Separation of uniform learning classes

Sandra Zilles

Fachbereich Informatik, Universität Kaiserslautern, Postfach 3049,
67653 Kaiserslautern, Germany

Abstract

Within the scope of inductive inference a recursion theoretic approach is used to model learning behaviour. The fundamental model considered is Gold’s identification of recursive functions in the limit. Modifying the corresponding definition has proposed several inference classes, which have been compared regarding the capacities of the relevant learners.

The present paper is concerned with a meta-version of this learning model. Given a description of a class of target functions, a uniform learner is supposed to develop a specific successful method for learning the represented class. The same modifications as in the elementary model are considered in the context of uniform learning, especially respecting identification capacities. It turns out that the former separations of inference classes are reflected on the meta-level, in particular finite classes of recursive functions – which constitute the most simple learning problems in the elementary model – are evidence of these separations.

1 Introduction

Various theoretical concepts can be used to model learning behaviour. In this context inductive inference is concerned with suitable techniques provided by recursion theory. The target objects to be identified are recursive functions represented by programs via a partial-recursive numbering called hypothesis space.

In Gold’s [8] basic model of identification in the limit, the learner, modelled by a partial-recursive function, identifies a recursive function \( f \), if it transfers a sequence of information about \( f \) into a sequence of hypotheses converging to a correct program for \( f \). A sequence of information about \( f \) is simply the sequence of output values returned by \( f \) in natural order. In general a class of

---

Email address: zilles@informatik.uni-kl.de (Sandra Zilles).

Preprint submitted to Elsevier Science 3 December 2002
recursive functions is considered learnable if there is a single learner identifying each element of the class. By weakening or strengthening the constraints in Gold’s definition – for example via additional demands respecting the quality of the intermediate hypotheses – several alternative inference classes have been defined, cf. [2–8,10,15,16]. On the one hand, it has been of particular interest, what price has to be paid for the quality of the intermediate hypotheses (i.e. how strengthening the constraints reduces the quantity of learnable classes), on the other hand it has been studied, in which cases it is advisable to loosen the demands (i.e. how weakening the constraints increases the quantity of learnable classes). The results of these studies, see [2–6,10,15], provide a hierarchy of inference classes.

A quite conceivable idea is to analyse structural properties that successful learners may have in common and thus hopefully to design universal methods for the uniform identification of infinitely many classes of target objects. Evidently such properties always go along with some common intrinsic structure of the classes to be learned and the corresponding adequate hypothesis spaces. For example a uniform method for learning all recursively enumerable sets of recursive functions in the limit is identification by enumeration as defined by Gold [8]. This strategy can be generalized to temporarily conform identification, cf. [6], which constitutes a successful uniform method in specific hypothesis spaces. These ideas suggest the formal definition of a uniform learning model; analysing the corresponding identification capacity is the scope of the present paper. The new model considers some kind of meta-learning, where the uniform learner is supposed to develop a specific learner for each target class represented via some description associated with the class. That means, the uniform learner is able to exploit the common structure in the identifiable target classes, to the extent that successful strategies for these classes can be computed by a uniform method. The analysis of meta-learning perhaps provides even more revelation about these common structures.

Uniform learning has also been investigated in the context of language identification, see [1,11,12]. Baliga, Case, and Jain [1] compare several inference classes in their uniform language learning model with plentiful results contributing to a more detailed understanding of general properties in Gold’s elementary model. For examples of rather simple classes of language families, which cannot be identified uniformly, see [11,12]. Jantke [9] has studied meta-learning of recursive functions with similar negative results, which are further strengthened in [17]. Yet this outcome has to be interpreted carefully; most often such simple classes are not themselves too complex for uniform learning, but an inadequate choice of descriptions representing these classes causes the failure of uniform strategies.

The present paper is mainly concerned with the comparison of inference classes – formerly analysed in Gold’s elementary model – now in the context of meta-
Learning. As it turns out, the known hierarchy remains valid in the new model, where each separation of two inference classes is achieved by a representation of finite classes of recursive functions – most often either singleton classes or classes consisting of two functions, depending on the restrictions in the choice of hypothesis spaces. In the elementary model, finite classes can never witness to an increase of learning capacity in the comparison of two inference classes, because they are identifiable with respect to any learning criterion considered here. So, although finite classes constitute trivial learning problems in the non-uniform model, specific descriptions of such classes are too difficult for meta-learners to cope with.

The reflection of the former hierarchy in the uniform model corroborates the intuition, that any pair of different inference classes creates a relationship of learning power universally valid in lots of learning models; i.e. the hierarchy of learning classes expresses some kind of natural relationships. So there might exist a general trade-off between quality constraints in the learning criteria and resulting identification capacities. Therefore also in the context of uniform learning it is sometimes advisable to loosen the restrictive demands concerning the inference criteria in order to exploit a more powerful learning model.

Moreover the proofs of the separations provide methods for constructing descriptions of target classes not suitable for uniform identification with respect to a given inference criterion. Hopefully a further analysis of these methods may give insight into structures which are generally inadequate for learning in the specific inference classes.

A preliminary version presenting parts of the results in this paper has appeared, cf. [18].

2 Preliminaries

2.1 Notations

For notions and concepts relating to recursion theory see [13]. Standard notions are used for the comparison of sets, where \(\subseteq\) always indicates a proper inclusion of sets and \(\#\) expresses incomparability. \(\emptyset\) is a symbol for the empty set. In order to refer to the cardinality of a set \(X\) the notion \(\text{card} X\) is used.

The basic concept needed for modelling a learning scenario in inductive inference is the concept of partial-recursive functions (cf. [13]). Inputs and outputs of these functions are non-negative integers, the set of which is denoted by \(\mathbb{N}\). The variables \(n, x, y\) always range over \(\mathbb{N}\). A partial-recursive function which
is total, i.e. defined for all inputs, is simply called recursive function. If $f$ is any partial-recursive function, then $f(n)$ denotes the value of $f$ on input $n$, where $f(n) \uparrow$ indicates that $f$ is undefined on input $n$. Similarly two-place functions, three-place functions, etc. are considered. $\overline{f}$ symbolizes a recursive function returning 1 on input 0 and 0 on all other inputs.

By means of a recursive bijective mapping, finite tuples over $\mathbb{N}$ are identified with non-negative integers. Thus, if $f$ is a partial-recursive function and $n$ any input value such that $f(0), f(1), \ldots, f(n)$ are defined, this bijective mapping yields a code number $f[n]$ to be identified with the finite tuple $(f(0), f(1), \ldots, f(n))$. Given another partial-recursive function $g$, the notions $g(f[n])$ and $g(f(0) \ldots f(n))$ may sometimes be used interchangeably. If for all but finitely many $n$ either $f(n)$ and $g(n)$ are both undefined or $f(n) = g(n)$, this is indicated by $f =^* g$. Identifying the function $f$ with the set $\{ (n, f(n)) \mid f(n) \text{ is defined} \}$ explains the use of notions like $f \subseteq g$ and $f \subset g$. But each partial-recursive function may also be identified with the corresponding sequence of output values. For example let $f(n) = 0$ for $n \leq 6$ and $f(n) \uparrow$ otherwise; $g(n) = 0$ for $n \leq 5$ and $g(n) = 1$ otherwise; $h(n) = 0$ for all $n$. This might be denoted for short by $f = 0^7 \uparrow \infty$, $g = 0^6 1^{\infty}$, $h = 0^{\infty}$. Here $f \not\equiv g$, $g \not\equiv h$, but $f \subset h$.

If $n$ is given, any $(n+1)$-place partial-recursive function $\psi$ enumerates the set $\{ \psi_i \mid i \in \mathbb{N} \}$ of $n$-place partial-recursive functions, where $\psi_i$ ($i \in \mathbb{N}$) is given by $\psi_i(x_1, \ldots, x_n) := \psi(i, x_1, \ldots, x_n)$ for all elements $x_1, \ldots, x_n$ of $\mathbb{N}$. Therefore such a function $\psi$ is also called a numbering. Assume $f$ belongs to $\{ \psi_i \mid i \in \mathbb{N} \}$. In this case any index $x$ satisfying $\psi_x = f$ is called a $\psi$-number or a $\psi$-program of $f$. As an example consider the function $\psi$, which is for any $x, y$ defined by $\psi(x, y) \uparrow$, if $x = 0$; $\psi(x, y) := 0$, if $x > 0$ and $y < x$; $\psi(x, y) := 1$, otherwise. Then $\psi$ is a numbering of the set $\{ \uparrow^{\infty} \} \cup \{ 0^i 1^{\infty} \mid i \geq 1 \}$; 0 is the (unique) $\psi$-number of $\uparrow^{\infty}$ and each index $i > 0$ is the $\psi$-number of $0^i 1^{\infty}$. Of course there are also numberings which provide more than one program for a single function.

### 2.2 Hierarchy of learning classes

A theoretical learning model is principally characterized by five components: a class of possible target objects, a method for communicating information about these objects, a set of possible learners developing a hypothesis from any feasible information about an object to be learned, a class of hypothesis spaces associating objects with such hypotheses, and finally a success criterion declaring the desired behaviour of the other components. In any inference class defined in this section four of these components are always specified the same: the target objects to be identified are recursive functions
with the corresponding information presented as a gradually growing infinite sequence \( f[0], f[1], f[2], \ldots \) of the tuples of its output values. **Learners** are partial-recursive functions, also called strategies; **hypothesis spaces** are partial-recursive numberings, enumerating at least all the functions which have to be identified. That means, each function to be learned has an index in the hypothesis space.

The different inference classes defined here thus result from different success criteria. In the basic model – **identification in the limit** or **explanatory identification**, cf. [8] – the learner is required to eventually return a single correct hypothesis for any target function.

The modifications of this model considered below are chosen such that three approaches are taken into account: firstly, modifying the requirements concerning the success of the sequence of hypotheses; secondly, modifying the demands regarding the quality of the hypotheses – independent of the amount of information known about the target function; thirdly, modifying the quality demands depending on the current information. Each approach will be represented by at least two inference types.

**Definition 1** A set \( U \) of recursive functions is identifiable in the limit (**Ex-identifiable**), iff there is some hypothesis space \( \psi \) and a strategy \( S \), such that for any \( f \in U \) the following conditions are fulfilled:

1. \( S(f[n]) \) is defined for all \( n \in \mathbb{N} \),
2. the sequence \( (S(f[n]))_{n \in \mathbb{N}} \) converges to a \( \psi \)-number of \( f \).

**Ex** denotes the class of all **Ex-learnable** sets \( U \).

For example any class of functions enumerated by a recursive numbering is **Ex-learnable** (see [8]), but there is no adequate strategy for the whole class of recursive functions (cf. [8,4]). Still it is conceivable that loosening the success criterion in Definition 1 might yield a learning model which allows identifiability of the whole set of recursive functions. In a first step the requirements concerning convergence of the sequence of hypotheses are weakened. In the model of **behaviorally correct identification**, as defined in [2] and also discussed in [3], convergence is no longer required; the learner eventually has to return correct programs, but is allowed to conjecture different programs for the same function.

**Definition 2** A set \( U \) of recursive functions is **Bc-identifiable**, iff there is some hypothesis space \( \psi \) and some learner \( S \), such that for any \( f \in U \) all values \( S(f[n]) \) (\( n \in \mathbb{N} \)) are defined and all but finitely many of them are \( \psi \)-numbers for \( f \). **Bc** is the class of all **Bc-learnable** sets.

This modification of Definition 1 yields an increase of learning power, i.e. **Ex** is
a proper subset of $Bc$ (see [2]), but the top of the hierarchy of learning classes is not yet reached. Permitting a few errors in the conjectures, as suggested in [5], results in an even stronger model, denoted by $Bc^*$.

**Definition 3** A set $U$ of recursive functions is $Bc^*$-identifiable, iff there is some hypothesis space $\psi$ and some learner $S$, such that for any $f \in U$ all values $S(f[n])$ ($n \in \mathbb{N}$) are defined and all but finitely many of them fulfil $\psi_S(f[n]) =^* f$. $Bc^*$ denotes the class of all $Bc^*$-learnable sets.

With this inference criterion the top of the hierarchy of identification power is definitely reached, since the whole set of recursive functions is $Bc^*$-learnable; the corresponding proof in [5] refers to a private communication to L. Harrington. So loosening the conditions in Definition 1 yields the hierarchy $Ex \subset Bc \subset Bc^*$ of increasing learning power. But it is also conceivable to strengthen the demands concerning $Ex$-identifiability; one idea is for example to modify the conditions regarding the aspect of mind change complexity in the sequence of hypotheses returned by the strategy.

**Definition 4** Let $S$ be a strategy which is additionally permitted to return the sign “?” . A set $U$ of recursive functions is $Ex_m$-identifiable by $S$, iff $U$ is $Ex$-learned by $S$ with respect to some hypothesis space $\psi$, such that for all $f \in U$ the following conditions hold:

1. there is some $k \in \mathbb{N}$, such that $S(f[n]) = ?$ iff $n < k$,
2. $\text{card} \{ n \mid ? \neq S(f[n]) \neq S(f[n+1]) \} \leq m$.

$Ex_m$ is the class of all sets which are $Ex_m$-identifiable by some learner $S$.

The advantage of identification with a bound $m$ on the number of mind changes is, that whenever this bound is actually reached in the identification process, the final correct hypothesis is already known. Note that the definition of identification in the limit never allows for certainty concerning the correctness of the current hypothesis. But the advantage achieved by the $Ex_m$-model goes along with a loss of identification power: $Ex_m \subset Ex_{m+1} \subset Ex$ for all $m \geq 0$, cf. [5]. A further approach to strengthening the demands of Definition 1 is to improve the quality of the intermediate hypotheses by additional constraints arising from a somewhat natural motivation. Definition 5 suggests some properties conceivably augmenting this quality; for more background on these properties and the corresponding learning models the reader is referred to [2–4,6–8,10,15,16].

Note that all modifications of $Ex$-learning defined above deal with requirements concerning the convergence of the sequence of hypotheses returned by the learner. The modifications to be defined next rather deal with the properties of the intermediate hypotheses themselves. In particular two types of properties are distinguished: first, properties in dependency of the information
the learner has currently received, i.e. the known initial segment of the target function; such properties are for example consistency or conformity. Second, it is also conceivable to consider properties neglecting the amount of information given about the target function, such as convergent incorrectness or totality of the intermediate hypotheses.

**Definition 5** Let \( f \) be any recursive function, \( S \) a strategy, \( \psi \) any hypothesis space. Fix some number \( n \), such that \( S(f[n]) \) is defined. Moreover let \( m \geq 0 \). The hypothesis \( S(f[n]) \) is called

- **consistent for** \( f[m] \) with respect to \( \psi \) iff, for all \( x \leq m \), \( \psi_{S(f[n])}(x) \) is defined and equals \( f(x) \);
- **conform for** \( f[m] \) with respect to \( \psi \) iff, for all \( x \leq m \), either \( \psi_{S(f[n])}(x) \) is undefined or \( \psi_{S(f[n])}(x) = f(x) \);
- **convergently incorrect for** \( f \) with respect to \( \psi \) iff \( \psi_{S(f[n])} \not\subseteq f \);
- **total with respect to** \( \psi \) iff \( \psi_{S(f[n])} \) is a total function.

Demanding that all hypotheses returned by a learner on relevant input sequences should be consistent with the information seen so far, is a quite natural approach. Yet these requirements might be too strong, taking into account that any inconsistency resulting from an undefined value may in general not be found by the learner. This motivates the approach of conformity.

It is also conceivable that a learner may try to maintain its hypotheses until they are evidently found to be wrong. To allow for such convergently justified mind changes, every incorrect guess should correspond to a function disagreeing with the target function in at least one defined value, i.e. no incorrect hypothesis describes a subfunction of \( f \).

Moreover, these requirements can be strengthened to a demand for total intermediate hypotheses, since in particular no non-total function can equal the target function.

**Definition 6** Let \( U \) be a set of recursive functions, \( S \) a strategy and \( \psi \) some hypothesis space, such that \( U \) is \( \text{Ex-} \)-learned by \( S \) with respect to \( \psi \). Then \( U \) is \( \text{Cons-} \), \( \text{Cex-} \), \( \text{Total-} \)-learned, resp.) by \( S \) with respect to \( \psi \), iff, for any \( f \in U \) and \( n \in \mathbb{N} \), \( S(f[n]) \) is consistent for \( f[n] \) (conform for \( f[n] \), either correct or convergently incorrect for \( f \), total, resp.) with respect to \( \psi \). The notions \( \text{Cons} \), \( \text{Conf} \), \( \text{Cex} \), \( \text{Total} \) are defined as usual.

The inference class \( \text{Cons} \) has especially been studied in \( [8,3,14,16] \); there it is verified that the demand for consistency yields a decrease of learning power. As the definitions already suggest, \( \text{Conf} \) is an inference class ranging between \( \text{Cons} \) and \( \text{Ex} \) in the hierarchy. For a proof of \( \text{Cons} \subseteq \text{Conf} \subseteq \text{Ex} \) see \( [15] \), moreover in particular the work of Fulk \( [7] \) is of interest regarding conform identification. Similar ideas as used for the separations of several inference
criteria in [6] yield Cons \# Ex_m and Conf \# Ex_m for all m \geq 1, whereas Ex_0 \subset Cons; details are omitted. The main work done regarding Cex-learning can be found in [6], including proofs for Cex \subset Ex, Cex \# Cons, and Cex \# Ex_m for all m \geq 1. Again Ex_0 \subset Cex is easily verified and for the proof of Cex \# Conf the ideas from [6] are helpful. For Total-identification and a proof of Total \subset Cons see [10]. Ex_0 \subset Total and Total \# Ex_m for all m \geq 1 can be verified with the help of the separations mentioned above. By definition Total is a subset of Cex; the proper subset relation Total \subset Cex is then obtained from Total \subset Cons and Cons \# Cex.

The notion \mathbb{I} refers to the set of all inference classes defined so far.

\[ \mathbb{I} := \{ \text{Ex}, Bc, Bc^*, \text{Cons}, \text{Conf}, \text{Cex}, \text{Total} \} \cup \{ \text{Ex}_m \mid m \geq 0 \} . \]

The following lemma summarizes some commonly used results, see for example [8,16].

**Lemma 7** Let \( I \in \mathbb{I}, U \in I \) and let \( \tau \) be any acceptable numbering. Then there exists a strategy \( I \)-learning the class \( U \) with respect to the hypothesis space \( \tau \). Moreover, if \( I \not\in \{ \text{Cons}, \text{Conf} \} \) and \( \psi \) is a hypothesis space, such that \( U \) is \( I \)-learnable with respect to \( \psi \), then there exists a total recursive \( I \)-learner identifying \( U \) with respect to \( \psi \).

A counterexample for the the criterion Cons in the second part of Lemma 7 is given in [16]. The results mentioned above are summarized in Theorem 8 and illustrated in Figure 1.

**Theorem 8** [2–6,10,15]

1. \( \text{Ex}_m \subset \text{Ex}_{m+1} \subset \text{Ex} \subset Bc \subset Bc^* \) for all \( m \geq 0 \), \( \{ f \mid f \text{ recursive} \} \in Bc^* \),
2. \( \text{Ex}_0 \subset \text{Total} \subset \text{Cons} \subset \text{Conf} \subset \text{Ex} \),
3. \( \text{Total} \subset \text{Cex} \subset \text{Ex} \),
4. \( \text{Cex} \# \text{Cons}, \text{Cex} \# \text{Conf} \),
5. \( \text{Ex}_m \# I \) for all \( m \geq 1 \) and all \( I \in \{ \text{Total}, \text{Cex}, \text{Cons}, \text{Conf} \} \).

Note that three kinds of inference types have been defined via modifications of the constraints in Ex-identification:

- types resulting from special constraints concerning the success criterion of the sequence of hypotheses, namely \( \text{Ex}_m \) for \( m \in \mathbb{N}, Bc, Bc^* \) (the latter also modifying the accuracy demands); these form the right axis and the upper left axis in Figure 1;
- types resulting from special constraints concerning the quality of the intermediate hypotheses, independent of the amount of information currently known about the target function, namely Total and Cex; these form the middle left axis in Figure 1;
types resulting from special constraints concerning the quality of the intermediate hypotheses, depending on the information currently known about the target function; namely \( Cons \) and \( Conf \); these form the lower left axis in Figure 1.

For each kind of inference type the separation results will be transferred to the context of uniform learning.

\[
\begin{array}{cccccc}
E_0 & \rightarrow & E_1 \\
\downarrow & & \\
\text{Total} & \rightarrow & \text{Cex} & \rightarrow & \text{Ex} & \rightarrow \text{Be} & \rightarrow \text{Be}^*
\end{array}
\]

Fig. 1. The hierarchy of learning classes. Vectors indicate proper inclusions; if two classes are not connected by a sequence of vectors in one direction, they are incomparable.

3 The model of uniform learning

3.1 Definitions

The learning models defined in the previous section will now be considered on a meta-level. Uniform learning is concerned with the existence of strategies, which simulate appropriate learners for infinitely many learning problems. In this context, any class of recursive functions constitutes a learning problem. So a uniform strategy – on input of a description for a class of recursive functions – must develop an appropriate learner for the class described.

The formal definition of the corresponding learning model first requires a clear explanation of how to describe learning problems. The descriptions are necessary, in order to inform a uniform learner of the actual learning problem to cope with. A quite simple method is to consider a class of recursive functions as a subset of a class of partial-recursive functions enumerated by an arbitrary numbering. Thus a family of numberings yields a family of learning problems. So from now on let \( \phi \) denote a fixed three-place acceptable numbering. This provides an effective enumeration \( (\phi^d)_{d \in \mathbb{N}} \) of all numberings, where \( \phi^d(i, x) \) equals \( \phi(d, i, x) \) for all \( d, i, x \in \mathbb{N} \). With each numbering \( \phi^d \) the recursive core \( R_d \) is associated as follows:

\[
R_d = \{ \phi^d_i \mid i \in \mathbb{N} \text{ and } \phi^d_i \text{ is recursive} \} \text{ for any } d \in \mathbb{N}.
\]
Hence any parameter \( d \in \mathbb{N} \) corresponds to a set \( R_d \) of recursive functions to be identified, i.e. \( d \) describes a learning problem. Consider for example the numbering \( \psi \), which is for any \( x, y \) defined by \( \psi(x, y) \uparrow \), if \( x = 0 \); \( \psi(x, y) := 0 \), if \( x > 0 \) and \( y < x \); \( \psi(x, y) := 1 \), otherwise. Then any integer \( d \) satisfying \( \varphi^d = \psi \) is a description of the recursive core \( R_d = \{0^d i^\infty \mid i \geq 1\} \). Of course the interpretation of such descriptions is influenced by the choice of \( \varphi \). Nevertheless, since \( \varphi \) is acceptable, all results obtained below hold independently, no matter what acceptable numbering is chosen.

Now note that any set \( D \subseteq \mathbb{N} \) corresponds to a series of classes of recursive functions and thus to a series of learning problems. Therefore such a set will be called a description set whenever it is considered as a set indexing a family of classes of recursive functions. For a uniform learner trying to cope with any learning problem described in a set \( D \), it is sufficient to develop from any parameter \( d \in D \) a suitable learner for the recursive core described by \( d \). More formally, if one input parameter of the uniform learner is fixed by \( d \), the resulting function must be a learner for \( R_d \).

**Definition 9** Let \( I \in \mathbb{I} \) and \( D \subseteq \mathbb{N} \). Fix an acceptable numbering \( \tau \). \( D \) is uniformly \( I \)-learnable iff there is a two-place strategy \( S \), such that, for any description \( d \in D \), the learner \( S_d \) \( I \)-identifies the set \( R_d \) with respect to \( \tau \). Uni\( I \) denotes the class of all uniformly \( I \)-learnable description sets.

Note that this definition is independent of the choice of \( \tau \). Of course it is quite natural to choose an acceptable numbering as the common hypothesis space to be used for uniform learning of the whole series of classes described in a set \( D \), cf. Lemma 7. Nevertheless other motivations might influence the choice of hypothesis spaces: as each description \( d \) of a recursive core also corresponds to a numbering \( \varphi^d \) which “contains” all functions in the recursive core, perhaps even the numberings \( \varphi^d \) might serve as hypothesis spaces. Hence the idea to demand correct identification with respect to the numberings associated to the descriptions also seems conceivable. Since \( \varphi^d \)-programs can be uniformly transformed into \( \tau \)-programs (for any acceptable numbering \( \tau \)), this idea yields a special case of the Uni\( I \)-model. Therefore the term restricted uniform learning will be used in this context.

**Definition 10** Let \( I \in \mathbb{I} \) and \( D \subseteq \mathbb{N} \). \( D \) is uniformly \( I \)-learnable with restricted choice of hypothesis spaces iff there is a two-place strategy \( S \), such that, for any description \( d \in D \), the learner \( S_d \) \( I \)-identifies the set \( R_d \) with respect to \( \varphi^d \). res Uni\( I \) denotes the class of all description sets which are uniformly \( I \)-learnable in this restricted model.

Another conceivable thought is to weaken the constraints concerning the choice of hypothesis spaces, such that the learner is just required to synthesize adequate strategies for the learning problems described, but no longer required to
synthesize the corresponding suitable hypothesis spaces. Thus the UniI-model is generalized to the so-called model of extended uniform learning.

**Definition 11** Let \( I \in \mathbb{I} \) and \( D \subseteq \mathbb{N} \). \( D \) is uniformly \( I \)-learnable with extended choice of hypothesis spaces iff there is a two-place strategy \( S \), such that, for any description \( d \in D \), the learner \( S_d \) \( I \)-identifies the set \( R_d \) with respect to some arbitrary hypothesis space \( \psi \). \( \text{ext UniI} \) denotes the class of all description sets which are uniformly \( I \)-learnable in this extended model.

Of course, for any \( I \in \mathbb{I} \), the inclusions \( \text{res UniI} \subseteq UniI \subseteq \text{ext UniI} \) follow immediately from the definitions. To show that in general \( \text{res UniI} \) really constitutes a restriction of \( UniI \), and \( \text{ext UniI} \) corresponds to a proper extension of \( UniI \), special descriptions of finite recursive cores are sufficient, as Proposition 13 states. Since this is not the only context where finite classes of recursive functions help to obtain interesting results within the scope of uniform learning, some further notation, concerning the identification of finite recursive cores, might be useful.

**Definition 12** Let \( I \in \mathbb{I} \). Then \( UniI[*] \) is the class of all description sets \( D \in UniI \) corresponding to a family of finite recursive cores. The notions \( \text{res UniI[*]} \) and \( \text{ext UniI[*]} \) are used analogously.

**Proposition 13**  
(1) \( \text{res UniI[*]} \subseteq UniI[*] \subseteq \text{ext UniI[*]} \) for \( I \in \mathbb{I} \setminus \{Bc^*\} \),
(2) \( \text{res UniBc*[*]} \subseteq \text{UniBc*[*]} \),
(3) \( \text{UniBc*} = \text{ext UniBc*} = \{D \mid D \subseteq \mathbb{N}\} \).

*Sketch of proof.* ad 1. Fix \( I \in \mathbb{I} \setminus \{Bc^*\} \). By the remarks above, it remains to verify \( \text{res UniI[*]} \neq UniI[*] \neq \text{ext UniI[*]} \). The set \( \{d \mid \text{card}R_d = 1\} \) is an example for a description set belonging to \( \text{ext UniI[*]} \setminus \text{UniI[*]} \). Uniform learning of this set in the extended model is trivial: since for every recursive function \( f \) there is a hypothesis space \( \psi \) satisfying \( \psi_0 = f \), the strategy constantly zero is an appropriate learner. \( \{d \mid \text{card}R_d = 1\} \notin \text{UniI[*]} \) follows from Theorem 24.1 for \( (I, I') = (Bc, Bc^*) \), so a proof will be given below.

Moreover there exists a set \( D \subseteq \{d \mid \text{card} \{i \mid \varphi^d_i \text{ is recursive} \} = 1\} \), which is not suitable for restricted uniform Bc-identification (see the proof of Theorem 24.1 for \( (I, I') = (Bc, Bc^*) \)). From such a set \( D \) a description set \( D' \) in \( \text{UniI[*]} \setminus \text{res UniI[*]} \) can be constructed in the following way: choose a recursive function \( g \), such that, for all \( d, i, x \),

\[
\varphi^g_{i(d)}(x) = \begin{cases} 0, & \text{if } \varphi^d_i(y) \text{ is defined for all } y \leq x, \\ \dagger, & \text{otherwise}. \end{cases}
\]

Then let \( D' = \{g(d) \mid d \in D\} \). Since each recursive core described by \( D' \) equals \( \{0^\infty\} \), the strategy constantly returning a fixed program for \( 0^\infty \) witnesses to \( D' \in \text{UniI[*]} \). If there was an appropriate \( I \)-learner \( S \) for \( D' \) in the restricted
uniform model, then defining

\[ T_d(f[n]) := S_{g(d)}(0^n) \] for all recursive functions \( f \) and all \( d, n \),

would yield a resUniI-learner for \( D \). To verify this, note that, for all \( d \in D \) and all \( i \), \( \varphi^d_i \) is recursive iff \( \varphi^d_i \) equals \( 0^\infty \). Since \( D \notin \text{resUniBc} \), this results in a contradiction. Hence \( D' \in \text{UniI}[\ast] \setminus \text{resUniI}[\ast] \).

\( \text{ad 2.} \) The description set \( \{d \mid \text{card}R_d = 1\} \) belongs to \( \text{UniBc}^* [\ast] \), but not to \( \text{resUniBc}^* [\ast] \) (cf. [17]).

\( \text{ad 3.} \) This follows immediately from Theorem 8 and Lemma 7, because the whole set of recursive functions is \( \text{Bc}^* \)-identifiable with respect to any acceptable numbering. So, in the context of \( \text{UniBc}^* \)- and \( \text{ext UniBc}^* \)-identification, even the “classical” learners suffice. \( \square \)

If \( I, I' \in \mathbb{I} \) are inference classes, such that \( I' \setminus I \neq \emptyset \), then also \( \text{UniI'} \setminus \text{UniI} \neq \emptyset \) and \( \text{ext UniI'} \setminus \text{ext UniI} \neq \emptyset \); any description of a recursive core in \( I' \setminus I \) can be used to verify this result. Similar results can be obtained for most inference criteria in the restricted model, if the descriptions are chosen carefully. The following lemma is used to show, that such descriptions exist for all uniform learning models considered here.

**Lemma 14** Let \( I \in \mathbb{I}, U \in I \). Then there exists a hypothesis space \( \psi \), such that \( U \subseteq \{\psi_i \mid i \geq 0\} \) and the recursive core of the numbering \( \psi \) is \( I \)-learnable.

**Proof.** First assume \( I = \text{Bc}^* \). Then the whole set of recursive functions is \( I \)-learnable with respect to any acceptable numbering, so the assertion holds.

Next let \( I = \text{Ex} \). In this case the following characterization from [15] can be used: let \( U \) be a set of recursive functions.

\( U \in \text{Ex} \) iff there is some partial-recursive numbering \( \psi \) and a recursive function \( h \) satisfying

\( \cdot U \subseteq \{\psi_i \mid i \geq 0\}, \)

\( \cdot \) if \( i, j \in \mathbb{N} \) and \( i \neq j \), then \( \{ (x, \psi_i(x)) \mid x \leq h(i, j) \text{ and } \psi_i(x) \text{ is defined} \} \neq \{ (x, \psi_j(x)) \mid x \leq h(i, j) \text{ and } \psi_j(x) \text{ is defined} \} \), i.e. \( \psi_i \) and \( \psi_j \) disagree on some input “below” \( h(i, j) \).

Now if \( U \in \text{Ex} \) and \( \psi, h \) are chosen accordingly, then also the recursive core of \( \psi \) matches this characterization. Hence \( \psi \) witnesses to the assertion of Lemma 14.

In the case \( I \in \{\text{Cons, Bc}\} \cup \{\text{Ex}_m \mid m \in \mathbb{N}\} \) the same approach as for \( I = \text{Ex} \) can be used. Details are omitted.
For the case $I = \text{Conf}$ let $U$ be a class in $\text{Conf}$, $\tau$ an acceptable numbering and $S$ any strategy $\text{Conf}$-identifying $U$ with respect to $\tau$. Similar ideas as in [15] are used to obtain the desired numbering $\psi$. Define a set $M$ of pairs by

$$
M := \{ (z,n) \mid \tau_z(x) \text{ and } S(\tau_z[x]) \text{ are defined for all } x \leq n \\
\quad \text{ and } S(\tau_z[n]) = z \}.
$$

Obviously $M$ is recursively enumerable, so let $g$ be a recursive function with range $M$. For any number $i$, if $g(i) = (z,n)$, let $\psi_i[n] := \tau_z[n]$. Moreover, for $x > n$, let $\psi_i(x) := \tau_z(x)$, if $S(\tau_z[n]) = S(\tau_z[n + 1]) = \ldots = S(\tau_z[x]) = z$ and if Condition A holds.

**Condition A.** None of the $x + 1$ initial hypotheses are found to be non-conform with respect to $\tau$ within $x$ steps of computation (formally: for all $y \leq x$ and all $m \leq y$, if $\tau_{S(\tau_z[y])}(m)$ is defined within $x$ steps of computation, then $\tau_{S(\tau_z[y])}(m) = \tau_z(m)$).

In any other case, let $\psi_i(x)$ be undefined. Now it remains to verify, that $\psi$ satisfies the desired properties.

To prove that $U$ is contained in the set of all functions $\psi_i$, $i \geq 0$, fix some arbitrary function $f$ in $U$. Then there exist numbers $z$ and $n$, such that $\tau_z$ equals $f$ and, for all $x \geq n$, $S(\tau_z[x]) = z$. Otherwise $S$ would not learn $f$ in the limit with respect to $\tau$. In addition, $S(\tau_z[x])$ must also be defined for any $x < n$. Moreover – since the conformity demands are fulfilled – if $\tau_{S(\tau_z[y])}(m)$ is defined for any $y \geq 0$ and any $m \leq y$, then $\tau_{S(\tau_z[y])}(m)$ equals $\tau_z(m)$. By definition of $M$ the pair $(z,n)$ is contained in $M$; hence there is some $i$ with $g(i) = (z,n)$. The argumentation above then implies $\psi_i = \tau_z = f$. Thus $U \subseteq \{ \psi_i \mid i \geq 0 \}$.

Finally it is possible to show, that $S$ learns the recursive core of $\psi$ conformly with respect to $\tau$. For that purpose fix some number $i$, such that $\psi_i$ is a recursive function. Let $g(i) = (z,n)$. Obviously $\psi_i = \tau_z$. As $\psi_i$ is a total function, all hypotheses $S(\tau_z[x])$ for $x \geq 0$ must be defined and, if $x \geq n$, must equal $z$. Thus $S$ learns $\psi_i$ in the limit with respect to $\tau$. Furthermore, if any intermediate hypothesis returned by $S$ on $\tau_z$ was non-conform with respect to $\tau$, then $\psi_i$ could not be total because of Condition A. This implies, that $\psi_i$ – and so the whole recursive core of $\psi$ – is $\text{Conf}$-learned by $S$ (with respect to $\tau$).

For the case $I = \text{Cex}$ fix some $U \in \text{Cex}$ and some total recursive strategy $S$ $\text{Cex}$-learning $U$ with respect to an acceptable numbering $\tau$. Define a set $M$ similarly to the method above. A pair $(z,n)$ belongs to $M$ iff $\tau_z(x)$ is defined for all $x \leq n$ and $S(\tau_z[n]) = z$. Choose a recursive function $g$, such that the range of $g$ equals the set $M$. If $g(i) = (z,n)$, let $\psi_i[n] := \tau_z[n]$. Given $x > n$, let $\psi_i(x) := \tau_z(x)$, if $S(\tau_z[n]) = S(\tau_z[n + 1]) = \ldots = S(\tau_z[x])$ and Condition A holds.
Condition A. All of the $x + 1$ initial hypotheses are either consistent or convergently incorrect for $\tau_z$ in an argument “below” $x$ (formally: for all $y \leq x$ either $\tau_{S(\tau_z, [y])}(m) = \tau_z(m)$ for all $m \leq y$ or there is some $m \geq 0$, such that $\tau_{S(\tau_z, [y])}(m)$ is defined and not equal to $\tau_z(m)$).

In any other case let $\psi(x)$ be undefined. A similar argumentation as for the case $I = Conf$ shows that $\psi$ fulfills the desired properties.

Finally, if $I = Total$, consider a set $U \in Total$ and a recursive strategy $S$ which learns $U$ with total intermediate hypotheses with respect to an acceptable numbering $\tau$. The proof proceeds as in the case $I = Cex$, where Condition A is replaced as follows.

Condition A. All of the $x + 1$ initial hypotheses correspond to functions defined for the initial segment of length $x + 1$ (formally: $\tau_{S(\tau_z, [y])}(m)$ is defined for all $y, m \leq x$.

The rest of the argumentation can be transferred as usual. □

Corollary 15 Suppose $I, I' \in \Pi$ are inference classes, such that $I' \setminus I \neq \emptyset$. Then there exists a description $d$ satisfying $\{d\} \in UniI' \setminus ext UniI$.

Proof. Choose $U \in I' \setminus I$. By Lemma 14 there is a description $d$, such that $U \subseteq R_d$ and $R_d \in I'$. Lemma 7 then implies $R_d \in I'$, for any acceptable numbering $\tau$. Moreover $R_d \notin I$, because $U \notin I$. Consequently, $\{d\} \in UniI' \setminus ext UniI$. □

Hence $UniI' \setminus UniI$ and $ext UniI' \setminus ext UniI$ are non-empty, if $I' \setminus I \neq \emptyset$. The more challenging question is, whether there are description sets, which (i) correspond to families of classes in $I$, (ii) are uniformly $I'$-learnable, but (iii) are not uniformly $I$-learnable. Of course this problem is also relevant for the restricted and extended models. The main concern of this paper is to show that, for most of the models, such description sets exist. Moreover most often families of finite classes suffice to verify the desired results.

3.2 Helpful results

In the subsequent proofs for separations of the kind $UniI \subset UniI'$ (for $I, I' \in \Pi$) description sets are constructed to disallow UniI-identification for any learner. Such constructions become much more accessible, if a diagonal argument defeating all recursive learners suffices. Fortunately, as Proposition 16 shows, this idea can be exploited in many cases.

Proposition 16 Let $I \in \Pi \setminus \{Cons, Conf\}$ and let $D$ be any description set.
Assume $D \in \text{UniI}$ ($D \in \text{ext UniI}$). Then there exists a total recursive function $S$, such that $D$ is UniI-identifiable by $S$ (ext UniI-identifiable by $S$, respectively). Moreover, if $I \notin \{\text{Total}, \text{Cex}\}$ and $D \in \text{res UniI}$, there exists some total recursive learner $S$, which $\text{res UniI}$-identifies $D$.

The idea of the proof is the same as for the corresponding claims in Lemma 7 and is therefore not demonstrated. Counterexamples for the cases excluded in the statement of Proposition 16 are proposed below in Examples 17 and 18.

**Example 17** Let $I \in \{\text{Cons, Conf, Cex, Total}\}$; fix a description set $D$ by

$$D := \{d \mid R_d = \{0^\infty\} \text{ and there is exactly one index } i \text{ such that } \varphi_i^d(0) = 0\}.$$ 

Then $D$ belongs to $\text{res UniI}$, but $D$ is not $\text{res UniI}$-identifiable by any total recursive strategy.

**Proof.** First let $I \in \{\text{Cons, Conf, Total}\}$. The case $I = \text{Cex}$ will be handled separately afterwards. Obviously, $D$ is $\text{res UniI}$-identifiable: given the parameter $d$ as a description of a recursive core, a learner just has to return a number $i$ satisfying $\varphi_i^d(0) = 0$. If $d$ belongs to the set $D$, such a number must exist and is a program for $0^\infty$, which is the only function in $R_d$.

It remains to prove that $D$ cannot be identified with respect to $\text{res UniI}$ by any recursive learner. For that purpose fix some arbitrary recursive strategy $S$. To verify that $S$ is not suitable for $\text{res UniI}$-identification of the whole set $D$, a description $d^*$ is constructed, such that the following two properties hold:

1. $d^*$ belongs to $D$, but
2. the recursive core described by $d^*$ is not $I$-learned by $S_{d^*}$ with respect to the hypothesis space $\varphi_{d^*}$.

For that purpose define for each number $d$ a two-place function $\psi$ as follows.

First compute $e := S_d(0) + 1$ and let $\psi_e = 0^\infty$. Moreover, define $\psi_i = 1 \uparrow^\infty$ for all programs $i \neq e$.

As $S$ is a total recursive function, this definition is uniformly effective in $d$. Hence there exists some fixed point value $d^*$, satisfying $\varphi_{d^*} = \psi$, for the numbering $\psi$ constructed from $S$ and $d^*$. This fixed point value shall be used to make the learner $S$ fail. 

End Construction of $d^*$

Now the desired properties can be verified.

ad 1. $d^*$ belongs to $D$.

This is an immediate consequence of the definitions.
ad 2. The recursive core described by \(d^*\) is not \(I\)-learned by \(S_{d^*}\) with respect to the hypothesis space \(\varphi^{d^*}\).

By construction, \(\varphi^{d^*}_{S_{d^*}}(0)\) equals \(1 \uparrow \infty\). So, on input of the first initial segment of \(0^\infty\), the learner \(S_{d^*}\) returns some \(\varphi^{d^*}\)-number of a non-total function, which is not conform. Note that \(0^\infty\) belongs to \(R_{d^*}\). Consequently, the recursive core described by \(d^*\) is not \(I\)-identified by \(S_{d^*}\) with respect to the hypothesis space \(\varphi^{d^*}\).

These two properties of \(d^*\) now imply that \(S\) is not an appropriate \(\text{resUniI}\)-learner for \(D\). Since \(S\) was chosen arbitrarily from all recursive learners, this proves the claim for \(I \in \{\text{Cons, Conf, Total}\}\).

Finally, if \(I = \text{Cex}\), the proof proceeds analogously, where “\(\psi_{i} = 1 \uparrow \infty\)” is replaced by “\(\psi_{i} = \uparrow \infty\)” for all \(i \neq e\). □

**Example 18** Let \(I \in \{\text{Cons, Conf}\}\) and define a description set \(D\) by

\[D := \{d \mid \varphi^{d}\text{ is a recursive function}\}\,.

Then \(D\) belongs to \(\text{resUniI}\), but \(D\) is not \(\text{ext UniI}\)-identifiable by any total recursive strategy.

*Proof.* It suffices to show, that the set \(D\) is \(\text{resUniCons}\)-learnable, but not \(\text{ext UniConf}\)-identifiable by any recursive learner.

Given a number \(d\) and some segment \(\alpha\), a \(\text{resUniCons}\)-learner for \(D\) just returns the minimal \(\varphi^{d}\)-index consistent for \(\alpha\). Since \(\varphi^{d}\) is recursive for each \(d \in D\), this yields a successful strategy (which has been defined as the method of “identification by enumeration” by Gold [8]).

In order to prove that \(D\) cannot be learned by any recursive strategy – even in the extended model \(\text{ext UniConf}\) – fix some recursive function \(S\). Now \(S\) is shown to be inappropriate for \(\text{ext UniConf}\)-learning of the whole class \(D\). This can be achieved by constructing a description \(d^*\) satisfying

1. \(d^*\) belongs to \(D\), but
2. the recursive core described by \(d^*\) is not \(\text{Conf}\)-learnable by \(S_{d^*}\) with respect to any hypothesis space.

For that purpose define for each number \(d\) a two-place function \(\psi\) by stages as follows.

*Stage 0.* Let \(\psi_0(0) := 0\). Go to stage 1.

In each stage \(k\) (\(k \geq 1\)), \(\psi_0(k)\) is defined by 0, if this forces the learner \(S_d\) into a mind change. Otherwise, \(\psi_0(k) := 1\). Furthermore, the function \(\psi_k\) is used
to make $S_d$ return some incorrect or non-conform hypothesis, if such a mind change on $\psi_0$ cannot be forced.

Stage $k$ ($k \geq 1$). Compute the values $S_d(\psi_0[k-1])$ and $S_d(\psi_0[k-1]0)$. If $S_d(\psi_0[k-1]) \neq S_d(\psi_0[k-1]0)$, then let $\psi_0(k) := 0$, otherwise $\psi_0(k) := 1$. Moreover let $\psi_k := \psi_0[k-1]0^{\infty}$. Go to stage $k + 1$.

As $S$ is recursive, this construction proceeds uniformly in $d$. Thus there is some fixed point value $d^\ast$ satisfying $\varphi^d = \psi$ for the numbering $\psi$ constructed from $S$ and $d^\ast$. This fixed point value will be used to show that $S$ is not an $ext\ UniConf$-learner for $D$.

End Construction of $d^\ast$

It remains to prove the desired properties.

ad 1. $d^\ast$ belongs to $D$.

This follows obviously from the construction, because all stages must be reached in the definition of the numbering $\psi$ corresponding to $S$ and $d^\ast$.

ad 2. The recursive core described by $d^\ast$ is not $Conf$-learnable by $S_d$, with respect to any hypothesis space.

Consider two cases.

Case 1. $S_d(\varphi_0^d[k-1]) \neq S_d(\varphi_0^d[k])$ for infinitely many $k \geq 1$.

Then $S_d$ cannot learn $\varphi_0^d$ correctly, because it fails to generate a convergent sequence of hypotheses.

Case 2. $S_d(\varphi_0^d[k-1]) = S_d(\varphi_0^d[k])$ for infinitely many $k \geq 1$.

For each such $k$, by the instructions in stage $k$, $\varphi_0^d[k] = \varphi_0^d[k-1]1$, $\varphi_k^d[k] = \varphi_0^d[k-1]0$, and

$$S_d(\varphi_k^d[k]) = S_d(\varphi_0^d[k-1]0) = S_d(\varphi_0^d[k-1]) = S_d(\varphi_0^d[k]) .$$

Now choose some arbitrary hypothesis space $\eta$.

Case 2.1. $\eta_{S_d, (\varphi_0^d)}(k)$ is defined for some $k \geq 1$ satisfying (1).

Then $\eta_{S_d, (\varphi_0^d)}(k) \neq \varphi_k^d(k)$ or $\eta_{S_d, (\varphi_0^d)}(k) \neq \varphi_0^d(k)$, although all these values are defined. Hence, for at least one of the functions $\varphi_0^d$ and $\varphi_k^d$, $S_d$ returns some hypothesis violating the conformity demands with respect to $\eta$. Consequently, $R_{d^\ast}$ is not $Conf$-learnable by $S_{d^\ast}$ with respect to $\eta$.

Case 2.2. $\eta_{S_d, (\varphi_0^d)}(k)$ is undefined for all $k \geq 1$ satisfying (1).

In particular $\eta_{S_d, (\varphi_0^d)}(k)$ is non-total for infinitely many $k \geq 1$. Thus, for the function $\varphi_0^d$, $S_d$ returns hypotheses incorrect with respect to $\eta$ infinitely often. Hence $S_{d^\ast}$ does not $Conf$-identify the class $R_{d^\ast}$ with respect to $\eta$.
4 Hierarchies of classes in uniform learning

As illustrated in Figure 1, the hierarchy of all inference classes has already been studied for the non-uniform learning model (cf. [2–6,10,15]). Now the scope of the subsequent theorems is to investigate the corresponding hierarchies for uniform identification – in the basic model as well as in the restricted and extended cases.

Actually hierarchies for the basic and the extended model can immediately be deduced from Corollary 15: since for any \( I, I' \in \mathbb{I} \) with \( I' \setminus I \neq \emptyset \) there is some description set in \( \text{Uni} I' \setminus \text{ext Uni} I \), both \( \text{Uni} I' \setminus \text{Uni} I \) and \( \text{ext Uni} I' \setminus \text{ext Uni} I \) must be non-empty. For a proof of Corollary 15 the required description set was chosen to represent a recursive core belonging to \( I' \setminus I \), which obviously disallows uniform \( I \)-learning. Together with a proof for \( \text{Uni} I \subseteq \text{Uni} I' \) this yields the same hierarchy for the \( \text{Uni} \)-model\(^1\) as has been verified in the non-uniform case – not a very astonishing result. It would be more remarkable to find description sets in \( \text{Uni} I' \setminus \text{Uni} I \) (and in parallel for the restricted and extended models), such that each recursive core described belongs to the class \( I \). Indeed the following results show that such description sets exist for nearly all the models. In particular, any separation verified here is achieved by descriptions of \emph{finite} recursive cores (most often even singletons or cores consisting of two elements). In the non-uniform model finite classes are the most simple sets regarding learnability: they can be identified with respect to any criterion \( I \in \mathbb{I} \) by a quite straightforward strategy. But despite their trivial role in the basic inference model these classes are complex enough to separate inference criteria in meta-learning.

Theorem 19 first summarizes the inclusions obtained for uniform learning of finite recursive cores; which of these are proper inclusions will be studied in the subsequent analysis.

**Theorem 19** Let \( I, I' \in \mathbb{I} \) be inference classes, such that \( I \subseteq I' \).

1. \( \text{Uni} I [\ast] \subseteq \text{Uni} I' [\ast] \).
2. If \( (I, I') \neq (\text{Ex}_0, \text{Total}) \), then \( \text{res Uni} I [\ast] \subseteq \text{res Uni} I' [\ast] \).
3. If \( (I, I') \neq (\text{Total}, \text{Cons}) \) and \( (I, I') \neq (\text{Total}, \text{Conf}) \), then \( \text{ext Uni} I [\ast] \subseteq \text{ext Uni} I' [\ast] \).

\(^1\) For most of the criteria parallels are observed easily in the restricted and extended cases.
Sketch of Proof. The first two claims can be verified easily for the pair \((I, I') = (Total, Cons)\): a uniform Cons-strategy just has to simulate a uniform Total-strategy and test its output for consistency. Any consistent intermediate hypothesis is returned without modification, any inconsistent hypothesis can be changed into an arbitrary consistent output.

The following idea for a proof of the second claim for the pair \((Ex_0, Cex)\) has been suggested by Jochen Nessel: if \(D\) is a description set belonging to \(res Uni Ex_0\) and \(S\) is a corresponding uniform strategy, then a \(res Uni Cex\)-learner \(T\) for \(D\) just has to replace the “?”-signs returned by \(S\) with correct or convergently incorrect intermediate hypotheses. Whenever \(S\) returns a hypothesis different from “?” , then \(T\) may do the same. So, if \(S_d(f[n]) = ?\) for some recursive function \(f\) and some \(d, n \geq 0\), then \(T_d\) (on input \(f[n]\)) looks for some pair \((i, m)\) of numbers, such that \(i\) is consistent for \(f[n]\) with respect to \(\varphi^d\) and \(S_d(\varphi^d_i[m]) = i\). As soon as such a pair \((i, m)\) is found, \(T_d\) returns \(i\). If \(f \in R_d\) and \(\varphi^d_i \neq f\), then \(S_d(f[m]) \neq i\) because of the choice of \(S\). So \(\varphi^d_i[m] \neq f[m]\), i.e. \(i\) is convergently incorrect for \(f\) with respect to \(\varphi^d\).

In order to prove Claim 3 for the pairs \((Ex_0, Total)\) and \((Ex_0, Cons)\) the hypothesis spaces used for \(Ex_0\)-learning have to be adjusted. To allow uniform Total-learning, an arbitrary total function (for example the function constantly zero) is added to the hypothesis space at a fixed index. This fixed index may be output, whenever the uniform \(Ex_0\)-learner returns “?” . This yields a uniform Total-strategy. For uniform Cons-learning the old hypothesis spaces are mixed with an enumeration of all recursive functions of finite support. If the \(Ex_0\)-learner returns “?” , then a Cons-learner may return some consistent hypothesis corresponding to a suitable function of finite support.

All other statements of the theorem follow immediately from the definitions of the corresponding learning classes. □

Figure 2 summarizes the results to be proved in the subsequent sections. Moreover it will turn out that

- all separations concerning the Uni-model are achieved via descriptions of singletons;
- all separations concerning the \(res Uni\)-model – except for \(res Uni Total \setminus res Uni E x_m \neq \emptyset (m \geq 0)\) – are achieved via descriptions of singletons;
- all separations concerning the ext Uni-model are achieved via descriptions of recursive cores consisting of no more than 2 functions.

Note that singleton recursive cores can never yield separations in the extended model of uniform learning: as for each recursive function \(f\) there is a numbering \(\psi\) with \(\psi_0 = f\), the strategy constantly zero witnesses to the fact that each description set representing singletons is \(ext Uni E x_0\)-identifiable and thus \(ext Uni I\)-identifiable for all \(I \in \mathbb{I}\). Therefore recursive cores consisting of two
functions constitute the optimal result in this context.

\[
\begin{align*}
UniEx_0[*] & \rightarrow UniEx_1[*] \\
UniTotal[*] & \rightarrow UniCex[*] \rightarrow UniEx[*] \rightarrow UniBc[*] \rightarrow UniBc^*[*] \\
UniCons[*] & \rightarrow UniConf[*] \\
resUniEx_0[*] & \rightarrow resUniEx_1[*] \\
resUniTotal[*] & \rightarrow resUniCex[*] \rightarrow resUniEx[*] \rightarrow resUniBc[*] \rightarrow resUniBc^*[*] \\
resUniCons[*] & \rightarrow resUniConf[*] \\
extUniEx_0[*] & \rightarrow extUniEx_1[*] \\
extUniTotal[*] & = extUniCex[*] = extUniEx[*] \rightarrow extUniBc[*] = extUniBc^*[*] \\
extUniCons[*] & \rightarrow extUniConf[*]
\end{align*}
\]

Fig. 2. The hierarchies for the three models of uniform learning of finite recursive cores. Vectors indicate proper inclusions; if two classes are not connected by a sequence of vectors in one direction, they are incomparable.

The corresponding proofs are the scope of the studies below.

4.1 Similarities between the hierarchies

Since all proofs regarding the hierarchies in Figure 2 meet a common structure, the criteria \(Ex\) and \(Bc\) are chosen for a first example. The corresponding separations are verified in detail, whereas the proofs for other inference classes are just sketched.

**Theorem 20** There exists a description set \(D \in resUniBc \setminus extUniEx\), such that each recursive core described by \(D\) consists of at most 2 functions.

**Proof.** The definition of \(D\) uses the following idea: first for each total recursive learner \(S\) and each number \(d\) a numbering \(\psi\) is constructed. The recursive core of this numbering \(\psi\) will consist of at most 2 functions and will not be
identifiable in the limit by $S_d$. Then the construction yields some fixed
point value $d^*$, such that $S_d$ fails to identify $R_{d^*}$. Moreover $R_{d^*}$ will have no more
than 2 elements. Finally these fixed point values are used as descriptions in
the set $D$. For each recursive learner $S$ such a fixed point $d^*$ is included in
$D$. Then $D$ is not suitable for extended uniform learning in the limit, because
each recursive strategy $S$ fails for at least one recursive core $R_{d^*}$. A careful
carrying out of this idea will still enable restricted uniform $Bc$-learning of the
constructed set $D$.

More formally: for any recursive learner $S$ and any number $d$ a partial-recursive
numbering $\psi$ is constructed by stages as follows.

Stage 0. Let $\psi_0(0) := 0$ and $n_1 := 0$. Go to stage 1.

In each stage $k$ ($k \geq 1$) the strategy $S_d$ is presented 2 different gradually
growing extensions of $\psi_0[n_k]$. As soon as $S_d$ changes its mind on at least one
of these segments (Case A), the function $\psi_0$ is extended accordingly. Otherwise
(not Case A) $\psi_{2k-1}$ and $\psi_{2k}$ become two different recursive functions, such that
the sequence of hypotheses returned by $S_d$ converges to the same program on
both $\psi_{2k-1}$ and $\psi_{2k}$.

The idea behind this is, that $S_d$ cannot Ex-identify the recursive core of the
numbering $\psi$: either Case A occurs in each stage or Case A fails at least
once. If Case A occurs in each stage, then $\psi_0$ becomes a recursive function, on
which $S_d$ changes its mind infinitely often. If Case A does not occur in stage $k$
($k \geq 1$), then $S_d$ guesses the same program for the two different functions
$\psi_{2k-1}$ and $\psi_{2k}$ in the limit.

Stage $k$ ($k \geq 1$). Let $\psi_{2k-1}[n_k] = \psi_{2k}[n_k] = \psi_0[n_k]$. Search for a number $z$
satisfying

\[ S_d(\psi_0[n_k](2k-1)^2) \neq S_d(\psi_0[n_k]) \text{ or } S_d(\psi_0[n_k](2k)^2) \neq S_d(\psi_0[n_k]) . \]

(2)

In parallel extend $\psi_{2k-1}$ with a sequence of the value $2k - 1$ and $\psi_{2k}$ with a
sequence of the value $2k$, until the search for $z$ is successful.

Case A. There exists a number $z$, such that (2) is fulfilled.
Then let $z_k$ be the minimal number $z$ satisfying (2). Moreover define $n_{k+1} := n_k + z_k$
and

\[
\psi_0[n_{k+1}] := \begin{cases} 
\psi_0[n_k](2k-1)^{z_k}, & \text{if } S_d(\psi_0[n_k](2k-1)^{z_k}) \neq S_d(\psi_0[n_k]) , \\
\psi_0[n_k](2k)^{z_k}, & \text{if } S_d(\psi_0[n_k](2k-1)^{z_k}) = S_d(\psi_0[n_k]) ,
\end{cases}
\]
as well as
\[
\psi_{2k-1} := \begin{cases} 
\psi_0, & \text{if } S_d(\psi_0[n_k](2k-1)^z_k) \neq S_d(\psi_0[n_k]), \\
\psi_0[n_k](2k-1)^z_k \uparrow^\infty, & \text{if } S_d(\psi_0[n_k](2k-1)^z_k) = S_d(\psi_0[n_k]).
\end{cases}
\]

\[
\psi_{2k} := \begin{cases} 
\psi_0, & \text{if } S_d(\psi_0[n_k](2k-1)^z_k) = S_d(\psi_0[n_k]), \\
\psi_0[n_k](2k)^z_k \uparrow^\infty, & \text{if } S_d(\psi_0[n_k](2k-1)^z_k) \neq S_d(\psi_0[n_k]).
\end{cases}
\]

Go to stage \(k + 1\).

Remark. If there is no number \(z\) satisfying (2), i.e. if Case A is not fulfilled, then stage \(k\) does not terminate. In this case \(\psi_{2k-1} = \psi_0[n_k](2k-1)^\infty\) and \(\psi_{2k} = \psi_0[n_k](2k)^\infty\). Furthermore, \(\psi_0(x)\) remains undefined for all \(x > n_k\), that means \(\psi_0 = \psi_0[n_k] \uparrow^\infty\); stage \(k + 1\) is not reached in the computation. In particular, for all \(i > 2k\), \(\psi_i\) is the empty function. \textit{End Construction of \(\psi\)}

Note that the whole construction is uniformly effective in \(S\) and \(d\). For any recursive learner \(S\) this implies the existence of some number \(d^*\), such that \(\varphi^{d^*}\) equals the numbering \(\psi\) constructed from \(S\) and \(d^*\). From now on, for any fixed recursive strategy \(S\), such a corresponding number \(d^*\) will be called a \textit{fixed point associated to \(S\)}. Thus the description set \(D\) can be defined as explained in the idea in the beginning of the proof:

\[
D := \{ d \mid d\text{ is a fixed point associated to some recursive function } S \}.
\]

The construction of the numberings \(\psi\) (by definition corresponding to the recursive cores described by \(D\)) provides two helpful observations:

\textit{Fact 1.} If \(d \in D\), then \(R_d = \{ \varphi_0^d \}\) or there are some \(k \geq 1\) and \(n \geq 0\), such that \(R_d = \{ \varphi_{2k-1}^d, \varphi_{2k}^d \} = \{ \varphi_0^d[n](2k-1)^\infty, \varphi_0^d[n](2k)^\infty \}\).

This can be verified easily: the construction of a numbering \(\psi\) either runs through all stages or there is some unique stage, which is never left. If all stages are reached, the corresponding recursive core consists of the function \(\psi_0\) only. Otherwise, where the number of the last stage reached is \(k\) \((k \geq 1)\), the recursive core of \(\psi\) contains exactly the functions \(\psi_{2k-1}\) and \(\psi_{2k}\) according to the remark below Case A.

\textit{Fact 2.} If \(d \in D\) and stage \(k\) \((k \geq 1)\) is reached in the construction of the corresponding numbering \(\psi = \varphi^d\) (with the value \(n_k\) accordingly), then \(\varphi_0^d(x) \leq \varphi_0^d(n_k) < 2k - 1\) for all \(x \leq n_k\).

This fact is verified by a simple induction.

It remains to prove the following claim.

\textit{Claim.}

(1) Each recursive core described by \(D\) consists of at most 2 functions,
(2) $D \in \text{res UniBc}$,
(3) $D \notin \text{ext UniEx}$.

ad 1. This is a direct consequence of Fact 1.

ad 2. Define a learner $T$ for any recursive function $f$ and all $n \geq 0$ by $T(f[n]) := \max \{ f(x) \mid x \leq n \}$. This learner $Bc$-identifies any recursive core $R_d$ described by $D$ with respect to its corresponding numbering $\varphi^d$. To verify this, fix some $d \in D$. By Fact 1 it suffices to consider two cases.

Case 1. $R_d = \{ \varphi^d_0 \}$.
Then each stage $k$ ($k \geq 0$) is reached in the construction of the corresponding numbering $\psi$. In particular, Case A occurs in each stage. For any $k \geq 1$ this implies that either $\varphi^d_{2k-1} = \varphi^d_0$ or $\varphi^d_{2k} = \varphi^d_0$. To be more concrete,

\begin{align*}
\varphi^d_{2k-1} = \varphi^d_0 & \iff \varphi^d_0(n_k + 1) = \cdots = \varphi^d_0(n_{k+1}) = 2k - 1 \quad \text{and} \quad \\
\varphi^d_{2k} = \varphi^d_0 & \iff \varphi^d_0(n_k + 1) = \cdots = \varphi^d_0(n_{k+1}) = 2k .
\end{align*}

Moreover, for any $n \in \{ n_k + 1, \ldots, n_{k+1} \}$, Fact 2 implies

$$T(\varphi^d_0[n]) = \max \{ \varphi^d_0(x) \mid x \leq n \} = \varphi^d_0(n_k + 1) .$$

As (3) holds for any $k \geq 1$, this proves $\varphi^d_{T(\varphi^d_0[n])} = \varphi^d_0$ for all $n \geq 0$. Hence $T$ is a $Bc$-learner for $R_d$ with respect to $\varphi^d$.

Case 2. $R_d = \{ \varphi^d_{2k-1}, \varphi^d_{2k} \}$ for some $k \geq 1$.
Then, by construction, $\varphi^d_{2k-1} = \varphi^d_0[n_k](2k - 1)^\infty$ and $\varphi^d_{2k} = \varphi^d_0[n_k](2k)^\infty$. Clearly, in the construction of the corresponding numbering $\psi$, stage $k$ must have been reached. Fact 2 then implies $\varphi^d_0(x) < 2k - 1$ for all $x \leq n_k$. So $T(\varphi^d_{2k-1}[n]) = \max \{ \varphi^d_{2k-1}(x) \mid x \leq n \} = 2k - 1$ and $T(\varphi^d_{2k}[n]) = \max \{ \varphi^d_{2k}(x) \mid x \leq n \} = 2k$ for all $n > n_k$. Consequently, the learner $T$ correctly $Bc$-identifies (even $Ex$-identifies) the class $R_d$ with respect to the numbering $\varphi^d$.

Since for any $d \in D$ the learner $T$ is a successful $Bc$-strategy for $R_d$ with respect to $\varphi^d$, the description set $D$ is suitable for uniform $Bc$-identification in the restricted model. So Claim 2 is verified.

ad 3. Assume to the contrary, that $D$ is suitable for extended uniform $Ex$-identification. Then by Proposition 16 there exists a recursive strategy $S$, such that each recursive core $R_d$ described by $D$ is identified in the limit by $S_d$. Now let $d^*$ be a fixed point associated to $S$. By definition this fixed point $d^*$ belongs to the set $D$. Therefore $R_{d^*}$ is $Ex$-identified by $S_{d^*}$. According to Fact 1 only the following two cases must be considered.

Case 1. $R_{d^*} = \{ \varphi^d_0 \}$.
Then each stage $k$ ($k \geq 0$) is reached in the construction of the corresponding
numbering $\psi$. In particular, Case A occurs in each stage. For any $k \geq 1$ this implies that $S_{d^*}(\wp_0 \upharpoonright n_k) \neq S_{d^*}(\wp_0 \upharpoonright n_{k+1})$. Since $n_{k+1} > n_k$ for all $k \geq 1$, the learner $S_{d^*}$ changes its hypothesis on $\wp_0$ infinitely often. Thus $S_{d^*}$ does not identify $R_{d^*}$ in the limit – a contradiction.

**Case 2.** $R_{d^*} = \{\wp_{2k-1}^*, \wp_{2k}^*\}$ for some $k \geq 1$.

Then, by construction, $\wp_{2k-1}^* = \wp_0^* \upharpoonright n_k(2k - 1)^\infty$ and $\wp_{2k}^* = \wp_0^* \upharpoonright n_k(2k)^\infty$. Furthermore, stage $k$ is the last stage reached in the definition of the corresponding numbering $\psi$. In particular, there does not exist any number $z$, such that (2) is fulfilled. Thus $S_{d^*}(\wp_0 \upharpoonright n_k(2k - 1)^z) = S_{d^*}(\wp_0 \upharpoonright n_k(2k)^z)$ for all $z \geq 0$. That means, that the sequences of hypotheses returned by $S_{d^*}$ on the two different functions $R_{d^*}$ converge to the same program. Consequently, $S_{d^*}$ does not identify $R_{d^*}$ in the limit. This yields a contradiction.

As both cases result in a contradiction, the assumption $D \in \text{ext UniEx}$ is wrong. This proves Claim 3. $\square$

**Corollary 21**

1. $\text{UniEx}[\ast] \subset \text{UniBc}[\ast]$.
2. $\text{res UniEx}[\ast] \subset \text{res UniBc}[\ast]$.
3. $\text{ext UniEx}[\ast] \subset \text{ext UniBc}[\ast]$.

Thus the separation of uniform Bc- and Ex-learning is verified for all three models; nevertheless Theorem 22 offers an interesting reinforcement of Theorem 20 for the case of res Uni-identification, namely that in this model singleton recursive cores are sufficient to obtain the desired separation.

**Theorem 22** There exists a description set $D \in \text{res UniBc} \setminus \text{UniEx}$, such that each recursive core described by $D$ is a singleton set.

**Proof.** Now the idea in the proof of Theorem 20 is adjusted to fit the UniEx-model: first for each acceptable numbering $\tau$, each recursive learner $S$, and each number $d$, a numbering $\psi$ is constructed. The recursive core of $\psi$ will be a singleton set and will not be Ex-identifiable by $S$ with respect to the hypothesis space $\tau$. The construction yields some fixed point value $d^*$, such that $S_{d^*}$ fails to identify $R_{d^*}$ with respect to $\tau$. Again for any acceptable numbering and any recursive learner these fixed point values are collected in the description set $D$.

More formally: for any acceptable numbering $\tau$, any recursive learner $S$, and any number $d$ a partial-recursive numbering $\psi$ is constructed by stages as follows.

**Stage 0.** Let $\psi_0(0) := 0$ and $n_1 := 0$. Go to stage 1.

In each stage $k$ ($k \geq 1$) the function $\psi_k$ first adapts the initial segment $\psi_0 \upharpoonright n_k$ constructed so far. This segment is extended, until either $S_d$ changes its mind.

24
on $\psi_k$ or the function computed by $\tau$ – for the program $S_d$ guesses – returns a value for some input greater than $n_k$. In the first case (Case A.1) the function $\psi_0$ is extended accordingly. In the second case (Case A.2) the function $\psi_0$ is extended with a value differing from the one returned by $\tau$. If neither Case A.1 nor Case A.2 occurs, then $\psi_k$ is extended ad infinitum.

The idea behind this is that $S_d$ cannot $E_x$-identify the recursive core of $\psi$ with respect to $\tau$: if in each step either Case A.1 or Case A.2 occurs, then $\psi_0$ becomes a recursive function, on which $S_d$ changes its mind infinitely often or returns incorrect programs infinitely often. If, at some stage $k$, neither Case A.1 nor Case A.2 occurs, then $\psi_k$ becomes a recursive function, but the program $S_d$ guesses for $\psi_k$ in the limit is wrong with respect to $\tau$.

**Stage $k$ ($k \geq 1$).** Search for a number $z$ satisfying

\[
S_d(\psi_0[n_k](k + 1)^z) \neq S_d(\psi_0[n_k]) \quad (4)
\]

or $\tau_{S_d(\psi_0[n_k])}(n_k + 1)$ is defined within $z$ steps of computation $\quad (5)$

In parallel extend $\psi_k$ with the value $k + 1$, until the search for $z$ is successful.

**Case A.** There exists a number $z$, such that (4) or (5) is fulfilled. Then let $z_k$ be the minimal number satisfying (4) or (5). Two cases are distinguished.

**Case A.1.** (4) is fulfilled for $z_k$.

Then define $n_{k+1} := n_k + z_k$ and $\psi_0[n_{k+1}] := \psi_0[n_k](k + 1)^z_k$ as well as $\psi_k := \psi_0$. Go to stage $k + 1$.

**Case A.2.** (4) is not fulfilled for $z_k$ (so (5) is fulfilled for $z_k$).

Then let $e_k := \pi_y(\tau_{S_d(\psi_0[n_k])}(n_k + 1))$. Moreover define $n_{k+1} := n_k + 1$ and $\psi_0[n_{k+1}] := \psi_0[n_k]e_k$ as well as $\psi_k = \psi_0[n_k](k + 1)^{z_k} \uparrow \infty$. Go to stage $k + 1$.

**Remark.** If there is no number $z$ satisfying (4) or (5), i.e. if Case A does not occur, then stage $k$ does not terminate. In this case $\psi_k := \psi_0[n_k](k + 1)^\infty$. Furthermore, $\psi_0(x)$ remains undefined for all $x > n_k$, i.e. $\psi_0 = \psi_0[n_k] \uparrow \infty$; stage $k + 1$ is not reached. In particular, for all $i > k$, $\psi_i$ is the empty function.

**End construction of $\psi$**

Note that the whole construction is uniformly effective in $\tau$, $S$, and $d$. Hence for any acceptable numbering $\tau$ and any recursive function $S$ there is some $d^*$, such that $\varphi^{d^*}$ is the numbering $\psi$ constructed from $\tau$, $S$, and $d^*$. Such a number $d^*$ is called a **fixed point associated to $\tau$ and $S$**. Finally, let

\[
D := \{ d \mid d \text{ is a fixed point associated to some acceptable numbering } \tau \text{ and some recursive function } S \}
\]
The definition of $D$ provides two helpful observations, both of which can be verified easily from the construction above.

**Fact 1.** Let $d$ be an element of $D$.

(1) If in each stage of the construction Case A occurs, then $R_d = \{ \varphi^d_0 \}$ and, for any $k \geq 1$, $\varphi^d_k = \varphi^d_0$ iff $\varphi^d_0(n_k + 1) = \cdots = \varphi^d_0(n_{k+1}) = k + 1$.

(2) If at some stage $k$ ($k \geq 1$, Case A does not occur, then $R_d = \{ \varphi^d_k \} = \{ \varphi^d_0(n_k)[k + 1] \}$.

**Fact 2.** If $d$ belongs to $D$ and stage $k$ ($k \geq 1$) is reached in the construction of $\varphi^d$ (with the corresponding value $n_k$), then $\varphi^d_0(x) < k + 1$ for all $x \leq n_k$.

It remains to prove the following claim.

**Claim.**

(1) Each recursive core described by $D$ is a singleton set,

(2) $D \in res Uni Bc$,

(3) $D \notin Uni Ex$.

**ad 1.** This is a direct consequence of Fact 1.

**ad 2.** Define a learner $T$ for any recursive function $f$ and any $n \geq 0$ by $T(0^{n+1}) := 0$ and $T(f[n]) := \max\{ f(x) \mid x \leq n \} - 1$, if $f[n] \neq 0^{n+1}$. This learner $Bc$-identifies any recursive core $R_d$ described by $D$ with respect to its corresponding numbering $\varphi^d$. To verify this, fix some $d \in D$. By Fact 1 it suffices to consider two cases.

**Case 1.** $R_d = \{ \varphi^d_0 \}$.

Then each stage $k$ ($k \geq 0$) is reached in the construction of the corresponding numbering $\psi$. In particular, Case A occurs in each stage. For any $k \geq 1$, Fact 1 implies that $\varphi^d_k = \varphi^d_0$ iff $\varphi^d_0(n_k + 1) = k + 1$. Applying Fact 2 evidences

$$\varphi^d_k = \varphi^d_0 \iff \max\{ \varphi^d_0(x) \mid x \leq n_k + 1 \} = k + 1$$

for all $k \geq 1$. Thus, for any $n \geq 0$, the learner $T$ satisfies $T(\varphi^d_0[n]) = 0$ or $T(\varphi^d_0[n]) = \max\{ \varphi^d_0(x) \mid x \leq n \} - 1 \in \{ k \mid \varphi^d_k = \varphi^d_0 \}$. This implies $\varphi^d_{T(\varphi^d_0[n])} = \varphi^d_0$ for all $n \geq 0$. Hence $T$ is a $Bc$-learner for $R_d$ with respect to $\varphi^d$.

**Case 2.** $R_d = \{ \varphi^d_k \}$ for some $k \geq 1$, such that $\varphi^d_k \neq \varphi^d_0$.

Then, by construction, $\varphi^d_k = \varphi^d_0[n_k][k + 1]$. Obviously, in the construction of the corresponding numbering $\psi$, stage $k$ must have been reached. Fact 2 implies $\varphi^d_0(x) < k + 1$ for all $x \leq n_k$. So $T(\varphi^d_k[n]) = \max\{ \varphi^d_k(x) \mid x \leq n \} - 1 = k$ for all $n \geq n_k + 1$. Consequently, the learner $T$ correctly $Bc$-identifies (even $Ex$-identifies) the class $R_d$ with respect to the numbering $\varphi^d$.  

26
Since for any $d \in D$ the learner $T$ is a successful $Bc$-strategy for $R_d$ with respect to $\varphi^d$, the description set $D$ is suitable for uniform $Bc$-identification in the restricted model. So Claim 2 is verified.

ad 3. Assume to the contrary, that $D$ is suitable for uniform $Ex$-identification. Then, by Proposition 16, there exist an acceptable numbering $\tau$ and a recursive strategy $S$, such that each recursive core $R_d$ described by $D$ is identified in the limit by $S_d$ with respect to $\tau$. Now let $d^* \in D$ be a fixed point associated to $\tau$ and $S$, so by assumption $R_{d^*}$ is $Ex$-identified by $S_{d^*}$ with respect to $\tau$. According to Fact 1, only the following two cases must be considered.

Case 1. $R_{d^*} = \{\varphi^d_{\psi_0}\}$.
Then each stage $k (k \geq 0)$ is reached in the construction of the corresponding numbering $\psi$. In particular, Case A occurs in each stage. For any $k \geq 1$ this implies that either $S_{d^*}(\varphi^d_{\psi_0}[n_k]) \neq S_{d^*}(\varphi^d_{\psi_0}[n_{k+1}])$ or $\tau_{S_{d^*}(\varphi^d_{\psi_0}[n_k])}(n_k + 1) \neq \tau_{\psi_0}(\varphi^d_{\psi_0}(n_k + 1))$ since $n_{k+1} > n_k$ for all $k \geq 1$, the learner $S_{d^*}$ changes its hypothesis on $\varphi^d_{\psi_0}$ infinitely often or returns incorrect hypotheses for $\varphi^d_{\psi_0}$ infinitely often. Thus $S_{d^*}$ does not identify $R_{d^*}$ in the limit – a contradiction.

Case 2. $R_{d^*} = \{\varphi^d_{\psi_k}\}$ for some $k \geq 1$, such that $\varphi^d_{\psi_k} \neq \varphi^d_{\psi_0}$.
Then by construction $\varphi^d_{\psi_k} = \varphi^d_{\psi_0}[n_k](k + 1)^\infty$ and stage $k$ is the last stage reached in the definition of the corresponding numbering $\psi$. In particular, there does not exist any number $z$, such that (4) or (5) is fulfilled. Thus $S_{d^*}(\varphi^d_{\psi_0}[n_k](k + 1)^z) = S_{d^*}(\varphi^d_{\psi_0}[n_k])$ for all $z \geq 0$. Moreover $\tau_{S_{d^*}(\varphi^d_{\psi_0}[n_k])}$ is undefined on input $n_k + 1$. In particular, $S_{d^*}(\varphi^d_{\psi_0}[n_k])$ is a $\tau$-program for $\varphi^d_{\psi_k}$. That means, that the sequence of hypotheses, returned by $S_{d^*}$ on the function in $R_{d^*}$, converges to a wrong $\tau$-number. Consequently, $S_{d^*}$ does not $Ex$-identify $R_{d^*}$ with respect to $\tau$. This yields a contradiction.

As both cases result in a contradiction, the assumption $D \in UniEx$ is wrong. This proves Claim 3. □

Evidently, recursive cores of no more than two functions are adequate for the separation of extended uniform $Bc$-learning from extended uniform $Ex$-learning; furthermore for the non-extended case singleton recursive cores meet the requirements. As Theorem 24 will show, this agrees with the results for most of the other separations in Figure 2. Yet considering description sets uniformly Total-learnable and not uniformly $Ex_m$-learnable (for $m \geq 1$), this observation only holds for $Uni$- and ext $Uni$-learning. Regarding the res $Uni$-model, a separation with recursive cores of $m + 2$ functions is the best result obtainable. The corresponding proof is just sketched.

**Theorem 23** Let $m \geq 0$.

1. There exists a description set $D \in UniTotal \setminus UniEx_m$, such that each
recursive core described by $D$ is a singleton set.

(2) There exists a description set $D \in \text{UniTotal} \setminus \text{ext UniEx}_m$, such that each recursive core described by $D$ consists of at most 2 functions.

(3) There exists a description set $D \in \text{res UniTotal} \setminus \text{ext UniEx}_m$, such that each recursive core described by $D$ consists of at most $m + 2$ functions.

(4) If $D \in \text{res UniTotal}$ and each recursive core described by $D$ consists of at most $m + 1$ functions, then $D \in \text{res UniEx}_m$.

Assertions 1 and 2 coincide with the corresponding results for other inference classes, whereas Assertions 3 and 4 imply that, in general, the separations in restricted uniform learning with total intermediate hypotheses cannot be witnessed by recursive cores consisting of one or two functions. To disallow $\text{ext UniEx}_m$-identification, cores of cardinality $m + 2$ suffice, moreover Assertion 4 states that in general this result cannot be improved.

**Sketch of Proof. ad 1.** For any recursive learner $S$ and any number $d$ a function $\psi$ is constructed as follows.

In stage 0 let $\psi_0(0) := 0$. Extend $\psi_0$ by 0’s, until $S_d(0^2) \neq ?$ for some minimal $x \geq 1$ and $\tau_{S_d(0^y)}(y) = 0$ for some $y \geq x$. If such a pair $(x, y)$ does not exist (not Case A), then stage 0 does not terminate and $\psi_0 = 0^\infty$; otherwise (Case A) extend $\psi_0$ by $(m + 2)^{\uparrow \infty}$ and go to stage 1 with $n_1 := y - 1$.

In each stage $k$, for $1 \leq k \leq m$, let $\psi_k[n_k] := \psi_{k-1}[n_k]$.

(* Note that $\tau_{S_d(\psi_{k-1})}(n_k + 1) = k - 1$.*)

Extend $\psi_k$ by k’s, until $S_d(\psi_k[n_k]) \neq S_d(\psi_k[n_k][k^x])$ for some minimal $x \geq 1$ and $\tau_{S_d(\psi_k[n_k][k^x])}(n_k + y + 1) = k$ for some $y \geq x$. If such a pair $(x, y)$ does not exist (not Case A), then stage $k$ does not terminate and $\psi_k = \psi_{k-1}[n_k][k^\infty]$; otherwise (Case A) let $n_{k+1} := n_k + y$, extend $\psi_k$ by $(m + 2)^{\uparrow \infty}$, and go to stage $k + 1$.

In stage $m + 1$ let $\psi_{m+1} = \psi_m[n_{m+1}](m + 1)^\infty$ and stop. All functions $\psi_i$, for $i > m + 1$, remain empty.

If in any stage $k$ ($k \leq m$) Case A is not fulfilled, then $\psi_k$ is the only recursive function enumerated by $\psi$, but, on input of the values of $\psi_k$, $S_d$ does not converge to a $\tau$-program of $\psi_k$. If Case A occurs in all stages $k$ ($k \leq m$), then stage $m + 1$ is reached and $\psi_{m+1}$ is the only recursive function enumerated by $\psi$. In this case, the learner $S_d$ must change its mind at least $m + 1$ times to identify $\psi_{m+1}$. Consequently, $S_d$ is no $\text{Ex}_m$-learner for the recursive core of the numberings $\psi$.

Defining $D$ by analogy with the proof of Theorem 22 yields a description set belonging to $\text{UniTotal}[\ast] \setminus \text{UniEx}_m[\ast]$ (details of the verification can be transferred). Moreover all recursive cores described by $D$ will be singleton sets.

**ad 2.** For any recursive learner $S$ and any number $d$ a function $\psi$ is constructed as follows. In stage 0 let $\psi_0(0) := 0$. Extend $\psi_0$ by 0’s, until $S_d(0^2) \neq ?$ for some
minimal $x \geq 1$. If such an $x$ does not exist, then $\psi_0 = 0^\infty$. Otherwise define
$n_1 := x - 1$, let $\psi_0[n_1] := 0^x$, $t_1 := 0$; go to stage 1.
In stage $k$ ($1 \leq k \leq m$) let $\psi_k[n_k] := \psi_{t_k}$. Extend $\psi_k$ with $k$’s, $\psi_{t_k}$, with
t$1$’s, until $S_d(\psi_k[n_k], k) \neq S_d(\psi_k[n_k])$ or $S_d(\psi_k[n_k]t_k^x) \neq S_d(\psi_k[n_k])$ for some
minimal $x \geq 1$. If $x$ does not exist (not Case A), then stage $k$ does not terminate
and $\psi_k = \psi_{t_k}[n_k]k^\infty$, $\psi_{t_k} = \psi_{t_k}[n_k]t_k^\infty$. Otherwise (Case A) let $\{z, t_{k+1}\} = \{k, t_k\}$, where $t_{k+1}$ is chosen to satisfy $S_d(\psi_k[n_k]t_{k+1}) \neq S_d(\psi_k[n_k])$. Then
define $n_{k+1} := n_k + x$, $\psi_{t_{k+1}}[n_{k+1}] := \psi_{t_k}[n_k]t_{k+1}$, extend $\psi$ by $(m + 2)^\uparrow \infty$,
and go to stage $k + 1$.
In stage $m+1$ define $\psi_{m+1} := \psi_{t_{m+1}}[n_{m+1}] (m+1)^\infty$, $\psi_{m+1} := \psi_{t_{m+1}}[n_{m+1}] t_{m+1}^\infty$,
and stop.

If stage 1 is not reached, then $\psi_0 = 0^\infty$, but $S_d$ always returns $\psi_0$. If in any
stage $k$ ($1 \leq k \leq m$) Case A is not fulfilled, then the recursive core of $\psi$
equals
$\{\psi_k, \psi_{t_k}\}$, but $S_d$ does not $\text{Ex}_m$-identify this set with respect to any hypothesis
space. If Case A occurs in all stages $k$ ($1 \leq k \leq m$), then stage $m+1$ is reached
and the recursive core of $\psi$ equals $\{\psi_{t_{m+1}}[n_{m+1}] (m+1)^\infty$, $\psi_{t_{m+1}}[n_{m+1}] t_{m+1}^\infty\}$. Since
$S_d$ changes its mind on $\psi_{t_{m+1}}[n_{m+1}]$ at least $m$ times, $S_d$ cannot $\text{Ex}_m$-
identify this set with respect to any hypothesis space. Note that in any case
the recursive core of $\psi$ has no more than 2 elements.

Defining $D$ as usual yields a description set belonging to $\text{UniTotal}[\ast]$, but not
to $\text{ext UniEx}_m[\ast]$. Details are omitted.

$\text{ad 3.}$ Here the construction proceeds by analogy. The only difference is, that
in Case A at stage $k$ the function $\psi_z$ is extended by $(m + 2)^\infty$ instead of
$(m + 2)^\uparrow \infty$. This makes the hypothesis $z$ total with respect to $\psi$. The price
paid for this is an increase in the number of functions contained in the recursive
core constructed; in the worst case $m+2$ functions ($\psi_0, \ldots, \psi_{m+1}$) are obtained.

$\text{ad 4.}$ If $D$ fulfils the conditions above and $S$ is a strategy appropriate for
$\text{res UniTotal}$-identification of $D$, then a $\text{res UniEx}_m$-learner $T$ for $D$ is obtained
from the following idea: assume $d \in D$. Since $S_d$ returns only total hypotheses
for the $m+1$ functions in $R_d$, there are at most $m + 1$ functions (but perhaps
more programs), which $S_d$ may guess during the learning process for some
$f \in R_d$. So let $T_d$ simulate $S_d$. In order to avoid superfluous mind changes,
$T_d$ will only change its hypothesis, if its old guess is no longer consistent and
the current guess of $S_d$ is consistent. Consistency tests are possible, because
all intermediate hypotheses returned by $S_d$ on any $f \in R_d$ correspond to total
functions.

Formally: for any recursive function $f$ and any description $d$ let $T_d(f[0]) := \psi$,
if $\varphi^d_{S_d(f[0])} (0) \neq f(0)$, and let $T_d(f[0]) := S_d(f[0])$ otherwise. For $n \geq 1$ compute
$S_d(f[n])$ and $T_d(f[n - 1])$. If $S_d(f[n])$ is inconsistent for $f[n]$ with respect
to $\varphi^d$ or $T_d(f[n - 1])$ is consistent for $f[n]$ with respect to $\varphi^d$, then define
\[ T_d(f[n]) := T_d(f[n - 1]). \text{ Otherwise let } T_d(f[n]) := S_d(f[n]). \] Now it is easy to show, that \( T \) learns \( D \) according to the model \( res \Uni Ex_m. \) \( \square \)

Theorem 24 summarizes the remaining cases, for which the hierarchy of uniform learning power looks similar to the hierarchy in the non-uniform model. As the structure of the corresponding proofs is close to the verification of Theorems 20 and 22, just the specific parts concerning the constructions of the required fixed point values for the separating description sets are outlined.

**Theorem 24** Let \( I, I' \) be inference classes in \( \mathbb{I} \), such that \( I' \setminus I \neq \emptyset \). Moreover assume \( (I, I') \neq (Ex, Total) \) for any \( m \geq 0 \).

1. There exists a description set \( D \in \text{res Uni} I' \setminus \text{Uni} I \), such that each recursive core described by \( D \) is a singleton set.

2. If \( (I, I') \neq (B_c, Bc^*) \) and \( I \notin \{Cx, Total\} \), then there exists a description set \( D \in \text{res Uni} I' \setminus \text{ext Uni} I \), such that each recursive core described by \( D \) consists of at most 2 functions.

**Sketch of Proof.** Any of these claims can be verified by a fixed point construction as in the proofs of Theorem 20 and Theorem 22. The main difference in the various proofs consists of the specific ideas used to construct the numberings \( \psi \). Fix some acceptable numbering \( \tau \) for the proof of the first part.

ad 1.

- \((I, I') = (Ex, Bc)\). For this pair of learning classes see Theorem 22.

- \((I, I') = (Bc, Bc^*)\). Again for any recursive learner \( S \) and any number \( d \) a function \( \psi \) is defined by stages. In stage 0, let \( \psi_0(0) := 0 \), let \( n_1 := 0 \) and go to stage 1. In each stage \( k \) \( (k \geq 1) \), let \( \psi_k[n_k] := \psi_0[n_k] \). Then \( \psi_k \) is extended by a sequence of \( 0 \)'s, until a number \( x \) is found, such that \( \tau_{S_d(\psi_0[n_k]0^x)}(n_k + x + 1) = 0 \).

If such an \( x \) does not exist (not Case A), this yields \( \psi_k = \psi_0[n_k]0^\infty \) and stage \( k \) does not terminate. Otherwise (Case A) let \( n_{k+1} := n_k + x + 1 \) and \( \psi_0[n_{k+1}] := \psi_0[n_k]0^{\tau_{S_d(\psi_0[n_k]0^x)}(n_k + x + 1)} \). The first value of \( \psi_k \) which has not yet been defined, will remain undefined (to exclude \( \psi_k \) from the recursive core constructed). All further values of \( \psi_k \) will be defined as the corresponding values of \( \psi_0 \) in the following stages (such that \( \psi_k =^* \psi_0 \)); go to stage \( k + 1 \).

If in any stage \( k \) \( (k \geq 1) \) Case A is not fulfilled, then \( \psi_k = \psi_0[n_k]0^\infty \) is the only recursive function enumerated by \( \psi \). In this case the output of the learner \( S_d \) on any segment \( \psi_0[n_k]0^x \) does not correspond to a \( \tau \)-program for \( \psi_k \), because \( \tau_{S_d(\psi_0[n_k]0^x)}(n_k + x + 1) \) is not equal to \( 0 = \psi_k(n_k + x + 1) \). If Case A occurs in all stages, then \( \psi_0 \) is the only recursive function enumerated by \( \psi \), but, for infinitely many initial segments of \( \psi_0 \), \( S_d \) returns \( \tau \)-programs of functions different from \( \psi_0 \): \( \psi_0(n_{k+1}) = 1 \neq 0 = \tau_{S_d(\psi_0[n_{k+1}]0^x)}(n_{k+1}) \) for all \( k \geq 1 \) (note that \( n_{k+1} > n_k \)). Hence \( S_d \) is not suitable for \( Bc \)-identification of the recursive

30
core of the numbering \( \psi \) with respect to \( \tau \).

Defining \( D \) by analogy with the proof of Theorem 22 yields a description set belonging to \( \text{res UniBc}^*[\bar{s}] \setminus \text{UniBc}[\bar{s}] \). Moreover all recursive cores described by \( D \) will be singleton sets.

- \((I, I') \in \{(E_x, E_{x-1}), (E_x, Cex), (E_x, Cons)\}\) for arbitrary \( m \geq 0 \.

Here the description set \( D \) used in the proof of Theorem 23.1 is sufficient.

- \((I, I') \in \{(Conf, E_1), (Conf, Cex)\}\). Here all partial-recursive learners have to be considered in the construction of the numberings \( \psi \). If \( S \) and \( d \) are fixed, start the definition of \( \psi \) in stage 0 with \( n_1 = 0 \) and \( \psi_0(0) = 0 \); then go to stage 1. In each stage \( k, k \geq 1 \), proceed as follows.

Let \( \psi_0(n_k + 2) := 0 \) (this will allow \( Cex \)-learning) and let \( \psi_k[n_k + 1] := \psi_d[n_k]0 \).

Moreover extend \( \psi_k \) by a sequence of the value \( k + 1 \), until the computations of \( S_d(\psi_0[n_k]) \) and \( S_d(\psi_d[n_k]0) \) terminate. The value \( k + 1 \) will help the desired \( E_{x+1} \)-learner to identify \( \psi_k \), if necessary.

**Remark 1.** If \( S_d(\psi_0[n_k]) \) is undefined or \( S_d(\psi_0[n_k]0) \) is undefined (i.e. neither Case A nor Case B below occurs), then stage \( k \) does not terminate. This yields \( \psi_k = \psi_0[n_k]0(k + 1)^\infty \) as the only element in the recursive core of \( \psi \), but \( S_d \) does not identify \( \psi_k \).

**Case A.** \( S_d(\psi_0[n_k]) \) and \( S_d(\psi_0[n_k]0) \) are defined and \( S_d(\psi_0[n_k]) \neq S_d(\psi_0[n_k]0) \). Then let \( n_{k+1} := n_k + 2 \), \( \psi_0(n_k + 1) := 0 \); go to stage \( k + 1 \).

(* Note that in this case \( \psi_k \) remains initial and \( S_d \) changes its mind on the extension of \( \psi_0 \) constructed in stage \( k \). *)

**Case B.** \( S_d(\psi_0[n_k]) \) and \( S_d(\psi_0[n_k]0) \) are defined and equal.

In this case let \( \psi_0(n_k + 1) := 1 \) and extend \( \psi_0 \) with a sequence of zeros, until the computation of \( S_d(\psi_0[n_k]1) \) terminates.

**Remark 2.** If \( S_d(\psi_0[n_k]1) \) is undefined (i.e. neither Case B.1 nor Case B.2 below occurs), then stage \( k \) does not terminate. Hence the recursive core of \( \psi \) consists of the function \( \psi_0 = \psi_0[n_k]10^\infty \) only, but \( S_d \) does not identify \( \psi_0 \).

**Case B.1.** \( S_d(\psi_d[n_k]1) \) is defined within \( x \) steps of computation and differs from \( S_d(\psi_d[n_k]) \).

In this case let \( n_{k+1} := n_k + 1 + x \) and \( \psi_0[n_{k+1}] := \psi_0[n_k]10^x \); go to stage \( k + 1 \).

(* Note that \( \psi_k \) remains initial and \( S_d \) changes its mind on the extension of \( \psi_0 \) constructed in this case. *)

**Case B.2.** \( S_d(\psi_0[n_k]1) \) is defined within \( x \) steps of computation and equal to \( S_d(\psi_0[n_k]0) \) and \( S_d(\psi_0[n_k]) \).

Then extend \( \psi_0 \) with a further sequence of zeros, until the computation
of $\tau_{S_d(\psi_0[n_k])}(n_k + 1)$ stops or until some number $z$ is found, such that $S_d(\psi_0[n_k]1) \neq S_d(\psi_0[n_k]10^x)$.

Remark 3. If the extension in Case B.2 never stops (i.e. none of the cases B.2.1, B.2.2, B.2.3 below occur), then stage $k$ does not terminate. This yields $\psi_0 = \psi_0[n_k]10^\infty$ as the only element of the recursive core of $\psi$. As $\tau_{S_d(\psi_0[n_k])}(n_k + 1)$ is undefined, the hypothesis $S_d(\psi_0[n_k]) = S_d(\psi_0[n_k]1)$ is not a $\tau$-program for $\psi_0$. But the output of $S_d$ on $\psi_0$ converges to $S_d(\psi_0[n_k])$, i.e. $S_d$ does not identify $\psi_0$ with respect to $\tau$.

Case B.2.1. The extension in Case B.2 is stopped, because the computation of $\tau_{S_d(\psi_0[n_k])}(n_k + 1)$ stops within $y$ steps and the result is different from 1. Then let $\psi_0 := \psi_0[n_k]10^\infty$ be the only function in the recursive core. (* Now the hypothesis $S_d(\psi_0[n_k + 1])$ is not conform for $\psi_0[n_k + 1]$ with respect to $\tau$. Hence $S_d$ does not identify the function $\psi_0$ conformly with respect to $\tau$. *)

Case B.2.2. The extension in Case B.2 is stopped, because the computation of $\tau_{S_d(\psi_0[n_k])}(n_k + 1)$ stops within $y$ steps and the result equals 1. Then let $\psi_0$ remain initial and let $\psi_k := \psi_0[n_k]0(k + 1)^\infty$ be the only element in the recursive core of $\psi$. (* Now the hypothesis $S_d(\psi_k[n_k + 1]) = S_d(\psi_0[n_k]0) = S_d(\psi_0[n_k])$ is not conform for $\psi_k[n_k + 1]$ with respect to $\tau$. *)

Case B.2.3. The extension in Case B.2 is stopped, because some $z$ satisfying $S_d(\psi_0[n_k]1) \neq S_d(\psi_0[n_k]10^x)$ has been found within $y$ steps. Let $y'$ be the maximum of $z$ and $y$ and define $n_{k+1} := n_k + 1 + y'$. Moreover $\psi_0[n_{k+1}] := \psi_0[n_k]10^{y'}$. Go to stage $k + 1$. (* In this case $\psi_k$ remains initial and $S_d$ changes its mind on the extension of $\psi_0$ constructed in stage $k$. *)

End stage $k$

Now $S_d$ does not learn the recursive core of $\psi$ conformly with respect to $\tau$: if one of the cases A, B.1, B.2.3, occurs infinitely often, then the recursive core of $\psi$ consists of the function $\psi_0$ only, but $S_d$ changes its mind on $\psi_0$ infinitely often. If one of the cases B.2.1, B.2.2 is fulfilled once, then, by the notes above, the recursive core of $\psi$ is not Conf-learned by $S_d$ with respect to $\tau$ either. Otherwise, by the remarks 1, 2, and 3, the same fact is observed. Furthermore the core constructed is a singleton set in any case.

Defining $D$ as usual yields a description set, which belongs to $res\ UniEx_1[\*]$ as well as to $res\ UniCex[\*]$, but not to $UniConf[\*]$. Further details are omitted.
• \((I, I') = (\text{Cons}, \text{Conf})\). Again all partial-recursive learners have to be considered. For each strategy \(S\) and each number \(d\) construct a two-place function \(\psi\) by stages. In stage 0 let \(\psi_0(0) := 0\) and go to stage 1. In each stage \(k\) \((k \geq 1)\) proceed as follows.

Let \(\psi_{2k-1}[k+1] := \psi_0[k-1]0(k+1)\) and \(\psi_{2k}[k+1] := \psi_0[k-1]1(k+1)\) (the value \(k+1\) will help the uniform Conf-learner to identify the functions \(\psi_{2k-1}\) and \(\psi_{2k}\), if necessary). Then extend \(\psi_{2k-1}\) with a sequence of the value \(k+1\), until the computations of \(S_d(\psi_0[k-1])\) and \(S_d(\psi_0[k-1]0)\) terminate.

**Remark 1.** If \(S_d(\psi_0[k-1])\) or \(S_d(\psi_0[k-1]0)\) is undefined (i.e., neither Case A nor Case B below occurs), then stage \(k\) does not terminate. This yields \(\psi_{2k-1} = \psi_0[k-1]0(k+1)^\infty\) as the only element of the recursive core of \(\psi\), but \(S_d\) does not identify \(\psi_{2k-1}\).

**Case A.** \(S_d(\psi_0[k-1])\) and \(S_d(\psi_0[k-1]0)\) are defined and \(S_d(\psi_0[k-1]) \neq S_d(\psi_0[k-1]0)\).

In this case let \(\psi_0(k) := 0\); go to stage \(k + 1\). \(\psi_{2k-1}\) and \(\psi_{2k}\) remain initial.

(* Note that \(S_d\) changes its mind on the extension of \(\psi_0\) constructed in this case. *)

**Case B.** \(S_d(\psi_0[k-1])\) and \(S_d(\psi_0[k-1]0)\) are defined and equal.

Then extend \(\psi_{2k-1}\) with a sequence of the value \(k + 1\), until the computation of \(\tau_{S_d(\psi_0[k-1])(k)}\) stops with the result 0.

**Remark 2.** If \(\tau_{S_d(\psi_0[k-1])}(k)\) is undefined or differs from 0 (i.e., Case B.1 below does not occur), then stage \(k\) does not terminate. This yields \(\psi_{2k-1} = \psi_0[k-1]0(k+1)^\infty\) as the only element of the recursive core of \(\psi\), but the hypothesis \(S_d(\psi_{2k-1}[k]) = S_d(\psi_0[k-1])\) is not consistent for \(\psi_{2k-1}[k]\) with respect to \(\tau\).

**Case B.1.** \(\tau_{S_d(\psi_0[k-1])}(k) = 0\).

Then let \(\psi_{2k-1}\) remain initial and extend \(\psi_{2k}\) with a sequence of the value \(k + 1\), until the computation of \(S_d(\psi_0[k-1]1)\) terminates.

**Remark 3.** If \(S_d(\psi_0[k-1]1)\) is undefined (i.e. neither Case B.1.1 nor Case B.1.2 below occurs), then stage \(k\) does not terminate. Hence \(\psi_{2k} = \psi_0[k-1]1(k+1)^\infty\) is the only function in the recursive core of \(\psi\), but \(S_d\) does not identify \(\psi_{2k}\).

**Case B.1.1.** \(S_d(\psi_0[k-1]1) = S_d(\psi_0[k-1])\).

Let \(\psi_{2k} = \psi_0[k-1]1(k+1)^\infty\) be the only element of the recursive core of the numbering \(\psi\).

(* Here \(S_d(\psi_{2k}[k]) = S_d(\psi_0[k-1])\) is not consistent for \(\psi_{2k}[k]\) with respect to \(\tau\) (according to Case B.1). *)

33
Case B.1.2. $S_d(\psi_0[k-1]1)$ is defined and differs from $S_d(\psi_0[k-1])$. Then define $\psi_0(k) := 1$; go to stage $k + 1$. $\psi_k$ remains initial.

(* Note that $S_d$ changes its mind on the extension of $\psi_0$ constructed in this case. *)

End stage $k$

If in the construction of $\psi$ one of the cases A or B.1.2 occurs infinitely often, then the recursive core of $\psi$ equals $\{\psi_0\}$, but $S_d$ changes its mind on $\psi_0$ infinitely often. If Case B.1.1 is fulfilled once, then the recursive core of $\psi$ consists of one function, which is not Cons-learned by $S_d$ with respect to $\tau$. Otherwise the remarks 1, 2, 3 above imply the same fact. Hence in any case $S_d$ does not identify the recursive core of $\psi$ with consistent intermediate hypotheses with respect to $\tau$.

Defining $D$ as usual yields a description set, which belongs to $\text{resUniConf}[\ast]$, but not to $\text{UniCons}[\ast]$. Further details are omitted.

• $(I, I') \in \{(\text{Cex, Ex}_1), (\text{Cex, Cons})\}$. For any recursive learner $S$ and any number $d$ define a function $\psi$ by stages. In stage 0 let $\psi_0(0) := 0$, $n_1 := 0$ and go to stage 1. In each stage $k$ ($k \geq 1$) proceed in the following way.

Define $\psi_k[n_k] := \psi_0[n_k]$ and $h_k := S_d(\psi_0[n_k])$. Then extend $\psi_k$ with the value $k + 1$, until (i) or (ii) is found true.

(i) there is some $y_k > n_k$, such that $\tau_{h_k}(y_k)$ is defined,

(ii) there is some $y_k \leq n_k$, such that $\tau_{h_k}(y_k)$ is defined

and $\tau_{h_k}(y_k) \neq \psi_0(y_k)$.

The value $k + 1$ will help the desired $\text{Ex}_1$- and Cons-learners to identify $\psi_k$, if necessary.

Remark 1. If neither (i) nor (ii) is found true (i.e. neither Case A nor Case B below occurs), then stage $k$ does not terminate. Hence the recursive core of $\psi$ equals $\{\psi_k\} = \{\psi_0[n_k](k + 1)^{\infty}\}$, but $\tau_{S_d(\psi_0[n_k])} = \tau_{h_k} \subseteq \psi_0[n_k]^{\infty} \subset \psi_k$. As the hypothesis $h_k$ returned by $S_d$ on $\psi_k[n_k]$ is a $\tau$-program of a proper subfunction of $\psi_k$, the learner $S_d$ does not Cex-identify $\psi_k$ with respect to $\tau$.

Case A. The extension of $\psi_k$ is stopped, because (i) is found true.

Then let $n_{k+1} := y_k$ and $\psi_0[n_{k+1}] := \psi_0[n_k]00 \ldots \overline{0}\overline{\sigma}(\tau_{h_k}(y_k))$; go to stage $k + 1$.

(* Note that in this case $S_d(\psi_0[n_k]) (= h_k)$ is not a $\tau$-number of $\psi_0$. *)

Case B. The extension of $\psi_k$ is stopped, because (ii) is found true.

Then extend $\psi_0$ with 0’s, until $S_d(\psi_0[n_k]0^x) \neq h_k$ is fulfilled for an extension of $\psi_0$ with $0^x$ for some $x \geq 1$. 

34
Remark 2. If \( S_d(\psi_0[n_k]^0) = h_k \) for all \( x > 0 \) (i.e. Case B.1 below does not occur), then stage \( k \) does not terminate. This implies that the recursive core of \( \psi \) equals \( \{\psi_0\} = \{\psi_0[n_k]^0\} \), but the output sequence of \( S_d \) on \( \psi_0 \) converges to \( h_k \), which is incorrect for \( \psi_0 \) with respect to \( \tau \) (because of (ii)). Thus \( S_d \) does not identify \( \psi_0 \) with respect to \( \tau \).

Case B.1. \( S_d(\psi_0[n_k]^0) \neq h_k \) for some minimal \( x > 0 \).
In this case let \( n_{k+1} := n_k + x \) and \( \psi_0[n_{k+1}] := \psi_0[n_k]^0 \); go to stage \( k + 1 \).
(* Note that \( S_d \) changes its hypothesis on the extension of \( \psi_0 \) defined in this case. *)

End stage \( k \)

If Case A occurs infinitely often in the construction of \( \psi \), then the recursive core of \( \psi \) consists of the function \( \psi_0 \) only, but on \( \psi_0 \) the strategy \( S_d \) returns incorrect hypotheses for \( \psi_0 \) with respect to \( \tau \) infinitely often. If Case B.1 occurs infinitely often, then again the recursive core equals \( \{\psi_0\} \), but \( S_d \) makes infinitely many mind changes on \( \psi_0 \). Otherwise, by the remarks 1 and 2 above, \( S_d \) does not \( Cex \)-identify the only function in the recursive core of \( \psi \) with respect to \( \tau \). Altogether this proves that \( S_d \) is not suitable for \( Cex \)-learning of the recursive core constructed.

Defining \( D \) as usual yields a description set, which belongs to \( res\text{UniEx}_1[*] \cap res\text{UniCons}[*] \), but not to \( Uni\text{Cex}[*] \). Further details are omitted.

For the other pairs \((I, I')\) satisfying the required conditions the corresponding claim follows from Theorem 19 and those parts of the claim which have already been verified.

ad 2.

• \((I, I') = (Ex, Bc)\). See Theorem 20.

• \((I, I') \in \{(Ex_m, Ex_{m+1}), (Ex_m, Cons)\} \) for arbitrary \( m \geq 0 \). Here the description set used in the proof of Theorem 23.2 is sufficient.

• \((I, I') \in \{(Conf, Ex_1), (Conf, Cex)\} \). For these pairs also non-total strategies have to be considered. For each learner \( S \) and each number \( d \) define a numbering \( \psi \) by stages. In stage 0 let \( \psi_0(0) := 0 \) and \( \psi_0(2) := 0 \). Furthermore define \( n_1 := 0 \) and go to stage 1. In each stage \( k \) \( (k \geq 1) \) proceed as follows. Let \( \psi_{2k-1}[n_k + 2] := \psi_0[n_k]0(k + 1) \), \( \psi_{2k}[n_k + 2] := \psi_0[n_k]1(k + 1) \). The value \( (k + 1) \) will prevent the desired \( Cex \)-learner form returning programs of proper subfunctions in the relevant cases. Moreover it helps the desired \( Ex_1 \)-learner to identify \( \psi_{2k-1} \) and \( \psi_{2k} \), if necessary. Extend \( \psi_{2k-1} \) and \( \psi_{2k} \) by the value \( k + 1 \), until the computations of all the values \( S_d(\psi_0[n_k]), S_d(\psi_0[n_k]) \), and \( S_d(\psi_0[n_k]) \) terminate.
Remark 1. If one of the values $S_d(\psi_0[nk])$, $S_d(\psi_0[nk]0)$, $S_d(\psi_0[nk]1)$ is undefined (i.e. neither Case A nor Case B below occurs), then stage $k$ does not terminate. Hence the recursive core of the numbering $\psi$ is equal to the set $\{\psi_{2k-1}, \psi_{2k}\} = \\{\psi_0[nk]0(k+1)^\infty, \psi_0[nk]1(k+1)^\infty\}$, but at least one of the functions $\psi_{2k-1}$, $\psi_{2k}$ is not identified by $S_d$.

Case A. $S_d(\psi_0[nk])$, $S_d(\psi_0[nk]0)$, and $S_d(\psi_0[nk]1)$ are defined and there is some $t \in \{0, 1\}$ satisfying $S_d(\psi_0[nk]) \neq S_d(\psi_0[nk]t)$.
Then leave the functions $\psi_{2k-1}$ and $\psi_{2k}$ initial, let $n_{k+1} := n_k + 2$, $\psi_0[n_{k+1}] := \psi_0[nk]t0$, $\psi_0(n_{k+1} + 2) := 0$, and go to stage $k + 1$.
(* Note that in Case A the learner $S_d$ changes its mind on the extension of $\psi_0$ just defined. *)

Case B. $S_d(\psi_0[nk])$, $S_d(\psi_0[nk]0)$, and $S_d(\psi_0[nk]1)$ are defined and equal.
In this case extend $\psi_0$ by 0’s and $\psi_{2k}$ by the value $k + 1$, until some $x \geq 0$ is found, such that $S_d(\psi_0[nk]0^{x+2}) \neq S_d(\psi_0[nk])$ is fulfilled.

Case B.1. This extension stops within $y$ steps; $S_d(\psi_0[nk]0^{x+2}) \neq S_d(\psi_0[nk])$ for some $x \leq y$.
Then let $n_{k+1} := n_k + 2 + y$ and define $\psi_0[n_{k+1}] := \psi_0[nk]0^{y+2}$, $\psi_0(n_{k+1} + 2) := 0$; go to stage $k + 1$.
(* Note that in Case B.1 the learner $S_d$ changes its mind on the extension of $\psi_0$ just defined. *)

Remark 2. If $S_d(\psi_0[nk]0^{x+2}) = S_d(\psi_0[nk])$ for all $x \geq 0$ (i.e. Case B.1 does not occur), then stage $k$ does not terminate. Hence the recursive core of $\psi$ equals $\{\psi_0, \psi_{2k}\} = \{\psi_0[nk]0^\infty, \psi_0[nk]1(k+1)^\infty\}$. Now let $\eta$ be any adequate hypothesis space for the recursive core of $\psi$ and consider two cases.

(i) $\eta_{S_d(\psi_0[nk])}(n_k + 1)$ is defined.
Then $S_d(\psi_0[nk]) = S_d(\psi_0[n_k + 1]) = S_d(\psi_{2k}[n_k + 1])$ is non-conform for at least one of the segments $\psi_0[n_k + 1], \psi_{2k}[n_k + 1]$ with respect to $\eta$. Thus $S_d$ does not Conf-learn the recursive core of $\psi$ with respect to $\eta$.

(ii) $\eta_{S_d(\psi_0[nk])}(n_k + 1)$ is undefined.
In this case the index $S_d(\psi_0[nk])$ is incorrect for $\psi_0$ with respect to $\eta$, but according to the condition in Remark 2 the output sequence of $S_d$ on the function $\psi_0$ converges to $S_d(\psi_0[nk])$. Therefore $S_d$ does not identify $\psi_0$ with respect to $\eta$.

End stage $k$

If Case A or Case B.1 occur infinitely often in the construction of $\psi$, then the recursive core of $\psi$ equals $\{\psi_0\}$, but $S_d$ changes its mind on $\psi_0$ infinitely often. If in some stage $k$ both Case A and Case B.1 fail, then, by Remark 1,
the recursive core consists of the functions $\psi_{2k-1}$ and $\psi_{2k}$, but $S_d$ does not identify this set. Otherwise, Remark 2 above shows that $S_d$ does not Conf-learn the recursive core $\{\psi_0, \psi_{2k}\}$ of $\psi$ with respect to any hypothesis space $\eta$. Consequently, in any case, $S_d$ does not identify the recursive core of $\psi$ with conform intermediate hypotheses.

Defining $D$ by analogy with the proof of Theorem 20 yields a description set belonging to $\text{res Uni Ex}_1[*]$ and $\text{res Uni Cex}_1[*]$, but not to $\text{ext Uni Conf}_1[*]$. Details are left out.

- $(I, I') = (\text{Cons}, \text{Conf})$. Again all partial-recursive learners have to be considered. For any strategy $S$ and any number $d$ construct a partial-recursive function $\psi$ in the following way. In stage 0 let $\psi_0(0) := 0$ and go to stage 1. In each stage $k$ ($k \geq 1$) proceed according to the following instructions.

Let $\psi_{2k-1}[k +1 ] := \psi_0[k -1]0(k+1)$, $\psi_{2k}[k +1 ] := \psi_0[k -1]1(k+1)$ and extend the functions $\psi_{2k-1}$ and $\psi_{2k}$ by the value $k+1$, until the computations of all the values $S_d(\psi_0[k -1])$, $S_d(\psi_0[k -1]0)$, and $S_d(\psi_0[k -1]1)$ terminate. The value $k+1$ will help the desired Conf-learner to identify $\psi_{2k-1}$ and $\psi_{2k}$, if necessary.

**Remark 1.** If one of the values $S_d(\psi_0[k -1])$, $S_d(\psi_0[k -1]0)$, $S_d(\psi_0[k -1]1)$ is undefined (i.e. neither Case A nor Case B below occurs), then stage $k$ does not terminate. This yields $\{\psi_{2k-1}, \psi_{2k}\} = \{\psi_0[k -1]0(k+1)\infty, \psi_0[k -1]1(k+1)\infty\}$ as the recursive core of $\psi$, but at least one of the functions $\psi_{2k-1}$, $\psi_{2k}$ is not identified by $S_d$.

**Case A.** $S_d(\psi_0[k -1])$, $S_d(\psi_0[k -1]0)$, and $S_d(\psi_0[k -1]1)$ are defined and equal. In this case let $\psi_{2k-1} := \psi_0[k -1]0(k+1)\infty$, $\psi_{2k} = \psi_0[k -1]1(k+1)\infty$ and leave all other functions enumerated by $\psi$ initial.

(* Since $S_d$ returns the same output for the segments $\psi_0[k -1]0$ and $\psi_0[k -1]1$, this hypothesis must be inconsistent (with respect to any hypothesis space) for at least one of the segments $\psi_{2k-1}[k]$, $\psi_{2k}[k]$. Hence $S_d$ is not an appropriate Cons-strategy for the recursive core of $\psi$. *)

**Case B.** $S_d(\psi_0[k -1])$, $S_d(\psi_0[k -1]0)$, and $S_d(\psi_0[k -1]1)$ are defined and there is some $t \in \{0, 1\}$ satisfying $S_d(\psi_0[k -1]) \neq S_d(\psi_0[k -1]t)$. Then let the functions $\psi_{2k-1}$ and $\psi_{2k}$ remain initial, let $\psi_0(k) := t$ and go to stage $k +1$.

(* Note that in Case B the learner $S_d$ changes its mind on the extension of $\psi_0$ just defined. *)

*End stage $k$*

If in the construction of $\psi$ Case B occurs infinitely often, then the recursive core of $\psi$ equals $\{\psi_0\}$, but $S_d$ changes its mind on $\psi_0$ infinitely often. If Case A is once fulfilled, then the note above implies that $S_d$ is not suitable for Cons-identification of the recursive core of $\psi$ (which equals $\{\psi_{2k-1}, \psi_{2k}\}$ for some $k \geq$
1). Otherwise, by Remark 1 above, the same fact is observed. Consequently, the recursive core of \( \psi \) is not \( \text{Cons} \)-learned by \( S_d \) with respect to any hypothesis space.

Defining \( D \) as usual yields a description set, which belongs to \( \text{res UniConf}[^*] \), but not to \( \text{ext UniCons}[^*] \). Further details are omitted.

- For the other pairs \( (I, I') \) satisfying the required conditions the corresponding claim follows from Theorem 19 and those parts of the claim which have already been verified. \( \square \)

This yields the following strict version of Theorem 19.

**Corollary 25** Let \( I, I' \in \mathbb{I} \) be inference classes, such that \( I \subset I' \).

1. \( \text{Uni}I[^*] \subset \text{Uni}I'[^*] \).
2. If \( (I, I') \neq (\text{Ex}_0, \text{Total}) \), then \( \text{res Uni}I[^*] \subset \text{res Uni}I'[^*] \).
3. If \( (I, I') \neq (Bc, \text{Be}^*) \), \( I \notin \{\text{Cex, Total}\} \), then \( \text{ext Uni}I[^*] \subset \text{ext Uni}I'[^*] \).

Moreover the following incomparability results are obtained from Theorem 23 and Theorem 24.

**Corollary 26**

1. \( \text{UniCex}[^*] \# \text{UniCons}[^*] \) and \( \text{UniCex}[^*] \# \text{UniConf}[^*] \) (analogously for \( \text{res Uni} \) instead of \( \text{Uni} \)).
2. \( \text{UniEx}_m[^*] \# \text{Uni}I[^*] \) for all \( I \in \{\text{Total, Cex, Cons, Conf}\} \) and all \( m \geq 1 \) (analogously for \( \text{res Uni} \) instead of \( \text{Uni} \)).
3. \( \text{ext UniEx}_m[^*] \# \text{ext UniCons}[^*] \) and \( \text{ext UniEx}_m[^*] \# \text{ext UniConf}[^*] \) for all \( m \geq 1 \).

### 4.2 Discrepancies between the hierarchies

With the preceding theorems all parts of the hierarchies for uniform learning, which agree with the corresponding parts of the hierarchy for the elementary learning model, have been verified. It remains to consider those cases, in which a change in the hierarchy has been claimed in Figure 2:

- \( \text{res UniEx}_0[^*] \# \text{res Uni Total}[^*], \)
- \( \text{ext UniBc}[^*] = \text{ext UniBc}[^*], \)
- \( \text{ext UniEx}[^*] = \text{ext UniCex}[^*] = \text{ext UniTotal}[^*] \).

The first of these claims is a consequence of Theorem 27, which furthermore states that the required separation is obtained with singleton recursive cores.

**Theorem 27** There exists a description set \( D \in \text{res UniEx}_0 \setminus \text{res Uni Total} \), such that each recursive core described by \( D \) is a singleton set.
Proof. The structure of the proof results from ideas similar to those used in the proof of Theorem 22. For any learner $S$ and any number $d$ a partial-recursive numbering $\psi$ is constructed as follows.

Let $\psi_i := \uparrow^\infty$ for all $i \geq 2$. Start computing $S_d(0)$. For each $x$, if the computation of $S_d(0)$ takes more than $x$ steps, let $\psi_0(x) := 0$.

**Case A.** $S_d(0)$ is defined.
If $S_d(0) \neq 0$, let $\psi_0 := 0^\infty$ and $\psi_1 := \uparrow^\infty$. Otherwise, leave $\psi_0 := 0^x \uparrow^\infty$ for some $x$, and $\psi_1 := 01^\infty$.

**Remark.** If $S_d(0)$ is undefined (i.e. if Case A does not occur), then $\psi_0$ equals $0^\infty$ and $\psi_1$ equals $\uparrow^\infty$.

As this construction is uniformly effective in $S$ and $d$, there is some number $d^*$, such that $\varphi^{d^*}$ equals the numbering $\psi$ constructed from $S$ and $d^*$. Such a number $d^*$ is called a fixed point associated to $S$. Now let

$$D := \{d \mid d \text{ is a fixed point associated to some partial-recursive function } S\}.$$ 

Note that, for any $d \in D$,

$$\text{either } R_d = \{\varphi_0^d\} = \{0^\infty\} \text{ or } R_d = \{\varphi_1^d\} = \{01^\infty\}. \quad (6)$$

It remains to verify the following claim.

**Claim.**

1. Each recursive core described by $D$ is a singleton set,
2. $D \in res\text{UniEx}_0$,
3. $D \notin res\text{UniTotal}$.

**ad 1.** This follows immediately from (6).

**ad 2.** By (6), an $Ex_0$-learner for any class described by $D$ has to return “?” on any initial segment consisting of just one value. Furthermore, it suffices to return 0 on input of any segment $0^n$ ($n \geq 2$), and to return 1, otherwise. Clearly this verifies $D \in res\text{UniEx}_0$.

**ad 3.** Assume to the contrary, that $D \in res\text{UniTotal}$. Then there exists some strategy $S$, such that each recursive core $R_d$ described by $D$ is identified by $S_d$ with total intermediate hypotheses with respect to $\varphi^d$. Now let $d^* \in D$ be a fixed point associated to $S$, so by assumption $R_{d^*}$ is Total-learned by $S_{d^*}$ with respect to $\varphi^{d^*}$. In the construction of the numbering $\psi$ corresponding to $d^*$ either Case A is fulfilled or not.
If Case A occurs, then $\psi_{S_{d^*}}^{d^*}(0)$ equals $\infty$ or $0^k \uparrow \infty$ for some $k$. So, for the only function $f \in R_{d^*}$, the hypothesis $S_{d^*}(f[0])$ is a $\varphi^{\alpha^*}$-number of a non-total function. Consequently, $S_{d^*}$ does not Total-identify $R_{d^*}$ with respect to $\varphi^{\alpha^*}$. This yields a contradiction.

If Case A does not occur, then, by the remark above, $\psi_{d^*}^{d^*} = 0^\infty$ and $S_{d^*}(\varphi^{\alpha^*}(0))$ is undefined. Clearly this implies, that $R_{d^*}$ is not learned by $S_{d^*}$, which again leads to a contradiction.

Thus the assumption $D \in res \ Uni \ Total$ is wrong. □

**Corollary 28** res $Uni \ Ex_0[*] \neq res \ Uni \ Total[*].$

The scope of the next two theorems is to verify the remaining claims concerning extended uniform learning. The proof of $ext \ Uni \ Be[*] = ext \ Uni \ Be^*[*]$ is based on the fact that the set of all descriptions of $Be$-classes is uniformly $Be$-learnable in the extended model.

**Theorem 29** $ext \ Uni \ Be[*] = ext \ Uni \ Be^*[*].$

*Proof.* It is possible to show even more:

$$ext \ Uni \ Be = \{D \subseteq N \mid R_d \in Be \text{ for each } d \in D\}.$$  \hfill (7)

As each finite class belongs to $Be$ and $ext \ Uni \ Be^*[*] = \{D \subseteq N \mid R_d \text{ is finite}\}$ (cf. Proposition 13), this implies the claim of Theorem 29. Therefore it remains to prove (7). Note that by definition $ext \ Uni \ Be \subseteq \{D \subseteq N \mid R_d \in Be \text{ for each } d \in D\}$. For the opposite inclusion, fix some description set $D$, such that each recursive core described by $D$ belongs to $Be$. The aim is to verify $D \in ext \ Uni \ Be$.

For that purpose, let $\tau$ be any acceptable numbering. By Lemma 7, there exists a class $\{T[d] \mid d \in D\}$ of recursive strategies, such that, for any $d \in D$, the strategy $T[d]$ $Be$-identifies $R_d$ with respect to $\tau$. Now define a class $\{\psi[d] \mid d \in D\}$ of hypothesis spaces by

$$\psi_i^{[d]} := \tau_{T[d]}(i) \text{ for all } d \in D \text{ and all } i.$$  

Moreover let a learner $S$ be given by $S_d(f[n]) := f[n]$ for all recursive functions $f$ and all $d, n$. As can be verified easily, $S$ is appropriate for uniform $Be$-identification of $D$ in the extended model with respect to the hypothesis spaces $\psi[d]$ ($d \in D$). Consequently, $D \in ext \ Uni \ Be$. □

Finally the proof of Figure 2 is completed by showing that for extended uniform learning of finite recursive cores the criteria $Ex$, $Cex$, and $Total$ coincide. In particular the inference types resulting from special properties concerning
the quality of the intermediate hypotheses (independent of the amount of information known about the target function) yield an exception in the separations – compared to the non-uniform model.

**Theorem 30** \( \text{ext UniEx}[\ast] = \text{ext UniCex}[\ast] = \text{ext UniTotal}[\ast] \).

**Proof.** Since \( \text{ext UniTotal}[\ast] \subseteq \text{ext UniCex}[\ast] \subseteq \text{ext UniEx}[\ast] \) by definition, it remains to prove \( \text{ext UniEx}[\ast] \subseteq \text{ext UniTotal} \). For that purpose choose a description set \( D \in \text{ext UniEx}[\ast] \). Then

1. each recursive core described by \( D \) is finite,
2. there is a strategy \( S \), such that for any \( d \in D \) the recursive core \( R_d \) is \( Ex \)-identified by \( S_d \) with respect to some hypothesis space \( \psi[d] \).

Note that the hypothesis spaces \( \psi[d] \) do not have to be computable uniformly in \( d \). In order to prove that \( D \) belongs to \( \text{ext UniTotal} \) the given strategy \( S \) is used as an appropriate learner. This requires a change of the hypothesis spaces \( \psi[d] \) for the descriptions \( d \) in the set \( D \).

The idea can be explained as follows: fix some description \( d \in D \). Since \( S_d \) identifies the finite class \( R_d \) in the limit, there are only finitely many initial segments of functions in \( R_d \), which force the strategy \( S_d \) to guess a non-total function. If the functions in \( \psi[d] \) associated with these inadequate guesses are replaced by some total function, a suitable hypothesis space for \( \text{Total-identification} \) of \( R_d \) by \( S_d \) is obtained.

More formally: Let \( d \) be an element of \( D \). For all functions \( f \in R_d \) statement (2) above implies that the set \( \{ n \geq 0 \mid \psi_{S_d(f[n])}^{[d]} \text{ is not total} \} \) is finite. Define the set of “forbidden” hypotheses on “relevant” initial segments by

\[
H^{[d]} := \{ i \geq 0 \mid \psi_i^{[d]} \text{ is not total and there is some function } f \in R_d \\
\text{ and some number } n \geq 0 \text{ such that } S_d(f[n]) = i \} .
\]

By (1) and (2) the set \( H^{[d]} \) is finite. Consider a new hypothesis space \( \eta^{[d]} \):

\[
\eta_i^{[d]} := \begin{cases} 
\psi_i^{[d]}, & \text{if } i \notin H^{[d]} \\
0^{\infty}, & \text{if } i \in H^{[d]} \end{cases} \text{ for all } i \in \mathbb{N} .
\]

Since \( H^{[d]} \) is finite, \( \eta^{[d]} \) is computable. Then \( R_d \) is \( \text{Total-identified} \) by \( S_d \) with respect to \( \eta^{[d]} \). As \( d \in D \) was chosen arbitrarily, this yields \( D \in \text{ext UniTotal} \). \( \square \)
Gold’s [8] model for identification of recursive functions in the limit has been investigated on a meta-level. As in the elementary model, several inference classes resulting from modifications of the constraints in Gold’s model have been studied, particularly concerning the comparison of the corresponding identification power. The hierarchy known from the elementary model has been manifested using finite classes of recursive functions for separating each pair of different inference classes. As finite classes are not appropriate for separations in the elementary model, this is evidence to the immense influence of the specific descriptions chosen to represent the target classes to the learner. Moreover – by analysing three variants of the uniform learning model – the impact of suitable hypothesis spaces is revealed. It turns out that the known hierarchy of inference classes witnesses to some kind of universal relationship. In particular, for each inference class considered there must be characteristic structures arranging the learnable classes and the adequate hypothesis spaces.

Acknowledgements

The author is grateful for the helpful hints of the anonymous referees. Additionally, many thanks are due to Steffen Lange for his comments improving the overall presentation.

References


