TECH REPORT

The Power and the Limitations of Concepts for Adaptivity and Personalization Characterized by Benchmarks of Inductive Inference

ADICOM® SOFTWARE KG
Frauentorstraße 11
99423 Weimar

Email software@adicom-group.de
Tel +49 (0) 3643 85594-0

KLAUS P. JANTKE
SEBASTIAN DREFAHL
OKSANA ARNOLD

Version 1.00
31.10.2018

ÖFFENTLICH / PUBLIC

01-2018

Klaus P. Jantke, Sebastian Drefahl
ADICOM Software KG
Frauentorstr. 11
99423 Weimar
Germany

Oksana Arnold
Erfurt University of Applied Sciences
Altonaer Str. 25
99085 Erfurt
Germany

Impressum

ADICOM TECH REPORT

ISSN (Print) 2627-0749
ISSN (Online) 2627-0757

ADICOM Software KG
Frauentorstr. 11
99423 Weimar

info@adicom-group.de
www.adicom-group.de

Geschäftsführer: Christian Hölzer
AG Jena: HRA501731

Cover Page Design: Robert Krause
Picture Credits:
World Map 45° Lines Vector
Author: pajhonka
https://www.vecteezy.com/map-vector/24225-world-map-45-lines-vector

Contents

Foreword

This is a first written version of the contribution "Characterizing the Power and the Limitations of Concepts for Adaptivity and Personalization by Benchmark Results from Inductive Inference" by *Klaus P. Jantke*, *Sebastian Drefahl* and *Oksana Arnold* recently presented at the international conference ISIP 2018, May 14/15, 2018, in Fukuoka, Japan.

The authors have been invited to submit a version to a book of selected papers to be published within the series CCIS of Springer Verlag, in which every submission of a chapter is limited to 15 pages of LNCS format.

Because of the complexity of the authors' results, especially constructions and proves relying on KLEENE's s-m-n theorem and on the recursion theorem (see [Rj 1967]), on the one hand, and the intriguing interdigitation of different disciplines involved, on the other hand, this page limit is considered inappropriate.

In response, the authors of the contribution decided to write an extended version containing some essentials that would be missing in a manuscript limited to 15 Springer LNCS pages.

Überhaupt hat der Fortschritt das an sich,
daß er viel größer ausschaut, als er wirklich ist. [1]
[Johann Nepomuk Nestroy, “Der Schützling”, 1847]

1 Introduction

In Computer Science and Technology, in general, and in Artificial Intelligence, in particular, innovations are mushrooming. At least, authors say so.

There is no doubt at all, digitalization pervades nearly every sphere of life. Humans are facing more and more digital systems at their workplaces, in everyday education, in their spare time, and in health care. With the US Food & Drug Administration's approval of aripriprazole tablets containing sensors in November 2017, digitalization reaches the inside of the human body.

Frequently, the process of digitalization is placing on humans the burden of learning about new digital systems and how to use them appropriately. More digital systems do not necessarily ease the human life. To use them effectively, users need to become acquainted with software tools, have to understand the interfaces and have to learn how to wield the tools. This bears abundant evidence of the need for a paradigmatic shift from tools to intelligent assistant systems, an issue discussed by the authors in earlier publications such as [JGL 2003], [JIS 2005], [JM 2005], [JSSB 2005], [JIS 2007], [JM 2007], [Jan 2011], [Jan 2013], [Jan 2016b], [Jan 2016a], [Jan 2018], and others over a period of more that one and a half decades.

A digital assistant tries to understand its human user for the sake of adaptation. Intelligent digital assistant systems are, by nature, learning systems.

The authors are working on concept development, on implementation and on application of intelligent assistant systems in varying areas. These assistant systems are aiming at advanced user models on the level of theories of mind. Based on these models, digital assistant systems “believe in understanding users”, personalize themselves accordingly and adapt in different ways.

However to adapt, everything depends on the hypothesized theories of mind that are induced from subsequent observations of human-computer interaction. In other words, theories of mind are learned inductively.

Throughout the process of concept development, design, implementation, and application, ideas of inductive learning are continuously emerging. In conditions of software and communication structures as well as programming languages, the essentials of ideas are incrusted by syntactic sugar. It is diffcult to separate the wheat from the chaff.

According to [Lak 1987], basic concepts[2] are transcendental. They may occur in largely varying areas. The authors take this point of view as a launch pad for an endeavor of mapping learning concepts that occur in system design and implementation to theoretical concepts in recursion-theoretic inductive inference. The mapped concepts are investigated and compared to each other by means of well-established research methods and technologies.

This results in infinite hierarchies of learning concepts characterizing the different power and limitations of varying practical approaches.

[1] English: “In the main it is a quirk of progress to appear bigger than it really is.”

[2] Possibly, some of them may be seen as memes according to Dawkins' seminal book [Daw 1976]. This would allow for bridging the gap to memetic software technologies [Tan 2003], an issue beyond the limits of this report.

2 Characteristics of the Contribution

The paper presentation at ISIP 2018 was delivered by the second author. The stepwise appearance of the presentation title as on display in fig. 1 reveals the characteristics of the contribution. The work is application-oriented.

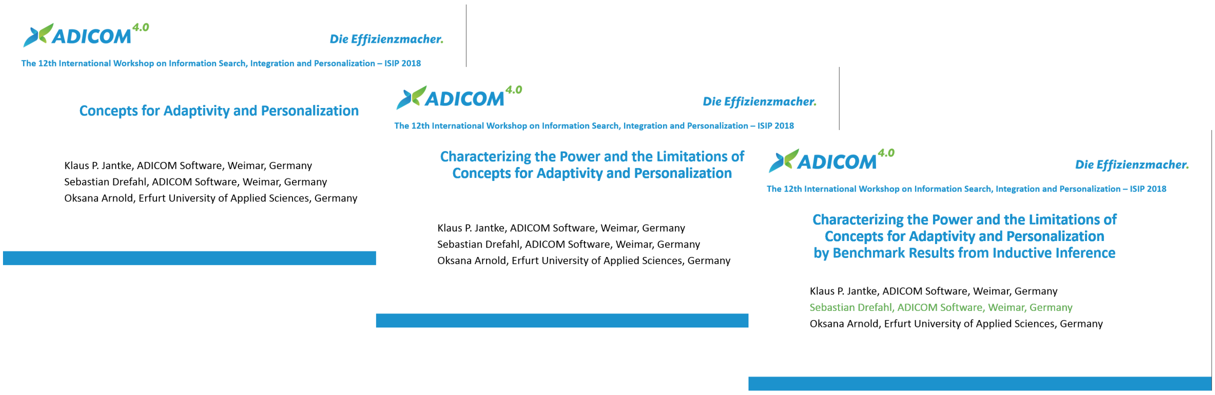

Figure 1: Stepwise Unfolding of the Authors' Title Slide of their ISIP 2018 Presentation

Concepts for Adaptivity and Personalization are in focus. Because the authors' approach is a novelty, it raises the key question for its *Power and Limitations*. To arrive at precise as well as clear answers of mathematical strength, the authors adopt and adapt a series of techniques and *Results of Benchmarking in Inductive Inference*.

Invoked concepts of inductive inference are found in publications such as [JB 1981], [AS 1983], and [JORS 1999].

3 Motivation

There are – four, at least – mutually related motivations underlying this approach, its presentation and its publication.

First of all, the authors would like to take up the cudgels for the paradigmatic shift from a flood of software systems to intelligent digital assistant systems. This brings with it the need for personalization and for context adaptivity by systems that learn.

Second, when speaking about systems that learn, the authors' intention is to propagate theory of mind modeling and induction. Theories of mind are of an enormous expressiveness beyond the limits of conventional user modeling such as overlay models in technology enhanced learning.

Third, it is the authors' intention to demonstrate the utility of benchmarking by means of recursion-theoretic inductive inference. Classes of recursive functions can sharply discriminate between two different learning concepts. Benchmarks tell, so to speak, what makes a difference.

Fourth, the authors – in a few of their projects, at least – aim at sophisticated theory induction without the need to program sophisticated learning algorithms. How does that work? Roughly speaking, there is one almost trivial learning algorithm deployed: identification by enumeration. The art, so to speak, is to provide suitable spaces of hypotheses. Then, learning is performed by a logic programming engine.

4 Theories of Mind for Intelligent System Assistance

The present authors have introduced *theory of mind modeling and induction* as an approach to human user modeling [Jan 2012b, Jan 2012a] of an unprecedented expressiveness [Jan 2016a]. A few application case studies as in [JSS 2016] and [ADFJ 2017] demonstrate the conceptual superiority in comparison to conventional approaches.

4.1 Theories of Mind in Behavioral Studies

The term *theories of mind* names a quite well-established concept of behavioral studies [CS 1996].

As sketched earlier by the authors in [AJ 2018], there is much evidence that certain animals reflect about intentions and behaviors of other animals [EDC 2004, EC 2009]. Birds of the species Aphelocoma californica – the Western scrub jay, esp. the California scrub jay – are food-caching. They do not only cache food, but also colorful and shiny objects such as plastic toys. In case such a bird, let's name it A, is caching food or other treasures and if it is watched by another bird of its species, we name it B, then – with high probability – A returns shortly after to unearth the treasures cached before. The interpretation is, loosely speaking, that the bird A "thinks" about the possibly "malicious thoughts" of the bird B. Bird A builds its own theory of mind. More generally speaking, *thinking about another one's thoughts* means to build a *theory of mind*.

4.2 Theories of Mind in Artificial Intelligence

The two reports [Jan 2012b] and [Jan 2012a] are intended to carry over the theory of mind conceptualizations from behavioral studies in animals to Artificial Intelligence (AI). Digital agents A shall be enabled to build theories of mind about their human users denoted by B.

Apparently, this is an issue of *user modeling*. From a methodological point of view, modeling a human user by generating hypothetical theories of mind is a case of *theory induction*.

Theories are collections of statements expressed in a certain language as known, by way of illustration, from physics. [Pop 1934] discusses essentials of theory induction and provides a firm methodological basis of user modeling by means of theories of mind.

When a digital agent A "tries to understand" a human user B, it forms hypothetical theories. Every particular theory is a *user profile*. In the course of human-computer interaction (HCI), human user profiles evolve over time. This appears, when seen from the process perspective, as *inductive modeling* as well as, when seen from the data perspective, as *mining HCI data* [AJ 2018].

The above-mentioned reports introduce theories of mind to AI by means of a digital games case study that is documented in detail in another report [Jan 2016b] a few years later. Before, the digital game Gorge underlying this study has been used for varying research purposes as in [Jan 2010] and [JHLN 2010]. Carrying on the earlier studies, [JSS 2016] is paving the road for theory of mind learner modeling.

4.3 Theories of Mind Modeling and Induction

As said very recently, "data mining is a creative process of model formation based on incomplete information. In brevity, data mining is inductive modeling" (see [AJ 2018], section 3). As an immediate serious consequence, *data mining results are only hypothetical.* As George E. P. Box put it nicely, "all models are wrong, but some are useful" (see, e.g., [BD 1987], p. 424).

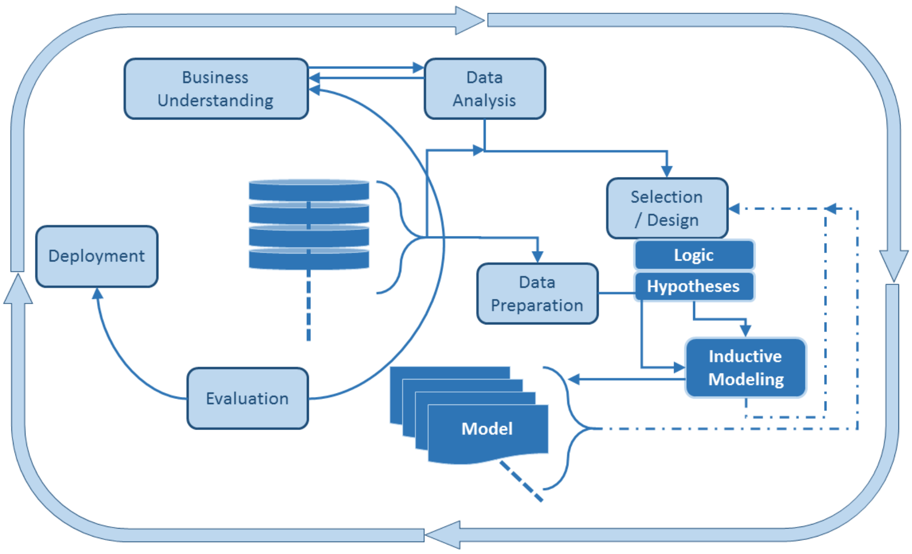

Figure 2: Inductive Modeling as a Process over Time (from [AJ 2018])

This lead the present authors to a refined model of data mining that stresses the feature of induction, because "the thinking about emerging sequences of hypotheses is badly underestimated in contemporary data mining investigations. Pondering model concepts is not sufficient. We need to put emphasis on the investigation of suitable spaces of model hypotheses" (ibid., page 50).

Within the framework of the author's present approach, models are theories. And the models are human user profiles or, vice versa, user profiles are models.

Because user models are digital, the statements that constitute a theory, i.e., a user profile, must adhere to certain rules of syntax. In terminology of logics, there is an underlying signature of the logical language in use. The signature determines a language of logic such that (sets of) logical formulas may be used to explain a human user's behavior. As said before, explanations are hypothetical. Under the assumption of a particular logic, a digital assistant observing a human user's behavior collects observations and tries to find explanations in logical terms. In applications such as [JSS 2016], there exists a fixed space of logical theories. The learning system searches this space of hypotheses to find the first one consistent with the system's observations. To deal with more complex applications, [ADFJ 2017] introduce the *dynamic* generation of spaces of hypotheses. The dynamics is further *generalized* by [ADF$^+$ 2017] introducing what is called NUM^*_{part} below. This generalization goes as far as possible.

Varying ideas, concepts, and implementations of user modeling by means of theory of mind induction are encrusted by large amounts of software-technological details. It is rather difficult to compare alternative approaches and to assess their possibly different power and limitations. This contribution aims at a clarification by stripping ideas of algorithmic learning to the essentials.

5 Concepts of Inductive Inference

The present section provides the technicalities to relieve the key ideas and concepts of [JSS 2016], [ADFJ 2017], and [ADF$^+$ 2017] from, so to speak, the syntactic sugar encrusting them. In terms of recursion theory, the essentials become clear and considerably easy to compare and to assess.

5.1 Notions and Notations

[Rj 1967] is our standard reference for notions and notations of recursion theory. The usage in inductive inference is similar to [JB 1981], [AS 1983], and [JORS 1999].

$\mathcal{P}^n$ is the class of all n-ary partial recursive functions over the set of natural numbers $I\!N$. For any function $f \in \mathcal{P}^1$ and for any argument $x \in I\!N$, the notation $f(x)\!\downarrow$ indicates that f is defined on x. $\mathcal{R}^n$ denotes the subclass of all fully defined functions called general recursive. Learners are functions $\mathcal{L} \in \mathcal{P}^1$. Any function $f \in \mathcal{R}^1$ is given by its graph $f(0)$, $f(1)$, ... where $f[n]$ encodes the sample from $f(0)$ to $f(n)$. Given a GÖDEL numbering φ of $\mathcal{P}^1$, a learner gets fed in data $f[0]$, $f[1]$, $[2]$, $f[3]$, ... and generates sequences of hypotheses $h(n) = \mathcal{L}(f[n])$ in response. Such a hypothesis is interpreted as a program for the function $\varphi_{h(n)}$. $\mathcal{L}$ learns f, if and only if the hypotheses converge to a correct program for f. The family of all function classes learnable in this way is denoted by *EX* and all functions $\mathcal{L}$ can learn form a class named $EX(\mathcal{L})$.

5.2 Conventional Concepts

More formally, assume any function class $C \subseteq \mathcal{R}^1$. C belongs to *EX*, if and only if there exists some learner $\mathcal{L} \in \mathcal{P}^1$ satisfying the conditions **[i]**, **[ii]**, and **[iii]** in the following definition: $\forall f \in C \; \forall n \in I\!N \; ([\mathbf{i}] \; \mathcal{L}(f[n])\!\downarrow \; \wedge \exists k \in I\!N \; \forall m \in I\!N \;\; ([\mathbf{ii}] \, (k \leq m \rightarrow \mathcal{L}(f[k]) = \mathcal{L}(f[m])) \wedge$ $[\mathbf{iii}] \; \varphi_{\mathcal{L}(f[m])} = f \,)$

The definition of *EX* is a bit restrictive, because information is fed in to the learner in the fixed standard order of natural numbers. To generalize, consider arbitrary arrangements $X = \{ x_0, x_1, x_2, x_3, \dots \}$ with the only requirement that every element of $I\!N$ occurs at least once. $\mathcal{X}$ denotes the set of all those orderings[3]. For any $X \in \mathcal{X}$, the information sequence from $(x_0, f(x_0))$ to $(x_n, f(x_n))$ is encoded by $f_X[n]$. The above definition of *EX* is easily generalized to EX^{arb} by substituting f_X for f in the formula $\forall f \in C \; \forall X \in \mathcal{X} \; \forall n \in I\!N \; (\dots)$ as above.

For completeness of this treatment: $\forall f \in C \; \forall X \in \mathcal{X} \; \forall n \in I\!N \; (\mathcal{L}(f_X[n])\!\downarrow \; \wedge \; \exists k \in I\!N \; \forall m \in I\!N \; ((k \leq m \rightarrow \mathcal{L}(f_X[k]) = \mathcal{L}(f_X[m])) \wedge \varphi_{\mathcal{L}(f_X[m])} = f \,)$. It is folklore in inductive inference that $EX^{arb} = EX$.

Initially, Gold studied inductive inference of formal languages [Gol 1967]. His approach and his first results attracted astonishing attention [Joh 2004]. Among his key contributions is the principle of *identification by enumeration*. Inductive inference of recursive functions allows for a particularly transparent explanation.

Assume any total recursive function $h \in \mathcal{R}^1$ used as an enumeration of some function class $\mathcal{P}^1_h = \{\varphi_{h(i)}\}_{i \in I\!N}$. Furthermore, assume that all functions $\varphi_{h(i)}$ are general recursive. In $\mathcal{P}^1_h$, the term $\mathcal{R}^1_h$ denotes the subset of all total functions. Thus, the latter assumption may be rewritten to $\mathcal{P}^1_h = \mathcal{R}^1_h$. Under these assumptions the following learner $\mathcal{L}^{IdbyEn}_h$ is able to learn all functions of $\mathcal{R}^1_h$ according to *EX*. μ denotes the minimum operator.

$\mathcal{L}^{IdbyEn}_h(f[n]) := h(\mu m[\, \varphi_{h(m)}[n] = f[n] \,])$

[3]Notice that $\mathcal{X}$ is uncountably infinite. It is of the same size as the set of real numbers.

In another more precise terminology, it holds $\mathcal{R}^1_h = EX(\mathcal{L}^{IdbyEn}_h)$. It is obvious that Gold's principle works for arbitrary arrangements $X \in \mathcal{X}$ as well; just the formula is more cumbersome.

$\mathcal{L}^{IdbyEn}_h(f_X[n]) := h(\mu m[\,\varphi_{h(m)_X}[n] = f_X[n]\,])$

Consequently, $\mathcal{R}^1_h = EX^{arb}(\mathcal{L}^{IdbyEn}_h)$. The abbreviation *NUM* denotes the family of all classes of general recursive functions contained in an enumeration as above. In a slightly more formal terminology, $C \subseteq \mathcal{R}^1$ belongs to *NUM*, if and only if $\exists h \in \mathcal{R}^1$ ($C \subseteq \mathcal{P}^1_h = \mathcal{R}^1_h \subseteq \mathcal{R}^1$).

Here are two examples of function classes contained in *NUM*. The functions they contain are called functions of finite support. Except finitely many so-called points of support, they return always the value 0.

$C_{\textit{fin-supp}} = \{\ f \mid f \in \mathcal{R}^1\ \wedge\ \exists n \in I\!N\ \forall x \in I\!N (n \leq x \rightarrow f(x) = 0)\ \}$

$C_{\textit{init-supp}} = \{\ f \mid f \in \mathcal{R}^1\ \wedge\ \exists n \in I\!N\ \forall x \in I\!N (n \leq x \leftrightarrow f(x) = 0)\ \}$

Every enumeration h generates some space of hypotheses. When information $f[n]$ or $f_X[n]$, respectively, is provided, the learner $\mathcal{L}^{IdbyEn}_h$ searches for the first hypothesis in the enumeration that is consistent with this information. Formal terms describing consistency are $\varphi_{h(m)}[n] = f[n]$ and $\varphi_{h(m)_X}[n] = f_X[n]$, resp. In recent applications such as human-computer co-operative knowledge discovery [ADFJ 2017], the authors consider consistency to be a necessity.

The appreciation for consistency leads to some refinements of *EX* and EX^{arb}.

Identification by enumeration as above succeeds, because consistency is a recursively decidable property. And this key property is decidable, because h enumerates only general recursive functions. So, what about computable learners that generate only general recursive hypotheses, but possibly in a different way?

TOTAL and $TOTAL^{arb}$ are two families of function classes defined like *EX* and EX^{arb} before with the additional demand on the learner $\mathcal{L}$ to always satisfy $\varphi_{\mathcal{L}(f[n])} \in \mathcal{R}^1$ or $\varphi_{\mathcal{L}(f_X[n])} \in \mathcal{R}^1$, resp., for any f and n (and X, when it applies).

By definition, $NUM \subseteq TOTAL \subseteq EX$ and $NUM \subseteq TOTAL^{arb} \subseteq EX^{arb}$. Due to [JB 1981], e.g., we know $TOTAL = TOTAL^{arb}$ as well as $NUM \subset TOTAL \subset EX$.

This leads to the following research question that has been left open so far.

How does a learner function operationally, if it generates general recursive hypotheses exclusively, but deals with learning problems beyond the limits of *NUM* ...? Practically, how to implement such a learner ...?

Apparently, the generation of only general recursive function hypotheses is *sufficient* to decide consistency. But is it *necessary* as well?

The answer is found by introducing the requirement of consistency into the definitions of *EX* and EX^{arb}. After the terms $\mathcal{L}(f[n])\!\downarrow$ and $\mathcal{L}(f_X[n])\!\downarrow$, one inserts into the corresponding formula the new requirement $\varphi_{\mathcal{L}(f[n])}[n] = f[n]$ or $\varphi_{\mathcal{L}(f_X[n])_X}[n] = f_X[n]$, respectively. The families of classes of general recursive functions learnable consistently are named *CONS* and $CONS^{arb}$. Accordingly, the class of functions learnable by $\mathcal{L}$ is $CONS(\mathcal{L})$ resp. $CONS^{arb}(\mathcal{L})$.

The proper set inclusions $TOTAL \subset CONS^{arb} \subset CONS \subset EX$ are known [JB 1981].

This is by far not the end of dealing with consistent learning. Readers may have noticed that in inductive inference, a learner can hardly be sure that the job is done. By way of illustration, consider the example function class $C_{\textit{fin-supp}}$. When getting fed in the data $f(0)$, $f(1)$, $f(2)$, $f(3)$, $f(4)$, $f(5)$, ..., the learner can not know whether or not it has already seen the final point of support. In contrast, when learning functions from $C_{\textit{init-supp}}$, the occurrence of the first value 0 clearly indicates that sufficient information is provided. In a certain sense, the class $C_{\textit{init-supp}}$ is simpler than $C_{\textit{fin-supp}}$.

When learning functions from $C_{init\text{-}supp}$, a learner is able to decide effectively when the job is done. This is expressed by means of an add-on requirement to condition **[ii]** in the definition of EX and EX^{arb}. There exists some $\delta \in \mathcal{P}^1$ that is defined on $f[n]$ or on $f_X[n]$, resp., exactly if $\mathcal{L}$ is defined. δ decides termination of the learning process by meeting the requirement $(\,\delta(f[n]) = 1 \leftrightarrow k \leq n\,)$ or $(\,\delta(f_X[n]) = 1 \leftrightarrow k \leq n\,)$, resp. The corresponding families of function classes are FIN and FIN^{arb}. Trivially, it holds $FIN = FIN^{arb}$ [JB 1981].

Classes of total recursive functions such as $C_{fin\text{-}supp}$ serve as benchmarks for discriminating varying principles of inductive learning. $C_{fin\text{-}supp} \in NUM \setminus FIN$ tells, so to speak, half of the story about the relationship between NUM and FIN.

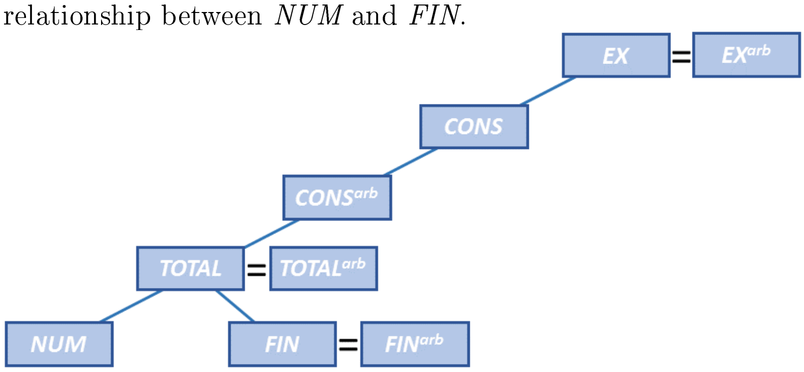

Figure 3: Hierarchy of Selected Principles of Inductive Learning

For the counterpart, another benchmark class is necessary. So-called quines[4] and quine-like functions may be used conveniently. The basic benchmark class of general recursive functions is $C^0_{q\text{-}like} = \{\ f \mid f \in \mathcal{R}^1 \wedge \forall x \in I\!N(f(x) > 0) \wedge \varphi_{f(0)} = f\ \}$.

For getting an impression of the abundance of $C^0_{q\text{-}like}$, assume an arbitrary unary general recursive function $g \in \mathcal{R}^1$. Let us define in a uniform and effective way a sequence of functions $\{\psi_i\}_{i \in I\!N}$ as follows.

$$\psi_i(x) = \begin{cases} i & \textit{if} \ \ x = 0 \\ g(x-1) & \textit{otherwise} \end{cases}$$

Intuitively, every function ψ_i has an identifier at poistion 0 followed by the full graph of function g.

By KLEENE's s-m-n theorem, the sequence $\{\psi_i\}_{i \in I\!N}$ may be effectively translated into any GÖDEL numbering φ. In formal terms, there exists some $c \in \mathcal{R}^1$ such that $\psi_i = \varphi_{c(i)}$ holds for all $i \in I\!N$. Next, one applies the recursion theorem to the compiler function $c \in \mathcal{R}^1$. According to this theorem, there exists some fixed point $n \in I\!N$ with the property $\varphi_n = \varphi_{c(n)}$.

For every fixed point n, the function $\varphi_n = \varphi_{c(n)} = \psi_n$ apparently belongs to the class $C^0_{q\text{-}like}$, because its value at position 0 is an index of the function in φ.

Furthermore, notice two important corollaries from the proof of the recursion theorem [Rj 1967]. Every function in $\mathcal{R}^1$ (a) has *infinitely many* fixed points and (b) the set of fixed points *is not effectively enumerable*. Though this is folklore in recursion theory, it helps to get an idea of the abundance of $C^0_{q\text{-}like}$.

To sum up, any computational behavior represented by a general recursive function occurs in the graph of some quine-like function. Moreover, it occurs in infinitely many of these functions that form a class being so complex that it is beyond the limits of effective enumerability.

Thus, $C^0_{q\text{-}like}$ is not effectively enumerable and it follows $C^0_{q\text{-}like} \in FIN \setminus NUM$.

[4] Quines – named after the philosopher Willard Van Orman Quine – are self-replicating programs; the web is full of quines such that further details may be dropped here [Tho 1999].

5.3 Dynamic Identification by Enumeration

For human player modeling by theories of mind as studied in [JSS 2016], the simple principle of identification by enumeration works well. In contrast, more ambitious tasks of human-computer collaboratory knowledge generation as in [ADFJ 2017] lead to situations in which the expressive limits of a single enumeration are exceeded.

The authors invented what they call *dynamic identification by enumeration.* [ADF$^+$ 2017], section 9.3, p. 80, contains what is called *NUM** below. Beyond the limits of a single enumeration h, the novel key idea is to generate enumerations on demand. For this purpose, an extra generator function $\gamma \in \mathcal{P}^1$ is introduced. There are three practically motivated requirements of *operational appropriateness*, *conversational appropriateness*, and *semantic appropriateness* (ibid., section 8.3, pp. 75/76; see also section 9.3, p. 80). Now, these concepts will be rephrased in terms of recursion theory. Beyond [ADF$^+$ 2017], the present treatment provides a novel infinite hierarchy exhausting *NUM**.

For a lucid comparison, recall the original concept of learning recursive functions by means of identification by enumeration on a given enumeration h:

$\mathcal{L}_h^{IdbyEn}(f_X[n]) := h(\mu m[\,\varphi_{h(m)_X}[n] = f_X[n]\,])$

Instead, generator γ provides – in response to an observation – an ad hoc enumeration deemed appropriate. For given data $f_X[n]$, $\gamma(f_X[n])$ is an index of the generated enumeration and $\varphi_{\gamma(f_X[n])}$ is the enumeration itself. The hypotheses enumerated are $\varphi_{\varphi_{\gamma(f_X[n])}(0)}$, $\varphi_{\varphi_{\gamma(f_X[n])}(1)}$, $\varphi_{\varphi_{\gamma(f_X[n])}(2)}$, $\varphi_{\varphi_{\gamma(f_X[n])}(3)}$, $\varphi_{\varphi_{\gamma(f_X[n])}(4)}$, and so on.

$\mathcal{L}_\gamma^{\mathrm{IdByEn}}(f_X[n]) := \varphi_{\gamma(f_X[n])}(\mu m[\,\varphi_{\varphi_{\gamma(f[n])}(m)_X}[n] = f_X[n]\,])$

This specification of the learner looks as before. The only difference is that the formerly static enumeration h is now replaced by a dynamically generated enumeration $\varphi_{\gamma(f_X[n])}$ in response to observed data $f_X[n]$.

To work effectively, γ has to obey the criteria of appropriateness above. For readability, the *EX*-type definition comes first. For brevity, a notation is adopted and adapted from [Gri 2008]. For any $C \subseteq \mathcal{R}^1$, the term $[C] = \{f[n] \mid f \in C \wedge n \in I\!N\}$ denotes the set of all initial segments of functions. $\forall f \in C\,\forall n \in I\!N\ (\,\gamma(f[n])\!\downarrow\ \wedge\ \varphi_{\gamma(f[n])} \in \mathcal{R}^1\ \wedge\ \mathcal{P}^1_{\gamma(f[n])} \subseteq \mathcal{R}^1\ \wedge\ f[n] \in [\mathcal{P}^1_{\gamma(f[n])}]\,)$ specifies the operational appropriateness necessary for effective identification by enumeration. $\forall f \in C\,\exists k \in I\!N\,\forall m \in I\!N\ (\,k \leq m \rightarrow \gamma(f[k]) = \gamma(f[m])\,)$ is conversational appropriateness. It prevents the human-computer communication from, so to speak, a Babylonian explosion of terminology. Finally, semantic appropriateness with respect to this particular k for $f \in C$ means $f \in \mathcal{P}^1_{\gamma(f[k])}$.

Formally, any class $C \subseteq \mathcal{R}^1$ belongs to *NUM**, if and only if there exists some generator $\gamma \in \mathcal{P}^1$ meeting the three conditions above. This is the brief summary:

$$\begin{aligned}&\forall f \in C\,\forall n \in I\!N\ (\,\gamma(f[n])\!\downarrow\ \wedge\ \varphi_{\gamma(f[n])} \in \mathcal{R}^1\ \wedge\\ &\qquad\qquad \mathcal{P}^1_{\gamma(f[n])} \subseteq \mathcal{R}^1\ \wedge\ f[n] \in [\mathcal{P}^1_{\gamma(f[n])}]\,)\ \wedge\\ &\exists k \in I\!N\ (\,\forall m \in I\!N\ (\,k \leq m \rightarrow \gamma(f[k]) = \gamma(f[m])\,)\ \wedge\ f \in \mathcal{P}^1_{\gamma(f[k])}\,)\,)\end{aligned}$$

NUM^*_γ is the class of all functions f for which γ meets the above requirements.

γ may be seen as a possibly partial mapping from $I\!N^*$ to $I\!N$. $\gamma(\varepsilon)$ denotes the default enumeration. In case γ is a constant function, i.e. for all data it holds $\gamma(f[n]) = \gamma(\varepsilon)$, the novel concept *NUM** coincides with the prior concept *NUM.*

If a function $f \in C$ is subsequently presented by $f[0]$, $f[1]$, $f[2]$, $f[3]$, ..., the generator γ provides varying effective enumerations of spaces of hypotheses $\varphi_{\gamma(f[0])}$, $\varphi_{\gamma(f[1])}$, $\varphi_{\gamma(f[2])}$, ... on which the learner $\mathcal{L}_\gamma^{\mathrm{IdByEn}}$ can perform identification by enumeration.

Every generator γ as above can be generalized to a generator of type EX^{arb}.

$$\gamma(f_X[n]) = \begin{cases} \gamma(\varepsilon) & \textit{if} \quad (0,f(0)) \not\preceq f_X[n] \\ \gamma(f[m]) & \textit{otherwise} \quad \textit{where } m = \\ & \qquad max\{k \mid \forall x \leq k\ (\,(x,f(x)) \preceq f_X[n]\,)\} \end{cases}$$

where $(x,f(x)) \preceq f_X[n]$ means that an observation $(x,f(x))$ occurs in $f_X[n]$.

As a consequence, $\mathcal{L}^{\mathrm{IdByEn}}_{\gamma}$ works on arbitrary arrangements of information. The main new result of [ADF$^+$ 2017], section 9.4, is $NUM^* = TOTAL$. This is an answer to the research question about how to learn consistently in case the problem class is beyond the limits of NUM. Whensoever a learner is facing a learning problem from $TOTAL$, the present problem may be solved by identification by enumeration, but *dynamically*.

When a function $f \in NUM^*_{\gamma}$ is presented piece by piece, the learner $\mathcal{L}^{\mathrm{IdByEn}}_{\gamma}$'s generator function γ provides changing spaces of hypotheses from time to time. In applications of business intelligence as in [ADF$^+$ 2017], presenting a new space of hypotheses may bring with it a certain expansion of terminology. The more terminology is enriched, the more can be said explicitly.

This leads directly to another novel hierarchy (see figure 3 above).

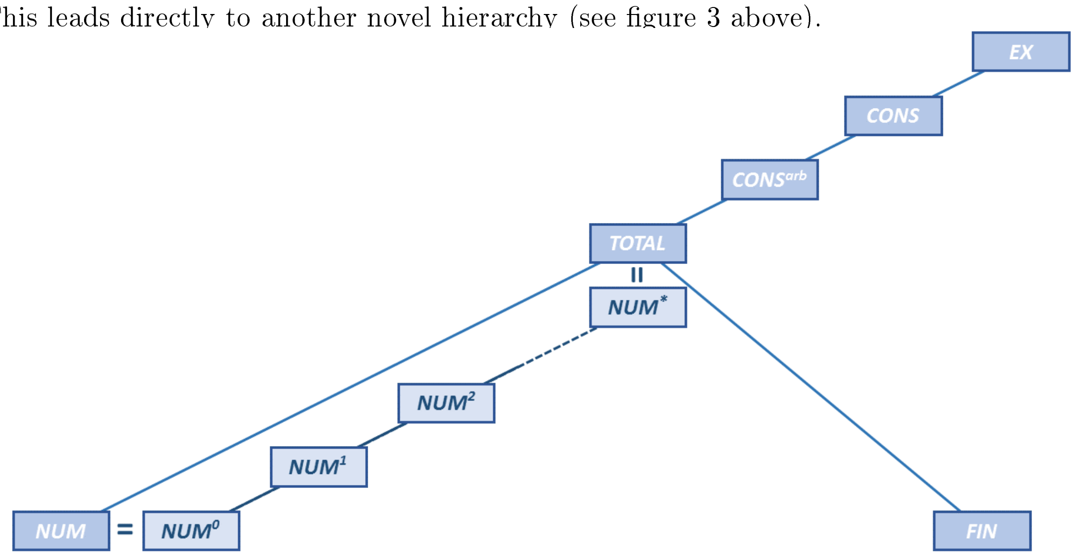

Figure 4: Infinite Hierarchy of Novel Learning Concepts Related to Earlier Established Concepts; every ascending line indicates a proper inclusion of the lower family in the upper one

For any natural number $k \in I\!N$, a class of general recursive functions $C \subseteq \mathcal{R}^1$ belongs to NUM^k, if and only if there exists a generator of spaces of hypotheses γ with $C \subseteq NUM^*_{\gamma}$ and it holds $|\{\gamma(f[n]) \mid f \in C \wedge n \in I\!N\} \cup \{\gamma(\varepsilon)\}| \leq k+1$.
Intuitively, $C \in NUM^0$ means that there is no need for the generator γ to provide any enumeration different from the default. Slightly generalized for any $k \in I\!N$, $C \in NUM^k$ means that γ succeeds with no more than k newly generated spaces of hypotheses per function $f \in C$.

By definition, it holds $NUM \subseteq NUM^0 \subseteq NUM^1 \subseteq NUM^2 \subseteq \ldots\ NUM^*$ and, trivially, $NUM = NUM^0$. Figure 3 above visualizes a variety of statements about the variants of dynamic identification by enumeration in this infinite chain of learning concepts such as (i) that the variants form a proper infinite hierarchy (ii) exhausting the space between NUM and $TOTAL$, and (iii) that all variants NUM^k, except NUM^*, are incomparable to FIN.

Within this paper, equalities such as $NUM = NUM^0$ and $NUM^* = TOTAL$ are seen trivial. By way of illustration, let us briefly discuss the second one.

First, $NUM^* \subseteq TOTAL$ is considered. Assume any class $C \in NUM^*$ and any function $f \in C$.

When f is presented piecewise with respect to any ordering and if the generator function γ provides spaces of hypotheses accordingly, identification by enumeration does always return an index of a general recursive function found in the current space of hypotheses. This demonstrates $NUM^* \subseteq TOTAL$ directly.

Second, $TOTAL \subseteq NUM^*$ is considered. Assume any class $C \in TOTAL$ and any learner $\mathcal{L} \in \mathcal{P}^1$ for C. There is a bit preposterous construction as follows. When $\mathcal{L}$ returns a hypothesis on whatsoever data $f_X[n]$, this hypothesis $\mathcal{L}(f_X[n])$ is an index of a general recursive function $\varphi_{\mathcal{L}(f_X[n])}$. The generator γ may be defined to always return some enumeration of the singleton class $\{\varphi_{\mathcal{L}(f_X[n])}\}$. Because the learner terminates on every presentation of a function, γ does change its spaces of hypotheses only finitely many times. Due to this ridiculous choice of an enumeration, the procedure of identification by enumeration does always find the same hypotheses as $\mathcal{L}$ does. This demonstrates $TOTAL \subseteq NUM^*$ directly.

In contrast to mostly trivial equalities, the task to discriminate variants of learning concepts is rather involved, as we will see in detail in section 7 below.

Within the final paragraphs of the present section, we focus the question whether or not $TOTAL$ characterizes the best we can do with computerized learning by means of identification by enumeration.

In business applications as investigated in [ADFJ 2017], a system's utterances about its (intermediate) learning results shall be sound with the data occuring in the course of human-system interaction. Fully defined hypotheses generated in accordance with the $TOTAL$ learning model allow for automatically checking consistency. The inclusions $TOTAL \subset CONS^{arb} \subset CONS$ lead to the question for a further *generalization of identification by enumeration.*

Theory of mind modeling and induction for purposes of business intelligence applications shall not go beyond the limits of consistent learning. But could we possibly exhaust the problem space $CONS$ or $CONS^{arb}$, at least, just by identification by enumeration in suitably generated more general spaces of hypotheses? How to generalize ...?

NUM^* is repeated in a tabular form. A class $C \subseteq \mathcal{R}^1$ belongs to NUM^*, if and only if there is a generator of hypotheses spaces $\gamma \in \mathcal{P}^1$ meeting this condition:

i	$\forall f \in C \;\; \forall X \in \mathcal{X} \;\; \forall n \in I\!N$	
ii	$(\quad \gamma(f_X[n])\!\downarrow$	$\wedge$
iii	$\varphi_{\gamma(f_X[n])} \in \mathcal{R}^1$	$\wedge$
iv	$\mathcal{P}^1_{\gamma(f_X[n])} \subseteq \mathcal{R}^1$	$\wedge$
v	$f_X[n] \in [\mathcal{P}^1_{\gamma(f_X[n])}]$	$\wedge$
vi	$\exists k \in I\!N$	
vii	$(\quad \forall m \in I\!N$	
viii	$(\quad k \leq m \;\rightarrow\; \gamma(f_X[k]) = \gamma(f_X[m]) \quad) \wedge$	
ix	$f \in \mathcal{P}^1_{\gamma(f_X[k])} \qquad)$	$)$

The crucial lines are ii, iii, iv, v, viii, and ix. For identification by enumeration, ii and iii are inevitable, viii guarantees termination, and finally ix guarantees correctness.

Aiming at a further generalization beyond the limits of dynamic identification by enumeration formalized by NUM^*, all that may be placed at disposal is condition iv above.

6 Generalized Dynamic Identification by Enumeration

The tabular definition of NUM^* on the page before is taken as a basis of a slight modification.

Recall that the goal is to learn by means of identification by enumeration on every generated space of hypotheses $\varphi_{\gamma(f_X[n])}$. The learner to be deployed on every space of hypotheses is $\mathcal{L}^{\mathrm{IdByEn}}_{\varphi_{\gamma(f_X[n])}}$, respectively.

Condition iv – which is part of operational appropriateness – is given up, i.e., a space of hypotheses may contain objects that do not allow for consistency checks in arbitrary conditions. Technically spoken, functions enumerated may be partial. To prevent $\mathcal{L}^{\mathrm{IdByEn}}_{\varphi_{\gamma(f_X[n])}}$ from running into non-terminating computations, the following necessary and sufficient condition – logically weaker than (iv) – is introduced.

$$\forall s \in I\!N\ (\ {\varphi_{\varphi_{\gamma(f_X[n])}(s)}}_X[n] \prec f_X[n]\ \rightarrow$$
$$\exists r \in I\!N\ (\ r < s\ \wedge\ {\varphi_{\varphi_{\gamma(f_X[n])}(r)}}_X[n] = f_X[n]\ )\ )$$

In this formula, the new symbol $\prec$ denotes the proper substring relation. For X, f, g and n, it holds $g_X[n] \preceq f_X[n]$ exactly if $g(x_m)\!\downarrow$ implies $f(x_m)\!\downarrow$ and $g(x_m) = f(x_m)$ for all $m \leq n$. The symbol $\prec$ denotes the irreflexive subset of the relation $\preceq$.

When for a class $C \subseteq \mathcal{R}^1$ some generator γ exists that satisfies the following condition, C belongs to a generalization of NUM^* called NUM^*_{Xpart}. The index letter X refers to arbitrary arrangements of information and the string *part* indicates that enumerations may contain partial functions.

i	$\forall f \in C\ \ \forall X \in \mathcal{X}\ \ \forall n \in I\!N$	
ii	$(\quad \gamma(f_X[n])\!\downarrow$	$\wedge$
iii	$\varphi_{\gamma(f_X[n])} \in \mathcal{R}^1$	$\wedge$
iv.a	$\forall s \in I\!N$	
iv.b	$(\quad {\varphi_{\varphi_{\gamma(f_X[n])}(s)}}_X[n] \prec f_X[n]\ \rightarrow$	
iv.c	$\exists r \in I\!N$	
iv.d	$(\quad r < s\ \wedge\ {\varphi_{\varphi_{\gamma(f_X[n])}(r)}}_X[n] = f_X[n]\quad)\ )$	$\wedge$
v	$f_X[n] \in [\mathcal{P}^1_{\gamma(f_X[n])}]$	$\wedge$
vi	$\exists k \in I\!N$	
vii	$(\quad \forall m \in I\!N$	
viii	$(\quad k \leq m\ \rightarrow\ \gamma(f_X[k]) = \gamma(f_X[m])\quad)\ \wedge$	
ix	$f \in \mathcal{P}^1_{\gamma(f_X[k])}\qquad)$	$)$

Analogously, one may define NUM^*_{part} by restriction to the standard ordering of natural numbers. The index X is dropped in the formula and in the notation of the family of function classes. By definition, it holds $NUM^*_{Xpart} \subseteq NUM^*_{part}$.

As before, one may define NUM^k_{Xpart} and NUM^k_{part} by limiting the number of newly generated spaces of hypotheses by a constant k.

Once again, the equalities are almost trivial. It holds $NUM^*_{Xpart} = CONS^{arb}$ and $NUM^*_{part} = CONS$. This does immediately imply $NUM^*_{Xpart} \subset NUM^*_{part}$. But the discrimination of only slightly varying variants of learning is involved.

Before going into the details, figure 4 below visualizes the entirety of the authors' novel results on learning by means of *dynamic* and *generalized dynamic identification by enumeration*. The lighter boxes with dark inscriptions on the right side show the authors' original hierarchies in comparison to conventional types of learning on the left side with white inscriptions.

The crude concepts NUM^0_{Xpart} and NUM^0_{part} defined by constant generators γ (see the preceding tabular definition) are of only theoretical interest, because a practically relevant generation of varying spaces of hypotheses in response to observations does not take place. They are dropped in the following survey.

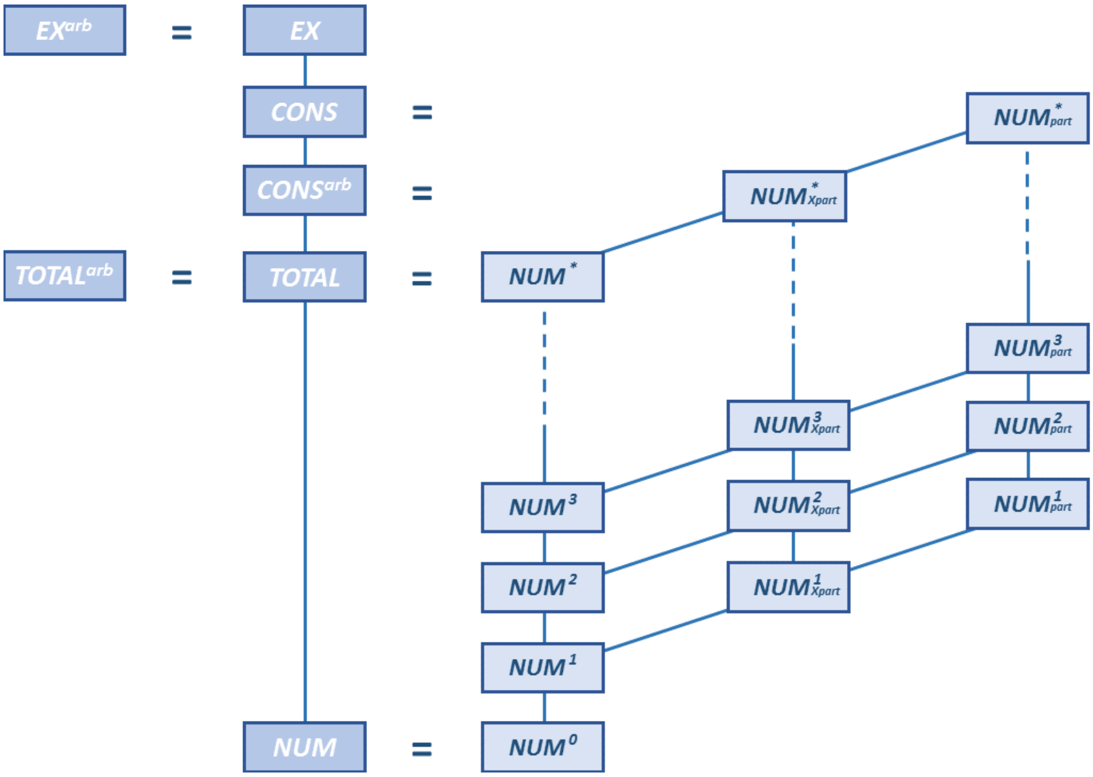

Figure 5: Two Hierarchies of Novel Learning Concepts Related to Earlier Established Concepts; every ascending line indicates a proper inclusion of the lower problems family in the upper one.

Not to distract the reader's attention from the paper's original topic which is benchmarking to discriminate closely related practical learning ideas, the authors confine themselves to just a few remarks about the equality $NUM^*_{part} = CONS$.

First, the statement $NUM^*_{part} \subseteq CONS$ is trivial, because the learner $\mathcal{L}^{\mathrm{IdByEn}}_{\varphi_{\gamma(f[n])}}$ works always consistently.

Second, for the statement $CONS \subseteq NUM^*_{part}$ works a somehow preposterous construction as before. Assume any class $C \in CONS$ with a related learner $\mathcal{L}$. On any data $f[n]$, the generator of spaces of hypotheses γ may be defined to return an enumeration of the singleton class $\{\varphi_{\mathcal{L}(f[n])}\}$. Due to the consistency of the learner $\mathcal{L}$, $\mathcal{L}^{\mathrm{IdByEn}}_{\varphi_{\gamma(f[n])}}$ running on an enumeration of $\{\varphi_{\mathcal{L}(f[n])}\}$ finds exactly the hypothesis generated by $\mathcal{L}$. Consequently, $\mathcal{L}^{\mathrm{IdByEn}}_{\varphi_{\gamma(f[n])}}$ behaves exactly like $\mathcal{L}$ and, thus, learns successfully.

Set inclusions such as $NUM^1 \subset NUM^1_{Xpart} \subset NUM^1_{part}$ as well as incomparabilities such as $NUM^2_{Xpart} \# NUM^1_{part}$ are considerably more involved.

7 Benchmarking

Assume any two families of function classes as on display in figure 5. For the sake of an abstract treatment, we call them FFC_1 and FFC_2. Without loss of generality, assume $FFC_1 \subseteq FFC_2$. To demonstrate $FFC_1 \subset FFC_2$, the key methodological idea is to construct a benchmark class that reflects *the gist of the learning principle* underlying the concept FFC_2 as closely as possible.

7.1 Benchmark Classes

Recall the benchmark classes $C_{\textit{init-supp}}$, $C_{\textit{fin-supp}}$ and $C^0_{\textit{q-like}}$ introduced earlier. The utility of the latter one for the demonstration of novel results on display in figure 5 is discussed in the following subsection. Furthermore, the concept is substantially advanced to allow for the demonstration of more involved results. $C^k_{\textit{w-q-like}}$ is called the class of *k-fold weakly quine-like functions*. In this context, *weakness* is that with every function in a class there are infintely many variations.

$$
\begin{aligned}
C^k_{\textit{w-q-like}} = \{\, f \mid\, & f \in \mathcal{R}^1 \,\wedge\, \exists n_1,\ldots,n_k \in I\!N \; (\, n_1 < \ldots\; < n_k \,\wedge\, f(n_1) = \ldots\; = f(n_k) = 0 \,\wedge \\
& \varphi_{f(n_k+1)} \in \mathcal{R}^1 \,\wedge\, \forall x \in I\!N \; (\, x \notin \{n_1,\ldots,n_k\} \,\rightarrow\, f(x) > 0\,) \,\wedge \\
& \forall n,x \in I\!N \; (\, n \in \{n_1,\ldots,n_{k-1}\} \,\wedge\, n < x \,\wedge\, \forall x' \in I\!N \; (\, n < x' \leq x \,\rightarrow\, f(x') > 0\,) \\
& \rightarrow\, f[x] = \varphi_{f(n+1)}[x]\,) \,\wedge \\
& \forall^\infty x \in I\!N \; (\, n_k + 1 < x \,\rightarrow\, f(x) = \varphi_{f(n_k+1)}(x)\,)\,)\,\} \\
C^{\leq k}_{\textit{w-q-like}} = & \bigcup_{i=1}^{k} C^i_{\textit{w-q-like}} \\
C^{*}_{\textit{w-q-like}} = & \bigcup_{i=1}^{\infty} C^i_{\textit{w-q-like}}
\end{aligned}
$$

$C^*_{\textit{w-q-like}}$ and its subclasses represent prototypical ideas of benchmarking. Learning functions in $C^*_{\textit{w-q-like}}$ on the standard ordering $X_0 = 0,1,2,\ldots$ is trivial. The difficulties of learning on arbitrary orderings are exemplified in the sequel. But first, in figure 6, it follows an illustration[5].

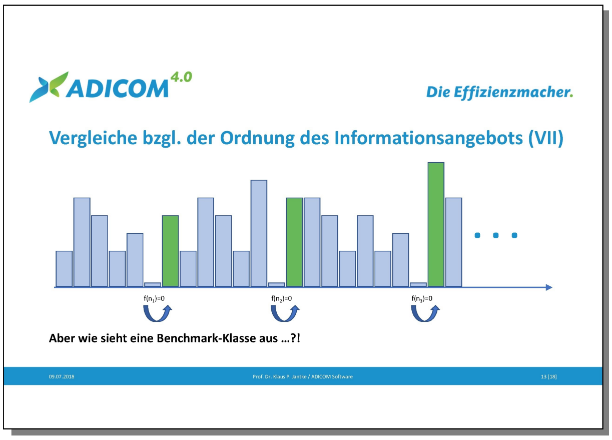

Figure 6: Visualization of the graph of a function in $C^*_{\textit{w-q-like}}$ with emphasis on indicators

Observations of the form $(n_i, 0)$ are *indicators* announcing, so to speak, key information to come on input point $n_i{+}1$ (green bars in the function graph of fig. 6). The *key value* $f(n_i{+}1)$ names a function $\varphi_{f(n_i+1)}$ that correctly describes the target function up to the next indicator. If n_k is the last indicator, $\varphi_{f(n_k+1)}$ equals the target function up to finitely many exceptions.

[5] This is a slide from the first author's lecture on "Learning Systems" at Erfurt University of Technology in the Summer term 2018. In this way the authors demonstrate the uptake of novel results in regular teaching.

7.2 Benchmarking Exemplified

By way of illustration, the authors use benchmarking to characterize the power and the limitations of their (i) dynamic and (ii) generalized dynamic variations of identification by enumeration.

The long-known benchmark class $C^0_{\textit{q-like}}$ (see section 5.2 above) belongs to FIN, but not to $\mathit{NUM} = \mathit{NUM}^0$. The inclusion $C^0_{\textit{q-like}} \in \mathit{NUM}^1$ is rather obvious and illustrates why dynamic identification by enumeration leads to an infinite hierarchy. The authors briefly provide a sketch.

Demonstration of Learning by means of Dynamic Identification by Enumeration

For learning according to the concept NUM^1, one may take an enumeration of $C_{\textit{init-supp}}$ as the initial default.

How to define a generator γ is obvious. Whenever a first observation of the form $(0, f(0))$ is made, γ abandons the default space of hypotheses and generates an enumeration of the singleton class $\{\varphi_{f(0)}\}$. Trivially, $\mathcal{L}^{\mathrm{IdByEn}}_{\varphi_{\gamma(f[0])}}$ is able to learn $\varphi_{f(0)}$ which is the only element of that class. This rediculously simple case demonstrates that the key to learning lies in the appropriate space of hypotheses.

The basic idea of learning by means of dynamic identification by enumeration demonstrated above works as well for the benchmark class $C^0_{\textit{w-q-like}}$. The only difference is that, in response to an observation $(0, f(0))$, γ generates an enumeration of all finite variations (on inputs greater than 0) of $\varphi_{f(0)}$. Thus, $C^0_{\textit{w-q-like}} \in \mathit{NUM}^1$. The understanding of $\mathit{NUM} = \mathit{NUM}^0 \subset \mathit{NUM}^1$ is a first step toward understanding the inifinite hierarchies on display in figure 5.

The proper set inculsion claimed relies on $C^0_{\textit{w-q-like}} \notin \mathit{NUM}^0$. This follows from the facts (i) that $C^0_{\textit{w-q-like}}$ contains $C^0_{\textit{q-like}}$ and (ii) $C^0_{\textit{q-like}} \notin \mathit{NUM}^0$ which is folklore in inductive inference (see, e.g., [JB 1981] or [AS 1983]).

Demonstration of the Power of Generalizing Dynamic Identification by Enumeration

The idea to further generalize dynamic identification by enumeration provides more learning power as may be demonstrated by means of the benchmark class $C^0_{\textit{q-like}}$. In contrast to the negative result $C^0_{\textit{q-like}} \notin \mathit{NUM}^0$, it holds $C^0_{\textit{q-like}} \in \mathit{NUM}^0_{\mathit{part}}$.

The following definition of a sequence of functions ψ_n leads directly to a suitable enumeration.

$$\psi_n(x) = \begin{cases} n & \text{if} \quad x = 0 \\ \varphi_n(x) & \text{if} \quad x > 0 \end{cases}$$

By Kleene's s-m-n theorem, there exists an enumeration $h \in \mathcal{R}^1$ such that for all indices $n \in I\!N$ it holds $\psi_n = \varphi_{h(n)}$. Apparently, $C^0_{\textit{q-like}} = \mathcal{P}^1_h \cap \mathcal{R}^1 \subset \mathcal{P}^1_h = \{\varphi_{h(n)}\}_{n \in I\!N}$.

Every function in $\mathcal{P}^1_h$ is defined on argument 0 and any two different functions have different values on input 0. Therefore, when $(0, f(0))$ is observed, identification by enumeration on h finds the hypothesis $\varphi_{f(0)}$ correctly. Therefore, $\mathcal{L}^{\mathrm{IdByEn}}_h$ learns every function from $C^0_{\textit{q-like}}$ without any need to change the space of hypotheses. This completes the demonstration of $C^0_{\textit{q-like}} \in \mathit{NUM}^0_{\mathit{part}}$.

The two separations $\mathit{NUM}^0 \subset \mathit{NUM}^1$ and $\mathit{NUM}^0 \subset \mathit{NUM}^0_{\mathit{part}}$ by means of the benchmark class $C^0_{\textit{q-like}}$ are prototypical cases of benchmarking. In the present report, they may be seen as a warming up for the more interesting and complex case studied in the sequel based on $C^*_{\textit{w-q-like}}$.

The Importance of Ordering for Generalized Dynamic Identification by Enumeration

The final part of section 7 is dedicated to a study of the importance of ordering observations in generalized dynamic identification by enumeration.

From the benchmark class $C^*_{w\text{-}q\text{-}like}$ introduced before, the subclass $C^{\leq 2}_{w\text{-}q\text{-}like}$ will be sufficient to demonstrate the following fact:

$$NUM^2_{part} \setminus NUM^{\infty}_{Xpart} \neq \emptyset$$

In learning by means of generalized dynamic identification by enumeration, to give up the requirement of a fixed ordering of observations can not be compensated by arbitrarily many more – even trillions – of changes of spaces of hypotheses.

Proof of $C^{\leq 2}_{w\text{-}q\text{-}like} \in NUM^2_{part}$

An identification by enumeration learner for $C^{\leq 2}_{w\text{-}q\text{-}like}$ may take any enumeration of $C_{fin\text{-}supp}$ as the initial default. As long as all function values observed are greater than 0, the learner searches the default enumeration.

As soon as an argument-value pair $(n, 0)$ is observed, the learner gets ready for a change of the space of hypotheses. Nevertheless, its current hypothis is still found in the default enumeration.

When $(n+1, f(n+1))$ is available, the generator γ computes an enumeration of the class of all finite variations of the, so to speak, skeleton function $\{\varphi_{f(n+1)}\}$. One may assume that the skeleton function occurs first in this enumeration. There are only variations enumerated that differ from the skeleton at input points greater than $n+1$ and that take values greater than 0. The generator function γ sticks to this enumeration as long as all values observed further on are greater than 0, i.e., no second indicator occurs.

In this case, convergence and correctness is obvious, because (i) the skeleton function $\varphi_{f(n+1)}$ is consistent with $f[n+1]$ and (ii) if variations are observed – what may take place only finitely many times – another consistent function is contained in the enumeration.

Proof of $C^{\leq 2}_{w\text{-}q\text{-}like} \notin NUM^{\infty}_{Xpart}$

To prove $C^{\leq 2}_{w\text{-}q\text{-}like} \notin NUM^{\infty}_{Xpart}$, the authors demonstrate a proof by contradiction. If $C^{\leq 2}_{w\text{-}q\text{-}like}$ were in NUM^{∞}_{Xpart}, there would be a consistent learner $\mathcal{L}$ able to learn every function of the class $C^{\leq 2}_{w\text{-}q\text{-}like}$ on arbitrary arrangements of data.

The idea of the proof consists in the effective definition of a sequence of functions $\{\psi^y\}_{y=1,2,3,\ldots}$ such that (i) there is a function ψ^n which belongs to $C^{\leq 2}_{w\text{-}q\text{-}like}$ and (ii) the assumed learner is changing its hypotheses on ψ^n infinitely many times.

Choose any number $m \geq 2$. Imagine the initial segment of functions constantly equal to 1 up to point $m-2$ and equal to 0 at point $m-1$ being an indicator. For every $y \in I\!N$, one defines

$$\psi^y(x) = \begin{cases} 1 & \textit{if} \quad x < m-1 \\ 0 & \textit{if} \quad x = m-1 \qquad\qquad \Longleftarrow \text{ the indicator} \\ y & \textit{if} \quad x = m \\ u & \textit{if} \quad x - m \equiv 0 \,\mathsf{mod} 3 \wedge \textit{exactly } \psi^y_{X^\Delta}[x-3] \textit{ is defined so far } \wedge \\ & \qquad u = \mu v[\, 0 < v \wedge \mathcal{L}(\psi^y[x-3]) \neq \mathcal{L}(\psi^y[x-3] \circ (x,v)) \,] \\ \psi^y(x+3-d)) & \textit{if} \quad x - m \equiv d \,\mathsf{mod} 3, \textit{where } d \neq 0 \quad \textit{otherwise} \end{cases}$$

extending this initial segment begining at point m. $\circ$ denotes the concatenation of (lists of) observations. $X^\Delta \in \mathcal{X}$ denotes 0,1,...,m,m+3,m+2,m+1,m+6,m+5,m+4,m+9,m+8,m+7, ...

The process of subsequently defining every ψ^y is illustrated by means of the following figure 7 assembled from the authors' ISIP 2018 presentation slides.

Figure 7: Stepwise Effective Definition of ψ^y in Dependence on the Assumed Consistent Learner; 6 Subsequent Slides from the Authors' Presentation at ISIP 2018, Fukuoka, Japan, in May 2018

The light blue boxes on the slides represent values of the function ψ^y under construction. The construction over time is visualized horizontally. Initially, the first values on arguments from 0 to $m-2$ are set to 1. This is visualized by a pile of boxes on the left. At position $m-1$ there occurs an indicator. This is a next step of construction and, therefore, it is shown slightly on the right of the pile of boxes. In the next step, it follows value y on argument m, i.e., $\psi^y(m) = y$.

After the occurence of a first indicator at position m–1, the turn-based construction proceeds on X^Δ in a uniform way by defining triplets of values on the arguments m+3, m+2, m+1, followed by $m+6, m+5, m+4$, then $m+9, m+8, m+7$, and so on.

The assignment on values to triplet elements begins on the second slide. The third slide shows the first complete triplet. The fourth and the fifth slide illustrate the emergence of the second triplet. Finally, the sixth slide in the lower right corner of the figure 7 is intended to give an impression of the overall infinite process.

When by the end of some turn, ψ^y is defined up to some point k, the next argument in focus is k+3. For any value u there is a value z such that the class $C^{\leq 2}_{w\text{-}q\text{-}like}$ contains functions f with four critical properties. First, $f[k] = \psi^y[k]$. Second, $f(k{+}1) = 0$, i.e., k+1 is an indicator. Third, $f(k{+}2) = z$ and $\varphi_z = f$. And fourth, $f(k{+}3) = u$. Appendix B demonstrates this crucial fact.

Consequently, when a function like ψ^y is observed up to point k, any value u may occur at point $k+3$ and, thus, the learner $\mathcal{L}$ must be able to generate a hypothesis when the data $(k+3, u)$ are coming in next. In particular, because $\mathcal{L}$ learns consistently, for different values u, the hypotheses must be mutually different to guarantee consistency in every case.

This justifies the line $u = \mu v[\,0 < v \wedge \mathcal{L}(\psi^y[x-3]) \neq \mathcal{L}(\psi^y[x-3] \circ (x, v))\,]$ when defining ψ^y. u is determined and may be computed effectively. It holds $\mathcal{L}(\psi^y[x-3]) \neq \mathcal{L}(\psi^y[x-3] \circ (x, u))$. When in the current turn u is found, ψ^y is extended by assigning the value u first to $\psi^y(k+3)$ and then to $\psi^y(k+2)$ and $\psi^y(k+1)$.

It follows the next turn, and so on. For every $y \in I\!N$, ψ^y is effectively defined in a uniform way depending on the parameter y exclusively. The learner $\mathcal{L}$ is used as a subroutine.

By KLEENE's s-m-n theorem, there exists a total recursive function $c \in \mathcal{R}^1$ – one may see c as a compiler that maps the enumeration $\{\psi^y\}_{y \in I\!N}$ into the underlying GÖDEL numbering φ – such that $\psi^y = \varphi_{c(y)}$ for all $y \in I\!N$.

By the recursion theorem, the compiler c has a fixed point z satisfying $\varphi_{c(z)} = \varphi_z$. Because of $\psi^z = \varphi_{c(z)} = \varphi_z$, this function is of a particular interest. It has exactly one indicator at point $m-1$ and its value at point m is an index of this function. Consequently, it belongs to the benchmark class $C^{\leq 2}_{\textit{w-q-like}}$.

Allegedly, $\mathcal{L}$ is a learner for all functions of $C^{\leq 2}_{\textit{w-q-like}}$. But when ψ^z is presented in the order X^{Δ}, this learner changes its hypothesis infinitely many times. This disproves the assumption and, thus, demonstrates $C^{\leq 2}_{\textit{w-q-like}} \notin NUM^{\infty}_{Xpart}$. The proof is completed.

All the separations of problem classes on display in figure 5, be it proper set inclusions or incomparabilities, can be demonstrated by means of suitable benchmark classes as exemplified above.

8 Summary & Conclusions

Due to its simplicity, to the clarity, and to the apparent relevance of a few key results, early work in inductive inference such as [Gol 1967] attracted an enormous attention in cognitive science [Joh 2004].

In contrast to Gold's investigation of learning formal languages, the learning of computable functions is even simpler and special cases such as polynomial interpolation are already well-understood for more than a century [Run 1901].

The theory of recursive functions and effective computability [Rj 1967] provides a collection of deep and sometimes seemingly abstruse results such as the recursion theorem, Rice's theorem, and certain consequences thereof such as the existence of self-describing functions named "quines" in software engineering. The latter phenomenon has challenged software practitioners who found out that all the deep theoretical results really have corresponding phenomena in software engineering practice [Tho 1999].

The above-mentioned state of the art encouraged the authors of the present report to map their *problems of learning theories of mind from observations* of human-computer interaction to the much simpler field of *learning general recursive functions.* The focus of the authors' investigations is on a single (!) inductive learning mechanism that solves all their learning problems provided there are spaces of hypotheses generated and presented appropriately. This novel principle of technology may be easily mapped to inductive inference of recursive functions.

8.1 Theory

The manuscript introduces the novel concepts NUM^*_{part} and NUM^*_{Xpart} and contributes two more infinite hierarchies on display in figure 5 on the right.

The key idea motivated by the authors' application projects (see also section 8.2) consists in focussing only a single (!) simple learning mechanism which is, so to speak, parametrized. The learning mechanism's parameters are spaces of hypotheses. The theoretical approach has been studied almost exactly 40 years ago in [Jan 1978] where the term *strategic operator* is coined and *identification by enumeration* as dealt with in the present report occurs as a specific case of a *consistent strategic operator* (ibid., page 489). To paraphrase the authors' intention, they aim at *uniform learning* [Zil 2003].

This work relates to [Wie 1991] and [Köt 2014]. Wiehagen expresses the belief that every problem of inductive inference can be solved by "an enumerative inference strategy" ([Wie 1991], p. 198), though the concept of an *enumerative inference strategy* remains unspecified (p. 199). [Köt 2014] provides a solution to Wiehagen's thesis for a variety of criteria. Kötzing's approach (ibid., Definition 2, p. 499) is weaker than the present one, because we are focussing a *single* learner $\mathcal{L}^{IdbyEn}$ running on varying enumerations, even varying in the course of solving a single learning problem. Kötzing, in contrast, takes varying learners into account, but requires a unique underlying enumeration.

8.2 Practice

The authors' papers on practical experiments in the areas of player modeling [JSS 2016] and data analysis [ADFJ 2017] are continued by the introduction and investigation of NUM^* [ADF$^+$ 2017] and of the infinite hierarchy exhausting the gap from NUM to NUM^* [AJ 2018].

In the latter one, which is a book chapter on data mining, the authors express the opinion that it should be possible "to generalize their recent approach to dynamic identification by enumeration even further. This requires a careful easing of one or more of the requirements named operational appropriateness, conversational appropriateness, and semantic appropriateness. The related open questions need some more research effort." (ibid., p. 61). With the results in the present report, the work is done.

The results explicate that – in practical applications – theory of mind induction does not so much *depend* on learning theorists and their sophisticated algorithms of high complexity, but *on domain experts and their ability to express what they are looking for*, i.e., spaces of hypotheses.

The authors' conference paper [ADFJ 2017] provides the discussion of an application case in business intelligence. Chapter 3 of [Jan 2016a] contains an even more detailed discussion supported by an appendix of 9 session screenshots.

Based on the recent results, the authors are able improve the system's intelligence by putting more emphasis on the internal generation of logical expressions that formalize domain knowledge and serve as spaces of hypotheses.

Future applications will teach us how to interpret human users' actions in human-computer interactive problem solving toward the generation of useful internal knowledge representations. A variety of challenges will occur such as, e.g., the usage of ontologies and of other knowledge sources for the ad hoc generation of system knowledge.

References

[ADF+ 2017] Oksana Arnold, Sebastian Drefahl, Jun Fujima, Klaus P. Jantke, and Christoph Vogler. Dynamic identification by enumeration for co-operative knowledge discovery. *IADIS Intl. J. on Computer Science and Information Systems*, 12(2):65–85, 2017.

[ADFJ 2017] Oksana Arnold, Sebastian Drefahl, Jun Fujima, and Klaus P. Jantke. Co-operative knowledge discovery based on meme media, natural language processing and theory of mind modeling and induction. In Piet Kommers and Pedro Isaias, editors, *Proc. 15th Intl. Conf. e-Society 2017*, pages 27–38. IADIS Press, 2017.

[AJ 2018] Oksana Arnold and Klaus P. Jantke. Mining HCI Data for Theory of Mind Induction. In Ciza Thomas, editor, *Data Mining*, chapter 4, pages 47–68. IntechOpen, 2018.

[AS 1983] Dana Angluin and Carl H. Smith. Inductive inference: Theory and methods. *ACM Comput. Surveys*, 15(3):237–269, 1983.

[BD 1987] George E. P. Box and Norman R. Draper. *Empirical Model-Building and Response Surfaces.* Hoboken, NJ, USA: John Wiley & Sons, 1987.

[CS 1996] Peter Carruthers and Peter K. Smith, editors. *Theories of Theories of Mind.* Cambridge University Press, 1996.

[Daw 1976] Richard Dawkins. *The Selfish Gene.* Oxford University Press, 1976.

[EC 2009] Nathan J. Emery and Nicola S. Clayton. Comparative social cognition. *Annual Review of Psychology*, 60:87–113, 2009.

[EDC 2004] Nathan J. Emery, Joanna M. Dally, and Nicola S. Clayton. Western scrub-jays (Aphelocoma californica) use cognitive strategies to protect their caches from thieving conspecifics. *Animal Cognition*, 7:37–43, 2004.

[Gol 1967] E. Mark Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.

[Gri 2008] Gunter Grieser. Reflective inductive inference of recursive functions. *Theoretical Computer Science*, 397:57–69, 2008.

[Jan 1978] Klaus P. Jantke. Universal methods for identification of total recursive functions. In *Rostocker Math. Kolloq.*, pages 63–69, 1978.

[Jan 2010] Klaus P. Jantke. The Gorge approach. Digital game control and play for playfully developing technology competence. In José Cordeiro, Boris Shishkov, Alexander Verbraeck, and Markus Helfert, editors, *CSEDU 2010. 2nd International Conference on computer Supported Education, Proc., Vol. 1, Valencia, Spain, April 7-10, 2010*, pages 411–414. INSTICC, 2010.

[Jan 2011] Klaus P. Jantke. Interdisciplinary advances in adaptive and intelligent assistant systems – foreword. In Gunther Kreuzberger, Aran Lunzer, and Roland Kaschek, editors, *Interdisciplinary Advances in Adaptive and Intelligent Assistant Systems. Concepts, Techniques, Applications, and Use*, pages vii–viii. Hershey, PA: Information Science Reference, 2011.

[Jan 2012a] Klaus P. Jantke. Theory of Mind Induction in User Modeling: An Introductory Game Case Study. Report KiMeRe-2012-06, Fraunhofer IDMT, Children's Media Dept., December 2012.

[Jan 2012b] Klaus P. Jantke. User Modeling with Theories of Mind: An Introductory Game Case Study. Report KiMeRe-2012-05, Fraunhofer IDMT, Children's Media Dept., November 2012.

[Jan 2013] Klaus P. Jantke. Digitale Assistenten für Schüler und Lehrer. In Andreas Jantowski, editor, *Thillm.2013 – Gemeinsam leben. Miteinander lernen.*, number 58 in Impulse, pages 264–274. Bad Berka: Thillm, 2013.

[Jan 2016a] Klaus P. Jantke. Theory of mind modeling and induction: Ausdrucksfähigkeit und Reichweite. ADISY Technical Report 03/2016, ADISY Consulting, Weimar, Germany, November 2016.

[Jan 2016b] Klaus P. Jantke. Theory of mind modeling and induction: Touch & feel. ADISY Technical Report 01/2016, ADISY Consulting, Weimar, Germany, September 2016.

[Jan 2018] Klaus P. Jantke. Mobile Learning – vom Werkzeug zum Assistenten. In Claudia de Witt and Christina Gloerfeld, editors, *Hanbuch Mobile Learning*, chapter 10, pages 195–218. Springer, 2018.

[JB 1981] Klaus P. Jantke and Hans-Rainer Beick. Combining postulates of naturalness in inductive inference. *EIK*, 17(8/9):465–484, 1981.

[JGL 2003] Klaus P. Jantke, Gunter Grieser, and Steffen Lange. Adaptivität – Vom Werkzeug zum Assistenten: Risiken der Autonomie. In Rüdiger Grimm, editor, *Proc. E-Learning: Beherrschbarkeit und Sicherheit, 1./2. Juli 2003, Ilmenau*, pages 25–41. TU Ilmenau, Institut für Medien- und Kommunikationswissenschaft, 2003.

[JHLN 2010] Klaus P. Jantke, Imke Hoppe, Denise Lengyel, and Alexandra Neumann. Time to play Gorge – Time to learn AI: A qualitative study. In Sybille Hambach, Alke Martens, Djamshid Tavangarian, and Udo Urban, editors, *eLearning Baltics 2010, Proc. 3rd Intl. eLBa Science Conference*, pages 99–110. Fraunhofer Verlag, 2010.

[JIS 2005] Klaus P. Jantke, Christoph Igel, and Roberta Sturm. Learner modeling and adaptive behavior toward a paradigmatic shift from e-learning tools to assistants. In S. Hartmann, R. Kaschek, Kinshuk, K.-D. Schewe, J.M. Turull Torres, and R. Whiddett, editors, *First International Workshop on Perspectives of Intelligent Systems' Assistance, PISA 2005, Palmerston North, New Zealand, March 3-5, 2005*, pages 60–71. Massey University, Dept. Information Systems, 2005.

[JIS 2007] Klaus P. Jantke, Christoph Igel, and Roberta Sturm. From e-learning tools to assistants by learner modelling and adaptive behaviour. In Roland H. Kaschek, editor, *Perspectives of Intelligent Systems' Assistance*, pages 212–231. Idea Group, 2007.

[JM 2005] Klaus P. Jantke and Carsten Müller. Wrapper induction programs as information extraction assistants. In S. Hartmann, R. Kaschek, Kinshuk, K.-D. Schewe, J.M. Turull Torres, and R. Whiddett, editors, *First International Workshop on Perspectives of Intelligent Systems' Assistance, PISA 2005, Palmerston North, New Zealand, March 3-5, 2005*, pages 86–101. Massey University, Dept. Information Systems, 2005.

[JM 2007] Klaus P. Jantke and Carsten Müller. Wrapper induction programs as information extraction assistants. In Roland H. Kaschek, editor, *Perspectives of Intelligent Systems' Assistance*, pages 35–63. Idea Group, 2007.

[Joh 2004] Kent Johnson. Gold's theorem and cognitive science. *Philosophy of Science*, 71:571–592, 2004.

[JORS 1999] Sanjay Jain, Daniel N. Osherson, James S. Royer, and Arun Sharma. *Systems That Learn.* Cambridge, MA, USA: MIT Press, 1999.

[JSS 2016] Klaus P. Jantke, Bernd Schmidt, and Rosalie Schnappauf. Next generation learner modeling by theory of mind model induction. In *Proc. 8th Intl. Conf. Computer Supported Education, CSEDU 2016, Rome, Italy, April 21-23, 2016, Vol. 1*, pages 499–506. SCITEPRESS, 2016.

[JSSB 2005] Klaus P. Jantke, Martin Steffens, Kevin Steffgen, and Jochen F. Böhm. The light assistants approach for e-learning in continuing education. In S. Hartmann, R. Kaschek, Kinshuk, K.-D. Schewe, J.M. Turull Torres, and R. Whiddett, editors, *First International Workshop on Perspectives of Intelligent Systems' Assistance, PISA 2005, Palmerston North, New Zealand, March 3-5, 2005*, pages 102–113. Massey University, Dept. Information Systems, 2005.

[Köt 2014] Timo Kötzing. A solution to Wiehagen's thesis. In Ernst W. Mayr and Natacha Portier, editors, *31st Intl. Symp. Theoretical Aspects of Computer Science (STACS 2014)*, volume 25 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 494–505. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2014.

[Lak 1987] George Lakoff. *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind.* Univ. Chicago Press, 1987.

[Pop 1934] Karl Popper. *Logik der Forschung.* Tübingen, 1934.

[Rj 1967] Hartley Rogers jr. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, 1967.

[Run 1901] Carl Runge. Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten. *Zeitschrift f. Mathematik und Physik*, 46:224–243, 1901.

[Tan 2003] Yuzuru Tanaka. *Meme Media and Meme Market Architectures: Knowledge Media for Editing, Distributing and Managing Intellectual Resources.* IEEE Press & Wiley-Interscience, 2003.

[Tho 1999] Gary P. Thompson II. The Quine Page. http://www.nyx. net/~gthompso/quine.htm [visited on October 31, 2018], 1999.

[Wie 1991] Rolf Wiehagen. A thesis in inductive inference. In Klaus P. Jantke, Jürgen Dix, and Peter H. Schmitt, editors, *Proc. 1st Intl. Workshop on Nonmonotonic and Inductive Logic, Karlsruhe, Germany, December 1990*, volume 543 of *Lecture Notes in Artificial Intelligence*, pages 184–207. Springer, 1991.

[Zil 2003] Sandra Zilles. *Uniform Learning of Recursive Functions.* Number 278 in DISKI. Berlin: Akademische Verlagsgesellschaft AKA, 2003.

Appendix A: The Benchmark Class $C^{\leq 2}_{\textit{w-q-like}}$

Readers less familiar with the *theory of recursive functions and effective computability* [Rj 1967] might have their doubts about constructions such as $C^{\leq 2}_{\textit{w-q-like}}$. This appendix provides a proof that this benchmark class does really exist and that it contains infinitely many functions.

The following simple construction works for every $n \in I\!N$ seen as a potential first indicator and, consequently, results in infinitely many elements of $C^{\leq 2}_{\textit{w-q-like}}$.

$$\chi^y_1(x) = \begin{cases} 1 & \textit{if} \quad x < n \\ 0 & \textit{if} \quad x = n \\ y & \textit{if} \quad x = n+1 \\ \alpha(x) & \textit{otherwise} \end{cases}$$

In this definition, α is an arbitrarily fixed total recursive function that computes exclusively values greater than 0. The choice of α allows for infinitely many variations of this approach.

When α is fixed (for simplicity, think of α as the one function that is constantly equal to 1), the definition of χ^y_1 does effectively depend on y only.

By Kleene's s-m-n theorem, there exists an effective enumeration $h_1 \in \mathcal{R}^1$ such that it holds $\varphi_{h_1(y)}(x) = \chi^y_1(x)$ for all $x, y \in I\!N$.

By the recursion theorem, h_1 has a fixed point z – in fact, there are always infinitely many fixed points; another aspect that reveals the existence of infinitely many variations – satisfying the equality $\varphi_{h_1(z)} = \varphi_z$. Recall that $\varphi_{h_1(z)} = \chi_z$ implies $\chi^z_1 = \varphi_z$.

Consequently, for any function α and for any fixed point z, the point n is an indicator within χ^z_1 and it holds $\chi^z_1 \in C^1_{\textit{w-q-like}} \subset C^{\leq 2}_{\textit{w-q-like}}$. Furthermore, χ^z_1 is a skeleton function and all its finite variations past point $n+1$ belong to the class of interest.

Appendix B: On Extensions of $\psi^y[k]$

There is a critical discussion in section 7.2 by the end of page 17. The function ψ^y under construction is defined up to some point k. The initial segment $\psi^y[k]$ is given. The construction proceeds by assigning a value for the argument $k+3$. For this purpose, it is assumed that the learner builds a hypothesis on potential observations such as $\psi^y[k] \circ (k+3, 1)$ and $\psi^y[k] \circ (k+3, 2)$. This assumption needs a justification.

Based on $\psi^y[k]$, one may define two sequences of functions $\{\chi^z_1\}_{z \in I\!N}$ and $\{\chi^z_2\}_{z \in I\!N}$ both extending $\psi^y[k]$, i.e. $\chi^z_1[k] = \chi^z_2[k] = \psi^y[k]$ as follows. One defines (ii) $\chi^z_1(k+1) = \chi^z_2(k+1) = 0$, and (iii) $\chi^z_1(k+2) = \chi^z_2(k+2) = z$. For values equal to or larger than $k+3$, both function sequences are distinguished. All functions $\{\chi^z_1\}_{z \in I\!N}$ become constantly equal to 1 whereas the functions $\{\chi^z_2\}_{z \in I\!N}$ are constantly equal to the value 2.

Because both constructions are effective and due to Kleene' s-m-n theorem, there are compilers h_1 and h_2 – for brevity, we use the notation h_b where b means 1 or 2, resp. – satisfying $\varphi_{h_b(z)} = \chi^z_b$. According to the recursion theorem, h_b has a fixed point z_b such that it holds $\varphi_{h_b(z_b)} = \varphi_{z_b}$. For every fixed point, the function $\varphi_{z_b} = \chi^{z_b}_b$ has a second indicator at point $k+1$ and its value at point $k+2$ is a correct index. Consequently, for all fixed points z_1 and z_2, the functions $\chi^{z_1}_1$ and $\chi^{z_2}_b 2$, resp., belong to the benchmark class $C^{\leq 2}_{\textit{w-q-like}}$. At point $k+3$, the values 1 and 2, resp., occur.

This implies that a learner for the benchmark class under consideration must be defined both on $\psi^y[k] \circ (k+3, 1)$ and $\psi^y[k] \circ (k+3, 2)$. This completes the necessary justification.