasing strategy. Already the results in Section 3.4 for learning from informant indicate that the less restrictive versions of learning by erasing as considered in Section 3, do not increase the learning power.

Therefore in this section we concentrate on learning *minimal* programs by erasing and learning minimal programs by strategies that result from learning by erasing. Moreover we will mostly concentrate on Gödel numberings. We show that learning of minimal programs by erasing, as originally defined in [10], is significantly weaker than learning minimal programs even in Gödel numberings. In order to enhance the learning power of erasing strategies, we generalize the concept in a manner analogous to Section 3. First, we allow the learning strategy to possibly not cancel out more than one of the programs for the input function, but we still require the strategy to cancel all incorrect programs of that function (*Super* in our notation). Secondly, we observe learning by erasing some incorrect programs only, including all programs that are smaller than the minimal correct one (*Sub* in our notation). We show that each of the above two types of learning minimal programs by erasing is considerably more powerful than erasing all but the minimal program. We only briefly consider *Arb* and *Equal* notions, since they turn out to be either identical to other notions or trivial. Then we exhibit various relationships and differences between the types of learning minimal programs by erasing and give some examples of classes learnable within each of those paradigms. In particular, we explore learning by erasing in *Kolmogorov* numberings that can be viewed as "natural" Gödel numberings of the partial recursive functions.

Learning of minimal programs by erasing naturally suggests a special strategy of *learning* minimal programs: each new hypothesis is *larger* than the prior one. In contrast, one can also consider learning minimal programs by strategies of the opposite type: each new hypothesis is *smaller* than the prior one. We show that these both types of learning minimal programs are weaker than learning minimal programs in the general case.

We derive necessary and sufficient conditions for the considered types of function learning by erasing. Therefore, we show that the types coincide with the type Ex of standard learning in the limit provided that arbitrary hypothesis spaces are allowed among them also non-Gödel numberings. Then we derive a pure numbering-theoretic characterization of all these types. Finally, we present a characterization which comes to the granularity of exhibiting necessary and sufficient learnability conditions for an arbitrary *pair* of a function class *and* a hypothesis space. Note that this is the first characterization result of such kind in function learning.

4.1. Definitions

In the following we present the necessary definitions both for standard function learning and for learning functions by erasing. Furthermore, we state some basic results which easily follow from these definitions.

For $f \in \mathcal{R}$ and $n \in \mathbb{N}$, the initial segment $(f(0), f(1), \ldots, f(n-1))$ is denoted by $f[n]$, or, more formally, the code of the tuple $(f(0), f(1), \ldots, f(n-1))$ in some fixed one-one computable encoding of all tuples of natural numbers onto $\mathbb{N}$. Let $\mathrm{SEG} = \{f[n] \mid f \in \mathcal{R}, n \in \mathbb{N}\}$. Λ denotes the empty segment $f[0]$. An inductive inference machine (IIM) (also called *learning machine*) is an algorithmic mapping from SEG to $\mathbb{N} \cup \{?\}$. Intuitively, we will interpret the output of a learning machine as a program. "?" then represents the case where the machine outputs "no conjecture". We let M, with or without, decorations range over learning machines.

An erasing learning machine (ELM) is similar to an IIM but, in a way similar to that of language learning, we interpret the output of an ELM in a different manner. We let M range over erasing learning machines, too. The context will determine whether M denotes IIM or ELM.

In the following definitions let ψ denote any numbering and f any recursive function. Let $\mathrm{Progs}_\psi(f)$ denote the set of ψ-programs for f, i.e., $\mathrm{Progs}_\psi(f) = \{i \mid \psi_i = f\}$. Let $min_\psi(f) = \min \mathrm{Progs}_\psi(f)$.

Definition 7 (Gold [14]). M *Ex_ψ-infers* f iff there exists an $i \in \mathbb{N}$ such that the sequence $(M(f[n]))_{n\in\mathbb{N}}$ converges to i and $\psi_i = f$.

M Ex_ψ-infers $\mathcal{C}$ iff *M Ex_ψ-infers* each $f \in \mathcal{C}$.

Let Ex_ψ denote the collection of all classes of functions for which there is an IIM M such that *M Ex_ψ-infers $\mathcal{C}$*.

It can be shown that $Ex_\psi = Ex_\varphi$, for all Gödel numberings ψ. Thus, the class Ex is invariant under Gödel numbering chosen to interpret the programs conjectured by the machines. Thus we often refer to Ex_φ as just Ex.

Definition 8 (Freivalds [6]). M *$MinEx_\psi$-infers* f iff $f \in \mathcal{R}_\psi$ and the sequence $(M(f[n]))_{n\in\mathbb{N}}$ converges to $min_\psi(f)$.

M $MinEx_\psi$-infers $\mathcal{C}$ iff *M $MinEx_\psi$-infers* each $f \in \mathcal{C}$.

Finally, $MinEx_\psi$ is defined analogously as above.

Unlike *Ex* inference, the class $MinEx_\psi$ depends on the Gödel numbering ψ (cf. [6]).

Definition 9 (Gold [14]). *M Fin_ψ-infers f* iff there exists an i such that $\psi_i = f$ and the sequence $(M(f[n]))_{n \in \mathbb{N}}$ contains only the program i (besides possibly "?").

M Fin_ψ-infers $\mathscr{C}$ iff *M Fin_ψ-infers* each $f \in \mathscr{C}$.

Finally, Fin_ψ is defined analogously as above.

The class Fin_ψ is the same for all Gödel numberings ψ. Thus we often refer to Fin_φ as just *Fin*.

Definition 10 (Freivalds [6]). *M $MinFin_\psi$-infers f* iff $f \in \mathscr{R}_\psi$ and the sequence $(M(f[n]))_{n \in \mathbb{N}}$ contains only the program $\min_\psi(f)$ (besides possibly "?").

M $MinFin_\psi$-infers $\mathscr{C}$ iff *M $MinFin_\psi$-infers* each $f \in \mathscr{C}$.

Finally, $MinFin_\psi$ is defined analogously as above.

The class $MinFin_\psi$ depends on the Gödel numbering ψ (cf. [6]).

We will now consider the different versions of learning by erasing. Similar to the definitions of *Ex* and *Fin*, we will first define the general version and then the minimal version. Again in the definitions below let ψ denote any numbering and f any recursive function.

Definition 11. Let $\text{ProgSet}(M, f) = \{M(f[n]) \mid n \in \mathbb{N} \wedge M(f[n]) \neq ?\}$, and let $\text{ProgSet}(M, f[n]) = \{M(f[m]) \mid m \leqslant n \wedge M(f[m]) \neq ?\}$.

We say that M *stabilizes* on f to i iff $i = \min(\mathbb{N} \backslash \text{ProgSet}(M, f))$. We say that M stabilizes on f iff there exists an i such that M stabilizes on f to i.

Definition 12. An ELM *M Arb_ψ-infers f* iff there exists an i such that $\psi_i = f$ and M stabilizes on f to i.

M Arb_ψ-infers $\mathscr{C}$ iff *M Arb_ψ-infers* each $f \in \mathscr{C}$.

Finally, Arb_ψ is defined analogously as above.

Definition 13. An ELM *M $MinArb_\psi$-infers f* iff $f \in \mathscr{R}_\psi$ and M stabilizes on f to $min_\psi(f)$.

M $MinArb_\psi$-infers $\mathscr{C}$ iff *M* $MinArb_\psi$-infers each $f \in \mathscr{C}$.

Finally, $MinArb_\psi$ is defined analogously as above.

The definition of All_ψ below was first given by Freivalds et al. [10].

Definition 14. An ELM *M*

(A) *Sub_ψ-infers f*,

(B) *$Equal_\psi$-infers f*,

(C) *$Super_\psi$-infers f*,

(D) *All_ψ-infers f*,

iff *M Arb_ψ-infers f* and the following corresponding condition is satisfied:

(A) $\mathrm{ProgSet}(M, f) \subseteq \mathbb{N} \backslash \mathrm{Progs}_{\psi}(f)$, i.e., M is only allowed to erase incorrect programs;
(B) $\mathrm{ProgSet}(M, f) = \mathbb{N} \backslash \mathrm{Progs}_{\psi}(f)$, i.e., M has to erase exactly all the incorrect programs;
(C) $\mathrm{ProgSet}(M, f) \supseteq \mathbb{N} \backslash \mathrm{Progs}_{\psi}(f)$, i.e., M has to erase all the incorrect programs;
(D) $|\mathbb{N} \backslash \mathrm{ProgSet}(M, f)| = 1$, i.e., M has to erase all but one program.

We can define Sub_{ψ} ($Super_{\psi}$, All_{ψ}, $Equal_{\psi}$)-identification of classes by an ELM M in a manner similar to above. Finally, the classes Sub_{ψ}, $Super_{\psi}$, All_{ψ}, and $Equal_{\psi}$ can be analogously defined.

Definition 15. An ELM M
(A) *$MinSub_{\psi}$-infers* f,
(B) *$MinEqual_{\psi}$-infers* f,
(C) *$MinSuper_{\psi}$-infers* f,
(D) *$MinAll_{\psi}$-infers* f,
iff M *$MinArb_{\psi}$-infers* f and the following corresponding condition is satisfied:
(A) $\mathrm{ProgSet}(M, f) \subseteq \mathbb{N} \backslash \mathrm{Progs}_{\psi}(f)$, i.e., M is only allowed to erase incorrect programs;
(B) $\mathrm{ProgSet}(M, f) = \mathbb{N} \backslash \mathrm{Progs}_{\psi}(f)$, i.e., M has to erase exactly all the incorrect programs;
(C) $\mathrm{ProgSet}(M, f) \supseteq \mathbb{N} \backslash \mathrm{Progs}_{\psi}(f)$, i.e., M has to erase all the incorrect programs;
(D) $|\mathbb{N} \backslash \mathrm{ProgSet}(M, f)| = 1$, i.e., M has to erase all but one program.

We can define $MinSub_{\psi}$ ($MinSuper_{\psi}$, $MinAll_{\psi}$, $MinEqual_{\psi}$)-identification of classes by an ELM M in a manner similar to above. Finally, the classes $MinSub_{\psi}$, $MinSuper_{\psi}$, $MinAll_{\psi}$, and $MinEqual_{\psi}$ can be analogously defined.

For Gödel numberings ψ, by Proposition 11 below, $Arb_{\psi} = Ex_{\psi}$. By Proposition 9, for all numberings ψ, $MinArb_{\psi} = MinSub_{\psi}$. Thus, we will not consider $MinArb_{\psi}, Arb_{\psi}$ further in this paper.

For Gödel numberings ψ, the criterion $Equal_{\psi}$ is essentially trivial, so we do not consider $Equal_{\psi}$ any further in this paper, too.

Learning by erasing in the sense of $MinSub_{\psi}$ suggests a natural strategy of *learning* minimal programs: output, in increasing order, all the indices smaller than the minimal one for the given function. We will observe below in Proposition 9 that this type of learning minimal programs and $MinSub_{\psi}$ are of the same power.

Definition 16. M *$MinIncEx_{\psi}$-infers* f iff M *$MinEx_{\psi}$-infers* f and the sequence $(M(f[n]))_{n \in \mathbb{N}}$ (except for initial sequence of ?'s) is monotonically non-decreasing.
M *$MinIncEx_{\psi}$-infers* $\mathscr{C}$ iff M *$MinIncEx_{\psi}$-infers* each $f \in \mathscr{C}$.
Finally, $MinIncEx_{\psi}$ is defined analogously as above.

As a dual to $MinIncEx_{\psi}$ criterion we consider the case where the machine is required to output its conjectures in a *decreasing* order.

Definition 17. *M $MinDecEx_\psi$-infers f* iff *M $MinEx_\psi$-infers f* and the sequence $(M(f[n]))_{n\in\mathbb{N}}$ (except for initial sequence of ?'s) is monotonically non-increasing.
M $MinDecEx_\psi$-infers $\mathscr{C}$ iff *M $MinDecEx_\psi$-infers* each $f\in\mathscr{C}$.
Finally, $MinDecEx_\psi$ is defined analogously as above.

For any learning type Lt above, we set $Lt=\bigcup_{\psi\in\mathscr{P}^2}Lt_\psi$.

It can easily be shown that there exists an r.e. sequence of machines $M_0,M_1,\ldots,$ such that, for any learning type Lt_ψ discussed in this section, if $\mathscr{C}\in Lt_\psi$ then, for some $i\in\mathbb{N}$, M_i Lt_ψ-infers $\mathscr{C}$. Intuitively, this enumeration of machines allows us to restrict our attention to just these machines in the diagonalizations.

Clearly, the minimal version of each of the criteria defined above can only be restrictive, thus:

Proposition 8. *For every $Lt\in\{Fin,Ex,Sub,Super,All\}$ and every numbering ψ, $MinLt_\psi\subseteq Lt_\psi$.*

The following proposition essentially follows directly from the definitions, we omit the details.

Proposition 9. *$MinIncEx_\psi=MinSub_\psi=MinArb_\psi=Sub_\psi$ and $MinIncEx_\psi\cap MinDecEx_\psi=MinFin_\psi$ for every numbering ψ.*

Hence in order to derive results on $MinSub_\psi$ and Sub_ψ it suffices to study $MinIncEx_\psi$ instead. We will use this approach at various places below.

Proposition 10. *$MinFin_\psi\subseteq MinAll_\psi\subseteq MinSuper_\psi\subseteq MinSub_\psi\subseteq MinEx_\psi$ and $MinFin_\psi\subseteq MinDecEx_\psi\subseteq MinEx_\psi$ for every numbering ψ.*

Proof. We show $MinSuper_\psi\subseteq MinSub_\psi$. All the other inclusions follow from the definitions.

Suppose M is given. Let M' be such that $\mathrm{ProgSet}(M',f)=\{i\mid(\forall j\leqslant i)[j\in\mathrm{ProgSet}(M,f)]\}$ for all f. It is easy to construct M' as above. It is now easy to see that M' $MinSub_\psi$-infers each function $MinSuper_\psi$-identified by M. □

Proposition 11. *$All_\psi=Super_\psi=Arb_\psi=Ex_\psi$ for every Gödel numbering ψ.*

Proof. Clearly, for all numberings ψ, $All_\psi\subseteq Super_\psi\subseteq Arb_\psi\subseteq Ex_\psi$. For Gödel numberings ψ, $Ex_\psi\subseteq All_\psi$ (cf. [10]). □

Since we are mainly interested in Gödel numberings and Kolmogorov numberings, due to Proposition 11, we will mostly be interested only in the *minimal* versions of the criteria of learning by erasing.

Note that in contrast to Proposition 11 the following holds.

$$\begin{tikzcd}
 & MinEx & \\
MinIncEx = MinSub \arrow[ur] & & MinDecEx \arrow[ul] \\
MinSuper \arrow[u] & & \\
MinAll \arrow[u, dashed] & & \\
 & MinFin \arrow[ul] \arrow[uuur] &
\end{tikzcd}$$

$\longrightarrow$ Proper containment in all Kolmogorov numberings

$\dashrightarrow$ Proper containment in some Kolmogorov numbering

Fig. 2. Learning in Kolmogorov numberings.

Proposition 12. *$Sub_\psi \subset Ex_\psi$ for every Gödel numbering ψ.*

Proof. Let ψ be any Gödel numbering. Then, by Proposition 9 and by Freivalds [6], $Sub_\psi = MinIncEx_\psi \subseteq MinEx_\psi \subset Ex_\psi$. □

4.2. Kolmogorov numberings as hypothesis spaces

In this section we show that all of the different minimal criteria defined above are separated in *every* Kolmogorov numbering. There is one exception, though, namely for $MinSuper_\psi$ and $MinAll_\psi$, at present we know only that these types are separated in *some* Kolmogorov numbering. The other "exception" concerns $MinSub_\psi$ and $MinIncEx_\psi$ which cannot be separated, since these types coincide by Proposition 9. Fig. 2 summarizes the inclusions of the types of both standard learning and learning by erasing of minimal programs in Kolmogorov numberings.

First, we will show the following result.

Theorem 24. *For every Kolmogorov numbering ψ,*
(1) $MinSuper_\psi \subset MinIncEx_\psi$,
(2) $MinAll_\psi \subset MinIncEx_\psi$,
(3) $MinSuper_\psi \# MinDecEx_\psi$,
(4) $MinAll_\psi \# MinDecEx_\psi$,
(5) $MinIncEx_\psi \# MinDecEx_\psi$.

The proof of Theorem 24 will be based on several other results which we will prove now. Some of these results are interesting in its own. Moreover, the proofs of these results demonstrate some techniques which turn out to be useful just in Kolmogorov numberings.

Let $\mathscr{H} = \{h_j \mid j \in \mathbb{N}\}$, where h_j is defined as follows.

$$h_j(x) = \begin{cases} 1 & \text{if } x = j, \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 25 (Freivalds [7, 8]). *For every Kolmogorov numbering ψ, there is a $\mathscr{C} \subseteq \mathscr{H}$ such that $|\mathscr{C}| = \infty$ and $\mathscr{C} \in MinFin_\psi$.*

Recall that $MinFin_\psi \subseteq MinIncEx_\psi$ by Proposition 9. We next consider the following proposition.

Proposition 13. *For every Gödel numbering ψ, all $\mathscr{C} \in MinIncEx_\psi$ and all $f \in \mathscr{R}$, $\mathscr{C} \cup \{f\} \in MinIncEx_\psi$.*

Proof. Suppose $\psi, \mathscr{C}, f$ are as given in the hypothesis. Let $i = min_\psi(f)$. Then there exists a $k \in \mathbb{N}$ such that, for all $g \in \mathscr{R}$, if $g[k] = f[k]$, then $min_\psi(f) \leqslant min_\psi(g)$. Suppose M $MinIncEx_\psi$ infers $\mathscr{C}$. Define M' as follows:

$$M'(g[n]) = \begin{cases} 0 & \text{if } n \leqslant k; \\ M(g[n]) & \text{if } n > k, \text{ and } f[k] \neq g[k]; \\ i & \text{if } n > k, \text{ and } g[n] = f[n]; \\ \max\{i, M(g[n])\} & \text{otherwise.} \end{cases}$$

It is easy to verify that M' $MinIncEx_\psi$-identifies $\mathscr{C} \cup \{f\}$. □

Proposition 13 does not hold for *MinDecEx* replacing *MinIncEx*. Let *ZERO* be the everywhere 0 function. Note that any machine $MinDecEx_\psi$-identifying *ZERO* can $MinDecEx_\psi$-identify only finitely many functions in $\mathscr{H}$. Thus:

Proposition 14. *For every numbering ψ and every $\mathscr{C} \subseteq \mathscr{H}$ with $|\mathscr{C}| = \infty$, $\mathscr{C} \cup \{ZERO\} \notin MinDecEx_\psi$.*

Freivalds' proof of Theorem 25 essentially also shows:

Theorem 26. *For every Kolmogorov numbering ψ, there are $\varepsilon > 0$, $\mathscr{C} \in MinFin_\psi$ and infinitely many $r \in \mathbb{N}$ such that $|\{i \leqslant r \mid h_i \in \mathscr{C}\}| \geqslant \varepsilon \cdot r$.*

Using Theorem 26 we derive the following result.

Theorem 27. *For every Kolmogorov numbering ψ, there is a $\mathscr{C} \subseteq \mathscr{H}$ with $|\mathscr{C}| = \infty$ and $\mathscr{C} \cup \{ZERO\} \in MinAll_\psi$.*

Proof. Let $M, \varepsilon > 0$, and $\mathscr{C}' \subseteq \mathscr{H}$, be such that M $MinAll_\psi$-infers $\mathscr{C}'$ and $(\overset{\infty}{\exists}\, r)$ $[|\{i \leqslant r \mid h_i \in \mathscr{C}'\}| \geqslant \varepsilon \cdot r]$. Note that there exists such $M, \varepsilon, \mathscr{C}'$ by Theorem 26 and Proposition 10.

Let M' be defined as follows. Suppose $z = min_\psi(ZERO)$.

$$\text{ProgSet}(M', ZERO[n]) = \{x \mid x \leqslant \varepsilon \cdot n/2 \wedge x \neq z\}$$
$$\text{ProgSet}(M', h_j) = \text{ProgSet}(M, h_j) \cup \text{ProgSet}(M', ZERO[j])$$

Note that such an M' can be easily constructed.

Note that $M' MinAll_\psi$-infers $ZERO$. It may however not $MinAll_\psi$-identify all the functions $MinAll_\psi$-identified by M (due to the extra programs output by M' on $h_j[j]$).

Let $\mathscr{C} = \{h_j \mid h_j \in \mathscr{C}' \text{ and } min_\psi(h_j) > \varepsilon \cdot j/2\}$. It is easy to verify that M' $MinAll_\psi$-infers each function in $\mathscr{C}$. Moreover, $|\mathscr{C}| = \infty$, since it contains at least $(\varepsilon - \varepsilon/2) \cdot r$ functions in $\{h_1, \ldots, h_r\}$ for infinitely many r. □

From Theorem 27 and Proposition 14, we have:

Theorem 28. *$MinAll_\psi \backslash MinDecEx_\psi \neq \emptyset$ for every Kolmogorov numbering ψ.*

The following corollary follows from Theorem 28. It can also be obtained directly from Theorem 25 and Propositions 13 and 14.

Corollary 29. *$MinIncEx_\psi \backslash MinDecEx_\psi \neq \emptyset$ for every Kolmogorov numbering ψ.*

Corollary 30. *$MinDecEx_\psi \subset MinEx_\psi$ for every Kolmogorov numbering ψ.*

We now prove a result which is complementary to Corollary 29.

Theorem 31. *$MinDecEx_\psi \backslash MinIncEx_\psi \neq \emptyset$ for every Kolmogorov numbering ψ.*

Proof. Suppose a Kolmogorov numbering ψ is given. We will construct a numbering τ as follows.

For each $i \in \mathbb{N}$, $j \leqslant i$, we will define $l_i^j, u_i^j \in \mathbb{N}$. Think of the programs (in the numbering τ) as being divided into intervals, I_i, and each interval I_i as being subdivided into $i+1$ subintervals I_i^j. The numbers l_i^j and u_i^j are the boundaries of the interval I_i^j.

$l_0^0 = 1$.

$l_{i+1}^0 = u_i^i + 1$. For $j < i$, $l_i^{j+1} = u_i^j + 1$.

For $j \leqslant i$, $u_i^j = l_i^j \cdot i \cdot 3 + 1$.

For $j \leqslant i$, let $I_i^j = \{p \mid l_i^j \leqslant p \leqslant u_i^j\}$.

Let $I_i = \bigcup_{j \leqslant i} I_i^j$.

Each τ_k will either be a total recursive function or the everywhere undefined function.

In the construction below, we will define the functions τ_k, for $k \in I_i$ (such a construction is carried out for each i separately). Intuitively, for each i, we plan to construct a collection $\mathscr{S}_i$ of functions, such that none of $M_0, M_1, \ldots, M_{i-1}$ $MinIncEx_\psi$-identifies any function in $\mathscr{S}_i$. These $\mathscr{S}_i$ will, in addition, satisfy some nice properties. This, in turn will allow us to construct the diagonalizing class witnessing the theorem.

Informally, in the construction below, we start with $j = i$ and define all τ_p for $p \in I_i^j$. Then, we search for one out of the i machines $M_0, \ldots, M_{i-1}$, which can be "diagonalized" against using (an appropriate initial segment of) one of the functions τ_p already defined (see steps 3 and 4). If the search is successful, the procedure is repeated with $j = i-1$, searching for one more machine which can be diagonalized against, and so on. Since, we do not know about the reduction from τ to ψ, the diagonalization mentioned above may not work for all i. It however works for all but finitely many i.

The diagonalization condition in step 4:

$$|\{r < i \mid (\exists q \geqslant i \cdot l_i^j)[q \in \mathrm{ProgSet}(M_r, \tau_p[y])]\}| > s$$

is due to the fact that a machine M_r, which has already been diagonalized against, will fulfill the requirement $(\exists q \geqslant i \cdot l_i^j)[q \in \mathrm{ProgSet}(M_r, \tau_p[y])]$ in the subsequent stages. Thus, after each (successful) stage s, we would have diagonalized against at least $s+1$ machines.

We now proceed formally. Actual definition of $\mathscr{S}_i$ and the diagonalization class witnessing the theorem will be given after the construction.
Definition of τ_k, for $k \in I_i$.

Let $\sigma_i^0(0) = i$ (σ_i^0 is of length 1). Go to stage 0.
Begin stage s.
1. Let $j = i - s$.
2. Let $m = |\sigma_i^s|$ (i.e., m is the least element not in the domain of σ_i^s).
3. For all $p \in I_i^j$, define τ_p as follows

$$\tau_p(x) = \begin{cases} \sigma_s(x) & \text{if } x < m, \\ p & \text{if } x = m, \\ 0 & \text{otherwise.} \end{cases}$$

4. Search for $p \in I_i^j$ and $y > m$, such that, $|\{r < i \mid (\exists q \geqslant i \cdot l_i^j)[q \in \mathrm{ProgSet}(M_r, \tau_p[y])]\}| > s$.
   (* Intuitively if $(\exists q \geqslant i \cdot l_i^j)[q \in \mathrm{ProgSet}(M_r, \tau_p[y])]$ then M_r has output a "large" program, and thus would become useless *)
5. If and when such p, y are discovered, let $\sigma_i^{s+1} = \tau_p[y]$ and go to stage $s+1$.

End stage s.
End of Definition of τ_k, for $k \in I_i$.

First note that τ is indeed a numbering. To compute $\tau_k(x)$, first determine the interval $I_i^{j'}$, such that $k \in I_i^{j'}$. Then, simulate the construction above, until the stage is reached where $j = j'$. If this stage is not reached, then $\tau_k(x)$ is undefined. Otherwise, the value of $\tau_k(x)$ can be computed from the assignment in step 3.

Next, note that there cannot be infinitely many stages, since the search in step 4 will, at the latest, not hold for $s = i$. Let s_i denote the last stage that is executed, and let $j_i = i - s_i$.

It is easy to observe that, τ_p is total, for $p \in \bigcup_{j=j_i}^{j=i} I_i^j$, and τ_p is everywhere undefined, for $p \in \bigcup_{j=0}^{j=j_i-1} I_i^j$. Moreover, all total functions in $\{\tau_k \mid k \in \mathbb{N}\}$ are pairwise different.

Let c be a constant such that, for all p, there exists a $p' \leqslant \max\{c \cdot p, c\}$, such that $\tau_p = \psi_{p'}$ (since ψ is a Kolmogorov numbering, there exists such a c). Let $i > c$. We note the following property about the functions τ_p, $p \in I_i^{j_i}$.

For all $r < i$, either ProgSet(M_r, τ_p) contains a ψ-program which is greater than or equal to $i \cdot l_i^{j_i+1} > i \cdot u_i^{j_i} \geqslant i \cdot p > c \cdot p$ (M_r was diagonalized against in some of the previous stages), or ProgSet(M_r, τ_p) contains only programs less than $i \cdot l_i^{j_i}$ (M_r could not be diagonalized against in any stage).

Thus M_r, $r < i$, can *MinIncEx*$_\psi$-identify only those τ_p in $\{\tau_p \mid l_i^{j_i} \leqslant p \leqslant u_i^{j_i}\}$, whose minimal ψ programs are $< i \cdot l_i^{j_i}$. Let

$$\mathscr{S}_i = \{\tau_p \mid l_i^{j_i} \leqslant p \leqslant u_i^{j_i} \wedge min_\psi(\tau_p) \geqslant i \cdot l_i^{j_i}\}.$$

It immediately follows that no machine M_r can *MinIncEx*$_\psi$-identify any function in $\mathscr{S}_i$, for $i > \max\{r, c\}$. We will construct our diagonalizing class as an infinite subset of $\bigcup_{i>c} \mathscr{S}_i$, using a trick used by Freivalds [7, 8]. For $i > c$, let

$$\mathscr{S}_i' = \{\tau_p \mid \tau_p \in \mathscr{S}_i \wedge |\{q \leqslant c \cdot u_i^{j_i} \mid \tau_p[|\sigma_i^{i-j_i}| + 1] \subseteq \psi_q\}| \leqslant 4c\}.$$

Note that $|\mathscr{S}_i'| \geqslant [u_i^{j_i} - l_i^{j_i}] - u_i^{j_i}/4 - i \cdot l_i^{j_i}$ ($i \cdot l_i^{j_i}$ term is for functions spoiled due to them having small ($< i \cdot l_i^{j_i}$) minimal programs in ψ; the $u_i^{j_i}/4$ term is due to the functions having more than $4c$ programs $\leqslant c \cdot u_i^{j_i}$ in the numbering ψ). Thus $\mathscr{S}_i'$ is non-empty (for $i > c$).

Let $\mathscr{C} = \bigcup_{i>c} \mathscr{S}_i'$. We now construct $M_1', M_2', \ldots, M_{4c}'$, such that at least one of these machines *MinDecEx*$_\psi$-infers an infinite subset of $\mathscr{C}$.

The idea of the construction of M_r' is as follows. Suppose the input function is $f \in \mathscr{S}_i'$ (i can be determined from $f(0)$). Thus f must be the same as τ_p, for some $p \in I_i^{j_i}$. Note that, $|\{x > 0 \mid \tau_p(x) \neq 0\}| = i - j + 1$, for $p \in I_i^j$, $j \geqslant j_i$. Moreover, if $\tau_p = f$, then $min_\psi(f)$ must lie in the interval $[i \cdot l_i^{j_i}, c \cdot u_i^{j_i}]$ (note that $c \cdot u_i^{j_i} \leqslant i \cdot u_i^{j_i} < i \cdot l_i^{j_i+1}$). This is what our construction uses. Note that σ_i^{i-j}, if defined, can be effectively determined. The following algorithm for M_r' will only use j (for further processing) such that σ_i^{i-j} is defined. So assume such a restriction on j. Let $n_i^j = |\sigma_i^{i-j}| + 1$ (note that $n_i^j - 1$ determines the point at which the functions in $\{\tau_p \mid p \in I_i^j\}$ differ).

Let $X_i^j = \{q \mid i \cdot l_i^j \leqslant q \leqslant c \cdot u_i^j$ and $f[n_i^j] \subseteq \psi_q\}$.

M_r' is defined as follows. $M_r'(f[n]) = ?$, for $n \leqslant 1$. For $n > 1$, let $j = i - |\{x \mid 0 < x < n \wedge f(x) \neq 0\}| + 1$; $M_r'(f[n])$ is then the rth element, if any, in a standard 1–1 enumeration of X_i^j.

It is easy to note that the conjectures of M_r' are monotonically decreasing (since $c \cdot u_i^j \leqslant i \cdot l_i^{j+1}$ – recall that $i > c$, and the j's as used in the definition of M_r' are monotonically decreasing). Moreover, at least one of the machines M_r', $1 \leqslant r \leqslant 4c$, *MinEx*$_\psi$-infers f (since $|X_i^{j_i}| \leqslant 4c$, for $f \in \mathscr{C}$, and $X_i^{j_i}$ contains a minimal program for f.) Thus for every function $f \in \mathscr{C}$, at least one of the machines $M_1', \ldots, M_{4c}'$ *MinDecEx*$_\psi$-infers f.

Since $\mathscr{C}$ is infinite, there exists a machine which $MinDecEx_\psi$-infers an infinite subset of $\mathscr{C}$. Since no infinite subset of $\mathscr{C}$ is in $MinIncEx_\psi$, the theorem follows. □

From Theorem 31 and Corollary 30 we immediately get the following result.

Corollary 32. *$MinIncEx_\psi \subset MinEx_\psi$ for every Kolmogorov numbering ψ.*

Theorem 33. *$MinIncEx_\psi \backslash MinSuper_\psi \neq \emptyset$ for every Kolmogorov numbering ψ.*

Proof. The idea of the proof is similar to that of the proof of Theorem 31 though there are subtle differences. Suppose a Kolmogorov numbering ψ is given. We will construct a numbering τ as follows.

For each $i \in \mathbb{N}$, $j \leqslant i$, we will define, l_i^j, u_i^j.

$l_0^0 = 1$.

$l_{i+1}^0 = u_i^i + 1$. For $j < i$, $l_i^{j+1} = u_i^j + 1$.

For $j \leqslant i$, $u_i^j = (l_i^0)^2(3i+1)^{5i+j}$.

For $j \leqslant i$, let $I_i^j = \{p \mid l_i^j \leqslant p \leqslant u_i^j\}$, and let $I_i = \bigcup_{j \leqslant i} I_i^j$.

Each τ_k will either be a total recursive function or the everywhere undefined function. We will now define the functions τ_k, for $k \in I_i$ (such a construction is carried out for each i separately). Intuitively, this will give us a collection of functions $\mathscr{S}_i$, such that none of the machines $M_0, M_1, \ldots, M_{i-1}$, will $MinSuper_\psi$-identify any of the functions in $\mathscr{S}_i$.

Definition of τ_k, for $k \in I_i$.
Let $\sigma_i^0(0) = i$ (σ_i^0 is of length 1). Go to stage 0.

Begin stage s.
1. Let $j = s$.
2. Let $m = |\sigma_i^s|$ (i.e. m is the least element not in the domain of σ_i^s).
3. For all $p \in I_i^j$, define τ_p as follows:

$$\tau_p(x) = \begin{cases} \sigma_s(x) & \text{if } < m, \\ p & \text{if } x = m, \\ 0 & \text{otherwise.} \end{cases}$$

4. Search for $p \in I_i^j$ and $y > m$, such that, $|\{r < i \mid |\{q \leqslant i \cdot u_i^i \mid q \notin \text{ProgSet}(M_r, \tau_p[y])\}| \leqslant \sqrt{i \cdot u_i^i}\}| > j$.
   (* Intuitively $|\{q \leqslant i \cdot u_i^i \mid q \notin \text{ProgSet}(M_r, \tau_p[y])\}| \leqslant \sqrt{i \cdot u_i^i}$ means that M_r can $MinSuper_\psi$-identify only "few" relevant functions *)
5. If and when such p, y are discovered, let $\sigma_i^{s+1} = \tau_p[y]$ and go to stage $s+1$.

End stage s.
End of Definition of τ_k, for $k \in I_i$.

First note that there cannot be infinitely many stages. In fact, if the construction reaches stage i, then the search at step 4 cannot succeed. Let s_i denote the last stage that is executed, and let $j_i = s_i$.

It is easy to observe that each τ_k is either a total function or the everywhere undefined function (for $p \in I_i^j$, τ_p is total iff $j \leqslant j_i$.) Moreover, total functions in the numbering τ are pairwise different.

Let c be a constant such that, for all p, there exists a $p' \leqslant \max\{c \cdot p, c\}$, such that $\tau_p = \psi_{p'}$ (since ψ is a Kolmogorov numbering, there exists such a c). Let $i > c$. We note the following property about the functions τ_p, $p \in I_i^{j_i}$.

For all $r < i$, either

(a) there exists an $S \subseteq \{x \mid x \leqslant i \cdot u_i^i\}$, such that $|S| < \sqrt{i \cdot u_i^i}$ and, for all $p \in I_i^{j_i}$, $\mathbb{N} \setminus \mathrm{ProgSet}(M_r, \tau_p) \subseteq S$ (M_r has been diagonalized against in previous stages) or

(b) for each $p \in I_i^{j_i}$, ProgSet(M_r, τ_p) does not contain at least $\sqrt{i \cdot u_i^i}$ programs $\leqslant i \cdot u_i^i$. (M_r was not diagonalized against in any of the stages)

In either case M_r, $r < i$, can *MinSuper*$_\psi$-identify at most $\sqrt{i \cdot u_i^i}$ of the functions in $\{\tau_p \mid l_i^{j_i} \leqslant p \leqslant u_i^{j_i}\}$ (in case (a), there are only $\sqrt{i \cdot u_i^i}$ possibilities for M_r to stabilize to; in case (b) there can be at most $\sqrt{i \cdot u_i^i}$ distinct programs for which M_r erases all the incorrect programs $\leqslant i \cdot u_i^i$).

Let $\mathscr{S}_i = \{\tau_p \mid l_i^{j_i} \leqslant p \leqslant u_i^{j_i} \wedge (\forall r < i)[\tau_p \notin \mathit{MinSuper}_\psi(M_r)]\}$.

It immediately follows that no machine M_r can *MinSuper*$_\psi$-identify any function in $\mathscr{S}_i$, for $i > \max\{r, c\}$.

We will construct our diagonalizing class as an infinite subset of $\bigcup_{i>c} \mathscr{S}_i$, using a trick used by Freivalds [7, 8]. For $i > c$, let

$$\mathscr{S}_i' = \{\tau_p \mid \tau_p \in \mathscr{S}_i \wedge \mathit{min}_\psi(\tau_p) \geqslant i \cdot l_i^{j_i} \wedge |\{q \leqslant c \cdot u_i^{j_i} \mid \tau_p[|\sigma_i^{j_i}| + 1] \subseteq \psi_q\}| \leqslant 4c\}.$$

Note that, for large enough i, $|\mathscr{S}_i'| \geqslant (u_i^{j_i} - l_i^{j_i}) - u_i^{j_i}/4 - i \cdot l_i^{j_i} - i\sqrt{i \cdot u_i^i} > 0$ ($i \cdot l_i^{j_i}$ term is for functions spoiled due to them having small ($< i \cdot l_i^{j_i}$) minimal program in ψ; the $u_i^{j_i}/4$ term is due to the functions having more than $4c$ programs $\leqslant c \cdot u_i^{j_i}$ in the numbering ψ; $i\sqrt{i \cdot u_i^i}$ term is for the functions which are *MinSuper*$_\psi$-identified by some M_r, $r < i$). Thus $\mathscr{S}_i'$ is non-empty for large enough $i > c$.

Let $\mathscr{C} = \bigcup_{i>c} \mathscr{S}_i'$. We now construct $M_1', M_2', \ldots, M_{4c}'$, such that at least one of these machines *MinIncEx*$_\psi$-infers an infinite subset of $\mathscr{C}$.

The idea of the construction of M_r' is as follows. Suppose the input function is $f \in \mathscr{S}_i'$ (i can be determined from $f(0)$). Thus f must be the same as τ_p, for some $p \in I_i^{j_i}$. Note that, $|\{x > 0 \mid \tau_p(x) \neq 0\}| = j + 1$, for $p \in I_i^j$, $j \leqslant j_i$. Moreover, for $f \in \mathscr{S}_i'$, $\mathit{min}_\psi(f)$ must lie in the interval $[i \cdot l_i^{j_i}, c \cdot u_i^{j_i}]$ (note that $c \cdot u_i^{j_i} \leqslant i \cdot u_i^{j_i} < i \cdot l_i^{j_i+1}$). This is what our construction uses.

Note that σ_i^j, if defined, can be effectively determined. The following algorithm for M_r' will only use j (for further processing) such that σ_i^j is defined. So assume such a restriction on j. Let $n_i^j = |\sigma_i^j| + 1$ (note that $n_i^j - 1$ determines the point at which the functions in $\{\tau_p \mid p \in I_i^j\}$ differ).

Let $X_i^j = \{q \mid i \cdot l_i^j \leqslant q \leqslant c \cdot u_i^j \wedge f[n_i^j] \subseteq \psi_q\}$.

M_r' is defined as follows. $M_r'(f[n]) = ?$, for $n \leqslant 1$. For $n > 1$, let $j = |\{x \mid 0 < x < n \wedge f(x) \neq 0\}| - 1$; $M_r'(f[n])$ is then the rth element, if any, in a standard 1–1 enumeration of X_i^j.

It is easy to note that the conjectures of M'_r are monotonically increasing (since $c \cdot u_i^j \leqslant i \cdot l_i^{j+1}$ – recall that $i > c$, and the j's as used in the definition of M'_r are monotonically increasing). Moreover, at least one of the machines M'_r, $1 \leqslant r \leqslant 4c$, MinEx_ψ-infers f (since $|X_i^{j_i}| \leqslant 4c$, for $f \in \mathscr{C}$, and $X_i^{j_i}$ contains a minimal program for f). Thus for every function $f \in \mathscr{C}$, at least one of the machines $M'_1, \ldots, M'_{4c}$ $\mathit{MinIncEx}_\psi$-infers f. Since $\mathscr{C}$ is infinite, there exists a machine which $\mathit{MinIncEx}_\psi$-infers an infinite subset of $\mathscr{C}$. Since no infinite subset of $\mathscr{C}$ is in $\mathit{MinSuper}_\psi$, the theorem follows. □

We are now ready to prove Theorem 24.

Proof of Theorem 24. Let ψ be any Kolmogorov numbering.
$\mathit{MinSuper}_\psi \subset \mathit{MinIncEx}_\psi$ can be seen as follows. By Proposition 10, $\mathit{MinSuper}_\psi \subseteq \mathit{MinSub}_\psi$. By Proposition 9, $\mathit{MinIncEx}_\psi = \mathit{MinSub}_\psi$. Thus, $\mathit{MinSuper}_\psi \subseteq \mathit{MinIncEx}_\psi$. Furthermore, by Theorem 33, $\mathit{MinIncEx}_\psi \backslash \mathit{MinSuper}_\psi \neq \emptyset$.
$\mathit{MinAll}_\psi \subset \mathit{MinIncEx}_\psi$ is proved as follows. By definition and Propositions 9 and 10, $\mathit{MinAll}_\psi \subseteq \mathit{MinSuper}_\psi \subseteq \mathit{MinSub}_\psi = \mathit{MinIncEx}_\psi$. By Theorem 33, we have $\mathit{MinIncEx}_\psi \backslash \mathit{MinSuper}_\psi \neq \emptyset$; thus $\mathit{MinIncEx}_\psi \backslash \mathit{MinAll}_\psi \neq \emptyset$.

Next we show $\mathit{MinSuper}_\psi \# \mathit{MinDecEx}_\psi$. By Theorem 28, we have $\mathit{MinAll}_\psi \backslash \mathit{MinDecEx}_\psi \neq \emptyset$. Since, by definition, $\mathit{MinAll}_\psi \subseteq \mathit{MinSuper}_\psi$, we obtain $\mathit{MinSuper}_\psi \backslash \mathit{MinDecEx}_\psi \neq \emptyset$. Conversely, by Theorem 31, $\mathit{MinDecEx}_\psi \backslash \mathit{MinIncEx}_\psi \neq \emptyset$. Since, by Propositions 9 and 10, $\mathit{MinSuper}_\psi \subseteq \mathit{MinSub}_\psi = \mathit{MinIncEx}_\psi$, we get $\mathit{MinDecEx}_\psi \backslash \mathit{MinSuper}_\psi \neq \emptyset$.

$\mathit{MinAll}_\psi \# \mathit{MinDecEx}_\psi$ follows, since, by Theorem 28, we have $\mathit{MinAll}_\psi \backslash \mathit{MinDecEx}_\psi \neq \emptyset$, and, by Theorem 31, $\mathit{MinDecEx}_\psi \backslash \mathit{MinIncEx}_\psi \neq \emptyset$. As above, $\mathit{MinAll}_\psi \subseteq \mathit{MinIncEx}_\psi$; hence $\mathit{MinDecEx}_\psi \backslash \mathit{MinAll}_\psi \neq \emptyset$.

$\mathit{MinIncEx}_\psi \# \mathit{MinDecEx}_\psi$ is a direct consequence of Corollary 29 and Theorem 31. □

From Theorem 24 parts (1), (2), (5) and Proposition 9 we immediately get the following corollary.

Corollary 34. *For every Kolmogorov numbering* ψ,
(1) $\mathit{MinSuper}_\psi \subset \mathit{MinSub}_\psi$,
(2) $\mathit{MinAll}_\psi \subset \mathit{MinSub}_\psi$,
(3) $\mathit{MinSub}_\psi \# \mathit{MinDecEx}_\psi$.

The only separation which is not given by Theorem 24 and Corollary 34 concerns the types *MinAll* and *MinSuper*. We conjecture that for every Kolmogorov numbering ψ, $\mathit{MinAll}_\psi \subset \mathit{MinSuper}_\psi$. However, at present we only know that these types are separated in *some* Kolmogorov numbering. In the following we exhibit such a numbering.

Theorem 35. *There is a Kolmogorov numbering* ψ *and* $\mathscr{C} \in \mathit{MinSuper}_\psi$ *such that for every Gödel numbering* η, $\mathscr{C} \notin \mathit{MinAll}_\eta$.

Proof. Without loss of generality, suppose that φ is a Kolmogorov numbering. We will construct a Kolmogorov numbering ψ and class $\mathscr{C}$ witnessing the theorem.

Let $\psi_{7i} = \varphi_i$. Note that this makes ψ a Kolmogorov numbering. Define h as follows: $h(0) = 1$, and for all $i \geqslant 0$, $h(i+1) = 3(h(i)+1)$.

Note that for any Gödel numbering η, there must be a 1–1, increasing, recursive function φ_j witnessing the reduction from ψ to η. Thus we will try to diagonalize against all pairs of machines, M_i, and potential reduction functions, φ_j.

Let $S_k = \{p \mid (\exists l \mid h(k) < l \leqslant h(k+1))(\exists r \mid 0 < r < 7)[p = 7l + r]\}$. Intuitively, S_k denotes the kth set of available programs for diagonalization. We will use the programs in the set S_k for diagonalization against machine M_i and reduction function φ_j, where $k = \langle i, j\rangle$. It will be the case that all functions computed by programs in S_k will be total functions.

Let $\mathscr{C} = \{\psi_p \mid p \in \bigcup_k S_k \wedge p = min_\psi(\psi_p)\}$.

Note that totality of all functions computed by ψ_p, $p \in \bigcup_k S_k$, immediately implies that $\mathscr{C} \in MinSuper_\psi$. For each $k = \langle i, j\rangle$, we will construct the functions computed by ψ_p, $p \in S_k$, in such a way that, if φ_j is an increasing function witnessing a reduction from ψ to Gödel numbering η, then, for at least one $p \in S_k$, we get $\psi_p \in \mathscr{C} \backslash MinAll_\eta(M_i)$. This would prove that $\mathscr{C} \notin MinAll_\eta$, for any Gödel numbering η.

For each $r \in \mathbb{N}$, let f_r denote the constant function $f_r(x) = r$. For $w \in \{1,2,3\}$, and l such that $h(k) < l \leqslant h(k+1)$, we will now define the functions ψ_{7l+w} and ψ_{7l+w+3}.

$$\psi_{7l+w} = f_{7l+w}.$$

$$\psi_{7l+w+3}(x) = \begin{cases} 7l+w & \text{if } x = 0, \\ f_{7l+w}(x) & \text{if } x > 0 \text{ and } \Phi_j(7l+w+3) > x, \\ f_{7l+w}(x) & \text{if } x > 0 \text{ and } \Phi_j(7l+w+3) \leqslant x \text{ and} \\ & |\{q \leqslant \varphi_j(7l+w+3) \mid q \notin \\ & \quad \mathrm{ProgSet}(M_i, f_{7l+w}[x])\}| > 1, \\ 7l+w+3 & \text{otherwise.} \end{cases}$$

It is easy to verify that all functions computed by programs in $\bigcup_k S_k$ are total. Suppose φ_j, is a 1–1, increasing, recursive function which witnesses the reduction between ψ and η. Then M_i does not $MinAll_\eta$-identify at least one of ψ_{7l+w} and ψ_{7l+w+3}. Moreover, for each k, there exists an l, $h(k) < l \leqslant h(k+1)$, and a $w \in \{1,2,3\}$, such that for all $x \leqslant h(k+1)$, $\varphi_x(0) \neq 7l+w$. (This holds since the number of such pairs l, w is $3(h(k+1) - h(k)) > h(k+1)+1$). It immediately follows that there exists a $p \in S_k$, such that $\psi_p \in \mathscr{C}$, and M_i does not $MinAll_\eta$-identify ψ_p. $\square$

As a corollary, we get the following separation.

Corollary 36. *There is a Kolmogorov numbering ψ such that $MinAll_\psi \subset MinSuper_\psi$.*

Proof. By definition, $MinAll_\psi \subseteq MinSuper_\psi$ for every numbering ψ. The proper inclusion follows from Theorem 35. $\square$

Our main aim in this section was to separate all the minimal identification criteria in each Kolmogorov numbering. As an aside we note that a variant of the proof of Theorem 11 in [15] can be used to show that there is an infinite class $\mathscr{C} \subseteq \mathscr{H}$ such that for every Kolmogorov numbering ψ, $\mathscr{C} \in \mathit{MinFin}_{\psi}$. Hence Corollary 29 can be strengthened in the following way.

Theorem 37. *There is $\mathscr{C} \subseteq \mathscr{R}$ such that for every Kolmogorov numbering ψ, $\mathscr{C} \in \mathit{MinIncEx}_{\psi} \setminus \mathit{MinDecEx}_{\psi}$.*

Thus the *same* class could be used for diagonalization for all Kolmogorov numberings. Such a result can also be obtained for Theorem 28. However, we do not yet know whether we could use the same class for other diagonalizations in this section.

We now note that a result similar to Theorem 35 can also be proved for *MinDecEx* versus *MinIncEx*.

Theorem 38. *There are a Kolmogorov numbering ψ and a $\mathscr{C} \in \mathit{MinDecEx}_{\psi}$ such that for every Gödel numbering η, $\mathscr{C} \notin \mathit{MinIncEx}_{\eta}$.*

Proof. Let $\mathscr{C} = \{f \in \mathscr{R} \mid (\forall x)[f(x) = f(0)]\}$, the class of constant functions. We first construct a Kolmogorov numbering ψ such that $\mathscr{C} \in \mathit{MinDecEx}_{\psi}$. Let ψ be defined as follows.

Without loss of generality, suppose that φ is a Kolmogorov numbering. Let $\psi_{2i+1} = \varphi_i$. Note that this makes ψ a Kolmogorov numbering. For all x, let $\psi_{2i}(x) = \varphi_i(0)$. Consider the following machine M.

$M(f[0]) = ?$. For $n > 0$,

$$M(f[n]) = \begin{cases} 2j & \text{if} (\exists i \leqslant n)[\Phi_i(0) \leqslant n \wedge \varphi_i(0) = f(0)] \textit{ and} \\ & j = \min\{i \leqslant n \mid \Phi_i(0) \leqslant n \wedge \varphi_i(0) = f(0)\}, \\ ? & \textit{otherwise.} \end{cases}$$

It is easy to verify that M $\mathit{MinDecEx}_{\psi}$-infers $\mathscr{C}$.

Let η be any Gödel numbering and M be any machine. We now show that M cannot $\mathit{MinIncEx}_{\eta}$-identify $\mathscr{C}$. This would prove the theorem.

By implicit use of Kleene's recursion theorem, there exists an e such that η_e may be defined as follows. Let f_c denote the function, $f_c(x) = c$, for all x.

Definition of η_e.
1. Search for $c, n \in \mathbb{N}$, such that $M(f_c[n]) > e$.
2. If and when such c, n are found, let $\eta_e = f_c$.

End of Definition of η_e.

Note that if step 1 does not succeed then M can $\mathit{MinIncEx}_{\eta}$-identify only finitely many functions in $\mathscr{C}$. On the other hand, if step 1 search succeeds then, clearly, $\mathit{min}_{\eta}(f_c) \leqslant e$. However, since M on f_c outputs a program larger than e, M cannot $\mathit{MinIncEx}_{\eta}$-identify f_c. It follows that M does not $\mathit{MinIncEx}_{\eta}$-identify $\mathscr{C}$. $\square$

We now present a result on $MinAll_\psi$-learning of *recursively enumerable* classes in Kolmogorov numberings.

Theorem 39. *For any infinite r.e. $\mathscr{U} \subseteq \mathscr{R}$, there are an infinite $\mathscr{V} \subseteq \mathscr{U}$ and a Kolmogorov numbering ψ such that $\mathscr{V} \in MinAll_\psi$.*

Proof. Let $\eta \in \mathscr{R}^2$ be a 1–1 numbering such that $\mathscr{U} = \{\eta_i \mid i \in \mathbb{N}\}$. Since η is 1–1, it is easy to see that there is a machine M $MinAll_\eta$-learning $\mathscr{U}$. Let β be a Kolmogorov numbering. Define ψ as follows. $\psi_{3i} = \beta_i$, $\psi_{3i+1} = \eta_{2i}$ and $\psi_{3i+2} = \eta_{2i+1}$. Let M' be such that $\mathrm{ProgSet}(M', f) = \{3i \mid i \in \mathbb{N}\} \cup \{3i + 1 \mid 2i \in \mathrm{ProgSet}(M, f)\} \cup \{3i + 2 \mid 2i + 1 \in \mathrm{ProgSet}(M, f)\}$.

Let $\mathscr{V} = \{\eta_{2i} \mid (\forall j \leqslant i)[\beta_j \neq \eta_{2i}]\} \cup \{\eta_{2i+1} \mid (\forall j \leqslant i)[\beta_j \neq \eta_{2i+1}]\}$. It is easy to verify that $\mathscr{V}$ is infinite and M' $MinAll_\psi$-infers $\mathscr{V}$. □

Finally, we compare the type *MinFin* with the types of learning by erasing.

Corollary 40. *For every Kolmogorov numbering ψ,*
(1) $MinFin_\psi \subset MinAll_\psi$,
(2) $MinFin_\psi \subset MinDecEx_\psi$.

Proof. $MinFin_\psi \subseteq MinAll_\psi$ by Proposition 10. Moreover, $MinFin_\psi \subseteq MinDecEx_\psi$ by Proposition 10 and $MinAll_\psi \setminus MinDecEx_\psi \neq \emptyset$ by Theorem 28. Hence $MinFin_\psi \subset MinAll_\psi$.

$MinFin_\psi = MinIncEx_\psi \cap MinDecEx_\psi$ by Proposition 9. Furthermore, we have $MinIncEx_\psi \# MinDecEx_\psi$ by Theorem 24, Assertion (5). Consequently, $MinFin_\psi \subset MinDecEx_\psi$. □

4.3. Gödel numberings as hypothesis spaces

In this section we prove that some of the diagonalizations shown for arbitrary Kolmogorov numberings in the previous section may not hold for every Gödel numbering.

Theorem 41. *There exists a Gödel numbering ψ such that $MinFin_\psi = MinAll_\psi = MinSuper_\psi = MinIncEx_\psi = MinDecEx_\psi = MinEx_\psi$.*

Proof. Taking a Gödel numbering ψ such that $MinEx_\psi$ contains only finite classes of functions (cf. Freivalds [6]) gives the theorem. □

In fact, using the above theorem, for every "reasonable" relationship between the minimal criteria considered in this paper, we can construct a Gödel numbering in which this relationship does hold. The essential idea is to interleave the needed diagonalizations in Theorems 43–47, with the numbering generated in Theorem 41 (cf. [9] used a similar trick to generate Gödel numberings for any reasonable relationship between $MinFin_\psi$, $MinAll_\psi$, $MinEx_\psi$). We omit the details.

Theorem 42. *Suppose $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \beta_1, \beta_2 \in \{\subset, =\}$ such that β_1 and β_2 are both '$=$' iff $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are all '$=$'. Then there exists a Gödel numbering ψ such that*
(1) *$MinFin_\psi$ α_1 $MinAll_\psi$ α_2 $MinSuper_\psi$ α_3 $MinIncEx_\psi$ α_4 $MinEx_\psi$, and*
(2) *$MinFin_\psi$ β_1 $MinDecEx_\psi$ β_2 $MinEx_\psi$.*

We now prove the theorems on non-Gödel numberings needed for the proof of Theorem 42.

Theorem 43. *There is a numbering ψ such that*
(1) *$\mathscr{R}_\psi \in MinIncEx_\psi$ and $|\mathscr{R}_\psi| = \infty$,*
(2) *for any $\mathscr{S} \subseteq \mathscr{R}_\psi$ with $|\mathscr{S}| = \infty$, $\mathscr{S} \notin MinDecEx_\psi \cup Super_\psi$.*

Proof. We exploit the fact that for $Super_\psi$-identification one needs to erase all the incorrect programs.

For $i \in \mathbb{N}$, define f_i as follows: $f_i(0) = i$, and, for $x > 0$, let $f_i(x) = 0$.
Let $\mathscr{C} = \{f_i \mid i \in \mathbb{N}\}$. Let $h(0) = 0$. Let $h(i+1) = h(i) + 2i + 1$.

We will make sure that (a), (b) and (c) are satisfied.
(a) For each i, exactly one of the programs in $S_i = \{p \mid h(i) < p \leqslant h(i+1)\}$ will compute f_i. All the other programs in S_i will compute non-total functions. This will make $\mathscr{R}_\psi = \mathscr{C}$.
(b) For $j < i$, M_j does not $MinDecEx_\psi$-identify or $Super_\psi$-identify f_i.
(c) $\mathscr{C} \in MinIncEx_\psi$.
This will prove the theorem.

Fix i. We now define ψ_j, for $j \in S_i$.

```
Definition of $\psi_j$, for $j \in S_i$.
Let $\mathrm{CancelS}_i^0 = \mathrm{CancelD}_i^0 = \emptyset$.
Go to stage 0.
Begin stage $s$
1. Let $p(i,s) = h(i) + 1 + |\mathrm{CancelS}_i^s| + |\mathrm{CancelD}_i^s|$.
2. For all $x \leqslant s$, let $\psi_{p(i,s)}(x) = f_i(x)$.
   (Intuitively, we want to make $\psi_{p(i,\infty)} = f_i$).
3. Let $\mathrm{CancelD}_i^{s+1} = \mathrm{CancelD}_i^s \cup \{j < i \mid p(i,s) \in \mathrm{ProgSet}(M_j, f_i[s])\}$.
4. Let $\mathrm{CancelS}_i^{s+1} = \mathrm{CancelS}_i^s \cup \{j < i \mid \{l \leqslant h(i+1) \mid l \neq p(i,s)\} \subseteq \mathrm{ProgSet}(M_j, f_i[s])\}$.
5. Go to stage $s+1$.
End stage $s$.
End of Definition of $\psi_j$, for $j \in S_i$.
```

It is easy to verify that $\mathrm{CancelD}_i^s$, $\mathrm{CancelS}_i^s$ are monotonically non-decreasing in s with respect to $\subseteq$. Let $\mathrm{CancelD}_i^\infty = \lim_{s\to\infty} \mathrm{CancelD}_i^s$ and $\mathrm{CancelS}_i^\infty = \lim_{s\to\infty} \mathrm{CancelS}_i^s$. Note that $p(i,s)$ is monotonically non-decreasing in s. Let $p(i,\infty) = \lim_{n\to\infty} p(i,s)$. Note that $h(i) < p(i,s) \leqslant h(i) + 1 + 2i \leqslant h(i+1)$. Also, $\psi_{p(i,\infty)} = f_i$, and for all $j \in S_i \setminus \{p(i,\infty)\}$, ψ_j is a non-total function. Thus (a) is satisfied.

Consider any $j<i$. If $j \in \mathrm{CancelD}_i^s$, then $\mathrm{ProgSet}(M_j, f_i)$ contains a program smaller than $p(i,s) \leqslant p(i,\infty)$. If $j \notin \mathrm{CancelD}_i^{\infty}$, then $\mathrm{ProgSet}(M_j, f_i)$ does not contain $p(i,\infty)$. In either case M_j does not $\mathit{MinDecEx}_{\psi}$-identify $\psi_{p(i,\infty)} = f_i$. Similarly, if $j \in \mathrm{CancelS}_i^s$, then $\mathrm{ProgSet}(M_j, f_i)$ contains $p(i,\infty)$. If $j \notin \mathrm{CancelS}_i^{\infty}$, then $\{l \leqslant h(i+1) \mid l \neq p(i,\infty)\} \nsubseteq \mathrm{ProgSet}(M_j, f_i)$. In either case M_j does not Super_{ψ}-identify $\psi_{p(i,\infty)} = f_i$. Thus (b) is satisfied.

To show that $\mathscr{C} \in \mathit{MinIncEx}_{\psi}$, note that $p(i,s)$ is a monotonically non-decreasing function of s, which converges to the minimal ψ-program for f_i. Thus $\mathscr{C} \in \mathit{MinIncEx}_{\psi}$, and (c) is satisfied. □

Theorem 44. *There is a numbering ψ such that*
(1) *$\mathscr{R}_{\psi} \in \mathit{MinSuper}_{\psi}$ and $|\mathscr{R}_{\psi}| = \infty$,*
(2) *for any $\mathscr{S} \subseteq \mathscr{R}_{\psi}$ with $|\mathscr{S}| = \infty$, $\mathscr{S} \notin \mathit{MinAll}_{\psi} \cup \mathit{MinDecEx}_{\psi}$.*

Proof. The proof of this theorem is similar to the proof of Theorem 43.

For $i \in \mathbb{N}$, define f_i as follows: $f_i(0) = i$, and, for $x>0$, let $f_i(x) = 0$.

Let $\mathscr{C} = \{f_i \mid i \in \mathbb{N}\}$. Let $h(0) = 0$. Let $h(i+1) = h(i) + 2i + 1$.

We will make sure that (a)–(c) are satisfied.

(a) For each i, there exists a $j \in S_i = \{p \mid h(i) < p \leqslant h(i+1)\}$, such that
(a.1) $(\forall j' \mid j \leqslant j' \leqslant h(i+1))[\psi_{j'} = f_i]$ and (a.2) $(\forall j' \mid h(i) < j' < j)[\psi_{j'} \notin \mathscr{R}]$. This will make $\mathscr{R}_{\psi} = \mathscr{C}$.
(b) For $j<i$, M_j does not $\mathit{MinDecEx}_{\psi}$-identify or MinAll_{ψ}-identify f_i.
(c) $\mathscr{C} \in \mathit{MinSuper}_{\psi}$.
This will prove the theorem.

Fix i. We now define ψ_j, for $j \in S_i$.
Definition of ψ_j, for $j \in S_i$.

```
Let $\mathrm{CancelA}_i^0 = \mathrm{CancelD}_i^0 = \emptyset$.
Go to stage 0.
Begin stage $s$
1. Let $p(i,s) = h(i) + 1 + |\mathrm{CancelA}_i^s| + |\mathrm{CancelD}_i^s|$.
2. For all $j'$ such that $p(i,s) \leqslant j' \leqslant h(i+1)$, for all $x \leqslant s$, let $\psi_{j'}(x) = f_i(x)$. (Intuitively,
   we want to make $p(i,\infty)$ as $j$ in clause (a)).
3. Let $\mathrm{CancelD}_i^{s+1} = \mathrm{CancelD}_i^s \cup \{j<i \mid p(i,s) \in \mathrm{ProgSet}(M_j, f_i[s])\}$.
4. Let $\mathrm{CancelA}_i^{s+1} = \mathrm{CancelA}_i^s \cup \{j<i \mid \{l \leqslant h(i+1) \mid l \neq p(i,s)\} \subseteq \mathrm{ProgSet}(M_j, f_i)\}$.
5. Go to stage $s+1$.
End stage $s$.
End of Definition of $\psi_j$, for $j \in S_i$.
```

It is easy to verify that $\mathrm{CancelD}_i^s$, $\mathrm{CancelA}_i^s$ are monotonically non-decreasing in s with respect to $\subseteq$. Let $\mathrm{CancelD}_i^{\infty} = \lim_{s\to\infty} \mathrm{CancelD}_i^s$ and $\mathrm{CancelA}_i^{\infty} = \lim_{s\to\infty} \mathrm{CancelA}_i^s$. Note that $p(i,s)$ is monotonically non-decreasing in s. Let $p(i,\infty) = \lim_{s\to\infty} p(i,s)$. Note that $h(i) < p(i,s) \leqslant h(i) + 2i + 1 \leqslant h(i+1)$. Also, for all j', such that $p(i,\infty) \leqslant j' \leqslant h(i+1)$, $\psi_{j'} = f_i$, and for all j' such that $h(i) < j' < p(i,\infty)$, $\psi_{j'}$ is a non-total function. Thus (a) is satisfied.

Consider any $j < i$. If $j \in \mathrm{CancelD}_i^s$, then $\mathrm{ProgSet}(M_j, f_i)$ contains a program smaller than $p(i,s) \leqslant p(i,\infty)$. If $j \notin \mathrm{CancelD}_i^\infty$, then $\mathrm{ProgSet}(M_j, f_i)$ does not contain $p(i,\infty)$. In either case M_j does not $\mathit{MinDecEx}_\psi$-identify $\psi_{p(i,\infty)} = f_i$. Similarly, if $j \in \mathrm{CancelA}_i^s$, then $\mathrm{ProgSet}(M_j, f_i)$ contains $p(i,\infty)$. If $j \notin \mathrm{CancelA}_i^\infty$, then $\{l \leqslant h(i+1) \mid l \neq p(i,\infty)\} \nsubseteq \mathrm{ProgSet}(M_j, f_i)$. In either case M_j does not MinAll_ψ-identify $\psi_{p(i,\infty)} = f_i$. Thus (b) is satisfied.

To show that $\mathscr{C} \in \mathit{MinSuper}_\psi$, note that $p(i,s)$ is a monotonically non-decreasing function of s, which converges to $\mathit{min}_\psi(f_i)$. Moreover, for all j', such that $p(i,\infty) \leqslant j' \leqslant h(i+1)$, $\psi_{j'} = f_i$. Thus $\mathscr{C} \in \mathit{MinSuper}_\psi$, and hence (c) is satisfied. □

Theorem 45. *There is a numbering ψ such that*
(1) $\mathscr{R}_\psi \in \mathit{MinDecEx}_\psi$ *and* $|\mathscr{R}_\psi| = \infty$,
(2) *for any* $\mathscr{S} \subseteq \mathscr{R}_\psi$ *with* $|\mathscr{S}| = \infty$, $\mathscr{S} \notin \mathit{MinIncEx}_\psi$.

Proof. The proof of this theorem is similar to the proof of Theorem 43, with some minor changes.

For $i \in \mathbb{N}$, define f_i as follows: $f_i(0) = i$, and, for $x > 0$, let $f_i(x) = 0$.

Let $\mathscr{C} = \{f_i \mid i \in \mathbb{N}\}$. Let $h(0) = 0$. Let $h(i+1) = h(i) + 2i + 1$.

We will make sure that (a)–(c) are satisfied.

(a) For each i, exactly one of the programs in $S_i = \{p \mid h(i) < p \leqslant h(i+1)\}$ will compute f_i. All the other programs in S_i will compute non-total functions. This will make $\mathscr{R}_\psi = \mathscr{C}$.

(b) For $j < i$, M_j does not $\mathit{MinIncEx}_\psi$-identify f_i.

(c) $\mathscr{C} \in \mathit{MinDecEx}_\psi$.

This will prove the theorem.
Fix i. We now define ψ_j, for $j \in S_i$.
Definition of ψ_j, for $j \in S_i$.

```
Let CancelI$_i^0 = \emptyset$.
Go to stage 0.
Begin stage $s$
1. Let $p(i,s) = h(i+1) - |\mathrm{CancelI}_i^s|$.
2. For all $x \leqslant s$, let $\psi_{p(i,s)}(x) = f_i(x)$.
   (Intuitively, we want to make $\psi_{p(i,\infty)} = f_i$).
3. Let $\mathrm{CancelI}_i^{s+1} = \mathrm{CancelI}_i^s \cup \{j < i \mid p(i,s) \in \mathrm{ProgSet}(M_j, f_i[s])\}$.
4. Go to stage $s+1$.
End stage $s$.
End of Definition of $\psi_j$, for $j \in S_i$.
```

It is easy to verify that $\mathrm{CancelI}_i^s$ is monotonically non-decreasing in s with respect to $\subseteq$. Let $\mathrm{CancelI}_i^\infty = \lim_{s\to\infty} \mathrm{CancelI}_i^s$. Note that $p(i,s)$ is monotonically non-increasing in s. Let $p(i,\infty) = \lim_{s\to\infty} p(i,s)$. Note that $h(i+1) \geqslant p(i,s) \geqslant h(i+1) - i > h(i)$. Also, $\psi_{p(i,\infty)} = f_i$, and for all $j \in S_i \backslash \{p(i,\infty)\}$, ψ_j is a non-total function. Thus (a) is satisfied.

Consider any $j<i$. If $j \in \mathrm{CancelI}_i^s$, then $\mathrm{ProgSet}(M_j, f_i)$ contains a program $> p(i,s) \geqslant p(i,\infty)$. If $j \notin \mathrm{CancelI}_i^\infty$, then $\mathrm{ProgSet}(M_j, f_i)$ does not contain $p(i,\infty)$. In either case M_j does not $\mathit{MinIncEx}_\psi$-identify $\psi_{p(i,\infty)} = f_i$. Thus (b) is satisfied.

To show that $\mathscr{C} \in \mathit{MinDecEx}_\psi$, note that $p(i,s)$ is a monotonically non-increasing function of s, which converges to $\mathit{min}_\psi(f_i)$. Thus $\mathscr{C} \in \mathit{MinDecEx}_\psi$, and hence (c) is satisfied. □

Theorem 46. *There is a numbering* ψ *such that*
(1) $\mathscr{R}_\psi \in \mathit{MinAll}_\psi$ *and* $|\mathscr{R}_\psi| = \infty$,
(2) *for any* $\mathscr{S} \subseteq \mathscr{R}_\psi$ *with* $|\mathscr{S}| = \infty$, $\mathscr{S} \notin \mathit{MinDecEx}_\psi$.

Proof. We will construct a numbering ψ, $\psi \in \mathscr{R}^2$, such that
(a) ψ is 1–1, and
(b) no infinite subset of $\mathscr{R}_\psi$ is in $\mathit{MinDecEx}_\psi$.

This would prove the theorem, since for any 1–1 total numbering ψ, $\mathscr{R}_\psi \in \mathit{MinSuper}_\psi = \mathit{MinAll}_\psi$.

If there exists a σ such that $M_0(\sigma) \neq ?$, then let σ_0 be the least such σ; otherwise, let $\sigma_0 = \Lambda$. For $i>0$, if there exists a $\sigma \supseteq \sigma_{i-1}$ such that $M_i(\sigma) \neq ?$, then let σ_i be the least such σ; otherwise, let $\sigma_i = \sigma_{i-1}$. Note that σ_i can be determined effectively in the limit. Let τ_i^j be such that
(i) τ_i^j can be determined effectively from i and j,
(ii) $\tau_i^j \subseteq \tau_{i+1}^j$, and
(iii) $\lim_{j\to\infty} \tau_i^j = \sigma_i$.
Note that there exist such τ_i^j.
Let ψ_i be defined as follows.

$$\psi_i(x) = \begin{cases} \tau_i^i(x) & \text{if } x < |\tau_i^i|, \\ i & \text{otherwise.} \end{cases}$$

It is easy to verify that each $\psi_i \in \mathscr{R}$, and ψ_i's are pairwise different. Hence (a) is satisfied. Note that for each i, for all but finitely many j, $\sigma_i \subseteq \tau_j^j$. Thus, for all i,
(c) for all but finitely many j, $\sigma_i \subseteq \psi_j$,
(d) either $M_i(\sigma_i) \neq ?$, or $(\forall \sigma \supseteq \sigma_i)[M_i(\sigma) = ?]$.

It follows immediately that M_i can $\mathit{MinDecEx}_\psi$-identify at most finitely many ψ_j. Thus (b) is satisfied. □

Theorem 47. *There is a numbering* ψ *such that*
(1) $\mathscr{R}_\psi \in \mathit{MinEx}_\psi$ *and* $|\mathscr{R}_\psi| = \infty$,
(2) *for any* $\mathscr{S} \subseteq \mathscr{R}_\psi$ *with* $|\mathscr{S}| = \infty$, $\mathscr{S} \notin \mathit{MinDecEx}_\psi \cup \mathit{MinIncEx}_\psi$.

Proof. This is a somewhat more complicated modification of the proof of Theorem 43. For $i \in \mathbb{N}$, define f_i as follows: $f_i(0) = i$, and, for $x>0$, let $f_i(x) = 0$.
Let $\mathscr{C} = \{f_i \mid i \in \mathbb{N}\}$. Let $h(0) = 0$. Let $h(i+1) = h(i) + 2^{2i+2} - 1$.
We will make sure that (a)–(c) are satisfied.

(a) For each i, exactly one of the programs in $S_i = \{p \mid h(i) < p \leqslant h(i+1)\}$ will compute f_i. All the other programs in S_i will compute non-total functions. This will make $\mathscr{R}_\psi = \mathscr{C}$.

(b) For $j < i$, M_j does not $\mathit{MinDecEx}_\psi$-identify or $\mathit{MinIncEx}_\psi$-identify f_i.

(c) $\mathscr{C} \in \mathit{MinEx}_\psi$.

This will prove the theorem.

Fix i. We now define ψ_j, for $j \in S_i$.

```
Definition of $\psi_j$, for $j \in S_i$.
Let $\mathrm{CancelI}_i^0 = \mathrm{CancelD}_i^0 = \emptyset$.
$l(i,0) = h(i) + 1$. $u(i,0) = h(i+1)$.
Go to stage 0.
Begin stage $s$
1. Let $p(i,s) = \frac{l(i,s)+u(i,s)}{2}$.
2. For all $x \leqslant s$, let $\psi_{p(i,s)}(x) = f_i(x)$.
   (Intuitively, we want to make $\psi_{p(i,\infty)} = f_i$).
3. if there exists a $j < i$ such that $j \notin \mathrm{CancelD}_i^s$ and $p(i,s) \in$
      $\mathrm{ProgSet}(M_j, f_i[s])$
   then let $l(i,s) = p(i,s) + 1$ and $\mathrm{CancelD}_i^{s+1} = \mathrm{CancelD}_i^s \cup \{j < i \mid p(i,s) \in$
      $\mathrm{ProgSet}(M_j, f_i[s])\}$
4. elseif there exists a $j < i$ such that $j \notin \mathrm{CancelI}_i^s$ and $p(i,s) \in$
      $\mathrm{ProgSet}(M_j, f_i[s])$
   then let $u(i,s) = p(i,s) - 1$ and $\mathrm{CancelI}_i^{s+1} = \mathrm{CancelI}_i^s \cup \{j < i \mid p(i,s) \in$
      $\mathrm{ProgSet}(M_j, f_i[s])\}$
   endif
5. Go to stage $s + 1$.
End stage $s$.
  End of Definition of $\psi_j$, for $j \in S_i$.
```

It is easy to verify that $\mathrm{CancelD}_i^s$, $\mathrm{CancelI}_i^s$ are monotonically non-decreasing in s with respect to $\subseteq$. Let $\mathrm{CancelD}_i^\infty = \lim_{s\to\infty} \mathrm{CancelD}_i^s$ and $\mathrm{CancelI}_i^\infty = \lim_{s\to\infty} \mathrm{CancelI}_i^s$. Also it is easy to verify that l_i^s is monotonically non-decreasing and u_i^s is monotonically non-increasing in s. Let $u_i^\infty = \lim_{s\to\infty} u_i^s$, and $l_i^\infty = \lim_{s\to\infty} l_i^s$. Let $p(i,\infty) = \lim_{s\to\infty} p(i,s) = (l_i^\infty + u_i^\infty)/2$. Note that $h(i) < l_i^s < p(i,s) \leqslant u_i^s \leqslant h(i+1)$. Also, $\psi_{p(i,\infty)} = f_i$, and for all $j \in S_i \backslash \{p(i,\infty)\}$, ψ_j is a non-total function. Thus (a) is satisfied.

Consider any $j < i$. If $j \in \mathrm{CancelD}_i^s$, then $\mathrm{ProgSet}(M_j, f_i)$ contains a program $< l(i,s) \leqslant l(i,\infty) < p(i,\infty)$. If $j \notin \mathrm{CancelD}_i^\infty$, then $\mathrm{ProgSet}(M_j, f_i)$ does not contain $p(i,\infty)$. In either case M_j does not $\mathit{MinDecEx}_\psi$-identify $\psi_{p(i,\infty)} = f_i$. Similarly, if $j \in \mathrm{CancelI}_i^s$, then $\mathrm{ProgSet}(M_j, f_i)$ contains a program $> u(i,s) \geqslant u(i,\infty) > p(i,\infty)$. If $j \notin \mathrm{CancelI}_i^\infty$, then $\mathrm{ProgSet}(M_j, f_i)$ does not contain $p(i,\infty)$. In either case M_j does not $\mathit{MinIncEx}_\psi$-identify $\psi_{p(i,\infty)} = f_i$. Thus (b) is satisfied.

To show that $\mathscr{C} \in \mathit{MinEx}_\psi$, note that $p(i,s)$ converges to $\mathit{min}_\psi(f_i)$. Thus $\mathscr{C} \in \mathit{MinEx}_\psi$, and hence (c) is satisfied. □

4.4. Characterizations

We now prove a characterization for all the considered types of function learning by erasing, when the hypothesis space may be chosen freely. It turns out that all these types coincide with Ex. In order to show this we need both allowing non-Gödel numberings as hypothesis spaces and a characterization of Ex in terms of non-Gödel numberings.

Theorem 48. *For all* $Lt \in \{All, Super, Sub\}$, $MinLt = Lt = Ex$.

Proof. Let $Lt \in \{All, Super, Sub\}$. We show $MinLt \subseteq Lt \subseteq Ex \subseteq MinLt$.

$MinLt \subseteq Lt$: By Proposition 8.

$Lt \subseteq Ex$: Let $\mathscr{C} \in Lt_\psi$ by a machine M. Let M' be a machine always outputting the least ψ-number which has not yet been erased by M. Clearly, $\mathscr{C} \in Ex_\psi$ by M'.

$Ex \subseteq MinLt$: We need the following result from Wiehagen [32].

Lemma 1. $\mathscr{C} \in Ex$ *iff there is a numbering* ψ *such that*
(1) $\mathscr{C} \subseteq \mathscr{R}_\psi$, *and*
(2) *there is* $d \in \mathscr{R}^2$ *such that, for any* $i \neq j$, $\psi_i[d(i,j)] \neq \psi_j[d(i,j)]$.

Now, suppose $\mathscr{C} \in Ex$. Let ψ, d be as given in the lemma. First note that ψ is 1–1. Hence ψ contains exactly one program for any function in $\mathscr{C}$. Thus showing that $\mathscr{C} \in MinAll_\psi$ suffices, since any machine witnessing $\mathscr{C} \in MinAll_\psi$ automatically witnesses both $\mathscr{C} \in MinSuper_\psi$ and $\mathscr{C} \in MinSub_\psi$. Let M be defined such that ProgSet $(M,f) = \{j \mid (\exists i \neq j)[f[d(i,j)] = \psi_i[d(i,j)]]\}$. It is easy to verify that M $MinAll_\psi$-infers $\mathscr{C}$. □

Note that the proof of Lemma 1 as given in [12] immediately shows that $\mathscr{R}_\psi \in MinIncEx_\psi$. Hence we also have the following characterization of $MinIncEx$.

Theorem 49. $MinIncEx = Ex$.

Lemma 1 also gives a pure numbering-theoretic characterization for All, $MinAll$, $Super$, $MinSuper$, Sub, $MinSub$, $IncEx$, $MinIncEx$ via Theorems 48 and 49.

Finally, we exhibit an alternative characterization of $MinIncEx_\psi$ (and thus, by Proposition 9, of $MinSub_\psi$ and Sub_ψ) which holds for *every* numbering ψ.

Theorem 50. *For every numbering* ψ, $\mathscr{C} \in MinIncEx_\psi$ *iff there is* $P \in \mathscr{P}$ *such that the following properties are satisfied*:
(1) $\mathscr{C} \subseteq \mathscr{R}_\psi$.
(2) *For all* $f \in \mathscr{C}$, $P(min_\psi(f))\downarrow$.
(3) *For all* $i \in \mathbb{N}$, $P(i)\downarrow$ *implies* (3a) *and* (3b):
(3a) *For all* $x < P(i)$, $\psi_i(x)\downarrow$.
(3b) *For all* $j < i$ *with* $\psi_j \in \mathscr{C}$, $\psi_j[P(i)] \neq \psi_i[P(i)]$.

Proof. Necessity: Suppose $\mathscr{C} \in MinIncEx_{\psi}$ as witnessed by M. (1) must clearly hold. Let P be defined as follows:

$$P(i) = \min\{x \mid (\forall y < x)[\psi_i(y)\downarrow] \wedge M(\psi_i[x]) = i\}.$$

It is easy to verify that (2) and (3) must hold.

Sufficiency: We use the fact that $MinSub_{\psi} = MinIncEx_{\psi}$ (Proposition 9). Suppose (1) holds and P is such that (2) and (3) hold. Then consider a machine M such that $\mathrm{ProgSet}(M, f) = \{j \mid (\exists i > j)[P(i)\downarrow = n \wedge f[n] = \psi_i[n]]\}$.

Note that such a machine M can easily be constructed. Using the properties of P above it is easy to verify that M $MinSub_{\psi}$-infers each function in $\mathscr{C}$. □

5. Conclusions

Different models of learning by erasing are defined. The capabilities of learning by erasing are investigated in relation to two factors: the choice of the overall hypothesis space itself *and* what sets of hypotheses must or may be erased. The power of the resulting learning types is related to one another as well as to those of standard learning types. These learning capabilities are studied for two fundamental kinds of objects to be learned, namely languages and functions.

For *language* learning by erasing, it turns out that all but the *EqualTxt* learning model are sensitive with respect to the particular choice of the hypothesis space, thus nicely contrasting learning in the limit and finite learning. Moreover, the learning power of the *SubTxt* model is even very dependent on the set of admissible hypothesis spaces.

A further interesting aspect is provided by Theorems 11 and 12. These results show that the process of elimination cannot be restricted to *incorrect* hypotheses for achieving its full learning power. On the other hand, all models of learning by erasing that are allowed to erase *correct* hypotheses, too, are as powerful as learning in the limit provided the hypothesis space is appropriately chosen (cf. Theorem 8). Consequently, in order to decide whether or not a particular indexed family can be *LtTxt*-learned, $Lt \in \{Arb, Super, All\}$, one can apply any of the known criteria for *ExTxt*-inferability (cf., e.g., [1, 29]).

These differences almost vanish if *absolute* learning is considered. Now, we have a somehow opposite effect. Erasing all but one guess turns out to be most restrictive with respect to the resulting learning capabilities.

The phenomena described above find their natural explanation in our characterization theorems. All models *ALtTxt* of absolute learning by erasing are constrained by the structural properties of the indexed families to be learned, i.e., they must be inclusion-free for $Lt \in \{Arb, Sub, Equal, Super, All\}$, and in case of *AAllTxt*, additionally, all hypothesis spaces must be equivalent with respect to reducibility, i.e., they must have a recursive equality problem.

Moreover, in Section 3.4 we study the problem whether or not information presentation may be traded versus learnability. The results obtained put the strength of

AAllInf-learning into the right perspective as displayed in Fig. 1. However, it remains open whether or not $AAllInf \subset ExTxt$ can be strengthened to $AAllInf \subset CConsvTxt$.

For *function* learning by erasing, we study three types of hypothesis spaces, Gödel numberings, Kolmogorov numberings and non-Gödel numberings. For Gödel numberings the same effect as in *language* learning by erasing can be observed, namely that erasing cannot be restricted to *incorrect* hypotheses in order to achieve full learning power (cf. Proposition 12). Since in Gödel numberings the other types of function learning by erasing yield the same power as *Ex*, the type of standard learning in the limit, (cf. Proposition 11), we turn over to investigate learning *minimal* programs.

For learning minimal programs by erasing, there are significant differences between arbitrary Gödel numberings and Kolmogorov numberings as hypothesis spaces. Whereas for all minimal learning criteria considered, any "reasonable" coincidence/inclusion between these criteria does hold in some Gödel numbering, as shown in Theorems 41 and 42, all these criteria (except for *MinAll* versus *MinSuper*) are separated in *every* Kolmogorov numbering (cf. Theorem 24). At present, $MinAll_\psi \subset MinSuper_\psi$ is proved only for *some* Kolmogorov numbering ψ. However, we conjecture that this separation is true for *every* Kolmogorov numbering. In order to achieve these results for Kolmogorov numberings some techniques are demonstrated which prove to be useful just in Kolmogorov numberings.

Non-Gödel numberings are used for both providing the necessary means to prove Theorem 42 and characterizing the types of learning by erasing. All the types of function learning by erasing considered coincide with *Ex* if also non-Gödel numberings are allowed as hypothesis spaces (cf. Theorem 48). Thus, the corresponding numbering-theoretic characterization of *Ex* given by Lemma 1 yields a "unique kind" of (non-Gödel) hypothesis spaces in which exactly every class from *Ex* can be learned in *all* of our *erasing* models. Furthermore, as in Theorems 5 and 6 for characterizing *language* learning by erasing, a characterization for $MinIncEx_\psi$, $MinSub_\psi$ and Sub_ψ (these types coincide by Proposition 9) is derived which holds for *every* numbering ψ, thereby exhibiting the first characterization of such type in function learning at all (cf. Theorem 50).

Finally, we want to point out a further possible line of research. In our opinion, it may also be interesting to investigate the *complexity* of learning by erasing. This includes the comparison of the complexity of both the different models of learning by erasing as well as of learning by erasing with standard learning. As a result of the first type we have the following comparison of the complexity of hypothesis spaces for language learning by erasing in the sense of *AllTxt* and *ArbTxt*, respectively. There is an infinite indexed family $\mathscr{L}$ such that

(1) for every $\psi \in \mathscr{R}^2_{0,1}$ such that $\mathscr{L} \in AllTxt_\psi$, all but one language from $\mathscr{L}$ must have *infinitely* many ψ-numbers,

(2) there exists $\psi \in \mathscr{R}^2_{0,1}$ such that $\mathscr{L} \in ArbTxt_\psi$ and every language from $\mathscr{L}$ has exactly *one* ψ-number.

This can be easily verified using the indexed family $\mathscr{L}$ defined in the proof of Theorem 11, Claim B, thus Property (2) follows. Property (1) is an immediate consequence of Theorem 13 in [13]. Hence in the sense of *AllTxt* this family $\mathscr{L}$ can be learned only with respect to hypothesis spaces possessing *infinite* "redundancy", whereas in the sense of *ArbTxt* it can be learned *without* redundancy.

Acknowledgements

We would like to thank the referees for many valuable comments and suggestions which have resulted in several improvements of the presentation of the paper.

References

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

[2] G. Baliga, J. Case, S. Jain, Synthesizing enumeration techniques for language learning, in: Proc. 9th Annual ACM Conf. on Computational Learning Theory, ACM Press, New York, 1996, pp. 169–180.

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

[4] M. Blum, A machine-independent theory of the complexity of recursive functions, J. ACM 14 (1967) 322–336.

[5] J. Case, C.H. Smith, Comparison of identification criteria for machine inductive inference, Theoret. Comput. Sci. 25 (1983) 193–220.

[6] R. Freivalds, Minimal Gödel numbers and their identification in the limit, in: Proc. Int. Conf. on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science, vol. 32, Springer, Berlin, 1975, pp. 219–225.

[7] R. Freivalds, Inductive inference of minimal programs, in: Proc. 3rd Annual Workshop on Computational Learning Theory, Morgan Kaufmann, San Mateo, 1990, pp. 3–20.

[8] R. Freivalds, Inductive inference of recursive functions: qualitative theory, in: Baltic Computer Science, Lecture Notes in Computer Science, vol. 502, Springer, Berlin, 1991, pp. 77–110.

[9] R. Freivalds, S. Jain, Kolmogorov numberings and minimal identification, in: Proc. 2nd European Conf. on Computational Learning Theory, Lecture Notes in Artificial Intelligence, vol. 904, Springer, Berlin, 1995, pp. 182–195.

[10] R. Freivalds, M. Karpinski, C.H. Smith, Co-learning of total recursive functions, in: Proc. 7th Annual Conf. on Computational Learning Theory, ACM Press, New York, 1994, pp. 190–197.

[11] R. Freivalds, D. Gobleja, M. Karpinski, C.H. Smith, Co-learnability and FIN-identifiability of enumerable classes of total recursive functions, in: Proc. 4th International Workshop on Analogical and Inductive Inference, Lecture Notes in Artificial Intelligence, vol. 872, Springer, Berlin, 1994, pp. 100–105.

[12] R. Freivalds, E.B. Kinber, R. Wiehagen, How inductive inference strategies discover their errors, Inform. Comput. 118 (1995) 208–226.

[13] R. Freivalds, T. Zeugmann, Co-learning of recursive languages from positive data, in: Proc. Perspectives of System Informatics, 2nd Int. Andrei Ershov Memorial Conf., Lecture Notes in Computer Science, vol. 1181, Springer, Berlin, 1996, pp. 122–133.

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

[15] S. Jain, An infinite class of functions identifiable using minimal programs in all Kolmogorov numberings, Int. J. Found. Comput. Sci. 6 (1995) 89–94.

[16] S. Jain, E. Kinber, R. Wiehagen, On learning and co-learning of minimal programs, Technical Report LSA-96-06E, Centre for Learning Systems and Applications, Department of Computer Science, University of Kaiserslautern, 1996.

[17] S. Jain, A. Sharma, Characterizing language learning by standardizing operations, J. Comput. System Sci. 49 (1994) 96–107.

[18] S. Kapur, G. Bilardi, Language learning without overgeneralization, Theoret. Comput. Sci. 141 (1995) 151–162.

[19] R. Klette, R. Wiehagen, Research in the theory of inductive inference by GDR mathematicians – a survey, Inform. Sciences 22 (1980) 149–169.

[20] M. Kummer, A learning-theoretic characterization of classes of recursive functions, Inform. Process. Lett. 54 (1995) 205–211.

[21] S. Lange, R. Wiehagen, T. Zeugmann, Learning by erasing, Technical Report RIFIS-TR-CS-122, Research Institute of Fundamental Information Science, Kyushu University, 1996.

[22] S. Lange, T. Zeugmann, Types of monotonic language learning and their characterization, in: Proc. 5th Annual ACM Workshop on Computational Learning Theory, ACM Press, New York, 1992, pp. 377–390.

[23] S. Lange, T. Zeugmann, Monotonic versus non-monotonic language learning, in: Proc. 2nd Int. Workshop on Nonmonotonic and Inductive Logic, December 1991, Lecture Notes in Artificial Intelligence, vol. 659, Springer, Berlin, 1993, pp. 254–269.

[24] S. Lange, T. Zeugmann, Language learning in dependence on the space of hypotheses, in: Proc. 6th Annual ACM Conf. on Computational Learning Theory, ACM Press, New York, 1993, pp. 127–136.

[25] S. Lange, T. Zeugmann, Learning recursive languages with bounded mind changes, Int. J. Found. Comput. Sci. 4 (1993) 157–178.

[26] S. Lange, T. Zeugmann, Characterization of language learning from informant under various monotonicity constraints, J. Exp. Theoret. Artificial Intell. 6 (1994) 73–94.

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

[28] H. Rogers, Theory of Recursive Functions and Effective Computability McGraw-Hill, New York, 1967, Reprinted, MIT Press, Cambridge, MA, 1987.

[29] M. Sato, K. Umayahara, Inductive inferability for formal languages from positive data, IEICE Trans. Inform. Systems E-75D (1992) 415–419.

[30] V.L. Selivanov, Enumerations of families of general recursive functions, Algebra i Logika 15 (1976) 205–226; English Translation, Algebra and Logic 15 (1976) 128–141.

[31] B.A. Trakhtenbrot, Ya.M. Barzdin, Finite Automata – Behavior and Synthesis, Fundamental Studies in Computer Science, vol. 1, North-Holland, Amsterdam, 1973.

[32] R. Wiehagen, Characterization problems in the theory of inductive inference, in: Proc. Int. Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science, vol. 62, Springer, Berlin, 1978, pp. 494–508.

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

[34] T. Zeugmann, S. Lange, S. Kapur, Characterizations of monotonic and dual monotonic language learning, Inform. Comput. 120 (1995) 155–173.