Intrinsic Complexity of Uniform Learning

Sandra Zilles

Universität Kaiserslautern,
FB Informatik, Postfach 3049, 67653 Kaiserslautern, Germany,
zilles@informatik.uni-kl.de

Abstract. Inductive inference is concerned with algorithmic learning of recursive functions. In the model of *learning in the limit* a learner successful for a class of recursive functions must eventually find a program for any function in the class from a gradually growing sequence of its values. This approach is generalized in *uniform learning*, where the problem of *synthesizing* a successful learner for a class of functions from a description of this class is considered.
A common reduction-based approach for comparing the complexity of learning problems in inductive inference is *intrinsic complexity*. In this context, reducibility between two classes is expressed via recursive operators transforming target functions in one direction and sequences of corresponding hypotheses in the other direction.
The present paper is the first one concerned with intrinsic complexity of uniform learning. The relevant notions are adapted and illustrated by several examples. Characterizations of complete classes finally allow for various insightful conclusions. The connection to intrinsic complexity of non-uniform learning is revealed within several analogies concerning firstly the role and structure of complete classes and secondly the general interpretation of the notion of intrinsic complexity.

1 Introduction

Inductive inference is concerned with algorithmic learning of recursive functions. In the model of learning in the limit, cf. [7], a learner successful for a class of recursive functions must eventually find a correct program for any function in the class from a gradually growing sequence of its values. The learner is understood as a machine – called inductive inference machine or IIM – reading finite sequences of input-output pairs of a target function, and returning programs as its hypotheses, see also [2]. The underlying programming system is then called a hypothesis space.

Studying the potential of such IIMs in general leads to the question whether – given a description of a class of functions – a corresponding successful IIM can be *synthesized* computationally from this description. This idea is generalized in the notion of *uniform learning*: we consider a collection $C_0, C_1, \ldots$ of learning problems – which may be seen as a decomposition of a class $C = C_0 \cup C_1 \cup \ldots$ – and ask for some kind of meta-IIM tackling the whole collection of learning problems. As an input, such a meta-IIM gets a description of one of the learning problems C_i (in our context a class C_i of recursive functions) in the collection. The meta-IIM is then supposed to develop a successful IIM for C_i. Besides studies on uniform learning of classes of recursive functions, cf. [12, 16], this topic has also been investigated in the context of learning formal languages, see in particular [1, 13, 14].

Since we consider IIMs as tackling a given problem, namely the problem of identifying all elements in a particular class of recursive functions, the complexity of such IIMs might express, how hard a learning problem is. For instance, the class of all constant functions allows for a simple and straightforward identification method; for other classes successful methods might seem more complicated. But this does not involve any rule allowing us to compare two learning problems with respect to their difficulty. So a formal approach for comparing the complexity of learning problems (i. e. of classes of recursive functions) is desirable.

Different aspects have been analysed in this context. One approach is, e. g., mind change complexity measured by the maximal number of hypothesis changes a machine needs to identify a function in the given class, see [3]. But since in general this number of mind changes is unbounded, other notions of complexity might be of interest.

Various subjects in theoretical computer science deal with comparing the complexity of decision problems, e. g. regarding decidability as such, see [15], or the possible efficiency of decision algorithms, see [5]. In general Problem A is at most as hard as Problem B, if A is *reducible* to B under a given reduction. Each such reduction involves a notion of complete (hardest solvable) problems. Besides studies concerning language learning, see [9–11], in [4] an approach for reductions in the context of learning recursive functions is introduced. This subject, *intrinsic complexity*, has been further analysed in [8] with a focus on complete classes. It has turned out that, for learning in the limit, a class is complete, iff it contains a dense r. e. subclass. Here the aspect of high topological complexity (density), contrasts with the aspect of low algorithmic complexity of r. e. sets, which is somehow striking and has caused discussions on whether this particular approach of intrinsic complexity is adequate.

The present paper deals with intrinsic complexity in the context of uniform learning. Assume some new reduction expresses such an idea of intrinsic complexity. If a class C of functions is complete in the initial sense, natural questions are *(i)* whether C can be decomposed into a uniformly learnable collection $C_0, C_1, \ldots$, which is *not* a hardest problem in uniform learning, and *(ii)* whether there are also inappropriate decompositions of C, i. e. collections of highest complexity in uniform learning.

Below a notion of intrinsic complexity for uniform learning is developed and the corresponding complete classes are characterized. The obtained structure of degrees of complexity matches recent results on uniform learning: it has been shown that even decompositions into singleton classes can yield problems too hard for uniform learning in Gold's model. This suggests that collections representing singleton classes may sometimes form hardest problems in uniform learning. Indeed, the notion developed below expresses this intuition, i. e. collections of singleton sets may constitute complete classes in uniform learning. Still, the characterization of completeness here reveals a weakness of the general idea of intrinsic complexity, namely – as in the non-uniform case – complete classes have a low algorithmic complexity (see Theorem 7). All in all, this shows that intrinsic complexity, as in [4], is on the one hand a useful approach, because it can be adapted to match the intuitively desired results in uniform learning. On the other hand, the doubts in [8] are corroborated.

2 Preliminaries

2.1 Notations

Knowledge of basic notions used in mathematics and computability theory is assumed, cf. [15]. $\mathbb{N}$ is the set of natural numbers. The cardinality of a set X is denoted by $card X$. *Partial-recursive functions* always operate on natural numbers. If f is a function, $f(n)\uparrow$ indicates that $f(n)$ is undefined. Our target objects for learning will always be *recursive functions*, i. e. total partial-recursive functions. $\mathcal{R}$ denotes the set of all recursive functions.

If α is a finite tuple of numbers, then $|\alpha|$ denotes its length. Finite tuples are coded, i. e. if $f(0), \dots, f(n)$ are defined, a number $f[n]$ represents the tuple $(f(0), \dots, f(n))$, called an *initial segment* of f. $f[n]\uparrow$ means that $f(x)\uparrow$ for some $x \leq n$. For convenience, a function may be written as a sequence of values or as a set of input-output pairs. A sequence $\sigma = x_0, x_1, x_2, \dots$ *converges* to x, iff $x_n = x$ for all but finitely many n; we write $\lim(\sigma) = x$. For example let $f(n) = 7$ for $n \leq 2$, $f(n)\uparrow$ otherwise; $g(n) = 7$ for all n. Then $f = 7^3\uparrow^\infty = \{(0,7),(1,7),(2,7)\}$, $g = 7^\infty = \{(n,7) \mid n \in \mathbb{N}\}$; $\lim(g) = 7$, and $f \subseteq g$. For $n \in \mathbb{N}$, the notion $f =_n g$ means that for all $x \leq n$ either $f(n)\uparrow$ and $g(n)\uparrow$ or $f(n) = g(n)$. A set C of functions is *dense*, iff for any $f \in C$, $n \in \mathbb{N}$ there is some $g \in C$ satisfying $f =_n g$, but $f \neq g$.

Recursive functions – our target objects for learning – require appropriate representation schemes, to be used as hypothesis spaces. Partial-recursive enumerations serve for that purpose: any $(n+1)$-place partial-recursive function ψ *enumerates* the set $P_\psi := \{\psi_i \mid i \in \mathbb{N}\}$ of n-place partial-recursive functions, where $\psi_i(x) := \psi(i, x)$ for all $x = (x_1, \dots, x_n)$. Then ψ is called a *numbering*. Given $f \in P_\psi$, any index i satisfying $\psi_i = f$ is a *ψ-program* of f.

Following [6], we call a family $(d_i)_{i\in\mathbb{N}}$ of natural numbers *limiting r. e.*, iff there is a recursive numbering $\overline{d}$ such that $\lim(\overline{d_i}) = d_i$ for all $i \in \mathbb{N}$.

2.2 Learning in the limit and intrinsic complexity

Below, let τ be a fixed acceptable numbering, serving as a hypothesis space. The learner is a total computable device called IIM (inductive inference machine) working in steps. The input of an IIM M in step n is an initial segment $f[n]$ of some f; the output $M(f[n])$ is interpreted as a τ-program. In learning in the limit, M is successful for f, if the sequence $M(f) := (M(f[n]))_{n\in\mathbb{N}}$ of hypotheses is *admissible* for f:

Definition 1 [4] *Let $f, \sigma \in \mathcal{R}$. σ is admissible for f, iff σ converges and* $\lim(\sigma)$ *is a τ-program for f.*

Now a class of recursive functions is learnable in the limit (*Ex*-learnable; *Ex* is short for *explanatory*), if a single IIM is successful for all functions in the class.

Definition 2 [7, 2] *A class $C \subseteq \mathcal{R}$ is Ex-learnable ($C \in Ex$), iff there is an IIM M such that, for any $f \in C$, the sequence $M(f)$ is admissible for f. M is then called an Ex-learner or an IIM for C.*

The class of constant functions and the class $C_{fsup} = \{\alpha 0^\infty \mid \alpha$ is an initial segment$\}$ of recursive functions of finite support are in *Ex*, but intuitively, the latter is harder to learn. A reduction-based approach for comparing the learning complexity is proposed in [4], using the notion of recursive operators.

Definition 3 [15, 8] *Let Θ be a total function operating on functions. Θ is a recursive operator, iff for all functions f, g and all numbers $n, y \in \mathbb{N}$:*

1. *if $f \subseteq g$, then $\Theta(f) \subseteq \Theta(g)$;*
2. *if $\Theta(f)(n) = y$, then $\Theta(f')(n) = y$ for some initial segment $f' \subseteq f$;*
3. *if f is finite, then one can effectively (in f) enumerate $\Theta(f)$.*

Reducing a class C_1 of functions to a class C_2 of functions requires two operators: the first one maps C_1 into C_2; the second maps any admissible sequence for a mapped function in C_2 to an admissible sequence for the associated original function in C_1.

Definition 4 [4] *Let $C_1, C_2 \in Ex$. C_1 is Ex-reducible to C_2, iff there are recursive operators Θ, Ξ such that all functions $f \in C_1$ fulfil the following conditions:*

1. *$\Theta(f)$ belongs to C_2,*
2. *if σ is admissible for $\Theta(f)$, then $\Xi(\sigma)$ is admissible for f.*

Note, if C_1 is *Ex*-reducible to C_2, then an IIM for C_1 can be deduced from any IIM for C_2; e. g. by [4], each class in *Ex* is *Ex*-reducible to C_{fsup}. As usual, this reduction yields complete classes, i. e. learnable classes of highest complexity.

Definition 5 [4] *A class $C \in Ex$ is Ex-complete, iff each class $C' \in Ex$ is Ex-reducible to C.*

By the remark above, the class C_{fsup} is *Ex*-complete. Note that C_{fsup} is r. e. and dense – a relevant property for characterizing *Ex*-complete classes:

Theorem 1 [8] *A class $C \in Ex$ is Ex-complete iff it has an r. e. dense subset.*

Ex-complete classes have subsets, which are dense, i. e. topologically complex, but r. e., i. e. algorithmically non-complex. The latter is astonishing, since there are dense classes, which are *not* *Ex*-complete, cf. [8], so they do *not* contain r. e. dense subsets. These classes are algorithmically more complex than C_{fsup}, but belong to a lower degree of intrinsic complexity. R. e. subsets as in Theorem 1 are obtained by mapping r. e. *Ex*-complete classes – such as C_{fsup} – to C with the help of an operator Θ. So perhaps this approach of intrinsic complexity just makes a class complete, if it is a suitable 'target' for recursive operators. This may be considered as a weakness of the notion of intrinsic complexity.

2.3 Uniform learning in the limit

Uniform learning views the approach of *Ex*-learning on a meta-level; it is not only concerned with the existence of methods solving specific learning problems, but with the problem to synthesize such methods. So the focus is on *families* of learning problems (here families of classes of recursive functions). Given a representation or description of a class of recursive functions, the aim is to effectively determine an adequate learner, i. e. to compute a program for a successful IIM learning the class.

For a formal definition of uniform learning it is necessary to agree on a scheme for describing classes of recursive functions (i. e. describing learning problems). For that purpose we fix a three-place acceptable numbering φ. If $d \in \mathbb{N}$, the numbering φ^d is the function resulting from φ, if the first input is fixed by d. Then any number d corresponds to a two-place numbering φ^d enumerating the set P_{φ^d} of partial-recursive functions. Now it is conceivable to consider the subset of all total functions in P_{φ^d} as a learning problem which is uniquely determined by the number d. Thus each number d acts as a description of the set R_d, where

$$R_d := \{\varphi_i^d \mid i \in \mathbb{N} \text{ and } \varphi_i^d \text{ is recursive}\} = P_{\varphi^d} \cap \mathcal{R} \text{ for any } d \in \mathbb{N} \ .$$

R_d is called *recursive core* of the numbering φ^d. So any set $D = \{d_0, d_1, \ldots\}$ can be regarded as a set of descriptions, i. e. a collection of learning problems R_{d_0}, R_{d_1}, ... In this context, D is called a description set.

A *meta-IIM* M is an IIM with two inputs: *(i)* a description d of a recursive core R_d, and *(ii)* an initial segment $f[n]$ of some $f \in \mathcal{R}$. Then M_d is the IIM resulting from M, if the first input is fixed by d. A meta-IIM M can be seen as mapping descriptions d to IIMs M_d; it is a successful uniform learner for a set D, in case M_d learns R_d for all $d \in D$; i. e. given any description in D, M develops a suitable learner for the corresponding recursive core.

Definition 6 *Let $D \subseteq \mathbb{N}$. D is uniformly Ex-learnable ($D \in UEx$), iff there is a meta-IIM M such that, for any $d \in D$, the IIM M_d is an Ex-learner for R_d.*

As a numbering φ^d enumerates a superset of R_d, a meta-IIM might also use φ^d as a hypothesis space for R_d. This involves a new notion of admissible sequences.

Definition 7 *Let $d \in \mathbb{N}$, $f \in R_d$, $\sigma \in \mathcal{R}$. σ is r-admissible for d and f, iff σ converges and $\lim(\sigma)$ is a φ^d-program for f.*

This approach yields just a special (*restricted*) case of uniform *Ex*-learning, because φ^d-programs can be uniformly translated into τ-programs.

Definition 8 *Let $D \subseteq \mathbb{N}$. D is uniformly Ex-learnable restrictedly ($D \in rUEx$), iff there is a meta-IIM M such that, for any $d \in D$ and any function $f \in R_d$, the sequence $M_d(f)$ is r-admissible for d and f.*

By the following result, special sets describing only singleton recursive cores are not uniformly *Ex*-learnable (restrictedly). For Claim 2 cf. a proof in [16].

Theorem 2 *1.* [12, 16] $\{d \in \mathbb{N} \mid card R_d = 1\} \notin UEx$.
2. Fix $s \in \mathcal{R}$. *Then* $\{d \in \mathbb{N} \mid R_d = \{s\}\} \notin rUEx$.

It has turned out, that even *UEx*-learnable subsets of these description sets are not in *UEx* (or *rUEx*), if additional demands concerning the sequence of hypotheses are posed, see [17]. This suggests that description sets representing only singletons may form hardest problems in uniform learning; analogously description sets representing only a fixed singleton recursive core may form hardest problems in restricted uniform learning. Hopefully, this intuition can be expressed by a notion of intrinsic complexity of uniform learning.

3 Intrinsic complexity of uniform learning

3.1 Intrinsic complexity of *UEx*-learning

The crucial notion now concerns the reduction between description sets D_1 and D_2. As in the non-uniform model, a meta-IIM for D_1 should be computable from a meta-IIM for D_2, if D_1 is reducible to D_2. We first focus on *UEx*-learning; the restricted variant will be discussed later on. A first idea for *UEx*-reducibility might be to demand the existence of operators Θ and Ξ such that for $d_1 \in D_1$ and $f_1 \in R_{d_1}$

$$\Theta \text{ transforms } (d_1, f_1) \text{ into a pair } (d_2, f_2) \text{ with } d_2 \in D_2 \text{ and } f_2 \in R_{d_2}\,;$$

where Ξ maps any admissible sequence for f_2 to an admissible sequence for f_1.

Unfortunately, this does not allow us to reduce every set in *UEx* to a set describing only singleton recursive cores: suppose $R_d = C_{fsup}$. As the set $D_1 = \{d\}$ is uniformly *Ex*-learnable, it should be reducible to a set D_2 representing only singleton recursive cores, say via Θ and Ξ as above. Now for any initial segment α, there are $d_2 \in D_2$ and $f_2 \in R_{d_2}$ such that $\Theta(d, \alpha 0^\infty) = (d_2, f_2)$. The usual notion of an operator yields an $n > 0$ and a subfunction $\sigma \subseteq f_2$ such that $\Theta(d, \alpha 0^n) = (d_2, \sigma)$. As $card R_{d_2} = 1$, this implies $\Theta(d, \alpha 0^n \beta 0^\infty) = (d_2, f_2)$ for all initial segments β. In particular, there are $f, f' \in R_d$ such that $f \neq f'$, but $\Theta(d, f) = \Theta(d, f') = (d_2, f_2)$. By assumption, Ξ maps each admissible sequence for f_2 to a sequence admissible for both f *and* f'. The latter is of course impossible, so this approach does not meet our purpose.

The problem above is that the description d_2, once it is output by Θ on input of $(d_1, f_1[m])$, can never be changed depending on the values of f_1 to be read. Hence, Θ should be allowed to return a *sequence* of descriptions, when fed a pair (d_1, f_1). As an improved approach, it is conceivable to demand, that for $d_1 \in D_1$ and $f_1 \in R_{d_1}$

$$\Theta \text{ transforms } (d_1, f_1) \text{ into a pair } (\delta_2, f_2)\,.$$

Here δ_2 is a sequence converging to some $d_2 \in D_2$ with $f_2 \in R_{d_2}$. Moreover, Ξ maps any admissible sequence for f_2 to an admissible sequence for f_1.

Still this approach bears a problem. Intuitively, reducibility should be transitive. In general, such a transitivity is achieved by connecting the operators of a first reduction with the operators of a second reduction. The idea above cannot guarantee that: assume D_1 is reducible to D_2 via Θ_1 and Ξ_1; D_2 is reducible to D_3 via Θ_2 and Ξ_2. If Θ_1 maps (d_1, f_1) to (δ_2, f_2), then which description d in the sequence δ_2 should form an input (d, f_2) for Θ_2? It is in general impossible to detect the limit d_2 of the sequence δ_2, and any description $d \neq d_2$ might change the output of Θ_2.

So it is inevitable to let Θ operate on *sequences* of descriptions and on functions, i. e., Θ maps pairs (δ_1, f_1), where δ_1 is a sequence of descriptions, to pairs (δ_2, f_2).

Definition 9 *Let Θ be a total function operating on pairs of functions. Θ is a recursive meta-operator, iff the following properties hold for all functions δ, δ', f, f':*

1. *if $\delta \subseteq \delta'$, $f \subseteq f'$, as well as $\Theta(\delta, f) = (\gamma, g)$ and $\Theta(\delta', f') = (\gamma', g')$, then $\gamma \subseteq \gamma'$ and $g \subseteq g'$;*
2. *if $n, y \in \mathbb{N}$, $\Theta(\delta, f) = (\gamma, g)$, and $\gamma(n) = y$ (or $g(n) = y$, resp.), then there are initial segments $\delta_0 \subseteq \delta$ and $f_0 \subseteq f$ such that $(\gamma_0, g_0) = \Theta(\delta_0, f_0)$ fulfils $\gamma_0(n) = y$ ($g_0(n) = y$, resp.);*
3. *if δ, f are finite, $\Theta(\delta, f) = (\gamma, g)$, one can effectively (in δ, f) enumerate γ, g.*

This finally allows for the following definition of *UEx*-reducibility.

Definition 10 *Let $D_1, D_2 \in UEx$. Fix a recursive meta-operator Θ and a recursive operator Ξ. D_1 is UEx-reducible to D_2 via Θ and Ξ, iff for any $d_1 \in D_1$, any $f_1 \in R_{d_1}$, and any initial segment δ_1 there are functions δ_2 and f_2 satisfying:*

1. *$\Theta(\delta_1 d_1^\infty, f_1) = (\delta_2, f_2)$,*
2. *δ_2 converges to some description $d_2 \in D_2$ such that $f_2 \in R_{d_2}$,*
3. *if σ is admissible for f_2, then $\Xi(\sigma)$ is admissible for f_1.*

D_1 is UEx-reducible to D_2, iff D_1 is UEx-reducible to D_2 via some Θ' and Ξ'.

Note that this definition expresses intrinsic complexity in the sense that a meta-IIM for D_1 can be computed from a meta-IIM for D_2, if D_1 is *UEx*-reducible to D_2. Moreover, as has been demanded in advance, the resulting reducibility is transitive:

Lemma 3 *If D_1, D_2, D_3 are description sets such that D_1 is UEx-reducible to D_2 and D_2 is UEx-reducible to D_3, then D_1 is UEx-reducible to D_3.*

The notion of completeness can be adapted from the usual definitions.

Definition 11 *A description set $D \in UEx$ is UEx-complete, iff each description set $D' \in UEx$ is UEx-reducible to D.*

The question is, whether this notion of intrinsic complexity expresses the intuitions formulated in advance, e. g., that there are *UEx*-complete description sets representing only singleton recursive cores. Before answering this question consider an illustrative example.

This example states that there is a single description d of an *Ex*-complete set such that the description set $\{d\}$ is *UEx*-complete. On the one hand, this might be surprising, because a description set consisting of just one index representing an *Ex*-learnable class might be considered rather simple and thus not complete for uniform learning. But on the other hand, this result is not contrary to the intuition, that the hardest problems in non-uniform learning may remain hardest, when considered in the context of meta-learning. The reason is that the complexity is still of highest degree, if the corresponding class of recursive functions is not decomposed appropriately.

Example 4 *Let $d \in \mathbb{N}$ fulfil $R_d = C_{fsup}$. Then the set $\{d\}$ is UEx-complete.*

Proof. Obviously, $\{d\} \in UEx$. To show that each description set in *UEx* is *UEx*-reducible to $\{d\}$, fix $D_1 \in UEx$ and let M be a corresponding meta-IIM as in Definition 6. It remains to define a recursive meta-operator Θ and a recursive operator Ξ appropriately.

Given initial segments δ_1 and α, let Θ just modify the sequence of hypotheses returned by the meta-IIM M, if the first input parameter is gradually taken from the sequence δ_1 and the second input parameter is gradually taken from the sequence α. The modification is to increase each hypothesis by 1 and to change each repetition of hypotheses into a zero output. A formal definition is omitted.

Moreover, given an initial segment $\sigma = (s_0, \ldots, s_n)$, let $\Xi(\sigma)$ look for the maximal $m \leq n$ such that at least one of the values $\tau_{s_m}(x)$, $x \leq n$, is defined within n steps and greater than 0. In case m does not exist, $\Xi(\sigma) = \Xi(s_0, \ldots, s_{n-1})$. Otherwise, let $y \leq n$ be maximal such that $\tau_{s_m}(y)$ has already been computed and is greater than 0. Then $\Xi(\sigma) = \Xi(s_0, \ldots, s_{n-1})\tau_{s_m}(y) - 1$.

Now D_1 is *UEx*-reducible to $\{d\}$ via Θ, Ξ; details are omitted. □

That decompositions of *Ex*-complete classes may also be *not UEx*-complete, is shown in Section 3.3. Example 4 moreover serves for proving the completeness of other sets, if Lemma 5 – an immediate consequence of Lemma 3 – is applied.

Lemma 5 *Let $D_1, D_2 \in UEx$. If D_1 is UEx-complete and UEx-reducible to D_2, then D_2 is UEx-complete.*

Lemma 5 and Example 4 simplify the proofs of further examples, finally revealing that there are indeed *UEx*-complete description sets representing singleton recursive cores only.

Example 6 *1. Let $(\alpha_i)_{i\in\mathbb{N}}$ be an r. e. family of all initial segments. Let $g \in \mathcal{R}$ fulfil $\varphi_0^{g(i)} = \alpha_i 0^\infty$ and $\varphi_{x+1}^{g(i)} = \uparrow^\infty$ for $i, x \in \mathbb{N}$. Then the description set $\{g(i) \mid i \in \mathbb{N}\}$ is UEx-complete.*

2. Let $g \in \mathcal{R}$ fulfil $\varphi_0^{g(i)} = \tau_i$ and $\varphi_{x+1}^{g(i)} = \uparrow^\infty$ for $i, x \in \mathbb{N}$. Then the description set $\{g(i) \mid i \in \mathbb{N}\}$ is UEx-complete.

Proof. ad 1. Obviously, $\{g(i) \mid i \in \mathbb{N}\} \in UEx$. Now we reduce the *UEx*-complete set $\{d\}$ from Example 4 to $\{g(i) \mid i \in \mathbb{N}\}$. Lemma 5 then proves Assertion 1.

It is easy to define Θ such that, if α does not end with 0, then $\Theta(\delta_1, \alpha 0^\infty) = (\delta_2, \alpha 0^\infty)$, where δ_2 converges to some $g(i)$ with $\alpha_i = \alpha$. Let $\Xi(\sigma) = \sigma$ for all σ. Then $\{d\}$ is *UEx*-reducible to $\{g(i) \mid i \in \mathbb{N}\}$ via Θ and Ξ. Details are omitted.

ad 2. Fix an r. e. family $(\alpha_i)_{i \in \mathbb{N}}$ of all initial segments; fix $h \in \mathcal{R}$ with $\tau_{h(i)} = \alpha_i 0^\infty$ for all $i \in \mathbb{N}$. Then $\varphi_0^{g(h(i))} = \alpha_i 0^\infty$ and $\varphi_{x+1}^{g(h(i))} = \uparrow^\infty$ for $i, x \in \mathbb{N}$. As above, the set $\{g(h(i)) \mid i \in \mathbb{N}\}$ is *UEx*-complete; so is its superset $\{g(i) \mid i \in \mathbb{N}\}$. □

Just as the properties of C_{fsup} are characteristic for *Ex*-completeness, the properties of description sets representing decompositions of C_{fsup} are characteristic for *UEx*-completeness, as is stated in Theorem 7 and Corollary 8.

Theorem 7 *Let $D \in UEx$. D is UEx-complete, iff there are a recursive numbering ψ and a limiting r. e. family $(d_i)_{i \in \mathbb{N}}$ of descriptions in D such that:*

1. *ψ_i belongs to R_{d_i} for all $i \in \mathbb{N}$;*
2. *P_ψ is dense.*

Proof. Fix a description set D in *UEx*.

Necessity. Assume D is *UEx*-complete. Fix any one-one recursive numbering χ such that $P_\chi = C_{fsup}$. Moreover fix $g \in \mathcal{R}$ which, given any $i, x \in \mathbb{N}$, fulfils $\varphi_0^{g(i)} = \chi_i$ and $\varphi_x^{g(i)} = \uparrow^\infty$, if $x > 0$. Then the description set $\{g(i) \mid i \in \mathbb{N}\}$ is *UEx*-complete, as can be verified similarly to Example 6. Lemma 5 then implies that $\{g(i) \mid i \in \mathbb{N}\}$ is *UEx*-reducible to D, say via Θ and Ξ.

Fix a one-one r. e. family $(\alpha_i)_{i \in \mathbb{N}}$ of all finite tuples over $\mathbb{N}$. For $i \in \mathbb{N}$, i coding the pair (x, y), define $(\delta_i, \psi_i) := \Theta(\alpha_y g(x)^\infty, \chi_x)$. By definition, ψ is a recursive numbering and, for all $i \in \mathbb{N}$, the sequence δ_i converges to some $d_i \in D$ such that $\psi_i \in R_{d_i}$. Hence $(d_i)_{i \in \mathbb{N}}$ is a limiting r. e. family of descriptions in D.

It remains to verify Property 2.

For that purpose fix $i, n \in \mathbb{N}$. By definition, if i encodes (x, y), we obtain $\Theta(\alpha_y g(x)^\infty, \chi_x) = (\delta_i, \psi_i)$. The properties of Θ yield some $m \in \mathbb{N}$ such that $\Theta(\alpha_y g(x)^m, \chi_x[m]) = (\delta_i', \alpha')$ for some δ_i', α' with $\delta_i' \subseteq \delta_i$ and $\psi_i[n] \subseteq \alpha' \subseteq \psi_i$.

Because of the particular properties of χ, there is some $x' \in \mathbb{N}$, $x' \neq x$, such that $\chi_{x'} =_m \chi_x$, but $\chi_{x'} \neq \chi_x$. Moreover, there is some $y' \in \mathbb{N}$ such that $\alpha_{y'} = \alpha_y g(x)^m$. If j encodes (x', y'), this yields $\Theta(\alpha_y g(x)^m g(x')^\infty, \chi_{x'}) = (\delta_j, \psi_j)$, where $\alpha' \subseteq \psi_j$. In particular $\psi_j =_n \psi_i$.

Assume $\psi_i = \psi_j$. Suppose σ is any admissible sequence for ψ_i. Then σ is admissible for ψ_j. This implies that $\Xi(\sigma)$ is admissible for both χ_x and $\chi_{x'}$. As $\chi_x \neq \chi_{x'}$, this is impossible. So $\psi_i \neq \psi_j$.

Sufficiency. Assume D, ψ, and $(d_i)_{i \in \mathbb{N}}$ fulfil the conditions of Theorem 7. Let $\overline{d}$ denote a numbering associated to the limiting r. e. family $(d_i)_{i \in \mathbb{N}}$. The results in the context of non-uniform learning help to show that D is *UEx*-complete:

By assumption, P_ψ is a dense r. e. subset of $\mathcal{R}$. Theorem 1 then implies that P_ψ is *Ex*-complete, so C_{fsup} is *Ex*-reducible to P_ψ, say via Θ', Ξ'.

Using Θ' and Ξ' one can show that the *UEx*-complete set $\{d\}$ from Example 4 is *UEx*-reducible to D. This implies that D is *UEx*-complete, too. Note that $R_d = C_{fsup}$.

It remains to define a recursive meta-operator Θ and a recursive operator Ξ appropriately. If δ_1 and α_1 are finite tuples over $\mathbb{N}$, define $\Theta(\delta_1, \alpha_1)$ as follows.

Compute $\Theta'(\alpha_1) = \alpha_2$ and $n = |\alpha_2|$.
For all $x < n$, let i_x be minimal such that $\alpha_2[x] \subseteq \psi_{i_x}$.
Return $\Theta(\delta_1\alpha_1) = ((\overline{d_{i_0}}(0), \overline{d_{i_1}}(1), \ldots, \overline{d_{i_{n-1}}}(n-1)), \alpha_2)$ (if $n = 0$, then the first component of $\Theta(\delta_1\alpha_1)$ is the empty sequence).

Clearly, if $f_1 \in \mathcal{R}$, then $\Theta(\delta_1, f_1) = (\delta_2, \Theta'(f_1))$ for some sequence δ_2.

Moreover, let $\Xi := \Xi'$.

Finally, to verify that $\{d\}$ is *UEx*-reducible to D, fix a sequence δ_1 and a function $f_1 \in R_d$.

First, note that $f_2 = \Theta'(f_1) \in P_\psi$. Let i be the minimal ψ-program of $\Theta'(f_1) = f_2$. As $\psi \in \mathcal{R}$, for all $x \in \mathbb{N}$ the minimal i_x satisfying $f_2[x] \subseteq \psi_{i_x}$ can be computed. Additionally, $\lim(i_x)_{x\in\mathbb{N}} = i$. Note that $\overline{d_i}$ converges to d_i. Hence $\Theta(\delta_1, f_1) = (\delta_2, f_2)$, where $f_2 \in P_\psi$ and δ_2 converges to d_i, given $f_2 = \psi_i$. In particular, $f_2 \in R_{d_i}$.

Second, if σ is admissible for f_2, then $\Xi'(\sigma)$ is admissible for f_1.

So $\{d\}$ is *UEx*-reducible to D via Θ and Ξ, and thus D is *UEx*-complete. □

Corollary 8 *Let $D \in UEx$. D is UEx-complete, iff there are a recursive numbering ψ and a limiting r. e. family $(d_i)_{i\in\mathbb{N}}$ of descriptions in D such that:*

1. *ψ_i belongs to R_{d_i} for all $i \in \mathbb{N}$;*
2. *P_ψ is Ex-complete.*

Proof. Necessity. The assertion follows from Theorem 1 and Theorem 7.
Sufficiency. Let $D \in UEx$. Assume ψ and $(d_i)_{i\in\mathbb{N}}$ fulfil the conditions above. Let $\overline{d}$ be a recursive numbering corresponding to the limiting r. e. family $(d_i)_{i\in\mathbb{N}}$. By Property 2, P_ψ is *Ex*-complete; thus, by Theorem 1, there exists a dense r. e. subclass $C \subseteq P_\psi$. Let ψ' be a one-one, recursive numbering with $P_{\psi'} = C$, in particular $P_{\psi'}$ is dense. It remains to find a limiting r. e. family $(d'_i)_{i\in\mathbb{N}}$ of descriptions in D such that $\psi'_i \in R_{d'_i}$ for all $i \in \mathbb{N}$. For that purpose define a corresponding numbering $\overline{d'}$. Given $i, n \in \mathbb{N}$, define $\overline{d'_i}(n)$ as follows.

Let $j \in \mathbb{N}$ be minimal such that $\psi'_i =_n \psi_j$. (* Note that, for all but finitely many n, the index j will be the minimal ψ-program of ψ'_i. *)
Return $\overline{d'_i}(n) := \overline{d_j}(n)$. (* $\lim(\overline{d'_i}) = d_j$, for j minimal with $\psi'_i = \psi_j$. *)

Finally, let d'_i be given by the limit of the function $\overline{d'_i}$, in case a limit exists.

Fix $i \in \mathbb{N}$. Then there is a minimal j with $\psi'_i = \psi_j$. By definition, the limit d'_i of $\overline{d'_i}$ exists and $d'_i = d_j \in D$. Moreover, as $\psi_j \in R_{d_j}$, the function ψ'_i is in $R_{d'_i}$. As ψ' and $(d'_i)_{i\in\mathbb{N}}$ allow us to apply Theorem 7, the set D is *UEx*-complete. □

Thus certain decompositions of *Ex*-complete classes remain *UEx*-complete, and *UEx*-complete description sets always represent decompositions of supersets of *Ex*-complete classes. Example 9 illustrates how to apply the above characterizations of *UEx*-completeness. A similar short proof may be given for Example 6.

Example 9 *Fix a recursive numbering χ such that P_χ is dense. Let $g \in \mathcal{R}$ fulfil $\varphi_0^{g(i)} = \chi_i$ and $\varphi_{x+1}^{g(i)} =\uparrow^\infty$ for $i, x \in \mathbb{N}$. Then $\{g(i) \mid i \in \mathbb{N}\}$ is UEx-complete.*

Proof. $(g(i))_{i\in\mathbb{N}}$ is a (limiting) r. e. family such that $\chi_i \in R_{g(i)}$ for all $i \in \mathbb{N}$ and P_χ is *Ex*-complete. Corollary 8 implies that $\{g(i) \mid i \in \mathbb{N}\}$ is *UEx*-complete. □

3.2 Intrinsic complexity of *rUEx*-learning

Adapting the formalism of intrinsic complexity for restricted uniform learning, we have to be careful concerning the operator Ξ. In *UEx*-learning, the current description d has no effect on whether a sequence is admissible for a function or not. For restricted learning this is different. Therefore, to communicate the relevant information to Ξ, it is inevitable to include a description from D_2 in the input of Ξ. That means, Ξ should operate on pairs (δ_2, σ) rather than on sequences σ only. Since only the limit of the function output by Ξ is relevant for the reduction, this idea can be simplified. It suffices, if Ξ operates correctly on the inputs d_2 and σ, where d_2 is the limit of δ_2. Then an operator on the pair (δ_2, σ) is obtained from Ξ by returning the sequence $(\Xi(\delta_2(0)\sigma[0]), \Xi(\delta_2(1)\sigma[1]), \ldots)$. Its limit will equal the limit of $\Xi(d_2\sigma)$.

Definition 12 *Let $D_1, D_2 \in rUEx$. Fix a recursive meta-operator Θ and a recursive operator Ξ. D_1 is rUEx-reducible to D_2 via Θ and Ξ, iff for any $d_1 \in D_1$, any $f_1 \in R_{d_1}$, and any initial segment δ_1 there are functions δ_2 and f_2 satisfying:*

1. *$\Theta(\delta_1 d_1^\infty, f_1) = (\delta_2, f_2)$,*
2. *δ_2 converges to some description $d_2 \in D_2$ such that $f_2 \in R_{d_2}$,*
3. *if σ is r-admissible for d_2 and f_2, then $\Xi(d_2\sigma)$ is r-admissible for d_1 and f_1.*

D_1 is rUEx-reducible to D_2, iff D_1 is rUEx-reducible to D_2 via some Θ' and Ξ'.

Completeness is defined as usual. As in the *UEx*-case, *rUEx*-reducibility is transitive; so the *rUEx*-completeness of one set may help to verify the *rUEx*-completeness of others.

Lemma 10 *If D_1, D_2, D_3 are description sets such that D_1 is rUEx-reducible to D_2 and D_2 is rUEx-reducible to D_3, then D_1 is rUEx-reducible to D_3.*

Lemma 11 *Let $D_1, D_2 \in rUEx$. If D_1 is rUEx-complete and rUEx-reducible to D_2, then D_2 is rUEx-complete.*

Recall that, intuitively, sets describing just one singleton recursive core may be *rUEx*-complete. This is affirmed by Example 12, the proof of which is omitted.

Example 12 *Let $s, g \in \mathcal{R}$ such that $\varphi_i^{g(i)} = s$ and $\varphi_x^{g(i)} =\uparrow^\infty$, if $i, x \in \mathbb{N}$, $x \neq i$. Then $\{g(i) \mid i \in \mathbb{N}\}$ is rUEx-complete, but not UEx-complete.*

Example 12 helps to characterize *rUEx*-completeness. In particular, it shows that the demand 'P_ψ is dense' has to be dropped.

Theorem 13 *Let $D \in rUEx$. D is $rUEx$-complete, iff there are a recursive numbering ψ and a limiting r. e. family $(d_i)_{i\in\mathbb{N}}$ of descriptions in D such that:*

1. *ψ_i belongs to R_{d_i} for all $i \in \mathbb{N}$;*
2. *for each $i, n \in \mathbb{N}$ there are infinitely many $j \in \mathbb{N}$ satisfying $\psi_i =_n \psi_j$ and $(d_i, \psi_i) \neq (d_j, \psi_j)$.*

Proof. Fix a description set D in $rUEx$.

Necessity. Assume D is $rUEx$-complete. Lemma 11 implies that the description set $\{g(i) \mid i \in \mathbb{N}\}$ from Example 12 is $rUEx$-reducible to D, say via Θ and Ξ.

Fix a one-one r. e. family $(\alpha_i)_{i\in\mathbb{N}}$ of all finite tuples over $\mathbb{N}$. For $i \in \mathbb{N}$, i coding the pair (x, y), define $(\delta_i, \psi_i) := \Theta(\alpha_y g(x)^\infty, s)$. By definition, ψ is a recursive numbering and, for all $i \in \mathbb{N}$, the sequence δ_i converges to some $d_i \in D$ such that $\psi_i \in R_{d_i}$. Hence $(d_i)_{i\in\mathbb{N}}$ is a limiting r. e. family of descriptions in D.

It remains to verify Property 2.

For that purpose fix $i, n \in \mathbb{N}$. By definition, if i encodes (x, y), we have $\Theta(\alpha_y g(x)^\infty, s) = (\delta_i, \psi_i)$. The properties of Θ yield some $m \in \mathbb{N}$ such that $\Theta(\alpha_y g(x)^m, s) = (\delta'_i, \alpha')$ for some δ'_i and α' with $\delta'_i \subseteq \delta_i$ and $\psi_i[n] \subseteq \alpha' \subseteq \psi_i$.

Now choose any $x' \in \mathbb{N}$ such that $x' \neq x$. Moreover, there is some $y' \in \mathbb{N}$ such that $\alpha_{y'} = \alpha_y g(x)^m$. If j encodes (x', y'), this yields $\Theta(\alpha_y g(x)^m g(x')^\infty, s) = (\delta_j, \psi_j)$, where $\alpha' \subseteq \psi_j$. In particular $\psi_j =_n \psi_i$.

Assume $(d_i, \psi_i) = (d_j, \psi_j)$. Suppose σ is any $rUEx$-admissible sequence for d_i and ψ_i. Then $\Xi(d_i\sigma)$ is $rUEx$-admissible for both $g(x)$ and s and $g(x')$ and s. As x is the only $\varphi^{g(x)}$-number for s and x' is the only $\varphi^{g(x')}$-number for s, the latter is impossible. So $(d_i, \psi_i) \neq (d_j, \psi_j)$.

Repeating this argument for any x' with $x' \neq x$ yields the desired property.

Sufficiency. First note: if $(d_i)_{i\in\mathbb{N}}$ is a limiting r. e. family and ψ any recursive numbering, such that $\{(d_i, \psi_i) \mid i \in \mathbb{N}\}$ is an infinite set, then there are a limiting r. e. family $(d'_i)_{i\in\mathbb{N}}$ and a recursive numbering ψ', such that $\{(d'_i, \psi'_i) \mid i \in \mathbb{N}\} \subseteq \{(d_i, \psi_i) \mid i \in \mathbb{N}\}$ and $i \neq j$ implies $(d'_i, \psi'_i) \neq (d'_j, \psi'_j)$. Details are omitted.

So let D, ψ, $(d_i)_{i\in\mathbb{N}}$ fulfil the demands of Theorem 7 and assume wlog that $i \neq j$ implies $(d_i, \psi_i) \neq (d_j, \psi_j)$. Let $\overline{d}$ be the numbering associated to the limiting r. e. family $(d_i)_{i\in\mathbb{N}}$. We show that the set $\{g(i) \mid i \in \mathbb{N}\}$ from Example 12 is $rUEx$-reducible to D; so Lemma 11 implies that D is $rUEx$-complete.

For that purpose fix a one-one numbering $\eta \in \mathcal{R}$ such that P_η equals the set $C_{const} := \{\alpha i^\infty \mid \alpha$ is a finite tuple over $\mathbb{N}$ and $i \in \mathbb{N}\}$ of all recursive finite variants of constant functions.

Using a construction from [8] we define an operator Θ' mapping P_η into P_ψ. In parallel, a function θ is constructed to mark used indices. Let $\Theta'(\eta_0) := \psi_0$ and $\theta(0) = 0$. If $i > 0$, let $\Theta'(\eta_i)$ be defined as follows.

> For $x < i$, let m_x be maximal with $\eta_i(m_x) \neq \eta_x(m_x)$. Let $m := \max_{x<i}(m_x)$. Let $k < i$ be minimal with $m_k = m$. (* Among the functions $\eta_0, \ldots, \eta_{i-1}$, none agree with η_i on a longer initial segment than η_k does. *)
> Compute the set $H := \{j \in \mathbb{N} \mid j \notin \{\theta(0), \ldots, \theta(i-1)\}$ and $\psi_j =_m \Theta'(\eta_k)\}$. (* H is the set of unused ψ-programs of functions agreeing with $\Theta'(\eta_k)$ on the first $m+1$ values. *)

Choose $h = \min(H)$; return $\Theta'(\eta_i) := \psi_h$, moreover let $\theta(i) := h$. (* Because of Property 2, the index h exists. As ψ is recursive, h is found effectively. *)

Note that Θ' is a recursive operator mapping P_η into P_ψ. θ is a recursive function, that maps each number i to the index h used in the construction of $\Theta'(\eta_i) = \psi_h$. θ is one-one, yet it may happen, that $\Theta'(\eta_i) = \Theta'(\eta_j)$, but $\theta(i) \neq \theta(j)$ for some $i, j \in \mathbb{N}$.

It remains to define a recursive meta-operator Θ and a recursive operator Ξ such that $\{g(i) \mid i \in \mathbb{N}\}$ is *rUEx*-reducible to D via Θ and Ξ. If δ is an infinite sequence, define $\Theta(\delta, s)$ as follows.

For each $x \in \mathbb{N}$, let $j_x \in \mathbb{N}$ be minimal such that $\eta_{j_x} =_x \delta$. Let $i_x := \theta(j_x)$. Return $\Theta(\delta, s) := ((\overline{d_{i_0}}(0), \overline{d_{i_1}}(1), \ldots), \Theta'(\delta))$.

Clearly, the output of Θ depends only on δ. If δ converges, then $\Theta(\delta, s) = (\delta', f')$, where $f' \in P_\psi$ and δ' converges to some description d_i such that $i = \theta(j)$ for the minimal number j satisfying $\eta_j = \delta$. To define an operator Ξ, compute $\Xi(d\sigma)$ for $d \in \mathbb{N}$, $\sigma \in \mathcal{R}$ as follows.

For $x \in \mathbb{N}$ let $X := \{y \leq x \mid \overline{d_y}(x) = d$ and, for all $z \leq x$, if $\varphi^d_{\sigma(x)}(z)$ is defined in x steps of computation, then $\varphi^d_{\sigma(x)}(z) = \psi_y(z)\}$. If X is empty, let $i_x = 0$, otherwise let i_x be the minimum of X. (* In the limit, the only i satisfying $d_i = d$ and $\varphi^d_{\lim(\sigma)} = \psi_i$ is found – provided that i exists. *)
For each $x \in \mathbb{N}$, compute $j_x \in \mathbb{N}$ with $\theta(j_x) = i_x$. (* In the limit, a number j with $\theta(j) = i$, $d_i = d$, and $\Theta'(\eta_j) = \psi_i$ is found – provided that j exists. *)
Return $\Xi(d\sigma) := (g^{-1}(\eta_{j_0}(0)), g^{-1}(\eta_{j_1}(1)), g^{-1}(\eta_{j_2}(2)), \ldots)$. (* g^{-1} denotes the function inverse to g. $\Xi(d\sigma)$ converges to $g^{-1}(l)$, where l is the limit of η_j with $\Theta'(\eta_j) = \psi_i$ and d_i equals d – provided that i and j exist. *)

To show that $\{g(i) \mid i \in \mathbb{N}\}$ is *rUEx*-reducible to D via Θ and Ξ, fix some $\delta_1 \in \mathcal{R}$ converging to some description $d \in \{g(i) \mid i \in \mathbb{N}\}$.

First, by the remarks below the definition of Θ, we obtain $\Theta(\delta_1, s) = (\delta_2, f_2)$, where $f_2 \in P_\psi$ and δ_2 converges to some description d_i such that $i = \theta(j)$ for the minimal j satisfying $\eta_j = \delta_1$. This implies $f_2 = \psi_i$. In particular, $f_2 \in R_{d_i}$.

Second, if $\sigma \in \mathcal{R}$ is r-admissible for d_i and ψ_i, then $\Xi(d_i\sigma)$ converges to $g^{-1}(d)$ (by the note in the definition of Ξ and $d = \lim(\eta_j)$). Recall that $g^{-1}(d)$ is the only φ^d-program of s, whenever $d \in \{g(i) \mid i \in \mathbb{N}\}$. Hence $\Xi(d_i\sigma)$ is r-admissible for d and s.

So $\{g(i) \mid i \in \mathbb{N}\}$ is *rUEx*-reducible to D and finally D is *rUEx*-complete. □

As an immediate consequence of Theorems 7 and 13 we have:

Corollary 14 *Let $D \in rUEx$. If D is UEx-complete, then D is rUEx-complete.*

3.3 Algorithmic structure of complete classes in uniform learning

Theorems 7 and 13 suggest a weakness of the notion of intrinsic complexity, similar to the non-uniform case: though *UEx-*/*rUEx*-complete sets involve a topologically complex structure, expressed by Property 2, this goes along with the demand for a limiting r. e. subset combined with an r. e. subset P_ψ of the union of all represented recursive cores. The latter again can be seen as a non-complex algorithmic structure.

Now Theorem 15 shows that there are non-complete description sets, for which the properties of Theorems 7 and 13 can be fulfilled, but only if the demand for limiting r. e. sets is dropped. These sets are algorithmically more complex than our examples of *UEx*-complete sets, but they belong to a lower degree of intrinsic complexity.

Theorem 15 *Let $C \subseteq \mathcal{R}$. Then there is a set $D \in rUEx$ such that*

1. *C equals the union of all recursive cores described by D,*
2. *D is not rUEx-complete (and hence not UEx-complete).*

Proof. Fix a list $A_0, A_1, \ldots$ of all infinite limiting r. e. sets such that $\varphi_0^d \in C$ and $\varphi_{x+1}^d =\uparrow^\infty$ for all $i, x \in \mathbb{N}$ and $d \in A_i$. Let $A := \bigcup_{i \in \mathbb{N}} A_i$ and $C = \{f_0, f_1, \ldots\}$. Define a set D_0 as follows.

> Fix the least elements d_0, d_0' of A_0, $d_0 < d_0'$. Let $I_0 := \{d_0\}$, $I_0' := \{d_0'\}$. Let $e_0 \in A \setminus (I_0 \cup I_0')$ be minimal such that $f_0 \in R_{e_0}$. (* e_0 exists, because A contains infinitely many descriptions d with $\varphi_0^d = f_0$. *)
> Let $D_0 := I_0 \cup \{e_0\}$. (* The disjoint sets D_0 and I_0' both intersect with A_0; some recursive core described by D_0 equals $\{f_0\}$. *)

Moreover, for any $k \in \mathbb{N}$, define a set D_{k+1} as follows.

> Fix the least elements d_{k+1}, d_{k+1}' of $A_{k+1} \setminus (D_k \cup I_k')$, $d_{k+1} < d_{k+1}'$. (* These have not been touched in the definition of $D_0, \ldots, D_k$ yet. *)
> Let $I_{k+1} := D_k \cup \{d_{k+1}\}$, $I_{k+1}' := I_k' \cup \{d_{k+1}'\}$. Let $e_{k+1} \in A \setminus (I_{k+1} \cup I_{k+1}')$ be minimal such that $f_{k+1} \in R_{e_{k+1}}$. (* e_{k+1} exists, because A contains infinitely many descriptions d with $\varphi_0^d = f_{k+1}$. *)
> Let $D_{k+1} := I_{k+1} \cup \{e_{k+1}\}$. (* The disjoint sets D_{k+1} and I_{k+1}' both intersect with A_{k+1}; some recursive core described by D_{k+1} equals $\{f_{k+1}\}$. *)

Choose $D := \bigcup_{k \in \mathbb{N}} D_k \subset A$, so D does not contain any infinite limiting r. e. set. As $\varphi_{x+1}^d =\uparrow^\infty$ for all $d \in D$, $x \in \mathbb{N}$, we have $D \in rUEx$. Moreover, C is the union of all cores described by D. It remains to prove that D is not *rUEx*-complete.

Assume D is *rUEx*-complete. Then some limiting r. e. set $\{d_i \mid i \in \mathbb{N}\} \subseteq D$ and some $\psi \in \mathcal{R}$ fulfil the conditions of Theorem 13. In particular, $\{(d_i, \psi_i) \mid i \in \mathbb{N}\}$ is infinite. As D does not contain any infinite limiting r. e. set, the set $\{d_i \mid i \in \mathbb{N}\}$ is finite. $card R_{d_i} = 1$ for $i \in \mathbb{N}$ implies that $\{\psi_i \mid i \in \mathbb{N}\}$ is finite, too; thus $\{(d_i, \psi_i) \mid i \in \mathbb{N}\}$ is finite – a contradiction. So D is not *rUEx*-complete. □

The reason why each *UEx*-/*rUEx*-complete set D contains a limiting r. e. subset representing a decomposition of an r. e. class is that certain properties of *UEx*-complete sets are 'transferred' by meta-operators Θ. This corroborates the possible interpretation that our approach of intrinsic complexity just makes a class complete, if it is a suitable 'target' for recursive meta-operators – similar to the non-uniform case.

By the way, Theorem 15 shows, that every *Ex*-complete class C has a decomposition represented by a description set which is *not* *UEx*-complete – answering a question in Section 3.1.

Acknowledgements. My thanks are due to the referees and to Steffen Lange for their comments correcting and improving a former version of this paper, moreover to Frank Stephan for a very helpful discussion on the technical details.

References

1. Baliga, G.; Case, J.; Jain, S. (1999); The synthesis of language learners, *Information and Computation* 152:16–43.
2. Blum, L.; Blum, M. (1975); Toward a mathematical theory of inductive inference, *Information and Control* 28:125–155.
3. Case, J.; Smith, C. (1983); Comparison of identification criteria for machine inductive inference, *Theoretical Computer Science* 25:193–220.
4. Freivalds, R.; Kinber, E.; Smith, C. (1995); On the intrinsic complexity of learning, *Information and Computation* 123:64–71.
5. Garey, M; Johnson, D. (1979); *Computers and Intractability – A Guide to the Theory of NP-Completeness*, Freeman and Company.
6. Gold, E. M. (1965); Limiting recursion, *Journal of Symbolic Logic* 30:28–48.
7. Gold, E. M. (1967); Language identification in the limit, *Information and Control* 10:447–474.
8. Jain, S.; Kinber, E.; Papazian, C.; Smith, C.; Wiehagen, R. (2003); On the intrinsic complexity of learning recursive functions, *Information and Computation* 184:45–70.
9. Jain, S.; Kinber, E.; Wiehagen, R. (2000); Language learning from texts: Degrees of intrinsic complexity and their characterizations, *Proc. 13th Annual Conference on Computational Learning Theory*, Morgan Kaufmann, 47–58.
10. Jain, S.; Sharma, A. (1996); The intrinsic complexity of language identification, *Journal of Computer and System Sciences* 52:393–402.
11. Jain, S.; Sharma, A. (1997); The structure of intrinsic complexity of learning, *Journal of Symbolic Logic* 62:1187–1201.
12. Jantke, K. P. (1979); Natural properties of strategies identifying recursive functions, *Elektronische Informationsverarbeitung und Kybernetik* 15:487–496.
13. Kapur, S.; Bilardi, G. (1992); On uniform learnability of language families, *Information Processing Letters* 44:35–38.
14. Osherson, D.; Stob, M.; Weinstein, S. (1988); Synthesizing inductive expertise, *Information and Computation* 77:138–161.
15. Rogers, H. (1987); *Theory of Recursive Functions and Effective Computability*, MIT Press.
16. Zilles, S. (2001); On the synthesis of strategies identifying recursive functions, *Proc. 14th Annual Conference on Computational Learning Theory*, LNAI 2111, pp. 160–176, Springer-Verlag.
17. Zilles, S. (2001); On the comparison of inductive inference criteria for uniform learning of finite classes, *Proc. 12th Int. Conference on Algorithmic Learning Theory*, LNAI 2225, pp. 251–266, Springer-Verlag.