Juris Smotrovs

Promocijas darbs

Slēgtības tipa īpašības induktīvajā izvedumā

Closedness Properties in Inductive Inference

Rīga, 1999

Anotācija

Šajā darbā tiek pētītas visur definēto rekursīvo funkciju un valodu induktīvā izveduma teorijā labi pazīstamu identifikācijas tipu īpašības, kas atgādina kopu slēgtību. Konkrētāk, aplūkotas tiek šādas īpašības: ja kādām n klasēm visi apvienojumi pa $n-1$ ir identificējami, tad arī visu n klašu apvienojums ir identificējams. Izrādās, ka šīs īpašības ir noteicošās, lai izšķirtu, kādi nosacījumi, kas uzlikti klašu apvienojumu identificējamībai, ir apmierināmi un kādi nav. Lielākajai daļai aplūkoto identifikācijas tipu šajā darbā ir konstatēts, pie kādiem n minētā īpašība ir spēkā, kā arī atsevišķos gadījumos ir izpētīts, pie kādiem n līdzīga īpašība ir spēkā diviem dažādiem identifikācijas tipiem. Darbā pētītie identifikācijas tipi ir dažādas modifikācijas identifikācijai robežā, ko definējis E. M. Golds.

Abstract

This work investigates properties resembling closedness that are characteristic to some well known identification types in inductive inference of total recursive functions and in language learning. The properties can be formulated as follows: if every union of $n-1$ classes out of n classes is identifiable, then the union of n classes is identifiable, too. It turns out that these properties are crucial for establishing which sets of requirements put on the identifiability of unions of classes are satisfiable and which are not. This work solves the problem of finding out for which n the mentioned property holds for most of the considered identification types, and in some cases: for which n a similar property holds for two different identification types. The identification types involved are different modifications of identification in the limit introduced by E. M. Gold.

Аннотация

В этой работе исследуются свойства напоминающие замкнутость, характерные некоторым общеизвестным идентификационным типам в теории индуктивного вывода рекурсивных функций и языков. Рассматриваемые свойства имеют следующий вид: если каждое объединение $n-1$ классов из n классов идентифицируемо, то и объединение всех n классов идентифицируемо. Как оказывается, свойства этого рода определяют, какие требования к идентифицируемости объединений классов удовлетворяемы, а какие — нет. В этой работе для большинства рассмотренных идентификационных типов решена проблема нахождения n, для которых упомянутое свойство имеет силу, а также в некоторых случаях найдены n аналогичного свойства для двух различных идентификационных типов. Рассмотренные идентификационные типы являются модификациями идентификации в пределе, введённой Э. М. Голдом.

Acknowledgements

Many people have contributed to the successful completion of this research. Firstly, many thanks to my supervisor, professor Rūsiņš Freivalds, whose energy and inexhaustible helpfulness in both scientific and unscientific matters have resulted in the success of many Latvian scientists.

Thanks to my coauthors Kalvis Apsītis, Rūsiņš Freivalds, Mārtiņš Kriķis and Raimonds Simanovskis, together with whom this research was begun and whose many ideas were further developed in this work, and to Andris Ambainis, cooperation with whom resulted in one of the theorems presented here.

This thesis has also benefited from discussions and comments by Richard Bonner, John Case, Kārlis Čerāns, Helger Lipmaa and Rolf Wiehagen as well as by the anonymous referees of the *Journal of Theoretical Computer Science* and at various conferences.

This research was supported by the Latvian Science Council Grants 93.599, 96.0282, Grant for Doctoral Studies 969, and by the fellowship "SWH izglītībai, zinātnei un kultūrai" of the Latvian Education Foundation. The results of this thesis were presented at the conferences AII'92, ICGI'96, EuroCOLT'97 and ALT'98. Thanks to the organizers of these conferences and to the IFIP Working Group 1.4 for providing the financial support for participating in these events.

Thanks to Dainis Kriķis and Agnis Andžāns for the love of mathematics, and my family for more than that. Finally, I express my gratitude to the colleagues, friends, relatives, all the people with whom this world is much brighter.

Contents

Chapter 1

Introduction

This work deals with a problem in inductive inference of recursive functions and languages. Inductive inference as a term for finding out an algorithm from sample computations was the first time used by E. M. Gold in [22].

E. M. Gold in [22] introduced the paradigm of identification in the limit: the identifying strategy receives more and more input data about the object to be learned and outputs a sequence of hypotheses about it (usually the hypotheses are algorithms for the identified object). Beginning with some place the strategy outputs only one and the same correct hypothesis. According to the current notation, this identification type is called **Ex** in case total recursive functions are the objects of learning, and **TxtEx** in case recursively enumerable languages are to be identified. E. M. Gold also proved that **TxtEx** is not closed under the set union: there are two **TxtEx**-identifiable language classes $\mathcal{L}_1$ and $\mathcal{L}_2$ such that $\mathcal{L}_1 \cup \mathcal{L}_2$ is not **TxtEx**-identifiable. Later, a similar non-union theorem for **Ex** was proved independently by J. Bārzdiņš in [8] and by L. Blum and M. Blum in [10].

Since then many identification types have been proposed as modifications or alternatives to the Gold's learning paradigm, such as *prediction* [9], *behaviourally correct* [8], *probabilistic* [15], and *consistent identification* [37], *co-learning* [18], *identification of minimal Gödel numbers* [16].

And always one of the first questions that arise after introducing a new identification type is: "Is it closed under the operation of set union?" Currently this problem is solved for most if not for all of the known identification types. So the problem of closedness of identification types seems to be closed. Nevertheless ...

Suppose some identification type, similarly as **Ex** or **TxtEx**, is not closed. Can we impose arbitrary requirements on the identifiablity of the classes and their unions and still find some classes that satisfy these requirements? Most of the identification types have a property: if some class is identifiable, then all its subsets are identifiable, too. So the requirements "$U_1 \cup U_2$ is identifiable, U_1 and U_2 are not identifiable" would be unsatisfiable. But if this property is obeyed? Suppose we have requirements "U_1, U_2, U_3, $U_1 \cup U_2$, $U_1 \cup U_3$, $U_2 \cup U_3$ are identifiable,

$U_1 \cup U_2 \cup U_3$ is not identifiable." Does the satisfiability of these requirements follow from the non-union theorem? It turns out that no, it does not. It was proved in [4] that the mentioned requirements cannot be obeyed for **Ex**. The property of **Ex** that implies the unsatisfiability of them is this: if the unions of two classes out of three are **Ex**-identifiable, then the union $U_1 \cup U_2 \cup U_3$ is **Ex**-identifiable, too. So **Ex** still has some property that resembles closedness! Moreover, this property allows us to distinguish between satisfiable and unsatisfiable requirements. In [4] such results were proved also for the identification types $\mathbf{Ex}_b$, where b denotes a bound on mindchanges (see [20] and [12]), and $\mathbf{Ex}_0^1$, where the superscript 1 denotes a bound on anomalies (see [10] and [12]). In this work this problem is solved for the general case of $\mathbf{Ex}_b^a$-identification of total recursive functions and $\mathbf{TxtEx}_b^a$-identification of recursively enumerable languages (the identification of partial recursive functions is equivalent to the identification of recursively enumerable languages, and usually is investigated in this form, so we complied with the tradition in this work).

To solve the problem, we extend the notion of closedness. We say that an identification type is n-closed if the identifiability of all unions of $n-1$ classes out of n classes implies the identifiability of the union of all n classes. We show that finding the minimal such n is sufficient for solving the problem of satisfiability of requirements for a very broad class of identification types.

It can be easily proved that, if $U_1, U_2 \in \mathbf{Ex}_0$, then $U_1 \cup U_2$ quite possibly is not in $\mathbf{Ex}_0$ and maybe even not in $\mathbf{Ex}_1$, but it always is in $\mathbf{Ex}_2$. Indeed, why should we necessarily require that the union is identifiable with the same mindchange (anomalies, ...) complexity? Often it would be sufficient that we identify the union with a larger, but still finite and estimable amount of mindchanges. So another generalization seems natural, the n-closedness in superset: if all the unions of $n-1$ classes out of n classes are identifiable in some sense, then the union of all n classes is identifiable in another, more general sense. As we shall also see, most of the results of n-closedness can be more easily proved with the help of this notion. The closedness in superset yields many interrelationships between different identification types. This work contains some results on n-closedness of $\mathbf{Ex}_b^a$ in $\mathbf{Ex}_d^c$.

Interestingly, n-closedness can be formulated in terms of team learning ([24] is a good survey on the team learning of both total recursive functions and recursively enumerable languages), so this work yields some new results in this area. Also, investigation of the n-closedness of the team learning identification types $[k,l]\mathbf{Ex}_b^a$ yields some interesting results. One section in this work is dedicated to these results.

The structure of this work is as follows. In Chapter 2 we introduce notation, define the identification in the limit of total recursive functions and recursively enumerable languages with bounds on mindchanges and anomalies, probabilistic and team learning as well as define the notion of identification type in general which we use in this work.

In Chapter 3 we define n-closedness and the closedness degree and formulate an equivalent problem in team learning. We also show how the closedness degree affects the power of team and probabilistic identification types with a high success ratio (probability); these results were obtained together with K. Apsītis. At the end of the chapter we prove that the closedness degree allows us to solve the problem of satisfiability of requirements for practically all "natural" identification types.

The next chapters contain the results on the closedness degrees for particular identification types. Chapter 4 considers the case when total recursive functions are identified. At first we find the closedness degrees of the identification types $\mathbf{Ex}_b^a$ (these results appeared in [4, 6, 7], they were coauthored with K. Apsītis, R. Freivalds, M. Kriķis and R. Simanovskis), then we investigate the closedness in superset among them, and finally we find the closedness degrees for some of the corresponding team learning identification types (these results appeared in [34]). We also notice that, if we do not consider anomalies, then the closedness degree in all the proved cases turns out to be finite, and we give a proof of such finiteness for a class of team identification types (this proof was obtained together with A. Ambainis).

Chapter 5 contains results on n-closedness in language learning. The results of this chapter have appeared in [5, 7], and they were coauthored with K. Apsītis, R. Freivalds and R. Simanovskis.

Chapter 6 summarizes the obtained results.

Chapter 2

Preliminaries

2.1 Notation

Any recursion theoretic notation not explained below is from [30]. $\mathbb{N}$ denotes the set of natural numbers, $\{0, 1, 2, \ldots\}$. $*$ denotes "an arbitrary finite (natural) number." In inequalities $(\forall n \in \mathbb{N})[n < * < \infty]$. $\forall^\infty$ means "for all but finitely many." $\exists^\infty$ means "there exist infinitely many." $\subset$ denotes proper subset. $\langle \cdot, \ldots, \cdot \rangle$ denotes a computable one-to-one numbering of all the tuples of natural numbers.

Let $\mathcal{R}$ denote the set of total recursivr functions of one argument and $\mathcal{P}$ the set of partial recursive functions of one argument. We fix a Gödel numbering of $\mathcal{P}$ and denote it by φ. The function computed by the program i we denote by φ_i. Its domain W_i is the recursively enumerable language accepted by φ_i. Let $\mathcal{E}$ denote the set of recursively enumerable languages.

If $f(x)$ is undefined, we write $f(x) \uparrow$. By $f(x) \downarrow = y$ we mean that $f(x)$ is defined and equal to y, $f(x) \downarrow$ means that $f(x)$ is defined. If $f, g \in \mathcal{P}$, $a \in \mathbb{N} \cup \{*\}$, then $f =^a g$ denotes the fact that $\text{card}(\{x \in \mathbb{N} \mid (f(x) \downarrow \wedge g(x) \downarrow \wedge f(x) \neq g(x)) \vee (f(x) \uparrow \wedge g(x) \downarrow) \vee (f(x) \downarrow \wedge g(x) \uparrow)\}) \leq a$. For $L_1, L_2 \in \mathcal{E}$, $a \in \mathbb{N} \cup \{*\}$, by $L_1 =^a L_2$ we mean that $card((L_1 - L_2) \cup (L_2 - L_1)) \leq a$. In both cases the up to a differences are called *anomalies.* If $f \in \mathcal{R}$, $n \in \mathbb{N}$, we define $f^{[n]} = \langle f(0), \ldots, f(n) \rangle$.

We shall consider finite and infinite sequences with values from $\mathbb{N} \cup \{\#\}$, where # means "no data." The length of a finite sequence σ is denoted by $|\sigma|$. For a sequence σ, the initial sequence of length n ($n \leq |\sigma|$ if σ is finite) is denoted by $\sigma[n]$. The *content* of a sequence σ is the set of natural numbers in the range of σ, denoted content(σ). An infinite sequence T is a text for a language L iff content$(T) = L$. We fix some computable one-to-one encoding of the finite sequences of this kind by natural numbers. The code of a sequence σ is denoted by $\overline{\sigma}$. $\sigma \subseteq \tau$ means that τ is an extension of σ, $\sigma \subset \tau$ means that τ is a proper extension of σ, $\sigma\tau$ denotes concatenation of the sequences (in the last two cases

σ must be finite). In concatenations α^n denotes the element α repeated n times.

The operator of assignment in descriptions of algorithms will be denoted as $\leftarrow$.

2.2 Identification of Recursive Functions

An identification strategy F is an arbitrary partial recursive function. It receives as input $f^{[n]}$ — the initial segment of the target function $f \in \mathcal{R}$. We shall refer to its output $F(f^{[n]})$ as *a hypothesis* on the function f. *A mindchange* is an event when $F(f^{[n]})$ and $F(f^{[n+1]})$ are both defined and different.

Definition 2.1 [22, 10, 20, 12] *Let* $a, b \in \mathbb{N} \cup \{*\}$. *A strategy* F $\mathbf{Ex}_b^a$-identifies *a function* $f \in \mathcal{R}$ *(*$f \in \mathbf{Ex}_b^a(F)$*) iff:*

1. $(\exists N)[(\forall n < N)[F(f^{[n]}) \uparrow] \wedge (\forall n \geq N)[F(f^{[n]}) \downarrow]]$;
2. $(\exists h)[(\forall^\infty n)[F(f^{[n]}) \downarrow= h] \wedge \varphi_h =^a f]$;
3. *the number of mindchanges made by* F *on* f *does not exceed* b.

Definition 2.2 [22, 10, 20, 12] *A class* $U \subseteq \mathcal{R}$ *is* $\mathbf{Ex}_b^a$-identifiable *(*$U \in \mathbf{Ex}_b^a$*) iff* $(\exists F \in \mathcal{P})[U \subseteq \mathbf{Ex}_b^a(F)]$.

The following relationship has been established between these identification types.

Theorem 1 [12] $(\forall a, b, c, d \in \mathbb{N} \cup \{*\})[\mathbf{Ex}_b^a \subseteq \mathbf{Ex}_d^c \Leftrightarrow a \leq c \wedge b \leq d]$.

2.3 Identification of Languages

A language identification strategy F is an arbitrary partial recursive function. It receives as input $\overline{T[n]}$ — the initial segment of a text T for the target language $L \in \mathcal{E}$. Note that there are infinitely many texts for any non-empty language. *A mindchange* is an event when $F(\overline{T[n]})$ and $F(\overline{T[n+1]})$ are both defined and different.

Definition 2.3 [22, 11, 27] *Let* $a, b \in \mathbb{N} \cup \{*\}$. *A strategy* F $\mathbf{TxtEx}_b^a$-identifies *a language* $L \in \mathcal{E}$ *(*$L \in \mathbf{TxtEx}_b^a(F)$*) iff for every text* T *for* L*:*

1. $(\exists N)[(\forall n < N)[F(\overline{T[n]}) \uparrow] \wedge (\forall n \geq N)[F(\overline{T[n]}) \downarrow]]$;
2. $(\exists h)[(\forall^\infty n)[F(\overline{T[n]}) = h] \wedge \varphi_h =^a f]$;
3. *the number of mindchanges made by* F *on* T *does not exceed* b.

Definition 2.4 [22, 11, 27] *A family of languages $\mathcal{L} \subseteq \mathcal{E}$ is* $\mathbf{TxtEx}^a_b$-identifiable *($\mathcal{L} \in \mathbf{TxtEx}^a_b$) iff $(\exists F)[\mathcal{L} \subseteq \mathbf{TxtEx}^a_b(F)]$.*

We sometimes omit the index a if $a = 0$ and b if $b = *$. Particularly, $\mathbf{TxtEx} = \mathbf{TxtEx}^0_*$.

The following basic relationship has been established between the defined identification types.

Theorem 2 [11, 27] $(\forall a, b, c, d \in \mathbb{N} \cup \{*\})[\mathbf{TxtEx}^a_b \subseteq \mathbf{TxtEx}^c_d \Leftrightarrow a \leq c \wedge b \leq d]$.

2.4 Probabilistic Identification

Probabilistic inductive inference was defined in [15, 28]. Informally, it allows the strategy to make probabilistic choices, for instance, to toss a coin.

Definition 2.5 [15, 28] *Let $\mathcal{I}$ be one of the identification types $\mathbf{Ex}^a_b$, $\mathbf{TxtEx}^a_b$ defined above. A probabilistic strategy F* identifies a class U with probability p according to the identification type $\mathcal{I}$ *($\langle p\rangle\mathcal{I}$-identifies U) iff for each $f \in U$: the probability that F $\mathcal{I}$-identifies f is at least p. Then we write $U \subseteq \langle p\rangle\mathcal{I}(F)$ and $U \in \langle p\rangle\mathcal{I}$.*

For a more formal and detailed definition with the basic proofs see [15, 28]. Actually, usually it is easy to define the probabilistic type $\langle p\rangle\mathcal{I}$ for any identification type $\mathcal{I}$, it only needs to be checked that the probability of successful identification is measurable.

2.5 Identification Types

In general, we define *an identification type* by the following scheme.

1. $\mathcal{I}$-identification is defined as a mapping $\mathcal{M} \to P(\mathcal{A})$, where $\mathcal{M}$ is the set of the subjects performing identification (in this work, the set of deterministic or probabilistic strategies or the set of teams of strategies), $\mathcal{A}$ is the set of objects to be identified (for instance, $\mathcal{A} = \mathcal{R}$ or $\mathcal{A} = \mathcal{E}$), and $P(\mathcal{A})$ is the set of all the subsets of $\mathcal{A}$; $\mathcal{I}(M)$ is the set of all the objects identified by $M \in \mathcal{M}$;

2. a class $U \subseteq \mathcal{A}$ is considered $\mathcal{I}$-identifiable iff $(\exists M \in \mathcal{M})[U \subseteq \mathcal{I}(M)]$;

3. the identification type is characterized by the set $\mathcal{I} = \{U \subseteq \mathcal{A} \mid U$ is $\mathcal{I}$-identifiable$\}$.

This definition takes into account only the set theoretical aspects of identification types, not the learning theoretical aspects. But we shall need mostly these aspects when speaking about identification types in general.

2.6 Team Learning

Team learning was suggested by Case and first investigated by Smith [31]. The general definition is due to [26]. According to this model, many strategies participate in the identification, and we require only a certain amount of them to be successful.

Definition 2.6 *Let $\mathcal{I}$ be an identification type. $U \subseteq \mathcal{R}$ is $\mathcal{I}$-identifiable by a team "k out of l" (we write $U \in [k,l]\mathcal{I}$, $1 \leq k \leq l$) iff there is a "team" of l strategies such that every function from U is $\mathcal{I}$-identified by at least k of these strategies.*

As we see, team learning allows to build new identification types from the existing.

Chapter 3

Closedness and Identification

Here we establish the basic properties of n-closedness and connections with the defined identification types.

3.1 n-Closedness

Definition 3.1 *Let A be a set with an associative and commutative binary operation $\circ$: $A \times A \to A$ defined in it, and let there be an element $e \in A$ equal to the empty $\circ$-product. A set $S_1 \subseteq A$ is* n-closed in a set $S_2 \subseteq A$ *($n \geq 1$) with respect to $\circ$ iff*

$$(\forall a_1, \ldots, a_n \in A)$$
$$[(\forall i \mid (1 \leq i \leq n))[a_1 \circ \ldots \circ a_{i-1} \circ a_{i+1} \circ \ldots \circ a_n \in S_1] \Rightarrow a_1 \circ \ldots \circ a_n \in S_2].$$

Definition 3.2 *Let $\circ$ be an associative and commutative binary operation: $A \times A \to A$, and let there be an element $e \in A$ equal to the empty $\circ$-product. A set $S \subseteq A$ is* n-closed *($n \geq 1$) with respect to $\circ$ iff S is n-closed in S.*

So "2-closed" is the same as "closed." In further the binary operation will be set union, A will be some family of sets, closed with respect to the set union and containing the empty set, which will be the element e (it is needed only in the exceptional case $n = 1$). The following statements concerning n-closedness can be easily proved.

Proposition 3.1 *If a set family S_1 is n-closed in a set family S_2, then $S_1 \subseteq S_2$.*

Proof. Suppose, $U \in S_1$. Define $U_1 = \ldots = U_n = U$. Since S_1 is n-closed in S_2, we get that $\bigcup_{j=1}^{n} U_i = U \in S_2$. □

Corollary 3.1 *If $\mathbf{Ex}_b^a$ is n-closed in $\mathbf{Ex}_d^c$, then $a \leq c$ and $b \leq d$.*

Corollary 3.2 *If $\mathbf{TxtEx}_b^a$ is n-closed in $\mathbf{TxtEx}_d^c$, then $a \leq c$ and $b \leq d$.*

Proposition 3.2 *If S_2 is n-closed in S_3, $S_1 \subseteq S_2$ and $S_3 \subseteq S_4$, then S_1 is n-closed in S_4.*

Proposition 3.3 *Let S_1 be n-closed in S_2 and satisfy the property $(\forall U \in S_1)$ $(\forall V \subseteq U)$ $[V \in S_1]$. Then S_1 is m-closed in S_2 for all $m \geq n$.*

Proof. Suppose S_1 is n-closed in S_2 and satisfies the mentioned property. If $n = 1$, then we have that $\emptyset \in S_1$ implies $U \in S_2$ for any $U \in A$. So $S_2 = A$, and S_1 is m-closed in S_2 for all $m \geq 1$.

Suppose $1 < n \leq m$, and sets $U_1, \ldots, U_m \in S$ satisfy the property $(\forall i \mid 1 \leq i \leq m)[\bigcup_{j=1, j\neq i}^{m} U_j \in S_1]$. Define $V_1 = U_1, \ldots, V_{n-1} = U_{n-1}, V_n = \bigcup_{j=n}^{m} U_j$. We have $V_n \in S_1$ because $V_n \subseteq \bigcup_{j=2}^{m} U_j \in S_1$, and $\bigcup_{j=1}^{n-1} V_j \in S_1$ because $\bigcup_{j=1}^{n-1} V_j \subseteq \bigcup_{j=1}^{m-1} U_j \in S_1$. Thus, $(\forall i \mid 1 \leq i \leq n)[\bigcup_{j=1, j\neq i}^{n} V_j \in S_1]$. Since S_1 is n-closed in S_2, $\bigcup_{j=1}^{n} V_j = \bigcup_{j=1}^{m} U_j \in S_2$. □

Note that the identification types built according to the scheme described in Section 2.5 satisfy the mentioned property.

The next proposition is by K. Apsītis.

Proposition 3.4 *Let S be n-closed and satisfy the property $(\forall U \in S)$ $(\forall V \subseteq U)$ $[V \in S]$. Let U be a set that can be expressed as a finite union $U = U_1 \cup \ldots \cup U_m$ so that*

$$(\forall I \subseteq \{1, \ldots, m\}) \left(|I| \leq n-1 \Rightarrow \bigcup_{i \in I} U_i \in S \right).$$

Then $U \in S$.

Proof. If $m < n$, we are done, since then we can choose $I = \{1, \ldots, m\}$. If $m \geq n$, we fix an arbitrary $I' = \{i_1, \ldots, i_n\} \subseteq \{1, \ldots, m\}$, $|I'| = n$. Since S is n-closed, it follows from the assumption that $\bigcup_{i \in I'} U_i \in S$. Since we chose I' arbitrarily, we can now use a stronger assumption:

$$(\forall I \subseteq \{1, \ldots, m\}) \left(|I| \leq n \Rightarrow \bigcup_{i \in I} U_i \in S \right).$$

Repeat the above reasoning replacing n by $n+1$ (it follows from Proposition 3.3 that S is $(n+1)$-closed). By induction we obtain that $\bigcup_{i \in I} U_i \in S$ for larger and larger I until we reach $|I| = m$, and thus get that $\bigcup_{i \in I} U_i = U \in S$. □

Proposition 3.3 shows that we need to find the minimal n for which S_1 is n-closed in S_2.

Definition 3.3 *Let a set family S_1 satisfy the property $(\forall U \in S_1)(\forall V \subseteq U)[V \in S_1]$. We say that n is* the closedness degree *of S_1* in superset *S_2 ($n =$* csdeg(S_1, S_2)*) iff n is the smallest number such that S_1 is n-closed in S_2.*

If such n does not exist, we define csdeg$(S_1, S_2) = \infty$.

We shall call cdeg$(S) =$ csdeg(S, S) the closedness degree *of S.*

The next two propositions follow from Proposition 3.2, Theorem 1 and Theorem 2.

Proposition 3.5 *If* $a_1 \leq a_2$, $b_1 \leq b_2$, $c_1 \leq c_2$ *and* $d_1 \leq d_2$, *then*

$$\mathrm{csdeg}(\mathbf{Ex}_{b_2}^{a_2}, \mathbf{Ex}_{d_1}^{c_1}) \geq \mathrm{csdeg}(\mathbf{Ex}_{b_1}^{a_1}, \mathbf{Ex}_{d_2}^{c_2}).$$

Proposition 3.6 *If* $a_1 \leq a_2$, $b_1 \leq b_2$, $c_1 \leq c_2$ *and* $d_1 \leq d_2$, *then*

$$\mathrm{csdeg}(\mathbf{TxtEx}_{b_2}^{a_2}, \mathbf{TxtEx}_{d_1}^{c_1}) \geq \mathrm{csdeg}(\mathbf{TxtEx}_{b_1}^{a_1}, \mathbf{TxtEx}_{d_2}^{c_2}).$$

3.2 Connection with Team and Probabilistic Learning

It turns out that the problem of finding the closedness degree is equivalent to a problem in team learning.

Proposition 3.7 *$\mathcal{I}_1$ is n-closed in $\mathcal{I}_2$ iff $[n-1,n]\mathcal{I}_1 \subseteq \mathcal{I}_2$.*

Proof. Suppose $\mathcal{I}_1$ is n-closed in $\mathcal{I}_2$. Let $U \in [n-1,n]\mathcal{I}_1$, and let $F_1, \ldots, F_n$ be a team that $[n-1,n]\mathcal{I}_1$-identifies U. We define $U_i = \{f \in U \mid (\forall j \neq i)[f \in \mathcal{I}_1(F_j)]\}$. Clearly, $(\forall j \mid 1 \leq j \leq n)[\bigcup_{i=1,i\neq j}^{n} U_i \subseteq \mathcal{I}_1(F_j)]$. Since $\mathcal{I}_1$ is n-closed in $\mathcal{I}_2$, $\bigcup_{i=1}^{n} U_i = U \in \mathcal{I}_2$.

Now, suppose $[n-1,n]\mathcal{I}_1 \subseteq \mathcal{I}_2$. Let $U_1, \ldots, U_n$ be such sets that $(\forall j \mid 1 \leq j \leq n)[\bigcup_{i=1,i\neq j}^{n} U_i \in \mathcal{I}_1]$. Let F_j be the strategy that identifies $\bigcup_{i=1,i\neq j}^{n} U_i$. Then the team $F_1, \ldots, F_n$ $[n-1,n]\mathcal{I}_1$-identifies $\bigcup_{i=1}^{n} U_i$. So $\bigcup_{i=1}^{n} U_i \in \mathcal{I}_2$. Therefore, $\mathcal{I}_1$ is n-closed in $\mathcal{I}_2$. □

Corollary 3.3 *Let $\mathcal{I}$ be an identification type, $n \in \mathbb{N}$, $n \geq 1$. Then* $\mathrm{cdeg}(\mathcal{I}) = n$ *iff n is the minimal number for which $[n-1,n]\mathcal{I} = \mathcal{I}$.* $\mathrm{cdeg}(\mathcal{I}) = \infty$ *iff for all $n \in \mathbb{N}$: $\mathcal{I} \subset [n-1,n]\mathcal{I}$.*

Proof. It is easy to see that $\mathcal{I} \subseteq [n-1,n]\mathcal{I}$. Indeed, let $U \subseteq \mathcal{I}(F)$. By defining $F_1 = \ldots = F_n = F$ we get a team that $[n-1,n]\mathcal{I}$-identifies (even more, $[n,n]\mathcal{I}$-identifies) U. The rest follows from the previous proposition and from the definition of closedness degree. □

Corollary 3.4 *Let $\mathcal{I}_1, \mathcal{I}_2$ be identification types, $n \in \mathbb{N}$, $n \geq 1$. The equality* $\mathrm{csdeg}(\mathcal{I}_1, \mathcal{I}_2) = n$ *holds iff n is the minimal number for which $[n-1,n]\mathcal{I}_1 \subseteq \mathcal{I}_2$. Otherwise* $\mathrm{csdeg}(\mathcal{I}_1, \mathcal{I}_2) = \infty$.

Suppose $n = \mathrm{cdeg}(\mathcal{I}) \geq 2$. Then Corollary 3.3 implies that $[n-1, n]\mathcal{I} = \mathcal{I}$ and $[n-2, n-1]\mathcal{I} \supset \mathcal{I}$. How about the team identification types $[r, s]\mathcal{I}$ with $(n-2)/(n-1) < r/s < (n-1)/n$ (the results in team learning have shown that the so-called success ratio r/s is often important in determining the learning power of a team $[r, s]$)? Are they equivalent to $\mathcal{I}$ or not?

Another question comes from the probabilistic learning which has proved to be closely related with team learning [13, 32, 24, 1]. The probabilistic identification types tend to form probabilistic hierarchies in the following sense: the segment $(0; 1]$ is divided into an enumerably infinite amount of disjoint intervals $(a; b]$ such that for any p_1, p_2 lying in the same interval: $\langle p_1\rangle\mathcal{I} = \langle p_2\rangle\mathcal{I}$, and for any $p_1 < p_2$ from different intervals: $\langle p_1\rangle\mathcal{I} \supset \langle p_2\rangle\mathcal{I}$. The endpoints of these intervals are said to be the elements of the probability hierarchy set of $\mathcal{I}$.

In many studied cases the maximal $p < 1$ from the probabilistic hierarchy turned out to be in the form $p = (n-2)/(n-1)$ where $n = \mathrm{cdeg}(\mathcal{I})$. Is this an accident? The following two results prove that it is not so. They were obtained by K. Apsītis and the author of this work.

Theorem 3 *Let $\mathcal{I}$ be an identification type with* $\mathrm{cdeg}(\mathcal{I}) = n \geq 2$. *Let $r, s \in \mathbb{N}$, $0 < r \leq s$. Then $[r, s]\mathcal{I} = \mathcal{I}$ iff $(n-2)/(n-1) < r/s \leq 1$.*

Proof. Suppose $r/s \leq (n-2)/(n-1)$ and $[r, s]\mathcal{I} = \mathcal{I}$. We shall prove that $[n-2, n-1]\mathcal{I} = \mathcal{I}$ in such case. Let $U \in [n-2, n-1]\mathcal{I}$, and let $F = \{F_1, \ldots, F_{n-1}\}$ be a team $[n-2, n-1]\mathcal{I}$-identifying U. Then we form a team $G = \{G_1, \ldots, G_s\}$ as follows: $G_i = F_{i \bmod n-1}$. For any $f \in U$, the number of unsuccessful learners among F_i is at most one, thus among G_i this number is at most $\lceil s/(n-1) \rceil$. The assumption implies that $r(n-1) \leq s(n-2) = s(n-1) - s$, so $s/(n-1) \leq s - r$. Since $s - r$ is integer, $\lceil s/(n-1) \rceil \leq s - r$. Thus G $[r, s]\mathcal{I}$-identifies U. Since $[r, s]\mathcal{I} = \mathcal{I}$, $U \in \mathcal{I}$. We have proved that $[n-2, n-1]\mathcal{I} = \mathcal{I}$, but that contradicts the equality $\mathrm{cdeg}(\mathcal{I}) = n$. Therefore, $[r, s]\mathcal{I} \supset \mathcal{I}$ for $r/s \leq (n-2)/(n-1)$.

Suppose $(n-2)/(n-1) < r/s \leq 1$. We shall prove that $[r, s]\mathcal{I} = \mathcal{I}$. Let $U \in [r, s]\mathcal{I}$, and let $F = \{F_1, \ldots, F_s\}$ be a team $[r, s]\mathcal{I}$-identifying U.

Every $f \in U$ is $\mathcal{I}$-identified by some subset of F. We declare two elements of U to be equivalent iff they are identified by exactly the same members of F. Denote the equivalence classes by $U_1, \ldots, U_m$. Let I be any subset of $\{1, \ldots, m\}$, $|I| \leq n-1$.

We claim that $\bigcup_{i \in I} U_i \in \mathcal{I}$. Indeed, each U_i is $\mathcal{I}$-identified by at least r members of F. Since $|F| = s$, at most $s - r$ members of F may fail to identify U_i. Therefore, at most $(n-1)(s-r)$ members of F can fail to identify the union $\bigcup_{i \in I} U_i$. But $r/s > (n-2)/(n-1)$ implies $(n-1)(s-r) < s = |F|$. Therefore at least one of the F_i $\mathcal{I}$-identifies the union $\bigcup_{i \in I} U_i$. By applying Proposition 3.4 we get $U = U_1 \cup \ldots \cup U_m \in \mathcal{I}$. □

Definition 3.4 *An identification type $\mathcal{I}$ is* team reducible *iff*

$$(\forall p \in (0; 1))(\forall \epsilon > 0)(\exists r, s \in \mathbb{N}) \left[\frac{r}{s} > p - \epsilon \wedge \langle p\rangle\mathcal{I} \subseteq [r, s]\mathcal{I}\right].$$

Additionally we require from the identification type a natural property: let $\{F_1, \ldots, F_n\}$ be a finite set of learners, and $p_1, \ldots p_n$ are recursive probabilities, $\sum_{i=1}^n p_i = 1$; then there exists a probabilistic learner G which initially chooses a learner F_i with probability p_i, and then simulates it (that is, any f is identified by G iff it is identified by F_i). All the identification types where the learners are algorithms or sets of algorithms, and the learning criterion involves only the output of these algorithms (not the algorithms themselves or their performance) satisfy this property, since any algorithm can be simulated by another one.

Theorem 4 *Let $\mathcal{I}$ be a team reducible identification type with* $\mathrm{cdeg}(\mathcal{I}) = n \geq 2$ *which satisfies the simulation property described above. Then $\langle p\rangle\mathcal{I} = \mathcal{I}$ iff* $(n-2)/(n-1) < p \leq 1$.

Proof. Suppose $p \leq (n-2)/(n-1)$ and $\langle p\rangle\mathcal{I} = \mathcal{I}$. Trivially, $\langle p_1\rangle\mathcal{I} \supseteq \langle p_2\rangle\mathcal{I}$ if $p_1 < p_2$. Hence, $\langle (n-2)/(n-1)\rangle\mathcal{I} = \mathcal{I}$. It is easy to see that $[r,s]\mathcal{I} \subseteq \langle r/s\rangle\mathcal{I}$: consider a team $[r,s]$ and a probabilistic learner which chooses initially any member of the team with probability $1/s$ and simulates it; clearly, its success probability is at least r/s. So $\mathcal{I} \subseteq [n-2,n-1]\mathcal{I} \subseteq \langle (n-2)/(n-1)\rangle\mathcal{I} = \mathcal{I}$, and $[n-2,n-1]\mathcal{I} = \mathcal{I}$. But that contradicts the assumption $\mathrm{cdeg}(\mathcal{I}) = n$.

Suppose $(n-2)/(n-1) < p \leq 1$. Take $\epsilon = (p-(n-2)/(n-1))/2$ and apply the definition of team reducibility to get $\langle p\rangle\mathcal{I} \subseteq [r,s]\mathcal{I}$ with $r/s > (n-2)/(n-1)$. From Theorem 3: $[r,s]\mathcal{I} = \mathcal{I}$. So $\langle p\rangle\mathcal{I} = \mathcal{I}$. □

3.3 Satisfiability of Requirements

Suppose we have a set of requirements on the $\mathcal{I}$-identifiability of every union of some classes out of U_1, U_2, …, U_k. We want to find a simple criterion for distinguishing if this set of requirements is satisfiable.

A convenient way for expressing such requirements is to use the Boolean functions. We shall write Boolean vectors in boldface and their components in italics with indices. A vector $\mathbf{x} \in \{0,1\}^k$ corresponds to the union $\bigcup_{x_i=1} U_i$. Let $f : \{0,1\}^k \to \{0,1\}$. If $f(\mathbf{x}) = 0$, we demand that the corresponding union is identifiable. If $f(\mathbf{x}) = 1$, the corresponding union must be unidentifiable.

Definition 3.5 *Let $a, b \in \mathbb{N} \cup \{*\}$. A Boolean function $f : \{0,1\}^k \to \{0,1\}$ is $\mathcal{I}$-satisfiable iff* $(\exists U_1, \ldots, U_k \subseteq \mathcal{A})(\forall \mathbf{x} \in \{0,1\}^k)[\bigcup_{x_i=1} U_i \in \mathcal{I} \Leftrightarrow f(\mathbf{x}) = 0]$.

Which of the properties of identification types $\mathcal{I}$ are relevant for the satisfiability of Boolean functions? Two properties are immediate: $\mathcal{I}$ contains the empty set and together with a set $\mathcal{I}$ contains all of its subsets. [4] showed that another property is relevant: the closedness degree. The following definition combines these three restrictions.

Definition 3.6 *A Boolean function $f : \{0,1\}^k \to \{0,1\}$ is n-convolutional iff*

1. $f(\mathbf{0}) = 0$;
2. $(\forall \mathbf{x}, \mathbf{y} \in \{0,1\}^k)[\mathbf{x} \leq \mathbf{y} \Rightarrow f(\mathbf{x}) \leq f(\mathbf{y})]$ *(monotonicity);*
3. $(\forall \mathbf{x} \in \{0,1\}^k)(\forall i_1, \ldots, i_n \mid 1 \leq i_1 < \ldots < i_n \leq k \wedge x_{i_1} = \ldots = x_{i_n} = 1)[(\forall r \mid 1 \leq r \leq n)[f(x_1, \ldots, x_{i_r - 1}, 0, x_{i_r + 1}, \ldots, x_k) = 0] \Rightarrow f(\mathbf{x}) = 0]$.

We shall prove by the next theorem that n-convolutionality is the desired criterion for all identification types satisfying two natural properties.

Definition 3.7 *Let t be an injective mapping $\mathcal{A} \times \mathbb{N} \to \mathcal{A}$ (we shall call such mapping* a tagging *of $\mathcal{A}$). An identification type $\mathcal{I}$ is t-*tag invariant *iff*

$$(\forall j \in \mathbb{N})[U \in \mathcal{I} \Leftrightarrow t(U, j) \in \mathcal{I}],$$

where $t(U, j)$ is the image of U under $t(\cdot, j)$.

Informally, $\mathcal{I}$ is t-tag invariant iff supplying a tag j to every element of a class does not affect its identifiability.

Definition 3.8 *Let t be a tagging of $\mathcal{A}$. An identification type $\mathcal{I}$ is t-*tagged union closed *iff*

$$(\forall n \in \mathbb{N})(\forall U_1, U_2, \ldots, U_n \in \mathcal{I})[\bigcup_{j=1}^{n} t(U_j, j) \in \mathcal{I}].$$

The "natural" identification types usually have these properties. We shall prove it for the types $\mathbf{Ex}_b^a$ and $\mathbf{TxtEx}_b^a$.

Proposition 3.8 *There exist taggings t_1, t_2 such that $(\forall a, b \in \mathbb{N} \cup \{*\})[\mathbf{Ex}_b^a$ is t_1-tag invariant and t_1-tagged union closed, and $\mathbf{TxtEx}_b^a$ is t_2-tag invariant and t_2-tagged union closed$]$.*

Proof. Define $t_1(f, j) = f'$, where $f'(x) = \langle f(x), j \rangle$, and $t_2(L, j) = L' = \{\langle x, j \rangle \mid x = 0 \vee x - 1 \in L\}$. It is easy to see that t_1 and t_2 are taggings for $\mathcal{R}$ and $\mathcal{E}$, and satisfy the corresponding condition of tag invariance for $\mathbf{Ex}_b^a$ and $\mathbf{TxtEx}_b^a$ (because the strategy can easily obtain f from f', L from L', and vice versa).

Suppose that $U_1, U_2, \ldots, U_n \in \mathcal{I}$. To identify $\bigcup_{j=1}^{n} t_i(U_j, j)$, $i = 1, 2$, in both cases the strategy obtains the tag j from the input and applies the strategy that identifies U_j. This proves the tagged union closedness. □

Theorem 5 *Let $\mathcal{I}$ be a t-tag invariant and t-tagged union closed identification type. If $\text{cdeg}(\mathcal{I}) = n \in \mathbb{N}$, then a Boolean function is $\mathcal{I}$-satisfiable iff it is n-convolutional.*

If $\text{cdeg}(\mathcal{I}) = \infty$, then a Boolean function f is $\mathcal{I}$-satisfiable iff $f(\mathbf{0}) = 0$ and f is monotone.

Proof. At first we prove the necessity. Suppose a function $f : \{0,1\}^k \to \{0,1\}$ is $\mathcal{I}$-satisfiable. Let $U_1, \ldots, U_k$ be the classes that satisfy the requirements. Then, because of the mentioned properties of identification types, $f(\mathbf{0}) = 0$ and f is monotone. Suppose $\mathrm{cdeg}(\mathcal{I}) = n \in \mathbb{N}$. Let $\mathbf{x}$ be an arbitrary vector from $\{0,1\}^k$. Let $i_1, \ldots, i_n$ be such that $1 \le i_1 < \ldots < i_n \le k$ and $x_{i_1} = \ldots = x_{i_n} = 1$. We define $\mathbf{y}^j$, $1 \le j \le n$, to be such vectors that

1. $y^j_{i_j} = 1$,
2. $y^j_{i_r} = 0$ for $r \ne j$, $1 \le r \le n$,
3. $y^j_s = x_s$ for $s \in \{1, \ldots, k\} - \{i_1, \ldots, i_n\}$.

Let V_j be the union of $U_1, \ldots, U_k$ corresponding to the vector $\mathbf{y}^j$. Then the vectors $(x_1, \ldots, x_{i_r - 1}, 0, x_{i_r + 1}, \ldots, x_k)$, $1 \le r \le n$, correspond to the unions of $n-1$ classes out of $V_1, \ldots, V_n$. If these are $\mathcal{I}$-identifiable, so is $\bigcup_{j=1}^n V_j$, because $\mathcal{I}$ is n-closed. Since $\bigcup_{j=1}^n V_j$ corresponds to the vector $\mathbf{x}$, we have proved that f is n-convolutional.

Now, sufficiency.

Definition 3.9 *A vector* $\mathbf{x}$ *is* a minimal 1-vector *for a Boolean function* f *iff*

1. $f(\mathbf{x}) = 1$ *and*

2. $(\forall \mathbf{y} < \mathbf{x})[f(\mathbf{y}) = 0]$.

Let $\mathbf{x}^j$, $1 \le j \le m$, be all the minimal 1-vectors for f. Let n_j be the number of components in $\mathbf{x}^j$ that are equal to 1. Suppose that $\mathrm{cdeg}(\mathcal{I}) = n \in \mathbb{N}$ and f is n-convolutional. Point 3 in the definition of n-convolutionality implies that $n_j < n$ for every $j \in \{1, \ldots, m\}$. Suppose $\mathrm{cdeg}(\mathcal{I}) = \infty$, $f(\mathbf{0}) = 0$ and f is monotone. Then, trivially, every $n_j < \infty$.

So, in both cases $\mathcal{I}$ is not n_j-closed, $j \in \{1, \ldots, m\}$, and there are such classes $U^j_1, \ldots, U^j_{n_j}$ that every union of $n_j - 1$ out of them is $\mathcal{I}$-identifiable, while $\bigcup_{i=1}^{n_j} U^j_i$ is not.

Now we construct the classes $U_1, \ldots, U_k$ that satisfy the requirements given by f. Suppose $x^j_i = 1$ for some $1 \le i \le k$ and $1 \le j \le m$, and suppose x^j_i is the l_j-th component of $\mathbf{x}^j$ that is equal to 1. Then we add the set $t(U^j_{l_j}, j)$ to U_i. So the class U_i is the union of these sets for all the values of j such that $x^j_i = 1$.

Suppose $f(\mathbf{x}) = 1$. Then for some j, $\mathbf{x}^j \le \mathbf{x}$, and the corresponding union contains as a subset the image of $\bigcup_{i=1}^{n_j} U^j_i \notin \mathcal{I}$ under the tagging $t(\cdot, j)$. Since $\mathcal{I}$ is t-tag invariant, this union is $\mathcal{I}$-unidentifiable.

Suppose $f(\mathbf{x}) = 0$. According to the monotonicity, for each j there is such s_j that $x_{s_j} = 0$ and $x^j_{s_j} = 1$. Suppose $x^j_{s_j}$ is the l_j-th component equal to 1 in $\mathbf{x}^j$. Then the union corresponding to $\mathbf{x}$ is a subset of $\bigcup_{j=1}^m t(V_j, j)$, where

$V_j = \bigcup_{i=1, i \neq l_j}^{n_j} U_i^j$ is $\mathcal{I}$-identifiable. Since $\mathcal{I}$ is t-tagged union closed, this union is $\mathcal{I}$-identifiable.

This proves $\mathcal{I}$-satisfiability. $\square$

Chapter 4

Identifying Total Recursive Functions

The results of the previous chapter imply that to solve the satisfiability problem for particular identification types, we have only to find the their closedness degrees, which we shall do in this and the next chapter. In the proofs we shall use diagonalization and simulation techniques. Another interesting approach was considered in [2], where the similarity of such proofs to games was explored.

4.1 Identification in the Limit

Here the case of $\mathbf{Ex}^a$-identification will be considered.

The first result in the whole area of the closedness of identification types (for total recursive functions) was Theorem 2 in [8].

Theorem 6 [8] *There are such classes $U_1, U_2 \subseteq \mathcal{R}$ that $U_1 \in \mathbf{Ex}$, $U_2 \in \mathbf{Ex}$, and $U_1 \cup U_2 \notin \mathbf{Ex}^*$.*

Proof. Define

$$U_1 = \{f \in \mathcal{R} \mid f = \varphi_{f(0)}\},$$
$$U_2 = \{f \in \mathcal{R} \mid (\exists N)[(\forall x \geq N)[f(x) = 0] \vee (\forall x \geq N)[f(x) = 1]]\}.$$

Strategy $F(\langle f(0), \ldots, f(n)\rangle) = f(0)$ identifies U_1. (We see that in fact $U_1 \in \mathbf{Ex}_0$.) Class U_2 is identified by a strategy that on the initial segment $f^{[n]}$ outputs a new hypothesis h such that

$$\varphi_h(x) = \begin{cases} f(x), & x \leq n, \\ 0, & x > n \text{ and } f(n) = 0, \\ 1, & x > n \text{ and } f(n) \neq 0, \end{cases}$$

if the previous hypothesis is invalid (if $f(n-1) = 0 \neq f(n)$, or $f(n-1) \neq 0$ and $f(n) \neq 1$, or $n = 0$); otherwise it outputs the previous hypothesis.

Now we shall prove that $U_1 \cup U_2 \notin \mathbf{Ex}^*$. Let F be an arbitrary strategy. We shall construct a function that belongs to $U_1 \cup U_2$ and is not identified by F.

Consider a family of functions $\{f_i \mid i \in \mathbb{N}\} \subseteq \mathcal{P}$ with the following algorithm for f_i.

- *Stage 0.*
  Define $f_i(0) = i$. Find x for which $F(\langle i0^x \rangle)$ is defined. Then define $f_i(1) = f_i(2) = \ldots = f_i(x) = 0$, let $\sigma_1 \leftarrow i0^x$ and go to stage 1.

- *Stage m ($m \geq 1$).*
  Compute, on which of the segments $\sigma_m 0^j$ and $\sigma_m 1^j$ ($j \in \mathbb{N}, j \geq 1$) strategy F changes its last hypothesis $F(\langle \sigma_m \rangle)$. If such segment $\sigma_m \alpha^k$ is found (α is 0 or 1), define $f_i(|\sigma_m|) = \ldots = f_i(|\sigma_m| + k - 1) = \alpha$, let $\sigma_{m+1} \leftarrow \sigma_m \alpha^k$, and go to stage $m + 1$.

According to the recursion theorem (see [30]) $(\exists i_0)[f_{i_0} = \varphi_{i_0}]$. Thus $f_{i_0}(0)$ is a correct Gödel number of f_{i_0}.

If $f_{i_0} \in \mathcal{R}$, then $f_{i_0} \in U_1$ and F makes infinitely many mindchanges on it, so $f_{i_0} \notin \mathbf{Ex}^*(F)$.

If $f_{i_0} \notin \mathcal{R}$, then either F did not output any hypothesis, or it did not change hypothesis on any of functions $\sigma_m 0^\infty$ and $\sigma_m 1^\infty$ for some m and remained in stage m forever. In the former case we choose function with the string of values $i_0 0^\infty$. In the latter case the last hypothesis made by F is incorrect for at least one of the segments $\sigma_m \alpha^\infty$ (α is 0 or 1), because they differ in infinitely many points. Choose the corresponding function. The chosen function belongs to U_2 and is not $\mathbf{Ex}^*$-identified by F. □

So, $\text{csdeg}(\mathbf{Ex}, \mathbf{Ex}^*) > 2$. Then, in team learning, the next result was obtained.

Theorem 7 [29] $(\forall a \in \mathbb{N} \cup \{*\})[[2,3]\mathbf{Ex}^a \subseteq \mathbf{Ex}^a]$.

Using Propositions 3.3 and 3.4 we get:

Theorem 8 $(\forall a \in \mathbb{N} \cup \{*\})[\text{cdeg}(\mathbf{Ex}^a) = 3]$.

4.2 A Bound on Mindchanges

Here we find the closedness degrees for the classes $\mathbf{Ex}_b$ and $\mathbf{Ex}_b^*$, $b \in \mathbb{N}$. These two identification types turn out to be similar in this aspect. Theorem 9 is a generalization of Theorem 4.2 in [4].

Theorem 9 $(\forall b \in \mathbb{N})(\forall a, a' \in \mathbb{N} \cup \{*\} \mid a' \geq 2^{b+1} a)[\text{csdeg}(\mathbf{Ex}_b^a, \mathbf{Ex}_b^{a'}) \leq 2^{b+2}]$.

The proof of the theorem is based on a lemma.

Lemma 4.1 *For each $b \in \mathbb{N}$ there is an algorithm that can $\mathbf{Ex}_b$-identify any function $f \in \mathcal{R}$ knowing (receiving as parameters) algorithms of $2^{b+2}-1$ strategies such that each of them produces at least one hypothesis on f and at least $2^{b+2}-2$ of them $\mathbf{Ex}_b$-identify f.*

Proof. Let strategies $F_1, F_2, \ldots, F_{2^{b+2}-1}$ and a function f satisfy the conditions. The algorithm F redirects its input to the strategies F_i until they output hypotheses h_i, $i = 1, 2, \ldots, 2^{b+2}-1$. Then F produces a hypothesis h such that $\varphi_h(x) = y$ iff at least 2^{b+1} of the values $\varphi_{h_i}(x)$, $i = 1, 2, \ldots, 2^{b+2}-1$, are y.

In case $b > 0$, F waits for $2^{b+1}-1$ of the strategies F_i to make a mindchange. Suppose it happens. Then, to $\mathbf{Ex}_b$-identify f, these strategies can make no more than $b-1$ mindchanges from now on. So F selects these $2^{b+1}-1$ strategies, disregards their hypotheses made before the mindchange and applies to them the algorithm corresponding to the case of $\mathbf{Ex}_{b-1}$-identification. This algorithm identifies f with no more than b additional mindchanges, so $f \in \mathbf{Ex}_b(F)$.

Suppose no more than $2^{b+1}-2$ strategies make a mindchange or $b = 0$. Then among h_i there are no more than $2^{b+1}-1$ incorrect hypotheses, and $\varphi_h = f$. □

Proof of Theorem 9. It is sufficient to prove that $\mathbf{Ex}_b$ is 2^{b+2}-closed.

Let $U_1, U_2, \ldots, U_{2^{b+2}} \subseteq \mathcal{R}$ be such classes that all the unions of $2^{b+2}-1$ classes out of them are $\mathbf{Ex}_b$-identifiable. Let $F_1, F_2, \ldots, F_{2^{b+2}}$ be the strategies that identify these unions. We shall construct a strategy F that identifies $\bigcup_{j=1}^{2^{b+2}} U_j$.

The strategy F redirects its input to the strategies F_i until $2^{b+2}-1$ of them output a hypothesis. Such an event happens because every function $f \in \bigcup_{j=1}^{2^{b+2}} U_j$ belongs to $2^{b+2}-1$ of the unions of $2^{b+2}-1$ classes, thus at most one of the strategies F_i does not identify f.

Then F selects these $2^{b+2}-1$ strategies, applies the algorithm from the previous lemma and identifies the input function. □

The next theorem is a generalization of Theorems 3.1 and 4.1 from [4].

Theorem 10 $(\forall b \in \mathbb{N})[\mathrm{csdeg}(\mathbf{Ex}_b, \mathbf{Ex}_b^*) > 2^{b+2}-1]$.

The method of proof of this and other theorems establishing lower bounds for csdeg makes use of the idea whose origin is the concept of "self-describing" functions used in [8, Theorem 2]. (Theorem 6 in this work). We shall use functions that output instructions for $\mathbf{Ex}_b^a$-identification of themselves. Even more, they will output many arrays of such instructions. The instructions will be of three kinds.

1. An elementary instruction $\langle 1, j, i, n\rangle$, $i, j \geq 1$. Informally, it proposes n as the i-th hypothesis in the j-th array of instructions.

2. A compound instruction $\langle 2, y_1, \ldots, y_p \rangle$, where y_i are elementary instructions. In this way many elementary instructions can be incorporated in one value output by a function.

3. A split instruction. It consists of two values, $\langle 3, i, y_1, y_2 \rangle$ and $\langle 4, i, y_3, y_4 \rangle$, where $y_1 - y_2 + y_3 - y_4$ is an elementary or a compound instruction, and i is a unique identifier for this pair of values. In this way an instruction can be split into two parts so that by changing any of these parts we can obtain a different instruction. (In fact, we could do this using only two numbers, y_1 and y_3; we have chosen the above form for the ease of writing the proof.)

Among the values $f(x)$ there must be exactly one value of kind $\langle 3, i, \cdot, \cdot \rangle$ and exactly one value $\langle 4, i, \cdot, \cdot \rangle$ to get a split instruction with identifier i. Naturally, other kinds of instructions can be designed to prove similar results for identification types not considered in this work.

Let $\mathrm{Instr}(f)$ be the set of elementary instructions output by f, including those that are contained in the compound and the split instructions.

Definition 4.1 *We shall say that a function $f \in \mathcal{R}$ is a* j-instructor *with respect to the $\mathbf{Ex}_b^a$-identification ($a, b \in \mathbb{N} \cup \{*\}$) iff there is an instruction $\langle 1, j, c, n \rangle \in \mathrm{Instr}(f)$ such that $\varphi_n =^a f$, $c \leq b + 1$ and, if $\langle 1, j, c', n' \rangle \in \mathrm{Instr}(f)$ for some c' and n', then $c' < c$ or $n' = n$.*

Let us denote the class of j-instructors with respect to $\mathbf{Ex}_b^a$ by $I_j^{\mathbf{Ex}_b^a}$.

Proposition 4.1 $I_j^{\mathbf{Ex}_b^a} \in \mathbf{Ex}_b^a$.

Proof. Receiving values of the input function the strategy extracts from them the elementary instructions and outputs the sequence of hypotheses corresponding to the j-th array of instructions. If its previous hypothesis was based on an instruction $\langle 1, j, i, n \rangle$ and it receives an instruction $\langle 1, j, i', n' \rangle$, it outputs n' iff $i' > i$. If it has no new hypothesis to output, it repeats the previous one. It follows from the definition of instructors that the strategy identifies the function. □

Proof of Theorem 10. Let us denote $k = 2^{b+2} - 1$. It is enough to prove that there are classes U_1, U_2, ..., U_k such that $(\forall i)[\bigcup_{j=1, j \neq i}^{k} U_j \in \mathbf{Ex}_b]$ and $\bigcup_{j=1}^{k} U_j \notin \mathbf{Ex}_b^*$.

Define $U_i = \bigcap_{j \neq i} I_j^{\mathbf{Ex}_b}$, where $i, j \in [1, k]$. Then $\bigcup_{i \neq j} U_i \subseteq I_j^{\mathbf{Ex}_b} \in \mathbf{Ex}_b$.

We have to prove that $\bigcup_{j=1}^{k} U_j \notin \mathbf{Ex}_b^*$. Suppose there is a strategy F that identifies this union. We shall use the multiple recursion theorem (see [35], it is a generalization of the recursion theorem used in the proof of Theorem 6). It allows us to construct functions that use each others Gödel numbers as parameters. We construct functions φ_{n_i}, one of which will be the function from $\bigcup_{i=1}^{k} U_i$ not identified by F.

The algorithm below uses a procedure new(x). It lets $x \leftarrow n_c$, and then $c \leftarrow c+1$, where c is a counter in the algorithm.

The algorithm for φ_{n_i} is as follows.

- *Stage 0.*
  $c \leftarrow 1$, $j \leftarrow 0$, $p \leftarrow k$, $D \leftarrow \{p\}$.
  Execute new(s_i) for $1 \leq i \leq p-1$. Output values as shown in the next table.

	0	$\dots$	$p-2$	$\dots$
$\varphi_{s_1}, \dots, \varphi_{s_{p-1}}$	$\langle 1,1,1,s_1\rangle$	$\dots$	$\langle 1,p-1,1,s_{p-1}\rangle$	$\langle\rangle$

  The leftmost column contains the functions defined, other columns show values output at the corresponding inputs. The rightmost column means that these values are output up to infinity unless the algorithm goes to the next stage.
  Let the variable y throughout this algorithm indicate the maximal value of argument at which the values have been output at the moment. We simulate the strategy F on the initial segments of φ_{s_1}. If a hypothesis is output on $\varphi_{s_1}^{[x]}$ for some x, we let $h \leftarrow F(\varphi_{s_1}^{[x]})$, $x_0 \leftarrow \max(x,y)+1$; we output $\langle\rangle$ up to $x_0 - 1$, if needed, and go to stage 1.

- *Stage m ($1 \leq m \leq b+1$).*
  $r \leftarrow \mathrm{card}(D)$, $l \leftarrow (p-1)/2$.
  Let $d_1, \dots, d_r$ be the elements of D. Execute new(t), new(u_i) for $1 \leq i \leq l-1$, new(t'), new(v_i) for $1 \leq i \leq l-1$. Output values as shown in the next table.

	x_0	$\dots$	x_0+r-1
$\varphi_{s_1}, \dots, \varphi_{s_l}, \varphi_t, \varphi_{u_1}, \dots, \varphi_{u_{l-1}}$	$\langle 1,d_1,m,t\rangle$	$\dots$	$\langle 1,d_r,m,t\rangle$
$\varphi_{s_{l+1}}, \dots, \varphi_{s_{p-1}}, \varphi_{t'}, \varphi_{v_1}, \dots, \varphi_{v_{l-1}}$	$\langle 1,d_1,m,t'\rangle$	$\dots$	$\langle 1,d_r,m,t'\rangle$
	x_0+r	$\dots$	
$\varphi_{s_1}, \dots, \varphi_{s_l}, \varphi_t, \varphi_{u_1}, \dots, \varphi_{u_{l-1}}$	$\langle 1,j+l+1,m+1,u_1\rangle$	$\dots$	
$\varphi_{s_{l+1}}, \dots, \varphi_{s_{p-1}}, \varphi_{t'}, \varphi_{v_1}, \dots, \varphi_{v_{l-1}}$	$\langle 1,j+1,m+1,v_1\rangle$	$\dots$	
	$x_0+r+l-2$	$\dots$	
$\varphi_{s_1}, \dots, \varphi_{s_l}, \varphi_t, \varphi_{u_1}, \dots, \varphi_{u_{l-1}}$	$\langle 1,j+2l-1,m+1,u_{l-1}\rangle$	$\langle\rangle$	
$\varphi_{s_{l+1}}, \dots, \varphi_{s_{p-1}}, \varphi_{t'}, \varphi_{v_1}, \dots, \varphi_{v_{l-1}}$	$\langle 1,j+l-1,m+1,v_{l-1}\rangle$	$\langle 0\rangle$	

  If $m = b+1$, the algorithm remains in this stage forever.
  If $m < b+1$, we simulate F on functions φ_{s_1} and $\varphi_{s_{l+1}}$.
  If F changes the current hypothesis h on $\varphi_{s_1}^{[x]}$ for some x, we let $h \leftarrow F(\varphi_{s_1}^{[x]})$, $x_0 \leftarrow \max(x,y)+1$, output $\langle\rangle$ up to x_0-1, add $j+1, \dots, j+l, j+p-1$ to D, let $s_i \leftarrow u_i$ for $1 \leq i \leq l-1$, $j \leftarrow j+l$, $p \leftarrow l$ and go to stage $m+1$.
  If F changes the current hypothesis h on $\varphi_{s_{l+1}}^{[x]}$ for some x, we let $h \leftarrow F(\varphi_{s_{l+1}}^{[x]})$, $x_0 \leftarrow \max(x,y)+1$, output $\langle 0\rangle$ up to x_0-1, add $j+l, \dots, j+p-1$ to D, let $s_i \leftarrow v_i$ for $1 \leq i \leq l-1$. $p \leftarrow l$ and go to stage $m+1$.

Let us explain the meanings of variables at the start of stage m. s_i are Gödel numbers that have been proposed as the m-th hypotheses in the instructions. The indices of these instructions begin with $j+1$ and their amount is $p-1 = 2^{b+3-m}-2$. D contains the indices of the arrays of instructions for which the m-th hypothesis has not been proposed yet.

At stage m two alternatives represented by φ_{s_1} and $\varphi_{s_{l+1}}$ are proposed for F. Since they differ at infinitely many points, the last hypothesis h cannot be $\mathbf{Ex}_b^*$-correct for both of them. If F does not make a mindchange on any of the two alternatives, the algorithm remains at stage m forever, $\varphi_{s_1}, \varphi_{s_{l+1}} \in \bigcup_{i=1}^k U_i$ and at least one of these two functions is not $\mathbf{Ex}_b^*$-identified by F. If F makes a mindchange on one of these alternatives, the algorithm switches to stage $m+1$, choosing this alternative for further consideration. At stage $b+1$ F cannot output a new hypothesis, since it already has made b mindchanges.

It is easy to check that, in whatever stage the algorithm stays forever, both proposed alternatives are in $\bigcup_{i=1}^k U_i$. So F does not identify this class. Contradiction. $\square$

Corollary 4.1 $(\forall b \in \mathbb{N})[\mathrm{cdeg}(\mathbf{Ex}_b) = \mathrm{cdeg}(\mathbf{Ex}_b^*) = 2^{b+2}]$.

4.3 Identification with Anomalies

Here we consider the case of $\mathbf{Ex}_b^a$-identification, where $a, b \in \mathbb{N}$, $a > 0$. The results turn out to be rather surprising. For $a = 1$, the closedness degree is finite and still grows exponentially relative to b, while for $a \geq 2$ the closedness degree is ∞.

Theorem 11 $(\forall b \in \mathbb{N})[\mathrm{cdeg}(\mathbf{Ex}_b^1) > \frac{7 \cdot 6^{b+1}-2}{5}]$.

Proof. Let us denote $k = \frac{7 \cdot 6^{b+1}-2}{5}$. It is enough to show that there are such classes $U_1, \ldots, U_k$ that the unions of $k-1$ classes out of them are identifiable, while $\bigcup_{j=1}^k U_j$ is not.

We define $U_i = \bigcap_{j=1, j\neq i}^k I_j^{\mathbf{Ex}_b^1}$, $1 \leq i \leq k$. Then $\bigcup_{i=1, i\neq j}^k U_i \subseteq I_j^{\mathbf{Ex}_b^1} \in \mathbf{Ex}_b^1$, $1 \leq j \leq k$.

We have to prove that $\bigcup_{i=1}^k U_i \notin \mathbf{Ex}_b^1$. We use diagonalization over the strategies. Let F be an arbitrary strategy. Using the multiple recursion theorem similarly as in Theorem 10 we construct functions φ_{n_i}, $1 \leq i \leq N$ (where N is a natural number determined only by b) that can use each others Gödel numbers incorporated in their values. With their help we shall construct a function $f \in \bigcup_{i=1}^k U_i$ not identified by F.

We remind that the procedure new(x) lets $x \leftarrow n_c$, $c \leftarrow c+1$, where c is a counter in the algorithm.

The algorithm for φ_{n_i} is as follows.

- *Stage 0.*
  Let $c \leftarrow 1$, $j \leftarrow 0$, $p \leftarrow (7 \cdot 6^{b+1} - 2)/5$, $D \leftarrow \{p\}$. Execute new(s_i) for $1 \le i \le p-1$. Output values as shown in the next table.

	0	...	$p-2$	...
$\varphi_{s_1}, \dots, \varphi_{s_{p-1}}$	$\langle 1,1,1,s_1 \rangle$	...	$\langle 1,p-1,1,s_{p-1} \rangle$	$\langle\rangle$
f_0	$\langle 1,1,1,s_1 \rangle$	...	$\langle 1,p-1,1,s_{p-1} \rangle$	$\langle\rangle$

  The function under the last horizontal line (f_0 in this case) is the function not identified by F in case the algorithm remains in this stage.
  Let the variable y throughout this algorithm indicate the maximal value of argument at which the values have been output at the moment. We simulate the strategy F on the initial segments of f_0. If a hypothesis is output on $f_0^{[x]}$, we let $h \leftarrow F(f_0^{[x]})$, $x_0 \leftarrow \max(x,y)+1$; we output $\langle\rangle$ up to $x_0 - 1$, if needed, and go to stage 1.

- *Stage m* $(1 \le m \le b+1)$.
  Let $r \leftarrow$ card(D). Let $d_1, \dots, d_r$ be the elements of D. Execute new(t). Go to substage 1.

  - *Substage 1.*
    Let $u \leftarrow (p-2)/2$, $y_1 \leftarrow \langle 3, 2m-1, 0, 0\rangle$, $z_2 \leftarrow \langle 2, \langle 1, d_1, m, t\rangle, \dots, \langle 1, d_r, m, t\rangle\rangle$, $y_2 \leftarrow \langle 4, 2m-1, z_2, 0\rangle$. Output values as shown in the next table.

	x_0	x_0+1	...
$\varphi_{s_1}, \dots, \varphi_{s_u}$	?	y_2	$\langle\rangle$
$\varphi_{s_{u+1}}, \dots, \varphi_{s_{p-2}}$	y_1	?	$\langle\rangle$
$\varphi_{s_{p-1}}$	?	?	$\langle\rangle$
φ_t	y_1	y_2	$\langle\rangle$
f_{7m-6}	y_1	y_2	$\langle\rangle$

    The question marks mean that the values are not output at these points as yet. We compute $\varphi_h(x_0)$, $\varphi_h(x_0+1)$ and the outputs of F on f_{7m-6}.
    If $m < b+1$ and F changes its current hypothesis on $f_{7m-6}^{[x]}$ for some x, we let $h \leftarrow F(f_{7m-6}^{[x]})$, replace question marks with the corresponding values of f_{7m-6}, let $x_0 \leftarrow max(x,y)+1$, output $\langle\rangle$ up to $x_0 - 1$, add $j+(p-2)/6+1, \dots, j+p-1$ to D, let $p \leftarrow (p-2)/6$ and go to stage $m+1$.
    If $\varphi_h(x_0) = y_1$, let $x_1 \leftarrow y+1$, and go to substage 2.
    If $\varphi_h(x_0+1) = y_2$, let $x_1 \leftarrow y+1$, and go to substage 5.

  - *Substage 2.*
    Let $v \leftarrow (p-2) \cdot 2/3$, $w \leftarrow (p-2) \cdot 5/6$. Execute new$(s_i')$ for $w+1 \le i \le p-3$. Let $y_3 \leftarrow \langle 3, 2m, 0, 0\rangle$, $z_4 \leftarrow \langle 2, \langle 1, j+w+1, m+$

$1, s'_{w+1}\rangle, \ldots, \langle 1, j+p-3, m+1, s'_{p-3}\rangle\rangle$, $y_4 \leftarrow \langle 4, 2m, z_4, 0\rangle$. Output values as shown in the next table.

	x_0	x_0+1	$\ldots$	x_1	x_1+1	$\ldots$
$\varphi_{s_1}, \ldots, \varphi_{s_u}, \varphi_{s_{p-1}}$	?	y_2	$\langle\rangle$	y_3	y_4	$\langle\rangle$
$\varphi_{s_{u+1}}, \ldots, \varphi_{s_v}$	y_1	y_2	$\langle\rangle$	?	y_4	$\langle\rangle$
$\varphi_{s_{v+1}}, \ldots, \varphi_{s_w}$	y_1	y_2	$\langle\rangle$	y_3	?	$\langle\rangle$
$\varphi_{s_{w+1}}, \ldots, \varphi_{s_{p-2}}$	y_1	y_2	$\langle\rangle$	?	?	$\langle\rangle$
$\varphi_t, \varphi_{s'_{w+1}}, \ldots, \varphi_{s'_{p-3}}$	y_1	y_2	$\langle\rangle$	y_3	y_4	$\langle\rangle$
f_{7m-5}	y_1	y_2	$\langle\rangle$	y_3	y_4	$\langle\rangle$

We compute $\varphi_h(x_1)$, $\varphi_h(x_1+1)$ and the outputs of F on f_{7m-5}. If $m < b+1$ and F outputs a new hypothesis on $f^{[x]}_{7m-5}$ for some x, we let $h \leftarrow F(f^{[x]}_{7m-5})$, $x_0 \leftarrow max(x,y)+1$, output $\langle\rangle$ up to x_0-1, add $j+1, \ldots, j+w$, $j+p-2$ and $j+p-1$ to D, let $s_i \leftarrow s'_{w+i}$ for $1 \le i \le (p-2)/6-1$, let $j \leftarrow j+w$, $p \leftarrow (p-2)/6$ and go to stage $m+1$.
If $\varphi_h(x_1) = y_3$, go to substage 3.
If $\varphi_h(x_1+1) = y_4$, go to substage 4.

- *Substage 3.*
  Execute new(t'), new(s'_i) for $v+1 \le i \le w-1$. Let $y_5 \leftarrow \langle 3, 2m-1, \langle 2, \langle 1, d_1, m, t'\rangle, \ldots, \langle 1, d_r, m, t'\rangle\rangle, z_2\rangle$, $y_6 \leftarrow \langle 3, 2m, \langle 2, \langle 1, j+v+1, m+1, s'_{v+1}\rangle, \ldots, \langle 1, j+w-1, m+1, s'_{w-1}\rangle\rangle, z_4\rangle$. Output values as shown in the next table.

	x_0	x_0+1	$\ldots$	x_1	x_1+1	$\ldots$
$\varphi_{s_1}, \ldots, \varphi_{s_u}, \varphi_{s_{p-1}}$	y_5	y_2	$\langle\rangle$	y_3	y_4	$\langle\rangle$
$\varphi_{s_{u+1}}, \ldots, \varphi_{s_v}$	y_1	y_2	$\langle\rangle$	y_6	y_4	$\langle\rangle$
$\varphi_{s_{v+1}}, \ldots, \varphi_{s_w}$	y_1	y_2	$\langle\rangle$	y_3	?	$\langle\rangle$
$\varphi_{s_{w+1}}, \ldots, \varphi_{s_{p-2}}$	y_1	y_2	$\langle\rangle$	y_6	y_4	$\langle\rangle$
$\varphi_{t'}, \varphi_{s'_{v+1}}, \ldots, \varphi_{s'_{w-1}}$	y_5	y_2	$\langle\rangle$	y_6	y_4	$\langle\rangle$
f_{7m-4}	y_5	y_2	$\langle\rangle$	y_6	y_4	$\langle\rangle$

  Compute outputs of F on f_{7m-4}. If $m < b+1$ and F outputs a new hypothesis on $f^{[x]}_{7m-4}$ for some x, we let $h \leftarrow F(f^{[x]}_{7m-4})$, $x_0 \leftarrow max(x,y)+1$, output $\langle\rangle$ up to x_0-1, add $j+1, \ldots, j+v, j+w, \ldots, j+p-1$ to D, let $s_i \leftarrow s'_{v+i}$ for $1 \le i \le (p-2)/6-1$, let $j \leftarrow j+v$, $p \leftarrow (p-2)/6$ and go to stage $m+1$.
- *Substage 4* is similar to substage 3.
- *Substages 5, 6, 7* are similar to substages 2, 3, 4, respectively.

End of stage m.

j in the algorithm is used as a base index for the arrays that have output their m-th hypotheses (s_i) before stage m was started. Note that the values are output so that the corresponding function f_i is a q-instructor for all $q \in \{1, \ldots, k\}$ except one, so $f_i \in \bigcup_{j=1}^{k} U_j$. Note also that there is no way out of the substages 3, 4, 6 and 7 of stage $b+1$. So the algorithm remains forever in some substage (or stage 0), and, as it is easy to see, the current hypothesis of F has at least two anomalies in comparison with the function f_i, corresponding to this substage (mindchanges after the b-th mindchange made by F are ignored). □

Theorem 12 $(\forall b \in \mathbb{N})[\mathrm{cdeg}(\mathbf{Ex}_b^1) \le \frac{7 \cdot 6^{b+1}+3}{5}]$.

Proof. Denote $k = \frac{7 \cdot 6^{b+1}+3}{5}$, $l = \frac{7 \cdot 6^{b}+3}{5}$.

Consider classes $U_1, \ldots, U_k$ such that the unions of $k-1$ classes out of them are $\mathbf{Ex}_b^1$-identified by strategies $F_1, \ldots, F_k$. We shall construct such strategy F that will identify $\bigcup_{j=1}^{k} U_j$ using $F_1, \ldots, F_k$ as subroutines.

Denote the input function by f. Strategy F simulates the strategies $F_1, \ldots, F_k$ on f. If $f \in \bigcup_{j=1}^{k} U_j$, then f is identified by at least $k-1$ of these strategies. So F waits until $k-1$ strategies make their first hypotheses. Suppose the strategies are $F_1, \ldots, F_{k-1}$, and their hypotheses are $h_1, \ldots, h_{k-1}$. Then F outputs its own first hypothesis h based on these strategies and their hypotheses.

Suppose $b > 0$ and $l-1$ out of these $k-1$ strategies output another hypothesis. Then F outputs its second hypothesis, based on these $l-1$ strategies together with their hypotheses, and we have reduced our problem to the case of $\mathbf{Ex}_{b-1}^1$-identification.

So it is enough to prove that, if no more than $l-2$ strategies make another hypothesis, or $b = 0$, then hypothesis h is correct.

In this case there is at most one strategy among $F_1, \ldots, F_{k-1}$ that does not identify f and at most $l-2$ strategies that identify f, but output another hypothesis. So no more than $l-1$ hypotheses among $h_1, \ldots, h_{k-1}$ are wrong.

Now we describe the algorithm for φ_h. It computes the following infinite table and the hypotheses made by the F_i's on all possible initial segments.

	0	$\ldots$	n	$\ldots$
φ_{h_1}	$\varphi_{h_1}(0)$	$\ldots$	$\varphi_{h_1}(n)$	$\ldots$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
$\varphi_{h_{k-1}}$	$\varphi_{h_{k-1}}(0)$	$\ldots$	$\varphi_{h_{k-1}}(n)$	$\ldots$

Let the weight of a value in a column be the number of occurrences of this value in the column. We shall say that values u and v in different columns are p-coordinated iff there are p rows that have u and v in the corresponding columns.

The aim is to find a consistent interpretation of the table, that is, such initial subtable, such $l_0 \le l$ and such initial segment $g^{[n]}$ that $l_0 - 2$ of strategies $F_1, \ldots, F_{k-1}$ output the second hypothesis on a subsegment of $g^{[n]}$ and there are at least $k - l_0$ rows in the subtable that have no more than one anomaly in comparison with $g^{[n]}$. Such interpretations will be found for all but finitely many n, because the initial segments of f give consistent interpretations starting with the segment on which the last of the second hypotheses is output.

When an interpretation is found, φ_h outputs values (those that are not already output) according to the following rules.

1. Value u is output if its weight is at least $(k-1)/2$ and it is l-coordinated with all the values already output.

2. Value u is output if its weight is at least $\lfloor (k - l_0 + 1)/2 \rfloor$, it is equal to the corresponding value of g and it is l-coordinated with all the values already output.

3. Value u_1 is output at point x_1 if it is l-coordinated with all the values already output and there is a column x_2 such that:

   (a) at point x_2 a value u_2 has been output;

   (b) there is another value $v_2 \neq u_2$ in column x_2 such that, denoting the numbers of rows that have the corresponding values at points x_1 and x_2 as in the table:

Number of rows	Value at x_1	Value at x_2
s_1	u_1	u_2
s_2	?!	u_2
s_3	u_1	?!
s_4	?!	?!
s_5	u_1	v_2
s_6	?!	v_2

   (here "?!" means "undefined value or value different from mentioned at this column"), the following inequalities are obeyed:

   $$s_1 + s_2 \geq \frac{k-l}{2}, \tag{4.1}$$

   $$s_2 + s_4 \leq l - 1; \tag{4.2}$$

4. Suppose there are columns x_1 and x_2 such that:

   (a) column x_1 contains two different values u_1 and v_1 (and maybe some other values);

   (b) either u_1 has been output at x_1, or no value has been output at x_1 and no value has been output according to rule 3;

(c) column x_2 contains two different values u_2 and v_2 (and maybe some other values) and u_2 has been output at x_2;

(d) denoting the numbers of rows that have the corresponding values at points x_1 and x_2 as in the table:

Number of rows	Value at x_1	Value at x_2
t_1	u_1	u_2
t_2	v_1	u_2
t_3	?!	u_2
t_4	u_1	?!
t_5	v_1	?!
t_6	?!	?!
t_7	u_1	v_2
t_8	v_1	v_2
t_9	?!	v_2

the following inequalities are obeyed:

$$t_1 + t_2 + t_3 \geq \frac{k-l}{2}, \tag{4.3}$$

$$t_2 + t_3 + t_5 + t_6 \leq l - 1, \tag{4.4}$$

$$t_8 \geq 2l - 1. \tag{4.5}$$

Then the algorithm outputs u_1 at x_1 if necessary, and further it outputs a value iff it is in at least l of t_8 rows that contain both v_1 and v_2 at x_1 and x_2, respectively (any output according to the previous rules is terminated).

To prove that $\varphi_h =^1 f$ in case no more than $l-2$ strategies change their hypotheses, we shall consider some cases.

1. *φ_h has output value according to rule 4.* We shall use the notation introduced in this rule. Since no more than $l-1$ hypotheses among $h_1, \ldots, h_{k-1}$ are wrong, if two values are l-coordinated, then no more than one of them is incorrect. We get from (4.3), (4.4) and the equality $k = 6l - 3$ that $t_1 \geq (k-l)/2 - l + 1 = 3l/2 - 1/2 \geq l$, so no more than one of the values u_1 and u_2 is incorrect. Since $t_8 \geq 2l - 1 \geq l$ (inequality 4.5), no more than one of the values v_1 and v_2 is incorrect. Combining these two conclusions we get that exactly one of the values u_1 and u_2 is incorrect and exactly one of the values v_1 and v_2 is incorrect. The latter implies that all of the $2l-1$ or more rows containing v_1 and v_2 except at most $l-1$ rows will contain the correct values at all other points, and according to rule 4 these values will be output.

Suppose an incorrect value u_3 has been previously output at some point x_3 different from x_1 and x_2. Then it was l-coordinated with u_2, therefore u_2 is correct, u_1 and v_2 are incorrect, and v_1 is correct. Hence $k-1-t_2$ rows have already at least one anomaly at columns x_1 and x_2, so the weight of u_3 does not exceed $t_2+l-1 \leq 2l-2 < (k-l)/2$, therefore u_3 was output according to rule 3. According to condition in rule 4, u_1 has been already output, so u_1 is l-coordinated with u_3. Since both these values are incorrect, we have a contradiction. Thus the only error made by φ_h is either u_1, or u_2.

2. *φ_h at least once has output value according to rule 3, but never according to rule 4.* Considering the first value output according to rule 3 we shall use the notation of this rule.

   (a) *The weight of u_1 in x_1 after all the defined values are computed in x_1 turns out to be no less than $(k-1)/2$.* Then u_1 satisfies the conditions of rule 1, and we can consider it to be output according to that rule. That case will be considered further below.

   (b) *The weight of u_1 in x_1 never exceeds $(k-1)/2-1$.* Let s_i denote the numbers defined in the table above in the situation when all the defined values in columns x_1 and x_2 have been computed. Note that, when new values are computed, s_1+s_2 cannot decrease, while s_2+s_4 cannot grow, so inequalities (4.1) and (4.2) remain satisfied. Our assumption implies

$$s_1+s_3+s_5 \leq \frac{k-1}{2}-1. \tag{4.6}$$

   Inequalities (4.1) and (4.2) imply

$$s_1 \geq \frac{k-l}{2}-l+1 = \frac{3l-1}{2}. \tag{4.7}$$

   Hence u_1 and u_2 are l-coordinated. Inequalities (4.2), (4.6) and equality $\sum_{i=1}^{6} s_i = k-1$ imply $s_4+s_6 \geq s_6 \geq (k-1)/2-l+2 = 2l \geq l$. So non-$u_1$ and non-u_2 values are also l-coordinated. Therefore, one of the values u_1 and u_2 is correct, and the other is incorrect.

      i. *u_1 is correct, u_2 is incorrect.* From (4.6) and (4.7) we get that $s_3+s_5 \leq (3l-5)/2$. So the amount of rows that have error in x_1 or x_2 is no less than $s_1+s_2+s_4+s_6 \geq k-1-(3l-5)/2 = (9l-3)/2$. At least $(9l-3)/2-(l-1) = (7l-1)/2 \geq (k-1)/2$ of these have correct values at all other columns, and using (4.1) we get that at least $s_1+s_2-(l-1) \geq l$ of them have u_1 at x_1 and u_2 at x_2, so the correct values are l-coordinated between themselves, with u_1 and u_2, while the incorrect values cannot be l-coordinated with u_2. Thus all the correct values are output according to rule 1, and φ_h has only one error, that is u_2.

ii. *u_1 is incorrect, u_2 is correct.*

A. $s_1 \geq 2l-1$. Then at least $(k-1)-s_2 \geq (k-1)-(l-1) = 5l-3$ rows have an error at x_1 or x_2, and at least $5l-3-(l-1) = 4l-2 \geq (k-1)/2$ of them have correct values at all other columns. Among them are at least $s_1-(l-1) \geq l$ rows that have u_1 at x_1 and u_2 at x_2. So the correct values are l-coordinated between themselves, with u_1 and u_2, and they are output according to rule 1 (the incorrect values cannot be output since they are not l-coordinated with u_1). φ_h has only one error, u_1.

B. $s_1 \leq 2l-2$. Let v_1 be the correct value at x_1, s_6' be the number of rows that have v_1 at x_1 and v_2 at x_2, $s_6'' = s_6 - s_6'$. Since no more than $l-1$ rows can have two errors, $s_3+s_4+s_5+s_6'' \leq l-1$. Applying this inequality as well as (4.2) and the assumption, we get $s_6' = (k-1)-s_1-s_2-s_3-s_4-s_5-s_6'' \geq (k-1) - 4\cdot(l-1) = 2l > 2l-1$. But then columns x_1 and x_2 satisfy the conditions of rule 4; this case was considered above.

3. *All the values output by φ_h satisfy the conditions of rule 1 or rule 2.*

(a) *There are two incorrect values output by φ_h.* That is impossible, since all output values are l-coordinated.

(b) *Among the values output by φ_h there is an incorrect value u_1 at some column x_1, and $\varphi_h(x_2)$ is undefined for some x_2.* Let $v_1 = f(x_1)$, $u_2 = f(x_2)$. Let w_1 be the weight of u_1 at x_1, w_2 the weight of v_1 at x_1, and $w_3 = (k-1)-w_1-w_2$. Then $w_1 \geq (k-l)/2$. So there are at least $w_1+w_3-(l-1) \geq (3l-1)/2 \geq l$ rows that have an error at x_1 and correct values at all other columns, including x_2. Therefore u_2 is l-coordinated with all the correct values (maybe except v_1) and with u_1 and thus satisfies conditions of one of the rules 1, 2 and 3. There is a problem, though. Maybe at every interpretation considered by φ_h u_2 was not l-coordinated with *the computed part* of a correct value output in some column. At further interpretations all the defined values at this column become computed, and u_2 becomes l-coordinated with the correct value, but now u_2 can have the same conflict with another column, etc.

Let us consider such interpretation applied by φ_h with initial segment $g^{[x]}$ modelling f that all the defined values at x_1, x_2 are computed, u_2 is not l-coordinated with some previously output value u_3 at x_3, and in the next interpretation considered by φ_h u_2 is already l-coordinated with u_3. If $g(x_1) \neq u_1$, with the same reasoning as above we get that u_2 and u_3 must be l-coordinated. So, $g(x_1) = u_1$.

i. $g(x_2) = v_2 \neq u_2$.

A. $g(x_3) \neq u_3$. Let w' be the weight of u_3 at x_3, $w' \geq (k-l)/2$. There are at least $w' - (l-1) \geq (3l-1)/2 \geq l$ rows whose only error in this interpretation is u_3 at x_3, so they have u_1 at x_1 and v_2 at x_2. Thus u_1 and v_2 are l-coordinated. That is a contradiction, since in fact u_1 and v_2 are both incorrect.

B. $g(x_3) = u_3$. Then the weight of u_2 at x_2 does not exceed $2l-2$, otherwise u_2 and u_3 would be l-coordinated, contrary to the assumption. Since $v_1 = f(x_1)$ and $u_2 = f(x_2)$, the number of rows that have not u_2 at x_2 and have v_1 at x_1, is at least $(k-1)-(2l-2)-(l-1) = 3l-1$. Since $v_1 \neq g(x_1)$ and $v_2 = g(x_2)$, the number of rows that have v_1 at x_1 and v_2 at x_2 is at least $3l-1-(l-1) = 2l$. But then columns x_2 and x_1 (in this order) satisfy the conditions of rule 4; this case was considered above.

ii. $g(x_2) = u_2$. Since $f(x_1) \neq u_1 = g(x_1)$, the weight of u_2 is at least $(w_1 - (l-1)) + (w_2 + w_3 - (l-1)) = 4l - 2$.

A. $g(x_3) \neq u_3$. Since the weight of u_3 exceeds $2l-1$, u_3 is l-coordinated with u_2, contrary to the assumption.

B. $g(x_3) = u_3$. Since u_3 is not l-coordinated with u_2, at least $4l-2-(l-1) = 3l-1$ rows have u_2 at x_2 and an error at x_3 (in this interpretation), and at least $3l-1-(l-1) = 2l \geq l$ of them have no other errors, so they have u_1 at x_1, u_2 at x_2 and values that are correct in relation to both f and g at all other columns, except x_3. According to the assumption, in the next interpretation u_2 becomes l-coordinated also with u_3, so it will be output then (according to the algorithm, the new columns of the new interpretation will be considered only after x_2).

(c) *There are two points x_1 and x_2 at which φ_h is undefined.*

Let α be an interpretation in which all the defined values at x_1 and x_2 have been computed, let $g^{[x]}$ be the initial segment modelling f in α. Let the number of strategies that have changed their hypothesis on $g^{[x]}$ be $l_0 - 2$ (we are interested only in the case $l_0 \leq l$), $u_1 = g(x_1)$, $u_2 = g(x_2)$. Then the number of rows that have no more than one error in α is at least $(k-1)-(l_0-1) = k - l_0$, and at least one of the values u_1 and u_2 have weight at least $\lfloor (k-l_0+1)/2 \rfloor$; let u_1 be this value.

i. *The weight of u_2 at x_2 is less than $\lfloor (k-l_0+1)/2 \rfloor$.* Then at least $(k-1)/2+1 = 3l-1$ rows have not u_2 at x_2, so at least $3l-1-(l-1) = 2l \geq l$ of them have α-correct values at all other columns, u_1 at x_1 among them. Since u_1 is not output, it is not l-

coordinated with some previously output u_3 at x_3, and $g(x_3) \neq u_3$. Since u_3 was output according to rule 1 or rule 2, it has weight at least $(k-l)/2 = (5l-3)/2$, so there are $(5l-3)/2 - (l-1) = (3l-1)/2 \geq l$ rows in which u_3 is coordinated with all the α-correct values, u_1 among them. Contradiction.

ii. *The weight of u_2 at x_2 is at least $\lfloor (k-l_0+1)/2 \rfloor$.* Then u_1 and u_2 both satisfy the conditions of rule 2. Since they are not output, they are not l-coordinated with some previously output value(s).

A. *Both u_1 and u_2 are not l-coordinated with some value u_3 output at x_3.* If $u_3 \neq g(x_3)$, then as previously we get that u_1 and u_2 are l-coordinated with u_3. So $u_3 = g(x_3)$. Suppose u_3 has weight at least $(k-1)/2$. Then the number of rows that have u_3 at x_3 and no more than one α-error, is at least $(k-1)/2 - (l-1) = 2l-1$. These rows have either u_1 at x_1, or u_2 at x_2, so u_3 is l-coordinated with at least one of these values.

Suppose u_3 has weight less than $(k-1)/2$. Then it was output according to rule 2. Let us consider interpretation α' with the initial segment $g'^{[x']}$ $(x' \leq x)$ at which u_3 was output. According to rule 2, $u_3 = g'(x_3)$. Suppose $(\exists x_0 \leq x')[g(x_0) \neq g'(x_0)]$. Let $u_0 = g'(x_0)$, $v_0 = g(x_0)$. Let r be the number of rows that have not u_3 at x_3, then $r \geq (k-1)/2+1$. According to α', at least $r-(l-1)$ of these rows have u_0 at x_0; according to α, at least $r-(l-1)$ of these rows have v_0 at x_0. Since $r > 2l-2$, we get a contradiction. Therefore such x_0 does not exist, and $g^{[x]}$ is an extension of $g'^{[x']}$. Some of the $l_0 - 2$ strategies that have changed their hypotheses at α have changed them already at α'. Let their number be $l_1 - 2$ $(l_1 \leq l_0)$. The weight of u_3 is at least $\lfloor (k-l_1+1)/2 \rfloor$. The number of rows that have u_3 at x_3 and no more than one α-error at all other columns is at least $\lfloor (k-l_1+1)/2 \rfloor - 1 - (l_0 - l_1) = \lfloor (k+l_1-1)/2 \rfloor - l_0 \geq (k+1)/2 - l = 2l-1$. Each of these rows have either u_1 at x_1, or u_2 at x_2, so u_3 is l-coordinated with at least one of these values, contrary to the assumption.

B. *u_1 is not l-coordinated with some previously output u_3 at x_3, u_2 is not l-coordinated with some previously output u_4 at $x_4 \neq x_3$.* As previously, if $u_3 \neq g(x_3)$, then u_3 would be l-coordinated with u_1. So $u_3 = g(x_3)$. Similarly, $u_4 = g(x_4)$. Since u_1 and u_3 are not l-coordinated, there are at least $(k-1)-(l-1)$ rows that have an α-error either at x_1, or at x_3. At least $(k-1) - 2\cdot(l-1) \geq l$ of them have no other α-errors, so they have u_2 at x_2 and u_4 at x_4, contrary to the assumption.

□

Corollary 4.2 $(\forall b \in \mathbb{N})[\mathrm{cdeg}(\mathbf{Ex}_b^1) = \frac{7 \cdot 6^{b+1}+3}{5}]$.

Theorem 13 $(\forall a \in \mathbb{N} \mid a > 1)(\forall b \in \mathbb{N})[\mathrm{cdeg}(\mathbf{Ex}_b^a) = \infty]$.

Proof. Let $a > 1, b$ be some natural numbers. It is enough to prove for an arbitrary $k \geq 2$ that there are such classes $U_1, \ldots, U_k$ that the unions of $k-1$ classes out of them are identifiable, while $\bigcup_{j=1}^k U_j$ is not.

We define $U_i = (\bigcap_{j=1, j\neq i}^k I_j^{\mathbf{Ex}_b^a})$, $1 \leq i \leq k$. Then $\bigcup_{i=1, i\neq j}^k U_i \subseteq I_j^{\mathbf{Ex}_b^a} \in \mathbf{Ex}_b^a$, $1 \leq j \leq k$.

We have to prove that $\bigcup_{i=1}^k U_i \notin \mathbf{Ex}_b^a$. We use diagonalization over the strategies. Let F be an arbitrary strategy. Using the multiple recursion theorem similarly as in Theorem 10 we construct functions φ_{n_i}, $1 \leq i \leq N$ (where N is a natural number determined only by a and b) that can use each others Gödel numbers incorporated in their values. With their help we shall construct a function $f \in \bigcup_{i=1}^k U_i$ not identified by F.

We remind that the procedure new(x) lets $x \leftarrow n_c$, $c \leftarrow c+1$, where c is a counter in the algorithm.

The algorithm for φ_{n_i} is as follows.

- *Stage 0.*
  Let $c \leftarrow 1$. Execute new(s_i) for $1 \leq i \leq k$. Output values as shown in the next table.

	0	$\ldots$	$k-1$	$\ldots$
$\varphi_{s_1}, \ldots, \varphi_{s_k}$	$\langle 1,1,1,s_1\rangle$	$\ldots$	$\langle 1,k,1,s_k\rangle$	$\langle\rangle$
f_0	$\langle 1,1,1,s_1\rangle$	$\ldots$	$\langle 1,k,1,s_k\rangle$	$\langle\rangle$

  Let the variable y throughout this algorithm indicate the maximal value of argument at which the values have been output at the moment. Simulate the strategy F on the initial segments of f_0. If a hypothesis is output on $f_0^{[x]}$, we let $h \leftarrow F(f_0^{[x]})$, $x_0 \leftarrow \max(x, y) + 1$; we output $\langle\rangle$ up to $x_0 - 1$, if needed, and go to stage 1.

- *Stage r $(1 \leq r \leq b+1)$.*

  - *Substage 0.*
    Output values as shown in the next table.

	x_0	$\ldots$	x_0+a-1	x_0+a	$\ldots$
φ_{s_1}	?	?	?	?	$\langle\rangle$
$\varphi_{s_2}, \ldots, \varphi_{s_k}$	?	?	?	$\langle\rangle$	$\langle\rangle$
$f_{r,0}$	$\langle\rangle$	$\langle\rangle$	$\langle\rangle$	$\langle\rangle$	$\langle\rangle$

    Compute $\varphi_h(x_0), \ldots, \varphi_h(x_0+a)$, and, if $r < b+1$, simulate F on $f_{r,0}$. If $\varphi_h(x) = \langle\rangle$ gets computed for $x_0 \leq x \leq x_0+a-1$, change $f_{r,0}$ to be

different from this value: $f_{r,0}(x) = \langle 0\rangle$, and restart simulating F on $f_{r,0}$ as well as computing φ_h at other points.
If F makes a mindchange on $f_{r,0}^{[x]}$ for some x, we let $h \leftarrow F(f_{r,0}^{[x]})$, $x_0 \leftarrow \max(x,y)+1$, $\sigma \leftarrow f_{r,0}^{[x_0]}$, and go to substage k.
If $\varphi_h(x_0+a) = \langle\rangle$ is computed, we let $x_1 \leftarrow x_0+a$, $x_2 \leftarrow y+1$, output $\varphi_{s_i}(x_0+a-2) = f_{r,0}(x_0+a-2)$, $\varphi_{s_i}(x_0+a-1) = f_{r,0}(x_0+a-1)$ for $1 \le i \le k$, $\langle\rangle$ up to x_2-1, if needed, and go to substage 1.

- *Substage t ($1 \le t \le k-2$).*
Output values as shown in the next table.

	x_t	$\ldots$	x_{t+1}	$x_{t+1}+1$	$x_{t+1}+2$	$\ldots$
$\varphi_{s_1},\ldots,\varphi_{s_t}$	?	$\langle\rangle$	?	$\langle\rangle$	$\langle\rangle$	$\langle\rangle$
$\varphi_{s_{t+1}}$	$\langle\rangle$	$\langle\rangle$	?	?	?	$\langle\rangle$
$\varphi_{s_{t+2}},\ldots,\varphi_{s_k}$	$\langle\rangle$	$\langle\rangle$	$\langle\rangle$	?	?	$\langle\rangle$
$f_{r,t}$	$\langle\rangle$	$\langle\rangle$	$\langle\rangle$	$\langle\rangle$	$\langle\rangle$	$\langle\rangle$

We compute $\varphi_h(x_0)$, $\ldots$, $\varphi_h(x_0+a-3)$, $\varphi_h(x_{t+1})$, $\varphi_h(x_{t+1}+1)$, $\varphi_h(x_{t+1}+2)$ and the outputs of F on $f_{r,t}$.
If $\varphi_h(x) = \langle\rangle$ gets computed for $x_0 \le x \le x_0+a-3$, change $f_{r,t}$ to be different from this value: $f_{r,t}(x) = \langle 0\rangle$, and restart simulating F on $f_{r,t}$ as well as computing φ_h at other points.
If $r < b+1$ and F changes its current hypothesis on $f_{r,t}^{[x]}$ for some x, we let $h \leftarrow F(f_{r,t}^{[x]})$, $x_0 \leftarrow \max(x,y)+1$, $\sigma \leftarrow f_{r,t}^{[x_0]}$, and go to substage k.
If $\varphi_h(x_{t+1}) = \langle\rangle$ is computed, we let $x_{t+2} \leftarrow y+1$, output $\varphi_{s_i}(x_t) = \varphi_{s_i}(x_{t+1}+1) = \varphi_{s_i}(x_{t+1}+2) = \langle\rangle$ for $1 \le i \le k$ (where the values have not been already output), $\langle\rangle$ up to $x_{t+2}-1$, if needed, and go to substage $t+1$.
The cases when $\varphi_h(x_{t+1}+1) = \langle\rangle$ or $\varphi_h(x_{t+1}+2) = \langle\rangle$ is computed are similar; we shall describe the first case.
Thus, if $\varphi_h(x_{t+1}+1) = \langle\rangle$ is computed, let $x' \leftarrow y+1$, and output values as in the next table.

	x_t	$\ldots$	$x_{t+1}+1$	$x_{t+1}+2$	$\ldots$	x'	$\ldots$
$\varphi_{s_1},\ldots,\varphi_{s_t}$	$\langle 0\rangle$	$\langle\rangle$	$\langle\rangle$	$\langle\rangle$	$\langle\rangle$	?	$\langle\rangle$
$\varphi_{s_{t+1}},\ldots,\varphi_{s_k}$	$\langle\rangle$	$\langle\rangle$	$\langle 0\rangle$	$\langle\rangle$	$\langle\rangle$	?	$\langle\rangle$
$f'_{r,t}$	$\langle 0\rangle$	$\langle\rangle$	$\langle 0\rangle$	$\langle\rangle$	$\langle\rangle$	$\langle\rangle$	$\langle\rangle$

Compute $\varphi_h(x_0),\ldots,\varphi_h(x_0+a-3)$, $\varphi_h(x')$ and the outputs of F on $f'_{r,t}$.
If $\varphi_h(x) = \langle\rangle$ gets computed for $x_0 \le x \le x_0+a-3$ or $x = x'$, change $f'_{r,t}$ to be different from this value: $f'_{r,t}(x) = \langle 0\rangle$, and restart simulating F on $f'_{r,t}$ as well as computing φ_h at other points.

If $r < b + 1$ and F changes its current hypothesis on $f_{r,t}'^{[x]}$ for some x, we let $h \leftarrow F(f_{r,t}'^{[x]})$, $x_0 \leftarrow \max(x, y)+1$, $\sigma \leftarrow f_{r,t}'^{[x_0]}$, and go to substage k.

– *Substage* $k - 1$.
Output values as shown in the next table.

	x_{k-1}	$x_{k-1}+1$	$x_{k-1}+2$	$\ldots$	x_k	$\ldots$
$\varphi_{s_1}, \ldots, \varphi_{s_{k-1}}$	$\langle 0\rangle$	$\langle\rangle$	$\langle\rangle$	$\langle\rangle$	?	$\langle\rangle$
φ_{s_k}	$\langle\rangle$	$\langle\rangle$	$\langle\rangle$	$\langle\rangle$	?	$\langle\rangle$
$f_{r,k-1}$	$\langle 0\rangle$	$\langle\rangle$	$\langle\rangle$	$\langle\rangle$	$\langle\rangle$	$\langle\rangle$

Compute $\varphi_h(x_0), \ldots, \varphi_h(x_0 + a - 3)$, $\varphi_h(x_{k-1} + 1)$, $\varphi_h(x_k)$ and the outputs of F on $f_{r,k-1}$.
If $\varphi_h(x) = \langle\rangle$ gets computed for $x_0 \le x \le x_0 + a - 3$, $x = x_{k-1} + 1$ or $x = x_k$, change $f_{r,k-1}$ to be different from this value: $f_{r,k-1}(x) = \langle 0\rangle$, and restart simulating F on $f_{r,k-1}$ as well as computing φ_h at other points.
If $r < b + 1$ and F changes its current hypothesis on $f_{r,k-1}^{[x]}$ for some x, we let $h \leftarrow F(f_{r,k-1}^{[x]})$, $x_0 \leftarrow \max(x, y) + 1$, $\sigma \leftarrow f_{r,k-1}^{[x_0]}$, and go to substage k.

– *Substage* k.
Execute $\mathrm{new}(s_i)$ for $1 \le i \le k$. Output values up to $x_0 - 1$ so that $\varphi_{s_i}^{[x_0-1]} = \sigma$ and further as shown in the next table.

	x_0	$\ldots$	$x_0 + k - 1$
$\varphi_{s_1}, \ldots, \varphi_{s_k}$	$\langle 1, 1, r+1, s_1\rangle$	$\ldots$	$\langle 1, k, r+1, s_k\rangle$
f_r	$\langle 1, 1, r+1, s_1\rangle$	$\ldots$	$\langle 1, k, r+1, s_k\rangle$

Let $x_0 \leftarrow x_0 + k$ and go to stage $r + 1$.

End of stage r.

Each stage in this algorithm deals with one hypothesis made by F. It makes the current hypothesis function to have at least $a + 1$ anomalies by forcing it to output values based on smaller and smaller evidence from the functions φ_{s_i} (substages 0 to $k - 2$). When F makes a mindchange, we disregard the previous functions φ_{s_i} by choosing new values for s_i, remembering only the segment σ on which F made the mindchange. So, either the last hypothesis output by F has at least $a + 1$ anomalies, or F makes at least $b + 1$ mindchanges. □

4.4 Closedness in Superset

In this section we shall turn our attention to the general case of establishing $\mathrm{csdeg}(\mathbf{Ex}_b^a, \mathbf{Ex}_d^c)$. According to Corollary 3.1 we have $\mathrm{csdeg}(\mathbf{Ex}_b^a, \mathbf{Ex}_d^c) = \infty$, in case $c < a$ or $d < b$. Some other results important in this section are contained in Propositions 3.5, 3.4, Theorems 6, 9 and 10. The next theorem have been obtained from a result in team learning.

Theorem 14 [19] $(\forall a, b \in \mathbb{N})(\forall c, d \in \mathbb{N} \cup \{*\} \mid c \geq a \wedge d \geq \lfloor \frac{a}{c-a+1} + 2 \rfloor (2b + 1))[\mathrm{csdeg}(\mathbf{Ex}_b^a, \mathbf{Ex}_d^c) = 2]$.

It shows that, given a and b, by choosing c and d large enough we can make the closedness degree for classes $\mathbf{Ex}_b^a$ and $\mathbf{Ex}_d^c$ reach its minimal value 2. From Theorems 9 and 10 we get the next result showing that, if we do not change the number of mindchanges, then the minimum reached by csdeg is different.

Theorem 15 $(\forall b \in \mathbb{N})(\forall a, a' \in \mathbb{N} \cup \{*\} \mid a' \geq 2^{b+1}a)[\mathrm{csdeg}(\mathbf{Ex}_b^a, \mathbf{Ex}_b^{a'}) = 2^{b+2}]$.

Now we shall turn our attention to estimating $\mathrm{csdeg}(\mathbf{Ex}_b, \mathbf{Ex}_d)$, $b < d < *$. By evaluating the influence of anomalies in the algorithms we shall obtain results for $\mathrm{csdeg}(\mathbf{Ex}_b^a, \mathbf{Ex}_d^c)$ for sufficiently large c relative to a.

At first, not a very exact upper bound.

Theorem 16 $(\forall k \geq 3)(\forall a, c \in \mathbb{N} \cup \{*\} \mid c \geq (\lfloor k/2 \rfloor + 1) \cdot a)(\forall b, d \in \mathbb{N} \mid d \geq 2b - 1 + \lfloor (2b+4)/(k-1) \rfloor)[\mathrm{csdeg}(\mathbf{Ex}_b^a, \mathbf{Ex}_d^c) \leq k]$.

Proof. At first, let us note that a similar result for $k = 2$: $\mathrm{csdeg}(\mathbf{Ex}_b^a, \mathbf{Ex}_d^c) = 2$ for $c \geq 2a$, $d \geq 4b + 2$, follows from Theorem 14.

Let $U_1, \ldots, U_k$, $k \geq 3$, be such classes that all the unions of $k - 1$ out of them are $\mathbf{Ex}_b^a$-identifiable. Let F_j, $1 \leq j \leq k$ be a strategy that $\mathbf{Ex}_b^a$-identifies $\bigcup_{i=1, i \neq j}^k U_i$. Note that each function from $\bigcup_{i=1}^k U_i$ is identified by at least $k - 1$ of the strategies F_j. Consider the following strategy F.

At first we describe, when F outputs hypotheses. F simulates the strategies F_j on the input function f. F waits until $k - 1$ strategies among F_j output their first hypotheses, we denote them $h_0^1, \ldots, h_0^{k-1}$, then F outputs hypothesis h_0 based on these hypotheses. In further, F computes the values output by the current hypotheses of F_j counting the number of incorrect values output by them and continues to simulate F_j on f. F does not consider $(b + 2)$-th and further hypotheses, in case such are output by some of the strategies F_j.

Suppose k is even. Suppose F has output its previous hypothesis h_{i-1} based on $k - 1$ hypotheses. F outputs a new hypothesis h_i iff at least $k/2$ of the strategies F_j output new hypotheses. h_i is based an all k current hypotheses of F_j.

Suppose the previous hypothesis h_{i-1} was based on k strategies. Then F outputs a new hypothesis h_i iff either $k/2 - 1$ of the strategies F_j output a new hypothesis and among the last hypotheses of the remaining $k/2 + 1$ strategies there is one, h', that has output incorrect values at least at $a + 1$ points, or if $k/2$ strategies output a new hypothesis. In the former case h_i is based on all the current hypotheses except h', in total on $k-1$ hypotheses. In the latter case h_i is based on all k current hypotheses.

Suppose k is odd. Then the previous hypothesis h_{i-1} was based on $k-1$ hypotheses. F outputs a new hypothesis iff either $(k-1)/2$ of the strategies on which hypotheses h_{i-1} was based output a new hypothesis, or if a new hypothesis is output by $(k-1)/2$ strategies including the strategy on whose hypothesis h_{i-1} was not based, and among the current hypotheses of the remaining $(k+1)/2$ strategies there is one, h', that has output incorrect values at least at $a+1$ points. In the former case h_i is based on the current hypotheses of the same strategies on which h_{i-1} was based. In the latter case h_i is based on all the current hypotheses except h'.

If at least 3 strategies output their $(b+1)$-th hypotheses, then F outputs its own last hypothesis based on them. If two strategies output their $(b+1)$-th hypotheses and one of these hypotheses output incorrect values at least at $a+1$ points, then F outputs its last hypothesis based on the other hypothesis.

Now we describe the algorithm for φ_{h_i}. Let l be the number of hypotheses it was based on, $l=k-1$ or $l=k$, and let $h^1,\ldots,h^l$ be these hypotheses. φ_{h_i}, receiving as input x, simulates $\varphi_{h^1}(x),\ldots,\varphi_{h^l}(x)$. When at least $\lfloor (l+1)/2 \rfloor$ among these functions output the same value y, $\varphi_{h_0}(x)$ outputs y, too. Note that, if l is even, $\lfloor (l+1)/2 \rfloor = l/2$, so the value output by φ_{h_0} could depend on the order in which the values $\varphi_{h^j}(x)$ are output.

It is easy to see that, if the conditions for outputting h_i are not obeyed, the previous hypothesis h_{i-1} is a correct **Ex***-hypothesis, since at all but a finite amount of points majority of the functions on which h_{i-1} was based output correct values. Similarly, the last hypothesis output by F is correct, since at most one of the $(b+1)$-th hypotheses of strategies F_j can be incorrect.

To conclude proof, we have to count, how many mindchanges F will have and how many anomalies its hypotheses h_i can have. Suppose k is even. Let us count, how many new hypotheses by F_j are needed for each hypothesis by F. For h_0 $k-1$ hypotheses are needed. For h_1 $k/2$ hypotheses are needed. In further, for each h_i at least $k/2-1$ hypotheses are needed, and for two consecutive hypotheses h_i, h_{i+1} $k-1$ hypotheses are needed. When F_j have output $bk+3$ hypotheses, at least 3 $(b+1)$-th hypotheses have appeared. So the number of mindchanges made by F does not exceed the minimal d for which the inequality $k-1+k/2+(d-1)(k-1)/2 \geq bk+3$ holds. The inequality implies $d \geq (2bk-2k+7)/(k-1)$ and, since d is integer,

$$d \geq \left\lceil \frac{2bk-2k+7}{k-1} \right\rceil = \left\lfloor \frac{2bk-k+5}{k-1} \right\rfloor = 2b-1+\left\lfloor \frac{2b+4}{k-1} \right\rfloor .$$

Suppose k is odd. Then for h_0 $k-1$ hypotheses are needed, and for h_i, $i>0$, $(k-1)/2$ hypotheses are needed. In this case d is determined by the inequality

$k-1+d\cdot(k-1)/2 \geq bk+3$. It implies $d \geq (2bk-2k+8)/(k-1)$ and

$$d \geq \left\lceil \frac{2bk-2k+8}{k-1} \right\rceil = \left\lfloor \frac{2bk-k+5}{k-1} \right\rfloor = 2b-1+\left\lfloor \frac{2b+4}{k-1} \right\rfloor .$$

(We used the fact that, for $k \geq 3$ an odd integer and l an integer, $\lceil l/(k-1)\rceil = \lfloor (l+k-3)/(k-1)\rfloor$.)

Suppose h_{i_0} is based on l hypotheses, and it is the last hypothesis output by F. Suppose l is odd. Then at least $(l+1)/2 = \lfloor l/2\rfloor + 1 > l/2$ of these l hypotheses are correct $\mathbf{Ex}^a$-hypotheses, and $\varphi_{h_{i_0}}$ can have anomalies only at no more than $(\lfloor l/2\rfloor + 1)\cdot a$ points where at least one of these correct hypotheses have an anomaly. Now, suppose l is even. Then at least $l/2$ hypotheses are correct $\mathbf{Ex}^a$-hypotheses, and at least one of the remaining hypotheses, h', outputs no more than a incorrect values, though it can be undefined at any number of points. But at the points this function is undefined, any incorrect value can gather no more than $l/2-1$ "votes" among the incorrect hypotheses. So $\varphi_{h_{i_0}}$ can have anomalies only at the points where either one of the correct hypotheses have an anomaly, or where $\varphi_{h'}$ outputs an incorrect value, that is at no more than $(l/2+1)\cdot a = (\lfloor l/2\rfloor+1)\cdot a$ points. □

This estimation has two flaws: (1) it gives upper bounds only for csdeg($\mathbf{Ex}_b$, $\mathbf{Ex}_d$) where $d \geq 2b-1$, and (2) as we shall see below these upper bounds are not simultaneously lower bounds. We shall see also that finding exact csdeg values is a rather difficult task.

Nevertheless the proved result is not that bad, too. First, it implies the following corollary.

Corollary 4.3 $(\forall b \in \mathbb{N})(\forall a, c \in \mathbb{N}\cup\{*\} \mid c \geq (b+4)\cdot a)[\mathrm{csdeg}(\mathbf{Ex}_b^a, \mathbf{Ex}_{2b-1}^c) \leq 2b+6]$.

We see that here estimation is linear in b, unlike results in Theorems 9, 12 and 19[1].

And, second, these upper bounds give exact csdeg values in some cases.

Theorem 17 $(\forall b,\ c \in \mathbb{N})$ $[\mathrm{csdeg}(\mathbf{Ex}_b,\ \mathbf{Ex}_{4b+1}^c) > 2]$. $(\forall b \in \mathbb{N})$ $[\mathrm{csdeg}(\mathbf{Ex}_b^*,\ \mathbf{Ex}_{4b+1}^*) > 2]$.

Proof. Let $U_1 = I_1^{\mathbf{Ex}_b^a}, U_2 = I_2^{\mathbf{Ex}_b^a}$, where $a=0$ or $a=*$. Then $U_1, U_2 \in \mathbf{Ex}_b^a$. We shall show that $U_1 \cup U_2 \notin \mathbf{Ex}_{4b+1}^c$, where $c \in \mathbb{N}$ if $a=0$, and $c=*$ if $a=*$, by using the multiple recursion theorem and constructing functions φ_{n_i} similarly as in previous such proofs.

Suppose $U_1 \cup U_2 \subseteq \mathbf{Ex}_{4b+1}^*(F)$ for some F. Procedure new(x) lets $x \leftarrow n_{\mathrm{count}}$, count $\leftarrow$ count $+\,1$. The algorithm for φ_{n_i} is as follows.

[1]Further in this section.

- *Stage 0.*
  Let count $\leftarrow 1$. Execute new(s_1). Output $\varphi_{s_1}(0) = \langle 1,1,1,s_1\rangle$. While in this stage, output $\varphi_{s_1}(x) = \langle\rangle$ for $x > 0$.
  Let the variable y throughout the algorithm indicate the maximal value of argument at which values have been output. We simulate F on φ_{s_1}. If a hypothesis is output on $\varphi_{s_1}^{[x]}$, we let $h \leftarrow F(\varphi_{s_1}^{[x]})$, $x_0 \leftarrow \max(x,y)+1$, $u \leftarrow 2$, $v \leftarrow 1$, we output $\langle\rangle$ up to $x_0 - 1$, let $f^{[x_0-1]} \leftarrow \varphi_{s_1}^{[x_0-1]}$, and go to stage 1.

- *Stage* $4m-3$ *(*$1 \le m \le b+1$*).*
  Execute new(t_1), new(t_2). Output $\varphi_{t_1}(x) = \varphi_{t_2}(x) = f(x)$ for $x < x_0$, and further as in the next table.

	x_0	$\dots$
φ_{s_v}	?	?
φ_{t_1}	$\langle 1,u,m,t_1\rangle$	$\langle\rangle$
φ_{t_2}	$\langle 1,u,m,t_2\rangle$	$\langle 0\rangle$

  Simulate F on φ_{t_1} and φ_{t_2}. If F outputs a new hypothesis on $\varphi_{t_1}^{[x]}$, we let $s_u \leftarrow t_1$. $h \leftarrow F(\varphi_{t_1}^{[x]})$, $\alpha \leftarrow \langle\rangle$ and output $\varphi_{s_v}(x_0) = \langle 1,u,m,t_1\rangle$.
  If F outputs a new hypothesis on $\varphi_{t_2}^{[x]}$, we let $s_u \leftarrow t_2$, $h \leftarrow F(\varphi_{t_2}^{[x]})$, $\alpha \leftarrow \langle 0\rangle$ and output $\varphi_{s_v}(x_0) = \langle 1,u,m,t_2\rangle$.
  In both cases we let $x_0 \leftarrow \max(x,y)+1$, output α up to x_0-1, let $f^{[x_0-1]} \leftarrow \varphi_{s_u}^{[x_0-1]}$, and go to stage $4m-2$.

- *Stage* $4m-2$ *(*$1 \le m \le b+1$*).*
  Output $\varphi_{s_u}(x) = \langle\rangle, \varphi_{s_v}(x) = \langle 0\rangle$ for $x \ge x_0$ while in this stage.
  If $m = b+1$, the algorithm remains in this stage. If $m < b+1$, simulate F on φ_{s_u} and φ_{s_v}. If F outputs a new hypothesis on $\varphi_{s_u}^{[x]}$, we let $h \leftarrow F(\varphi_{s_u}^{[x]})$, $\alpha \leftarrow \langle\rangle$.
  If F outputs a new hypothesis on $\varphi_{s_v}^{[x]}$, we let $h \leftarrow F(\varphi_{s_v}^{[x]})$, $\alpha \leftarrow \langle 0\rangle$, $(u,v) \leftarrow (v,u)$.
  In both cases we let $x_0 \leftarrow \max(x,y)+1$, output α up to x_0-1, extend f to x_0-1 by defining it equal to φ_{s_u} where f was undefined, and go to stage $4m-1$.

- *Stage* $4m-1$ *(*$1 \le m \le b$*).*
  Execute new(t_1), new(t_2). Output $\varphi_{t_1}(x) = \varphi_{t_2}(x) = f(x)$ for $x < x_0$ and further as in the next table.

	x_0	$\dots$
φ_{s_u}	?	?
φ_{t_1}	$\langle 1,v,m+1,t_1\rangle$	$\langle\rangle$
φ_{t_2}	$\langle 1,v,m+1,t_2\rangle$	$\langle 0\rangle$

  Simulate F on φ_{t_1} and φ_{t_2}. If F outputs a new hypothesis on $\varphi_{t_1}^{[x]}$, we let $s_v \leftarrow t_1$, $h \leftarrow F(\varphi_{t_1}^{[x]})$, $\alpha \leftarrow \langle\rangle$ and output $\varphi_{s_u}(x_0) = \langle 1,v,m+1,t_1\rangle$.

If F outputs a new hypothesis on $\varphi_{t_2}^{[x]}$, we let $s_v \leftarrow t_2$, $h \leftarrow F(\varphi_{t_2}^{[x]})$, $\alpha \leftarrow \langle 0 \rangle$ and output $\varphi_{s_u}(x_0) = \langle 1, v, m+1, t_2 \rangle$.
In both cases we let $x_0 \leftarrow \max(x,y)+1$, output α up to $x_0 - 1$, let $f^{[x_0-1]} \leftarrow \varphi_{s_v}^{[x_0-1]}$, and go to stage $4m$.

- *Stage* $4m$ $(1 \le m \le b)$.
The algorithm for this stage depends on a.
Suppose $a = 0$, then $c < *$. Output $\varphi_{s_u}(x) = \langle\rangle$ for $x \ge x_0$ while in this stage. Simulate $\varphi_h(x)$ for $x \ge x_0$ and F on φ_{s_u}. If $\varphi_h(x) = \langle\rangle$ for at least $c+1$ values of argument $x \ge x_0$, we output $\varphi_{s_v}(x) = \langle 0 \rangle$ for $x \ge x_0$ at the same time simulating F on φ_{s_v}. If F outputs a new hypothesis on $\varphi_{s_v}^{[x]}$, we let $h \leftarrow F(\varphi_{s_v}^{[x]})$, $x_0 \leftarrow \max(x,y)+1$, output $\langle 0 \rangle$ up to $x_0 - 1$, let $f^{[x_0-1]} \leftarrow \varphi_{s_v}^{[x_0-1]}$, and go to stage $4m+1$.
If F outputs a new hypothesis on $\varphi_{s_u}^{[x']}$, we output $\varphi_{s_v}(x) = \langle\rangle$ for $x_0 \le x \le x'$, let $h \leftarrow F(\varphi_{s_u}^{[x]})$, $x_0 \leftarrow \max(x,y)+1$, output $\langle\rangle$ up to $x_0 - 1$, let $f^{[x_0-1]} \leftarrow \varphi_{s_v}^{[x_0-1]}$, and go to stage $4m+1$.
Suppose $a = c = *$. Output one by one values $\varphi_{s_u}(x) = \langle\rangle$ and $\varphi_{s_v}(x) = \langle 0 \rangle$ for $x \ge x_0$. Simulate F on φ_{s_u} and φ_{s_v}. If F outputs a new hypothesis on $\varphi_{s_v}^{[x]}$, we let $h \leftarrow F(\varphi_{s_v}^{[x]})$, $x_0 \leftarrow \max(x,y)+1$, output $\langle 0 \rangle$ up to $x_0 - 1$, let $f^{[x_0-1]} \leftarrow \varphi_{s_v}^{[x_0-1]}$, and go to stage $4m+1$.
If F outputs a new hypothesis on $\varphi_{s_u}^{[x]}$, we let $h \leftarrow F(\varphi_{s_u}^{[x]})$, $x_0 \leftarrow \max(x,y)+1$, output $\langle\rangle$ up to $x_0 - 1$, let $f^{[x_0-1]} \leftarrow \varphi_{s_u}^{[x_0-1]}$, and go to stage $4m+1$.
(Note that in the latter case, though φ_{s_v}, the function that will be used in stage $4m+1$, differs from f, the function in which the segments feeded to F are recorded, φ_{s_v} still have only finite amount of anomalies with respect to f, and that is allowable in case $a = *$.)

Each stage is constructed so that the current hypothesis h is invalid for at least one of the considered functions that belong to $U_1 \cup U_2$. So F has to make a mindchange, and either the last hypothesis output by F is incorrect, or F makes at least $4b+2$ mindchanges. □

Corollary 4.4 $(\forall a, c \in \mathbb{N} \cup \{*\} \mid c \ge 2a \wedge (c = * \Rightarrow a = *))(\forall b, d \in \mathbb{N} \mid 3b+1 \le d \le 4b+1)[\mathrm{csdeg}(\mathbf{Ex}_b^a, \mathbf{Ex}_d^c) = 3]$.

Theorem 18 $(\forall b, c \in \mathbb{N})[\mathrm{csdeg}(\mathbf{Ex}_b, \mathbf{Ex}_{3b}^c) > 3]$. $(\forall b \in \mathbb{N})[\mathrm{csdeg}(\mathbf{Ex}_b^*, \mathbf{Ex}_{3b}^*) > 3]$.

Proof. Let $U_i = \bigcap_{j=1, j\ne i}^{3} I_j^{\mathbf{Ex}_b^a}$ for $1 \le i \le 3$ where $a = 0$ or $a = *$. Then union of any two of these classes is in $\mathbf{Ex}_b$. We shall show that $U_1 \cup U_2 \cup U_3 \notin \mathbf{Ex}_{3b}^c$ where $c \in \mathbb{N}$ if $a = 0$, and $c = *$ if $a = *$.

Suppose $U_1 \cup U_2 \cup U_3 \subseteq \mathbf{Ex}_{3b}^c(F)$ for some F. We again use the multiple recursion theorem to construct φ_{n_i}. Procedure new(x) lets $x \leftarrow n_{\mathrm{count}}$, count $\leftarrow$ count $+ 1$. The algorithm for φ_{n_i} is as follows.

- *Stage 0.*
  Let count $\leftarrow 1$. Execute $\mathrm{new}(s_1), \mathrm{new}(s_2)$. Output values as in the next table.

	0	1	...
$\varphi_{s_1}, \varphi_{s_2}$	$\langle 1,1,1,s_1\rangle$	$\langle 1,2,1,s_2\rangle$	$\langle\rangle$

  Let the variable y throughout the algorithm indicate the maximal value of argument at which values have been output. We simulate F on φ_{s_1}. If a hypothesis is output on $\varphi_{s_1}^{[x]}$, we let $h \leftarrow F(\varphi_{s_1}^{[x]})$, $x_0 \leftarrow \max(x,y)+1$, $u \leftarrow 1$, $v \leftarrow 2$, $w \leftarrow 3$, we output $\langle\rangle$ up to x_0-1, let $f^{[x_0-1]} \leftarrow \varphi_{s_1}^{[x_0-1]}$, and go to stage 1.

- *Stage $3m-2$ ($1 \le m \le b+1$).*
  Execute $\mathrm{new}(t_1), \mathrm{new}(t_2)$. Output $\varphi_{t_1}(x) = \varphi_{t_2}(x) = f(x)$ for $x < x_0$, and further as in the next table.

	x_0	...
$\varphi_{s_u}, \varphi_{t_1}$	$\langle 1,w,m,t_1\rangle$	$\langle\rangle$
$\varphi_{s_v}, \varphi_{t_2}$	$\langle 1,w,m,t_2\rangle$	$\langle 0\rangle$

  If $m = b+1$, the algorithm remains in this stage. If $m < b+1$, simulate F on φ_{t_1} and φ_{t_2}. If F outputs a new hypothesis on $\varphi_{t_1}^{[x]}$, we let $s_w \leftarrow t_1$, $h \leftarrow F(\varphi_{t_1}^{[x]})$, $\alpha \leftarrow \langle\rangle$, $(u,v) \leftarrow (v,u)$.
  If F outputs a new hypothesis on $\varphi_{t_2}^{[x]}$, we let $s_w \leftarrow t_2$, $h \leftarrow F(\varphi_{t_2}^{[x]})$, $\alpha \leftarrow \langle 0\rangle$.
  In both cases we let $x_0 \leftarrow \max(x,y)+1$, output α up to x_0-1, let $f^{[x_0-1]} \leftarrow \varphi_{s_w}^{[x_0-1]}$, and go to stage $3m-1$.

- *Stage $3m-1$ ($1 \le m \le b$).*
  Execute $\mathrm{new}(t_1), \mathrm{new}(t_2)$. Output $\varphi_{t_1}(x) = \varphi_{t_2}(x) = f(x)$ for $x < x_0$, and further as in the next table.

	x_0	...
$\varphi_{s_v}, \varphi_{t_1}$	$\langle 1,u,m+1,t_1\rangle$	$\langle\rangle$
$\varphi_{s_w}, \varphi_{t_2}$	$\langle 1,u,m+1,t_2\rangle$	$\langle 0\rangle$

  Simulate F on φ_{t_1} and φ_{t_2}. If F outputs a new hypothesis on $\varphi_{t_1}^{[x]}$, we let $s_u \leftarrow t_1$, $h \leftarrow F(\varphi_{t_1}^{[x]})$, $\alpha \leftarrow \langle\rangle$.
  If F outputs a new hypothesis on $\varphi_{t_2}^{[x]}$, we let $s_u \leftarrow t_2$, $h \leftarrow F(\varphi_{s_v}^{[x]})$, $\alpha \leftarrow \langle 0\rangle$, $(v,w) \leftarrow (w,v)$.
  In both cases we let $x_0 \leftarrow \max(x,y)+1$, output α up to x_0-1, $f^{[x_0-1]} \leftarrow \varphi_{s_u}^{[x_0-1]}$, and go to stage $3m$.

- *Stage $3m$ ($1 \le m \le b$).*
  Here we consider two cases.

1) $a = 0$, $c < *$. Execute new(t_1). Output $\varphi_{t_1}(x) = f(x)$ for $x < x_0$, and further as in the next table.

	x_0	$\dots$
φ_{s_u}	?	?
$\varphi_{s_v}, \varphi_{t_1}$	$\langle 1, w, m+1, t_1 \rangle$	$\langle\rangle$

Simulate $\varphi_h(x)$ for $x > x_0$ and F on φ_{t_1}. If $\varphi_h(x) = \langle\rangle$ for at least $c+1$ values of argument $x > x_0$, we execute new(t_2), output $\varphi_{t_2}(x) = f(x)$ for $x < x_0$, and further as in the next table.

	x_0	$\dots$
$\varphi_{s_u}, \varphi_{t_2}$	$\langle 1, v, m+1, t_2 \rangle$	$\langle 0 \rangle$

We simulate F on φ_{t_2}. If F outputs a new hypothesis on $\varphi_{t_2}^{[x]}$, we let $s_v \leftarrow t_2$, $h \leftarrow F(\varphi_{t_2}^{[x]})$, $x_0 \leftarrow \max(x, y) + 1$, output $\langle 0 \rangle$ up to $x_0 - 1$, let $f^{[x_0-1]} \leftarrow \varphi_{s_v}^{[x_0-1]}$, and go to stage $3m+1$.

If F outputs a new hypothesis on $\varphi_{t_1}^{[x']}$ (see the table before previous), we output $\varphi_{s_u}(x) = \varphi_{t_1}(x)$ for $x_0 \le x \le x'$, let $s_w \leftarrow t_1$, $(v, w) \leftarrow (w, v)$, $h \leftarrow F(\varphi_{t_1}^{[x']})$, $x_0 \leftarrow \max(x, y) + 1$, output $\langle\rangle$ up to $x_0 - 1$, let $f^{[x_0-1]} \leftarrow \varphi_{t_1}^{[x_0-1]}$, and go to stage $3m+1$.

2) $a = c = *$. Execute new(t_1), new(t_2). Output $\varphi_{t_1}(x) = \varphi_{t_2}(x) = f(x)$ for $x < x_0$, and further as in the next table.

	x_0	$\dots$
$\varphi_{s_u}, \varphi_{t_1}$	$\langle 1, v, m+1, t_1 \rangle$	$\langle\rangle$
$\varphi_{s_v}, \varphi_{t_2}$	$\langle 1, w, m+1, t_2 \rangle$	$\langle 0 \rangle$

Simulate F on φ_{t_1} and φ_{t_2}. If F outputs a new hypothesis on $\varphi_{t_1}^{[x]}$, we let $s_v \leftarrow t_1$, $h \leftarrow F(\varphi_{t_1}^{[x]})$, $x_0 \leftarrow \max(x, y) + 1$, output $\langle\rangle$ up to $x_0 - 1$, let $f^{[x_0-1]} \leftarrow \varphi_{t_1}^{[x_0-1]}$, and go to stage $3m+1$.

If F outputs a new hypothesis on $\varphi_{t_2}^{[x]}$, we let $s_w \leftarrow t_2$, $(v, w) \leftarrow (w, v)$, $h \leftarrow F(\varphi_{t_2}^{[x]})$, $x_0 \leftarrow \max(x, y) + 1$, output $\langle 0 \rangle$ up to $x_0 - 1$, let $f^{[x_0-1]} \leftarrow \varphi_{t_2}^{[x_0-1]}$, and go to stage $3m+1$. (Though in the latter case $\varphi_{s_u} \ne f$, still $\varphi_{s_u} =^* f$.)

Each stage is constructed so that the current hypothesis h is invalid for at least one of the considered functions that belong to $U_1 \cup U_2 \cup U_3$. So F has to make a mindchange, and either the last hypothesis output by F is incorrect, or F makes at least $3b+1$ mindchanges. □

Corollary 4.5 $(\forall a, c \in \mathbb{N} \cup \{*\} \mid c \ge 2a \wedge (c = * \Rightarrow a = *))(\forall b, d \in \mathbb{N} \mid \lfloor (8b+1)/3 \rfloor \le d \le 3b)[\text{csdeg}(\mathbf{Ex}_b^a, \mathbf{Ex}_d^c) = 4]$.

A question arises: what flaws has the algorithm described in Theorem 16? The proof of the next theorem shows the most significant one: this algorithm has equal confidence in the m-th hypothesis output by a strategy it simulates and in the $(m+1)$-th hypothesis. A later hypothesis is "nearer" to the $(b+1)$-th hypothesis that is always correct for a correct $\mathbf{Ex}_b^a$-strategy, so greater weight should be attached to later hypotheses than to earlier. On the other hand, as we shall see in the next theorem, the optimal weight function is rather complex, not depending only on which hyppothesis it is for the strategy.

Theorem 19 *Let k_n, $n \in \mathbb{N}$, be the smallest natural number for which $k_n \cdot 2^{k_n-3} \geq (2^{n+2}+1)/3$ holds. Then for all $b \in \mathbb{N}$:*

$$\mathrm{csdeg}(\mathbf{Ex}_b, \mathbf{Ex}_{b+1}) \leq \begin{cases} 2\left\lceil \frac{2^{b+2}+3\cdot 2^{k_b-3}}{k_b+1} \right\rceil - 1, & \textit{if } 2^{b+2} + 3 \cdot 2^{k_b-3} \equiv -1 \textit{ or } 0 \\ & \quad (\mathrm{mod}\ k_b+1); \\ 2\left\lceil \frac{2^{b+2}+3\cdot 2^{k_b-3}}{k_b+1} \right\rceil - 2, & \textit{otherwise.} \end{cases}$$

Proof. In proofs of the previous theorems proving upper bounds we saw that, if all the unions of $n-1$ classes out of $U_1, \ldots, U_n$ are $\mathbf{Ex}_b^a$-identifiable, then there are n strategies $F_1, \ldots, F_n$ such that every function $f \in \bigcup_{i=1}^n U_i$ is $\mathbf{Ex}_b^a$-identified by at least $n-1$ of them. Thus some strategy F identifying f can simulate all the strategies F_j obtaining n sequences of hypotheses at most one of which (the sequences) is incorrect in the $\mathbf{Ex}_b^a$-identification sense. In the process of identification different configurations arise in these sequences such as: F has output 5 different hypotheses, 7 sequences have produced 6 different hypotheses each, $n-9$ sequences have produced 5 different hypotheses each, and 2 sequences have produced 4 different hypotheses each. For a possibly better estimation of csdeg we are going to find possibly optimal *"winning"* configurations, that is configurations at which F can identify the input functions. For instance, a configuration in which 3 sequences have output $b+1$ different hypotheses each is a winning configuration, since at least two of the last hypotheses in these sequences are correct, so F can output a correct hypothesis based on the 3 last hypotheses. Similarly, substituting $b=1$ in the proof of Theorem 9 we get that a configuration in which 7 sequences have output their b-th hypotheses is winning if F can output two hypotheses. Since we are concerned with upper bounds, we shall not prove that some configuration is *not* a winning configuration (that would be more difficult).

At first some heuristics for eliminating the set of configurations to consider. First, if F can output no more than i hypotheses, and in some sequence i more hypotheses can be output not exceeding the bound on mindchanges, then this sequence shouldn't be considered at all by F. Indeed, each time F bases its new hypothesis on the current hypothesis in this sequence, a mindchange can occur in it making thus the previous hypothesis unreliable. Second, some hypotheses can prove to be incorrect by outputting values different from the values of f.

We should not consider configurations in which the current hypotheses of two or more sequences have proved to be incorrect. Indeed, since at most one sequence is incorrect, new hypotheses are guaranteed to appear, and the strategy can wait for them until it "sees" no more than one incorrect current hypothesis. Third, though a situation when between two hypotheses by F two or more mindchanges are made in some sequences would be "good" for F, our strategy should work also in the worst case when such events do not ocuur. So, if such event ocuurs: mindchanges from h_i to h_{i+1} and from h_{i+1} to h_{i+2} are made, we shall consider it equivalent to a situation in which only one mindchange: from h_i to h_{i+2} has been made. Since initially all the sequences are empty, we shall suppose that at the moment F outputs its i-th hypothesis no more than i hypotheses have been output in each of the sequences. Summarizing, since we are interested in csdeg$(\mathbf{Ex}_b, \mathbf{Ex}_b + 1)$ — F can make one mindchange more compared to F_j —, so we should consider configurations in which for each j: in the j-th sequence either the same amount of hypotheses has been output as by F, or one hypothesis more has been output than F has output, and there is no more than one sequence whose current hypothesis is known (to F) to be incorrect.

Now we proceed formally.

Definition 4.2 *We say that a quadruplet (s, t, α, i), where $i, s, t \in \mathbb{N}$, $i \geq 1$, $\alpha \in \{0, 1\}$, is a* winning configuration *iff there is an algorithm F such that*

1. *it receives as input the initial segments of $f \in \mathcal{R}$ and $s + t + \alpha$ sequences of hypotheses;*

2. *initially it receives one hypothesis in each sequence;*

3. *if $i = 1$, $s = 0$, otherwise s sequences explicitly marked can have no more than $i - 2$ mindchanges;*

4. *t other sequences explicitly marked can have no more than $i - 1$ mindchanges;*

5. *α other sequences explicitly marked have an incorrect first hypothesis and can have no more than $i - 1$ mindchanges;*

6. *at least $s + t + \alpha - 1$ sequences' last hypothesis is correct;*

7. *F identifies f making no more than $i - 1$ mindchanges (that is by producing i hypotheses).*

Since additional information cannot be harmful, the following is true.

Lemma 4.2 *If $s' \geq s$, $s' + \alpha' \geq s + \alpha$, $s' + t' + \alpha' \geq s + t + \alpha$, $i' \geq i$ and (s, t, α, i) is a winning configuration, then (s', t', α', i') is a winning configuration, too.*

Also, since the last allowed hypothesis must be correct for all sequences but one, we get the next lemma.

Lemma 4.3 $(0,3,0,1)$ *and* $(0,1,1,1)$ *are winning configurations.*

Lemma 4.4 *If* $(s,0,1,i)$, $i \geq 2$, *is a winning configuration, then* $\mathrm{csdeg}(\mathbf{Ex}_{i-2}, \mathbf{Ex}_{i-1}) \leq s+1$.

Proof. Let $U_1, \ldots, U_{s+1}$ be such classes that the unions of s classes out of them are $\mathbf{Ex}_{i-2}$-identified by strategies $F_1, \ldots, F_{s+1}$. Let F' be the algorithm from Definition 4.2 corresponding to the configuration $(s,0,1,i)$. Strategy F waits until s of the strategies F_j output their first hypotheses on the input function f. The previous hypothesis of the remaining one strategy cannot be considered correct, since it has not output any hypotheses, but it can yet output $i-1$ hypotheses, while the other s strategies can output only $i-2$ more hypotheses each. So F simulates F' marking the remaining strategy as a strategy whose last hypothesis is known to be incorrect (and substituting some function undefined at all points for its hypothesis), and proceeds further feeding the outputs of F_j to F' and producing hypotheses output by F' as its own. According to Definition 4.2, F will $\mathbf{Ex}_{i-1}$-identify f, if $f \in \bigcup_{j=1}^{s+1} U_j$. □

The proof of the theorem is based on the next lemma.

Lemma 4.5 *Let the sequence* $\{x_n\}$ *be defined by* $x_n = \lfloor \log_2((n+1)/3) \rfloor + 4$ *for* $n \geq 1$. *Let* $S_{m,0} = 2^{m+1} - 1$, $S_{m,n} = 2^{m+1} - 1 - \sum_{i=1}^{n} x_n$ *for* $m \geq 2$, $n \geq 1$. *Let* $A_m = (2^m + 3 \cdot 2^{k_{m-2}-3} + 1)/(k_{m-2}+1)$, $B_m = A_m - 1/(k_{m-2}+1)$ *for* $m \geq 2$. *Let*

$$C_m = \begin{cases} 2 \cdot \lceil A_m \rceil - 2, & \text{if } A_m \text{ or } A_m + 1/(k_{m-2}+1) \in \mathbb{Z}, \\ 2 \cdot \lceil A_m \rceil - 3, & \text{otherwise,} \end{cases}$$

$$D_m = \begin{cases} 2 \cdot \lceil B_m \rceil - 2, & \text{if } B_m \text{ or } B_m + 1/(k_{m-2}+1) \in \mathbb{Z}, \\ 2 \cdot \lceil B_m \rceil - 3, & \text{otherwise} \end{cases}$$

for $m \geq 2$.

Then $(n, S_{m,n}, 0, m)$ *for* $0 \leq n < C_m$, $(C_m, 0, 0, m)$, $(n, S_{m,n} - 2, 1, m)$ *for* $0 \leq n < D_m$, *and* $(D_m, 0, 1, m)$, *where* $m \geq 2$, *are winning configurations.*

Proof is by induction (applied to m).

Basis. We have $S_{2,0} = 7$, $S_{2,1} = 4$, $k_0 = 3$, $A_2 = 2$, $B_2 = 7/4$, $C_2 = 2$, $D_2 = 2$. So we have to prove that $(0,7,0,2)$, $(1,4,0,2)$, $(2,0,0,2)$, $(0,5,1,2)$, $(1,2,1,2)$ and $(2,0,1,2)$ are winning configurations.

Algorithm of Lemma 4.1 proves that $(0,7,0,2)$ is winning.

Let us consider the case $(1,4,0,2)$. Then one sequence already has output its last hypothesis. The identifying strategy F outputs its first hypothesis h_0 based on the 5 hypotheses similarly as in Theorem 16 (φ_{h_0} outputs a value if at least half of the hypotheses output this value). For h_0 to be incorrect, 2 more sequences must output last hypotheses. So we have configuration $(0,3,0,1)$ or better (in terms of Lemma 4.2). According to Lemma 4.3 this is a winning configuration.

Now, the case $(2,0,0,2)$. F outputs h_0 based on the hypotheses (last allowed) of both sequences. For h_0 to be incorrect, at least one of them must output an incorrect value at some point. F simulates both hypotheses on all inputs, so it sees after some time which hypothesis is the wrong one. At that moment we have configuration $(0,1,1,1)$ which is winning according to Lemma 4.3.

The case $(0,5,1,2)$. F outputs h_0 based on the 5 hypotheses that have not proved to be wrong as yet. For h_0 to be incorrect, new hypotheses must be output either in three of the corresponding sequences, or in two of them and in the sequence whose hypothesis was marked as wrong. In both cases three sequences output their last allowed hypothesis, and we have a winning configuration $(0,3,0,1)$.

The case $(1,2,1,2)$. F outputs h_0 based on the 3 hypotheses that have not proved to be wrong as yet. For h_0 to be incorrect, new hypotheses must be output at least in two of the three sequences that still can make one mindchange, and we have a winning configuration $(0,3,0,1)$.

The case $(2,0,1,2)$ follows from the case $(2,0,0,2)$ and Lemma 4.2.

Inductive step. Suppose that we have proved the statement for the case $m-1$, $m \geq 3$. Let us prove it for the case m. For the sake of correctness we should also prove that $S_{m,n} \geq 0$ for $0 \leq n \leq C_m$, and $S_{m,n} \geq 2$ for $0 \leq n \leq D_m$. In the course of the proof we shall see that it is so (it was true in the inductive basis).

It follows from Lemma 4.1 that $(0, 2^{m+1}-1, 0, m)$ is a winning configuration. It is also easy to see that $(0, 2^{m+1}-3, 1, m)$ is a winning configuration. Indeed, F outputs its first hypothesis h_0 based on the $2^{m+1}-3$ hypotheses that have not proved to be incorrect. Then at least 2^m-1 sequences must output new hypotheses, and we get a winning configuration $(0, 2^m-1, 0, m-1)$.

Now, let us consider a configuration $(n,t,0,m)$. We are going to estimate a sufficient value $t = t_n$ such that this configuration is winning. We shall consider all the values $n \geq 1$ until we get that $t_n = 0$. So our strategy F has access to n hypotheses in sequences that can make $m-2$ mindchanges, we shall call them β-hypotheses, and to t_n hypotheses in sequences that can make $m-1$ mindchanges, we shall call them γ-hypotheses. F outputs its first hypothesis h_0 based on these $n+t_n$ hypotheses. We are going to analyse in which cases $\varphi_{h_0}(x)$ outputs a value.

Suppose that i $(i \leq n)$ β-hypotheses output one and the same value at x. How large should be the amount of γ-hypotheses that output the same value (let us denote this amount by u_i), for $\varphi_{h_0}(x)$ to output it? Suppose this value is incorrect (so h_0 will be incorrect). That should imply that a winning $(m-1)$-configuration arises after some time. The fact that i β-hypotheses and u_i γ-hypotheses prove to be incorrect implies one of two alternatives. Either sequences of $i-1$ β-hypotheses and sequences of u_i γ-hypotheses make a mindchange (in case $i > 0$), then configuration $(i-1, u_i+n-i, 1, m-1)$ arises, or sequences of i

β-hypotheses and sequences of $\max(u_i - 1, 0)$ γ-hypotheses make a mindchange, then configuration $(i, \max(u_i - 1, 0) + n - i, 0, m - 1)$ arises. Let $s_k = S_{m-1,k}$ for $0 \le k < C_{m-1}$, and $s_k = 0$ for $k \ge C_{m-1}$. We ensure that both these configurations are winning (according to the inductive assumption) by letting $u_i = \max(s_{i-1} - 2 - n + i, s_i + 1 - n + i, 0)$ in case $s_i - n + i > 0$, $i > 0$, and $u_i = \max(s_{i-1} - 2 - n + i, 0)$ in case $s_i - n + i \le 0$, $i > 0$. But, if $s_i - n + i > 0$, then $0 < s_i \le s_{i-1} - 3$, so $u_i = \max(s_{i-1} - 2 - n + i, 0)$ in both cases. In case $i = 0$ we have only the second alternative, so we let $u_0 = \max(s_0 + 1 - n, 0)$.

Let us consider case $n < C_{m-1}$. Then $i < C_{m-1}$, too, and by using the inequality $S_{m-1,i-1} \ge S_{m-1,i} + 3$ we have: $u_i = s_{i-1} - 2 - n + i$ for $i > 0$, and $u_0 = s_0 + 1 - n$ (since $s_{i-1} - 2 - n + i > s_i - 2 - n + i + 1$ for $1 \le i < n$ and $s_{n-1} - 2 > 0$, the right sides in these equalities are positive). We have guaranteed that, if φ_{h_0} outputs an incorrect value, then a winning $(m-1)$-configuration arises after some time. We have to guarantee that also for the case when φ_{h_0} is undefined at some point. Let i be the amount of β-hypotheses that output the correct value at this point. Then no more than $u_i - 1$ γ-hypotheses output the correct value. So at least $n - i$ β-hypotheses and $t_n - u_i + 1$ γ-hypotheses are incorrect. At least $(n-i) + (t_n - u_i + 1) - 1$ of the corresponding sequences must make a mindchange. If $0 < i < n$, then, depending on which is the remaining incorrect hypothesis — is it β- or γ-hypothesis —, either the configuration $(n - i - 1, t_n - s_{i-1} + 2 + n - i + 1 + i + 1, 0, m - 1) = (n - i - 1, t_n - s_{i-1} + n + 4, 0, m - 1)$, or the configuration $(n - i, t_n - s_{i-1} + 2 + n - i + i, 0, m - 1) = (n - i, t_n - s_{i-1} + n + 2, 0, m - 1)$ is reached. By imposing on t_n inequalities $t_n \ge s_{i-1} + s_{n-i-1} - n - 4$ and $t_n \ge s_{i-1} + s_{n-i} - n - 2$ we achieve that these configurations are winning. Since $s_{n-i-1} \ge s_{n-i} + 3$, the second inequality follows from the first, and we can consider only the first.

If $i = 0$, we get either the configuration $(n - 1, t_n - s_0 + n + 1, 0, m - 1)$, or the configuration $(n, t_n - s_0 + n - 1, 0, m - 1)$, and impose inequalities $t_n \ge s_0 + s_{n-1} - n - 1$ and $t_n \ge s_0 + s_n - n + 1$. Since the first inequality implies the second, we shall consider only the first inequality.

If $i = n$, we get the configuration $(0, t_n - s_{n-1} + n + 2, 0, m - 1)$, and impose the inequality $t_n \ge s_{n-1} + s_0 - n - 2$, but it follows from the inequality of the case $i = 0$, so we shall not consider it.

So, if we find the minimal natural number t_n that satisfies the system of inequalities

$$t_n \ge s_{i-1} + s_{n-i-1} - n - 4 \text{ for } 0 < i < n, \tag{4.8}$$

$$t_n \ge s_0 + s_{n-1} - n - 1, \tag{4.9}$$

then the configuration $(n, t_n, 0, m)$ will be winning. In case $n = 1$ we can choose $t_1 = 2s_0 - 2 = 2^{m+1} - 4 = S(m, 1)$ (according to (4.9)). If $n > 1$, then (4.9) follows from (4.8) by substitution $i = 1$. Since x_k is a non-decreasing sequence, according to the definition of $S_{k,l}$ we have $s_{i-1} + s_{n-i-1} \le s_i + s_{n-i-2}$ for $0 < i \le (n-2)/2$. Let $n = 2k$ for some $k \ge 1$. Then we can choose $t_n = 2s_{k-1} - n - 4$. If $n = 2k + 1$

for some $k \geq 1$, then we can choose $t_n = s_{k-1} + s_k - n - 4$. By using the next lemma we obtain that $t_n = S_{m,n}$.

Lemma 4.6 $(\forall m \geq 3)(\forall k \geq 1)[S_{m,2k} = 2S_{m-1,k-1} - 2k - 4 \wedge S_{m,2k+1} = S_{m-1,k-1} + S_{m-1,k} - (2k-1) - 5]$.

Proof. Let us consider the sequence x_k. We have $x_{2k+1} = \lfloor \log_2((2k+2)/3) \rfloor + 4 = \lfloor \log_2((k+1)/3) \rfloor + 5 = x_k + 1$ for $k \geq 1$. Also $x_{2k+2} = \lfloor \log_2((2k+3)/3) \rfloor + 4 = \lfloor \log_2((k+1)/3 + 1/6) \rfloor + 5$. Since k is integer, we have $\lfloor \log_2((k+1)/3 + 1/6) \rfloor = \lfloor \log_2((k+1)/3) \rfloor$ and $x_{2k+2} = x_k + 1$ for $k \geq 1$. Let $y_k = \sum_{i=1}^{k} x_i$ for $k \geq 0$ ($y_0 = 0$). It follows from the proved that

$$y_{2k+1} - y_{2k} = y_{2k+2} - y_{2k+1} = y_k - y_{k-1} + 1 \text{ for } k \geq 1. \tag{4.10}$$

We prove that

$$y_{2k} = 2y_{k-1} + 2k + 5 \text{ and } y_{2k+1} = y_{k-1} + y_k + 2k + 6 \text{ for } k \geq 1. \tag{4.11}$$

It is easy to check that $y_2 = 2y_0 + 2 + 5$ and $y_3 = y_0 + y_1 + 2 + 6$. Suppose that the equalities (4.11) are true for some k. Then by applying (4.10) we get the corresponding equalities for y_{2k+2} and y_{2k+3}.

Using the definition of $S_{m,n}$ we automatically obtain from (4.11) the needed equalities. □

Now we consider the case $n = C_{m-1}$. The only difference is that the inequality $s_{n-1} - 2 < 0$ is possible. At the same time $s_{n-1} > 0$ according to our assumption, so $s_{n-2} - 2 > 0$, thus for the indexes not exceeding $n-2$ we get the same inequalities for t_n. Suppose $u_n = 0$ and n β-hypotheses output the correct value of f. Then φ_{h_0} outputs it, too, irrespective of the number of γ-hypotheses that output this value. So nothing is to be imposed on t_n in this case. If $u_n > 0$, then, similarly as before, we impose $t_n \geq s_{n-1} + s_0 - n - 2$, but this inequality, as previously, follows from the inequality (4.13) below anyway, so we obtain the same system

$$\begin{aligned} t_n &\geq s_{i-1} + s_{n-i-1} - n - 4 \text{ for } 0 < i < n, && (4.12)\\ t_n &\geq s_0 + s_{n-1} - n - 1 && (4.13) \end{aligned}$$

with the same solution for the minimal t_n: $t_n = S(m,n)$.

Now, let us consider the case $n > C_{m-1}$. Since $s_{C_{m-1}} = 0$, we have $u_{C_{m-1}+1} = 0$. Let j be the maximal number not exceeding C_{m-1} such that $u_j = s_{j-1} - 2 - n + j > 0$. Then $u_{j+1} = \ldots = u_{C_{m-1}+1} = 0$.

Suppose $n \geq j+2$ and $n-j-2$ or less β-hypotheses output the correct value. Then at least $j+2$ β-hypotheses are incorrect, and one of the configurations $(j+1, n-j-1, 0, m-1)$ or $(j+2, n-j-2, 0, m-1)$ (or better in the sense of Lemma

4.2) is reached. Either $s_{j+1} = 0$, or $s_{j+1} \leq s_j - 3 \leq 2 + n - j - 1 - 3 < n - j - 1$, according to our choice of j. Similarly we get that $s_{j+2} \leq n - j - 2$. So both these configurations are winning without any restriction on t_n.

Suppose only $n - j - 1$ β-hypotheses output the correct value and $s_j \leq n - j$. Then at least $j + 1$ β-hypotheses are incorrect, and one of the configurations $(j, n - j, 0, m - 1)$ or $(j + 1, n - j - 1, 0, m - 1)$ is reached. We already dealt with the second of them in the previous case, and the first of them is winning according to our assumption $s_j \leq n - j$. Hence, if $s_j \leq n - j$, then we have the system

$$t_n \geq s_{i-1} + s_{n-i-1} - n - 4 \text{ for } n - j - 1 < i < j + 1 \tag{4.14}$$

for t_n.

If $s_j > n - j$, then, according to the choice of j, either $j = C_{m-1}$, or $s_j \leq 2 + n - j - 1 = n - j + 1 \Rightarrow s_j = n - j + 1$. In this case the system is

$$t_n \geq s_{i-1} + s_{n-i-1} - n - 4 \text{ for } n - j - 2 < i < j + 1. \tag{4.15}$$

In both cases we get the solution $t_n = S(m, n)$ for the minimal t_n similarly as before if the system contains at least one inequality (we shall see that in this case $S(m, n) > 0$). If the system contains no inequalities, the solution is $t_n = 0$. We are going to find the minimal n for which $t_n = 0$ and show that this n is equal to C_m.

If $s_j \leq n - j$ and the system (4.14) has no inequalities, then $n - j \geq j + 1$, so $n \geq 2j + 1$. Suppose $n \geq 2j + 3$. If $s_j \leq n - j - 1$, then $t_{n-1} = 0$, too, so n is not minimal. If $s_j = n - j = (n - 1) - j + 1$, then we must consider the system (4.15) for t_{n-1}. Since there is no such i that $2j + 2 - j - 2 = j < i < j + 1$, we have $t_{n-1} = 0$ again. So, if n is minimal, then $n = 2j + 1$ or $n = 2j + 2$, and correspondingly either $s_j \leq j + 1$, or $s_j \leq j + 2$ holds. On the other hand, if $s_j \leq j + 1$, then $t_{2j+1} = 0$ according to (4.14); similarly, if $s_j = j + 2$, then $t_{2j+2} = 0$.

Suppose $s_j > n - j$ and the system (4.15) is empty. Then $n - j - 1 \geq j + 1$, so $n \geq 2j + 2$. We discard the case $j = C_{m-1}$, because, as we shall see, $t_n = 0$ for $n = C_m < 2C_{m-1} + 2$. So $s_j = n - j + 1$. If $n = 2j + 3$, then $s_j = j + 4$, and $s_{j+1} \leq s_j - 3 < (j + 1) + 1$, thus we get that $t_{2j+3} = 0$ according to the previous case, so we shall not consider this case further. If $n \geq 2j + 4$, let us denote by j' the number for the case of computing t_{n-1} that corresponds to the number j. Since $s_j > n - j > (n - 1) - j$ and $s_j = n - j + 1 \neq (n - 1) - j + 1$, we have $j' > j$. Since $s_{j+1} \leq s_j - 3 = (n - 1) - (j + 1)$, we have $j' = j + 1$ and the system (4.14) which is empty, because $(n - 1) \geq 2 \cdot (j + 1) + 1$, thus n is not minimal. So we have to consider only the case $n = 2j + 2$, when $s_j = j + 3$. On the other hand, if $s_j = j + 3$, then $t_{2j+2} = 0$ according to (4.15).

Summarizing, if $j + 2 \leq s_j \leq j + 3$, then $t_{2j+2} = 0$, and, if $s_j \leq j + 1$, then $t_{2j+1} = 0$. Let j be the minimal natural number for which $s_j \leq j + 3$

holds. Suppose $j+2 \le s_j \le j+3$. Then $t_{2j+1} = s_{j-1} + s_j - (2j+1) - 4 \ge (j+3) + (j+3) - (2j+1) - 4 = 1 > 0$. Suppose $s_j \le j+1$. Then $t_{2j} = 2s_{j-1} - 2j - 4 \ge 2 \cdot (j+3) - 2j - 4 = 2 > 0$. So these indeed will be the minimal values of n for which $t_n = 0$.

Now, let us find the minimal j for which $s_j \le j+3$. It is easy to notice from the definition that x_i is a non-decreasing sequence 3, 4, 4, 4, 5, 5, 5, 5, 5, 5, 6, ... containing the number 4 three times, the number 5 six times, etc., the number k $3 \cdot 2^{k-4}$ times for $k \ge 4$. Thus

$$s_j = 2^m - 1 - (3 + \sum_{i=4}^{k-1}(i \cdot 3 \cdot 2^{i-4}) + k \cdot (l+1))$$

for $j = 1 + \sum_{i=4}^{k-1}(3 \cdot 2^{i-4}) + l + 1 = 3 \cdot 2^{k-4} + l - 1$, some $k \ge 4$ and some l, $0 \le l \le 3 \cdot 2^{k-4} - 1$. Let us consider the sequence $z_i = i \cdot 2^i$. It satisfies the equality $z_{i+2} = 4z_{i+1} - 4z_i$. By taking the sum of these equalities for $2 \le i \le p-2$, where $p \ge 4$, we obtain the formula $\sum_{i=4}^{p} z_i = 4z_{p-1} - 4z_2 = (p-1) \cdot 2^{p+1} - 32$. Therefore $s_j = 2^m - 3 \cdot (k-2) \cdot 2^{k-4} - k \cdot (l+1) + 2$. We must solve the system

$$\begin{cases} s_{j-1} > (j-1) + 3, \\ s_j \le j + 3, \end{cases}$$

or applying the obtained formulae:

$$\begin{cases} 2^m - 3 \cdot (k-2) \cdot 2^{k-4} - kl + 2 > j + 2, \\ 2^m - 3 \cdot (k-2) \cdot 2^{k-4} - kl - k + 2 \le j + 3. \end{cases}$$

Substituting $l = j - 3 \cdot 2^{k-4} + 1$ and simplifying, we obtain

$$\begin{cases} (k+1) \cdot j < 2^m + 3 \cdot 2^{k-3} + 1 - (k+1), \\ (k+1) \cdot j \ge 2^m + 3 \cdot 2^{k-3} + 1 - 2 \cdot (k+1). \end{cases} \tag{4.16}$$

k is determined by the system

$$\begin{cases} 2^m - 1 - (3 + \sum_{i=4}^{k-1}(i \cdot 3 \cdot 2^{i-4})) > (3 \cdot 2^{k-4} - 2) + 3, \\ 2^m - 1 - (3 + \sum_{i=4}^{k}(i \cdot 3 \cdot 2^{i-4})) \le (3 \cdot 2^{k-3} - 2) + 3. \end{cases}$$

After simplifying we get

$$\begin{cases} 3 \cdot (k-1) \cdot 2^{k-4} < 2^m + 1, \\ 3k \cdot 2^{k-3} \ge 2^m + 1. \end{cases}$$

So $k = k_{m-2}$. From (4.16) we get $j = \lceil A_m \rceil - 2$. By imposing additional equality $s_j = j+3$ or $s_j = j+2$ we get by the same transformations that, respectively, $A_m \in \mathbb{Z}$ or $A_m + 1/(k_{m-2}+1) \in \mathbb{Z}$. So the minimal n for which $t_n = 0$ is $n = C_m$.

The analysis of the configurations $(n, t'_n, 1, m)$ is quite similar to that of the case $(n, t_n, 0, m)$. □

The statement of the theorem follows from the fact that $(D_m, 0, 1, m)$ is a winning configuration. □

Since the algorithm of the proved theorem seems to be optimal, it is a plausible hypothesis that the proved upper bounds are simultaneously lower bounds, though proving that seems to be a difficult task.

4.5 Team Learning

In Section 3.2 we pointed to the connection of our research with team learning. In this section we are going to investigate n-closedness of the team learning identification types themselves. To simplify matters, we shall consider only identification without anomalies. Thus we are interested in $\mathrm{cdeg}([k,l]\mathbf{Ex}_b)$ for $b \in \mathbb{N}\{*\}$, $1 \leq k \leq l$.

4.5.1 Ex-Identification

We shall begin with **Ex**-identification. We need some analogue of Theorem 1 to determine which of the identification types are really different.

Theorem 20 [31] $(\forall l \geq 1)[[1,l]\mathbf{Ex} \subset [1,l+1]\mathbf{Ex}]$.

Theorem 21 [29] $(\forall k,l \mid 1 \leq k \leq l)[[k,l]\mathbf{Ex} = [1,\lfloor l/k \rfloor]\mathbf{Ex}]$.

We see that the learning power of the class $[k,l]\mathbf{Ex}$ is determined by the ratio k/l. Before going further we notice that teams of teams can be introduced by substituting a team identification type for $\mathcal{I}$ in Definition 2.6.

Theorem 22 [3] $(\forall n \geq 1)[\mathrm{cdeg}([1,n]\mathbf{Ex}) \leq n+2]$.

Proof. $[n+1,n+2][1,n]\mathbf{Ex} \subseteq [n+1,n\cdot(n+2)]\mathbf{Ex} = [1,n]\mathbf{Ex}$, so according to Proposition 3.3 $[1,n]\mathbf{Ex}$ is $(n+2)$-closed. □

The exact cdeg value follows from the next theorem.

Theorem 23 $(\forall k,l \mid 1 \leq k \leq l)(\forall n \geq 1)[[k,l][1,n]\mathbf{Ex} = [k,ln]\mathbf{Ex}]$.

Proof. We have l teams of n strategies each, and for at least k of these teams at least one of these strategies succeed. So, clearly, $[k,l][1,n]\mathbf{Ex} \subseteq [k,ln]\mathbf{Ex}$.

Let $m = \lfloor ln/k \rfloor$. Then $[k,ln]\mathbf{Ex} = [1,m]\mathbf{Ex}$, so for each $[k,ln]\mathbf{Ex}$-identifiable class U there are m strategies $F_1,\ldots,F_m$ such that each function from U is identified by at least one of them. We now compose l teams $T_0,\ldots,T_{l-1}$ with n strategies in each. We put F_i for each i, $1 \leq i \leq m$, in the teams $T_{ki \bmod l}$, $T_{ki+1 \bmod l}$, $\ldots$, $T_{ki+k-1 \bmod l}$, where $x \bmod y$ for $y > 0$ is the smallest non-negative residue of x modulo y. Since $mk \leq ln$, $ln - mk$ vacancies are left; we fill them with F_1. Suppose $f \in U$, then $f \in \mathbf{Ex}(F_j)$ for some j. Therefore at least k teams $T_{kj \bmod l}, \ldots, T_{kj+k-1 \bmod l}$ contain F_j, and so $[1,n]\mathbf{Ex}$-identify f. Hence $U \in [k,l][1,n]\mathbf{Ex}$. We have proved that $[k,ln]\mathbf{Ex} \subseteq [k,l][1,n]\mathbf{Ex}$. □

Note that a more general result $[k,l][m,n]\mathbf{Ex} = [km,ln]\mathbf{Ex}$ is not true. Indeed, $[1,2][2,3]\mathbf{Ex} = [1,2][1,1]\mathbf{Ex} = [1,2]\mathbf{Ex}$, while $[1\cdot 2, 2\cdot 3]\mathbf{Ex} = [1,3]\mathbf{Ex}$.

Corollary 4.6 $(\forall n \geq 1)[\mathrm{cdeg}([1,n]\mathbf{Ex}) = n+2]$.

Proof. According to Theorem 23 $[n,n+1][1,n]\mathbf{Ex} = [1,n+1]\mathbf{Ex} \supset [1,n]\mathbf{Ex}$, so $\mathrm{cdeg}([1,n]\mathbf{Ex}) > n+1$. □

4.5.2 Fin-Identification

Now we turn our attention to $[k,l]\mathbf{Ex}_b$-identification with $b \in \mathbb{N}$. Since in classification of different learning powers of the identification types $[k,l]\mathbf{Ex}_b$ more or less significant results have been achieved only for the case $b=0$ (and even here not complete), we also restrict our attention to this case. In literature $\mathbf{Ex}_0$ is often referred to as **Fin**-identification; we shall use this notation here.

As the next theorems show, in this case the hierarchy of different learning powers among $[k,l]\mathbf{Fin}$ is very complicated.

Theorem 24 [15, 13]

$$(\forall n \geq 1)\left(\forall k,l \;\middle|\; \frac{n+1}{2n+1} < \frac{k}{l} \leq \frac{n}{2n-1}\right)[[k,l]\mathbf{Fin} = [n,2n-1]\mathbf{Fin}].$$

Theorem 25 [36] $[1,2]\mathbf{Fin} \subset [2,4]\mathbf{Fin}$.

Theorem 26 [23] $(\forall k \geq 1)[[2k-1,4k-2]\mathbf{Fin} = [1,2]\mathbf{Fin} \wedge [2k,4k]\mathbf{Fin} = [2,4]\mathbf{Fin}]$.

Definition 4.3 *The set $A \subseteq \mathbb{R} \cap [0,1]$ is* the hierarchy of success ratios *for* **Fin** *iff*

1. $(\forall p \in A)(\forall k,l,m,n \geq 1)[k/l < p < m/n \leq 1 \Rightarrow [m,n]\mathbf{Fin} \subset [k,l]\mathbf{Fin}]$, *and*

2. $(\forall p,q \in A \mid p < q \wedge (p,q) \cap A = \emptyset)(\forall k,l,m,n \geq 1)(\exists u,v \geq 1)[p < k/l \leq m/n < q \Rightarrow [ku,lu]\mathbf{Fin} = [mv,nv]\mathbf{Fin}]$.

Theorem 27 [1] *A, the hierarchy of success ratios for* **Fin**, *in decreasing ordering is order-isomorphic to an ordinal no less than* $\varepsilon_0 = \lim(\omega, \omega^\omega, \omega^{\omega^\omega}, \ldots)$.

We shall not go into details on ordinals here, we only draw a conclusion that the set A is of a very complex structure. The computed values $p \in A$ do not reach far under $1/2$, so we shall consider only the cases $[1,2]\mathbf{Fin}$ and $[n, 2n-1]\mathbf{Fin}$.

The next result was a surprise (as many things in team **Fin**-identification). For any $n \geq 1$ the automatic inclusion of the class $[n, n+1][1,2]\mathbf{Fin}$ that first comes into mind is $[n, n+1][1,2]\mathbf{Fin} \subseteq [n, 2n+2]\mathbf{Fin}$. Even if in fact $[n, n+1][1,2]\mathbf{Fin}$ due to its additional structuring would be equal to some class $[k,l]\mathbf{Fin}$ with higher success ratio k/l (it could also form some new class), it would seem improbable that it would be less powerful than $[2,4]\mathbf{Fin}$, a class with the success ratio $1/2$. So one could guess that $\mathrm{cdeg}([1,2]\mathbf{Fin}) = \infty$. Nevertheless it turns out not to be true.

Theorem 28 $\mathrm{cdeg}([1,2]\mathbf{Fin}) \leq 9$.

Proof. Suppose all the unions of 8 out of classes $U_1, \ldots, U_9$ are in $[1,2]\mathbf{Fin}$. Let $T_1, \ldots, T_9$ be the teams that identify these unions. Each of these teams consists of two strategies. We are going to construct an algorithm F that models strategies F_1 and F_2 $[1,2]\mathbf{Fin}$-identifying $\bigcup_{i=1}^{9} U_i$ using these 18 strategies as subroutines.

We shall denote by $h_{j,1}$ the first hypothesis output by any strategy of T_j, and by $h_{j,2}$ the hypothesis output by the other strategy from T_j, if any. The algorithm for F is as follows.

- *Stage 1.*
  Receiving $f^{[x]}$ in input perform x steps in computing the outputs of the strategies on $f^{[0]}, \ldots, f^{[x]}$ and for any hypothesis $h_{j,\alpha}$ computed perform x steps in computing $\varphi_{h_{j,\alpha}}(0), \ldots, \varphi_{h_{j,\alpha}}(x)$. This is done throughout all stages. Wait until in eight teams some hypothesis is produced by one of the strategies. Output h_1 based on these eight hypotheses and the 16 strategies of the corresponding teams as the hypothesis by F_1, discard the ninth team (do not consider it anymore) and go to stage 2.

- *Stage 2.*
  Wait until one of the two events happens.

  1. We see that φ_{h_1} outputs an incorrect value (algorithm for φ_{h_1} is described below) based on incorrect values output by four of the hypotheses h_1 was based on. Then go to stage 4.
  2. In four teams the second strategy outputs a hypothesis. Let $f^{[x_0]}$ be the input segment at which the last of them was discovered. Go to stage 3.

- *Stage 3.*
  Let $k(x)$ be the amount of the teams in which both strategies have produced their hypotheses when F has performed all the computations corresponding to the input $f^{[x]}$. Wait until one of the two events happens.

1. We see that φ_{h_1} outputs an incorrect value at $x \leq x_0$ based on four of the eight hypotheses h_1 was based on. Then go to stage 4.
2. We see for some $x_1 > x_0$ that the hypotheses of $8 - k(x_1) - 1$ teams among those in which only one hypothesis was produced output correct values at all points in the interval $[0, x_0]$. Then F_2 outputs hypothesis h_2 based on the $2k(x_1)$ hypotheses of the teams that produced two hypotheses.

- *Stage 4.*
  Wait until in three of the four teams whose hypotheses turned out to be incorrect the other strategy also produces its hypothesis. Output h_2' based on the three new hypotheses as the hypothesis by F_2.

The algorithm for φ_{h_1} follows.

- *Stage 1.*
  Let $h_{1,1}, \ldots, h_{8,1}$ be the hypotheses on which h_1 was based. φ_{h_1} outputs a value at point x only if it has already output values at points $0, \ldots, x-1$. It outputs all the values of f that were known at the moment h_1 was produced. After that it computes $\varphi_{h_{1,1}}(x), \ldots, \varphi_{h_{8,1}}(x)$. If four of them output some value y, φ_{h_1} outputs this value, too.

  Then φ_{h_1} performs x steps in computing the outputs of the 16 strategies on $\varphi_{h_1}^{[0]}, \ldots, \varphi_{h_1}^{[x]}$ and for any hypothesis $h_{j,\alpha}$ computed perform x steps in computing $\varphi_{h_{j,\alpha}}(0), \ldots, \varphi_{h_{j,\alpha}}(x)$. In this way φ_{h_1} learns about new hypotheses output by strategies and can simulate the strategy F (all this in case it has correctly guessed the values up to x).

  If the second hypotheses have been output in four teams, let us assume they are $h_{1,2}, h_{2,2}, h_{3,2}, h_{4,2}$, then go to stage 2.

- *Stage 2.*
  Output value at x if three hypotheses among $h_{j,\alpha}$, $1 \leq j \leq 4$, $\alpha \in \{1, 2\}$, belonging to different teams produce one and the same value at x.

  If the strategy F outputs the second hypothesis h_2, go to stage 3. Otherwise go on computing outputs at $x + 1$.

- *Stage 3.*
  Let us assume that the hypotheses on which h_2 was based are $h_{1,1}$, …, $h_{k,1}$, $h_{1,2}$, …, $h_{k,2}$, $k \geq 4$. At each input x perform computation of these hypothesis functions until it is clear that φ_{h_2} (see the algorithm below) outputs a value based on $k - 1$ hypotheses from different teams. Wait until one of two events happens.

1. $k-1$ other hypotheses from different teams output value y at x different from the value output by φ_{h_2} and the correct values (from φ_{h_1}'s standpoint) at all the previous points. Then output y and go to stage 4.
2. In total $2 \cdot (k-1)$ hypotheses produce the value output by φ_{h_2} at x and correct values at all the previous points. Then output the same value and continue by performing the computations for $x+1$.

- *Stage 4.*
  Output value at x if $k-1$ hypotheses among $h_{j,\alpha}$, $1 \le j \le k$, $\alpha \in \{1,2\}$, belonging to different teams produce one and the same value at x and correct values at all the previous points.

Now, the algorithm for φ_{h_2}.

- *Stage 1.*
  Assume that φ_{h_2} was based on $h_{j,\alpha}$, $1 \le j \le k$, $\alpha \in \{1,2\}$, $k \ge 4$, and that $h_{1,2}, \ldots, h_{4,2}$ were the first hypotheses output among $h_{j,2}$. Let x_0 and x_1 be as defined in the algorithm for F.

  Output the known values of f for $0 \le x \le x_1$. Simulate $\varphi_{h_{j,\alpha}}(x)$ for $1 \le j \le 4$, $\alpha \in \{1,2\}$, $x_0 < x \le x_1$ until it is clear if the first value output by three hypothesis functions from different teams is the correct value at each of these points. In other words, we check if φ_{h_1} outputs correct values at these points in case it outputs correct values for $x \le x_0$.

  If some incorrect value appears first at some point, go to stage 3. Otherwise go to stage 2.

- *Stage 2.*
  For each $x > x_1$, wait until at least $k-1$ hypotheses among $h_{1,1}, \ldots, h_{k,1}$, $h_{1,2}, \ldots, h_{k,2}$, $k \ge 4$, coming from different teams produce one and the same value y at x and the correct values (from φ_{h_2}'s standpoint) at all the previous points. Then output y.

- *Stage 3.*
  Three hypotheses among $\varphi_{h_{j,\alpha}}$, $1 \le j \le 4$, $\alpha \in \{1,2\}$, coming from different teams have proved to be incorrect. So by taking the other hypotheses from these teams, we obtain three hypotheses, at most one of which is incorrect. Thus by outputting the value produced by at least two of these hypotheses we always output the correct value.

At last, the algorithm for $\varphi_{h'_2}$ is the same as in stage 3 of the algorithm for φ_{h_2}: output the value produced by at least two of the three hypotheses h'_2 was based on.

We see that, unlike the identification algorithms in the previous sections, in team learning the hypotheses cooperate between themselves and also, in a sense, with the strategy that outputs them.

Let us analyse some cases to prove that the team composed by F_1 and F_2 identify any $f \in \bigcup_{i=1}^{9} U_i$.

1. *Hypothesis h_2 is never output.* There are two alternatives. First, no more than 3 of the 8 teams on whose hypotheses h_1 was based output another hypothesis. Hence at least $8-3-1=4$ of these first hypotheses are correct, and φ_{h_1} either is defined at all points and equal to f according to stage 1 of its algorithm, or outputs some incorrect value. In the latter case a correct hypothesis h_2' is output.

   The second alternative is that no more than $8-k-2$ of the $8-k$ teams that produce only one hypothesis output correct values at points between 0 and x_0. Then at least two teams do not identify f, so we have a contradiction.

2. *φ_{h_2}'s first anomaly is an undefined value.* Suppose we have both hypotheses of k teams. Since at least $k-1$ teams correctly identify f, at least $k-1$ hypotheses belonging to different teams are total recursive functions equal to f. Thus φ_{h_2} cannot become stuck in stage 1 or stage 2 of its algorithm. Clearly, it cannot become stuck also in stage 3. So this case is impossible.

3. *φ_{h_2}'s first anomaly is an incorrect output value at some point x.* Clearly, φ_{h_2} cannot output the incorrect value at stage 1 or stage 3. Thus it is output at stage 2. This value was produced by $k-1$ hypotheses that output correct values at all the previous points. There are other $k-1$ hypotheses among the $2k$ considered that output correct values at all the previous points *and* at x. So there are $k-2$ teams in which both hypotheses output are correct at the previous points. Hence among the 8 hypotheses on which h_1 was based there are at least $(k-2)+(8-k-1)=5$ hypotheses that produce correct outputs in interval $[0, x_0]$ ($8-k-1$ hypotheses being checked by F before producing h_2). So φ_{h_1} reaches stage 2 of its algorithm. Also, since φ_{h_2} reached stage 2, φ_{h_1} outputs correct values in interval $(x_0, x_1]$ and reaches stage 3. Since $k \geq 4$, we have $3\cdot(k-1) > 2k$, so φ_{h_1} has at most one alternative to choose from in stages 3 and 4. We have $2k-2$ hypotheses that output correct values in the interval (x_1, x), so φ_{h_1} output correct values in this interval, then produces the correct value at x, switches to stage 4, and produces correct values thereafter.

We see that at least one of the hypotheses is correct for f. □

It is interesting that every of the identification types not involving anomalies that were considered in this work has a finite closedness degree.

Theorem 29 cdeg([1, 2]**Fin**) > 8.

Proof. We shall modify the instructions a bit to make them useful in team learning. We shall also reduce the number of components, since some of them will not be necessary. The instructions will be of kind $\langle i, j, h\rangle$, where the first two components indicate that it is an instruction for the j-th strategy of the i-th team, and h is the proposed hypothesis. We define $I_i^{[k,l]\mathbf{Fin}}$ correspondingly (i is the number of team this time).

Let $U_i = \bigcap_{j=1,j\neq i}^{8} I_j^{[1,2]\mathbf{Fin}}$, $1 \le i \le 8$. Then $\bigcup_{i=1,i\neq j}^{8} U_i \in [1,2]\mathbf{Fin}$ for $1 \le j \le 8$.

The following algorithm constructs a function from $\bigcup_{i=1}^{8} U_i$ not identified by the given team T consisting of two strategies, F_1 and F_2.

- *Stage 1.*
  Output values as in the next table.

	0	...	6	...
$\varphi_{n_1}, \ldots, \varphi_{n_7}$	$\langle 1,1,n_1\rangle$	...	$\langle 7,1,n_7\rangle$	$\langle\rangle$

  Let y throughout the algorithm denote the maximal point at which values have been output. Simulate both strategies on φ_{n_1}. If a hypothesis h_1 is produced by one of them, let it be F_1, on $\varphi_{n_1}^{[x]}$, then let $x_0 \leftarrow \max(x, y) + 1$, output $\langle\rangle$ up to $x_0 - 1$, and go to stage 2.

- *Stage 2.*
  Output values as in the next table.

	x_0	x_0+1	x_0+2	x_0+3	...
$\varphi_{n_1}, \varphi_{n_2}, \varphi_{n_3}, \varphi_{n_8}, \ldots, \varphi_{n_{11}}$	$\langle 8,1,n_8\rangle$	$\langle 4,2,n_9\rangle$	$\langle 5,2,n_{10}\rangle$	$\langle 6,2,n_{11}\rangle$	$\langle\rangle$
$\varphi_{n_4}, \ldots, \varphi_{n_7}$	?	?	?	?	?

  Simulate $\varphi_{h_1}(x_0 + 4)$ and F_2 on φ_{n_1}.
  If $\varphi_{h_1}(x_0 + 4) = \langle\rangle$, go to stage 3.
  If a hypothesis h_2 is produced by F_2 on $\varphi_{n_1}^{[x']}$, output the values $\varphi_{n_4}(x), \ldots, \varphi_{n_7}(x)$ so that they are equal to the values $\varphi_{n_1}(x)$ as far as φ_{n_1} is defined at the moment, let $x_0 \leftarrow \max(x', y) + 1$, output $\langle\rangle$ up to $x_0 - 1$, and go to stage 5.

- *Stage 3.*
  Output values as in the next table.

	x_0	x_0+1	x_0+2	...
$\varphi_{n_4}, \ldots, \varphi_{n_7}, \varphi_{n_{12}}, \varphi_{n_{13}}, \varphi_{n_{14}}$	$\langle 8,1,n_{12}\rangle$	$\langle 1,2,n_{13}\rangle$	$\langle 2,2,n_{14}\rangle$	$\langle 0\rangle$

  Simulate F_2 on φ_{n_4}. If a hypothesis h_2 is produced by F_2 on $\varphi_{n_4}^{[x]}$, let $x_0 \leftarrow \max(x, y) + 1$, output $\langle 0\rangle$ up to $x_0 - 1$, and go to stage 4.

- *Stage 4.*
  Output values as in the next table.

	x_0	$\ldots$	x_0+5	$\ldots$
$\varphi_{n_{13}}$, $\varphi_{n_{15}},\ldots,\varphi_{n_{20}}$	$\langle 3,2,n_{15}\rangle$	$\ldots$	$\langle 8,2,n_{20}\rangle$	$\langle\rangle$
$\varphi_{n_{14}}$, $\varphi_{n_{21}},\ldots,\varphi_{n_{26}}$	$\langle 3,2,n_{21}\rangle$	$\ldots$	$\langle 8,2,n_{26}\rangle$	$\langle 0\rangle$

- *Stage 5.*
  Output values as in the next table.

	x_0	x_0+1	x_0+2	x_0+3	x_0+4	$\ldots$
$\varphi_{n_4},\varphi_{n_5}$, $\varphi_{n_{27}},\ldots,\varphi_{n_{31}}$	$\langle 1,2,n_{27}\rangle$	$\langle 2,2,n_{28}\rangle$	$\langle 3,2,n_{29}\rangle$	$\langle 7,2,n_{30}\rangle$	$\langle 8,2,n_{31}\rangle$	$\langle\rangle$
$\varphi_{n_6},\varphi_{n_9}$, $\varphi_{n_{32}},\ldots,\varphi_{n_{36}}$	$\langle 1,2,n_{32}\rangle$	$\langle 2,2,n_{33}\rangle$	$\langle 3,2,n_{34}\rangle$	$\langle 7,2,n_{35}\rangle$	$\langle 8,2,n_{36}\rangle$	$\langle 0\rangle$
$\varphi_{n_{10}},\varphi_{n_{11}}$, $\varphi_{n_{37}},\ldots,\varphi_{n_{41}}$	$\langle 1,2,n_{37}\rangle$	$\langle 2,2,n_{38}\rangle$	$\langle 3,2,n_{39}\rangle$	$\langle 7,2,n_{40}\rangle$	$\langle 8,2,n_{41}\rangle$	$\langle 1\rangle$

End of the algorithm.

Let m be the stage in which the algorithm remains forever. If $m=1$, noone of the strategies F_1 and F_2 produced any hypothesis. If $m=2$, φ_{h_1} was undefined at x_0+4, and F_2 did not output any hypothesis. If $m=3$, $\varphi_{h_1}(x_0+4)\neq\varphi_{n_4}(x_0+4)$, and F_2 did not output any hypothesis. If $m=4$, φ_{h_1} differs from both $\varphi_{n_{13}}$ and $\varphi_{n_{14}}$ at x_0+4, where we take the x_0 value at the beginning of stage 3; and φ_{h_2} cannot be equal to both $\varphi_{n_{13}}$ and $\varphi_{n_{14}}$. If $m=5$, for at least one of the functions φ_{n_4}, φ_{n_6}, $\varphi_{n_{10}}$ both hypotheses h_1 and h_2 are incorrect. □

Theorem 30 $(\forall n\geq 1)[\text{cdeg}([n,2n-1]\mathbf{Fin})\leq 2n+2]$.

Proof. Suppose all the unions of $2n+1$ out of classes $U_1,\ldots,U_{2n+2}$ are in $[n,2n-1]\mathbf{Fin}$. Let $T_1,\ldots,T_{2n+2}$ be the teams that identify these unions. Each of these teams consists of $2n-1$ strategies. We are going to construct an algorithm F that models strategies $F_1,\ldots,F_{2n-1}$ $[n,2n-1]\mathbf{Fin}$-identifying $\bigcup_{i=1}^{2n+2}U_i$ using the $(2n+2)(2n-1)$ strategies as subroutines.

We shall denote by $h_{j,i}$ the i-th hypothesis output in the team T_j, $1\leq i\leq 2n-1$. The algorithm for F is as follows.

- *Stage 1.*
  Receiving $f^{[x]}$ in input perform x steps in computing the outputs of the strategies on $f^{[0]},\ldots,f^{[x]}$ and for any hypothesis $h_{j,i}$ computed perform x steps in computing $\varphi_{h_{j,i}}(0),\ldots,\varphi_{h_{j,i}}(x)$. This is done throughout all stages. Wait until in $2n+1$ teams n hypotheses are produced. We can assume that

they are $h_{j,i}$ for $1 \leq j \leq 2n+1$ and $1 \leq i \leq n$. Output h_1 by F_1, h_2 by F_2, ..., h_n by F_n based on these $(2n+1) \cdot n$ hypotheses. and the $(2n+1)(2n-1)$ strategies of the corresponding teams, discard the $(2n+2)$-th team (do not consider it anymore), and go to stage 2.

- *Stage 2.*
  Let $k_i(x)$, $1 \leq i \leq n-1$, be the amount of the teams in which the $(n+i)$-th hypothesis has been output when F has performed all the computations corresponding to the input $f^{[x]}$. Clearly, $k_i(x) \geq k_{i+1}(x)$ for $1 \leq i \leq n-2$.

  Let m, $n \leq m \leq 2n-2$, be the number of hypotheses h_j already output. h_{m+1} is output by F_{m+1} at input $f^{[x]}$ if $k_i(x') \geq 2n+1-2i$ for some $i > m-n$ and $x' \leq x$, and in all teams but one it is computed that at least n hypotheses (among those known to F at $f^{[x]}$) output correct values in the interval $[0, x']$.

Now we describe the scheme according to which the hypotheses h_j cooperate. We assign priorities to the hypotheses h_j. The hypotheses that are output later have higher priority than those that were output sooner. If the hypotheses were output at the same time, then the lower index, the higher priority. The values are output one by one, at points $0, 1, 2, \ldots$. When φ_{h_j} outputs value at some point x, it simulates all the $(2n+1) \cdot (2n-1)$ strategies on $\varphi_{h_j}(x)$ and their hypotheses with the same procedure as in F. Thus φ_{h_j} can keep track of the new hypotheses (unknown when h_i was output), of what hypotheses are output by F, and of what values are output by other hypotheses $h_{j'}$.

Let H be a set of hypotheses h_j that have output the same values in the interval $[0, x-1]$. Let t_i, $n \leq t_i \leq 2n-1$ be the amount of hypotheses in the team T_i known to the hypotheses from H when they are computing what to output at x. Each value considered for output must satisfy the following condition: for each i, $1 \leq i \leq 2n+1$, except one, this value is output by at least $t_i - n + 1$ hypotheses in T_i, and these hypotheses output the same values in $[0, x-1]$ as φ_{h_j}, $h_j \in H$.

Suppose this condition is obeyed for $l \geq 1$ different values $y_1, \ldots, y_l$. When it becomes known for y_j, the hypothesis with the highest priority from H that has not output any value at x yet outputs y_j. Suppose $m \geq 1$ hypotheses from H have already output y_j. If for some $i \geq 0$: in $2n-2m-2i$ teams there are $m+i+1$ hypotheses that output y_j at x and the same values as φ_{h_j}, $h_j \in H$, in the interval $[0, x-1]$, then the next hypothesis from the priority queue formed in H outputs y_j. Note that the first hypothesis that outputs y_j also satisfies this condition with $m = i = 0$.

Naturally, if h_j was output at $f^{[x]}$ for some x, then φ_{h_j} outputs the known values of f in the interval $[0, x]$ and begins to cooperate with other hypotheses starting with point $x+1$.

Now, let us prove that at least n hypotheses among h_j are correct in case $f \in \bigcup_{i=1}^{2n+2} U_i$. Since in this case all teams among $T_1, \ldots, T_{2n+2}$ but one finally output at least n correct hypotheses and since, in the notation of stage 2 of the algorithm for F, $k_i(x)$ is a non-decreasing function, we have: if $k_i(x_0) \geq 2n+1-2i$ for some x_0, then the needed x' will be found sooner or later. Also, if the hypotheses from H have output the correct values in the interval $[0, x-1]$, then there are at least $t_i - n + 1$ correct hypotheses in the team T_i known to the hypotheses from H, for each i from $[1, 2n+1]$ except one.

Let $n+m$ be the total amount of hypotheses h_j output on f, $0 \leq m \leq n-1$, and let $l = n-1-m$. Then there are at most $2 \cdot (l-(i-n-m)+1)$ teams in which at least i hypotheses are produced, $n+m+1 \leq i \leq 2n-1$, and there are at least $2n+1-2l$ teams in which at most $n+m$ hypotheses are produced. Let us denote the set of the latter teams by S. At least $2n-2l$ of the teams from S have at least $n-m$ correct hypotheses among their first n hypotheses. Therefore, according to the conditions on which values are output by φ_{h_j}, hypotheses h_j, $1 \leq j \leq n$, at each input see sufficient information for $n-m$ hypotheses from the priority queue to output the correct values. When h_j, $j > n$, is output, in at least $2n+1-2i$ teams there are at least $n+i$ hypotheses for some $i \geq j-n$. Then at least $2n+1-2l-2i$ of these teams are in S, and among their $n+i$ hypotheses known at the moment there are at least $n+i-m$ correct ones. Therefore, when $j > n$ hypotheses have been output, there is sufficient information for $j-m$ hypotheses from the priority queue to output the correct values. For $j = n+m$ that gives us n correct hypotheses. The only problem is that, when a correct value is to be output, the queue may turn out to be empty. There are two possible reasons for that: some hypotheses might output incorrect values, and some hypotheses might infinitely wait at some previous argument — where less hypotheses are known, and therefore the information is insufficient for another hypothesis to output the correct value.

To show that these possibilities do not occur, let us count how many of h_j can be incorrect due to producing some incorrect value. Since the values at previous points are checked, one incorrect hypothesis cannot "cheat" the hypotheses h_j twice. Let p be the amount of incorrect values output (for each hypothesis h_j we choose only its first error, if any), and let s_i be the amount of hypotheses among h_j that output the i-th incorrect value, $1 \leq i \leq p$. Then, according to the algorithm for φ_{h_j}, there are at least $2n+1-2(s_i-1)-1$ teams in which at least s_i hypotheses output the i-th incorrect value. So, there are at least $2n+1-\sum_{i=1}^{p}(2s_i-1)$ teams with $\sum_{i=1}^{p} s_i$ incorrect hypotheses. If $p \geq 1$, that gives us at least $2n+2-2\sum_{i=1}^{p} s_i$ such teams. If $\sum_{i=1}^{p} s_i$ reaches the value $m+1$, we have at least $2n-2m$ such teams. In at least $2n-2m-1$ of them there are at least n correct hypotheses, so the total amount of hypotheses in each of them is at least $n+m+1$. But, according to our definitions of m and l, the amount of such teams does not exceed $2l = 2n-2m-2$. Contradiction. So no more than m of the hypotheses h_j output incorrect values.

We are going to prove that, when $u \geq n$ hypotheses h_j are output, there are at least $u - m$ hypotheses in the priority queue. Initially $u = n$, there are no more than m of h_j output incorrect values, and there is sufficient information for $n - m$ hypotheses to output the correct values, so this is true for $u = n$. This can become untrue only if for some $u > n$ there are $u - m$ hypotheses in the priority queue, and one of them outputs an incorrect value at some point x. Let it be the i-th time when one of h_j outputs an incorrect value. Then, for some $i' \geq 0$, there are $2n + 2 - 2(i + i')$ teams in which $i + i'$ hypotheses output this incorrect value, while at the previous points they output the correct values. $2n + 2 - 2(i + i' + l)$ of these teams are in S (note that $i + i' \leq m$). Then among the first n hypotheses of $2n + 2 - 2(i + i' + l)$ teams from S there are at least $n-m+i+i'$ such that output correct values at least in the interval $[0, x-1]$. Thus there is sufficient information for at least $n-m+i+i'$ hypotheses h_j to output the correct values up to x. Let $i'' \leq i - 1$ be the amount of hypotheses h_j producing incorrect outputs in the interval $[0, x - 1]$. Then at least $n - m + i + i' - i''$ hypotheses among $h_1, \ldots, h_n$ output the correct values until h_{n+1} is produced. As we showed above, that implies there is sufficient information for one more strategy to output correct values, so we have at least $n + 1 - m + i + i' - i''$ hypotheses among $h_1, \ldots, h_{n+1}$ that output correct values until h_{n+2} is produced, etc., we have at least $u - m + i + i' - i''$ hypotheses among $h_1, \ldots, h_u$ that output correct values up to the point $x - 1$ including. At most $i - 1 - i''$ hypotheses have an error at x before the considered error, so at the moment of this error there are at least $u - m + i + i' - i'' - (i - 1 - i'') = u - m + i' + 1 > u - m$ hypotheses in the priority queue. Contradiction.

So, after h_{n+m} is output, at each point there are always at least n hypotheses in the priority queue, and there is sufficient information for at least n hypotheses to output the correct value. Therefore, at least n of h_j are correct hypotheses. □

Theorem 31 $(\forall n \geq 1)[\mathrm{cdeg}([n, 2n - 1]\mathbf{Fin}) > 2n + 1]$.

Proof. $n = 1$ yields the class $[1, 1]\mathbf{Fin} = \mathbf{Fin}$ that was considered in Theorem 10. So, we suppose that $n > 1$. It is enough to show that there are such classes $U_1, \ldots, U_{2n+1}$ that the unions of $2n$ classes out of them are identifiable, while $\bigcup_{j=1}^{2n+1} U_j$ is not.

We define $U_i = (\bigcap_{j=1, j\neq i}^{2n+1} I_j^{[n,2n-1]\mathbf{Fin}})$, $1 \leq i \leq 2n + 1$. Then $\bigcup_{i=1, i\neq j}^{2n+1} U_i \subseteq I_j^{[n,2n-1]\mathbf{Fin}} \in [n, 2n - 1]\mathbf{Fin}$, $1 \leq j \leq 2n + 1$.

We have to prove that $\bigcup_{i=1}^{2n+1} U_i \notin [n, 2n - 1]\mathbf{Fin}$. As usually, we construct functions φ_{n_i} that use each other's Gödel numbers. Let T be an arbitrary team of $2n - 1$ strategies, $F_1, \ldots, F_{2n-1}$.

The procedure new(x) is the same as in previous such proofs. The algorithm for φ_{n_i} is as follows.

- *Stage 0.*
  Let $c \leftarrow 1$. Execute new(s_j^i) for $1 \leq i \leq 2n$, $1 \leq j \leq n$. Informally, s_j^i

will be used as the j-th hypothesis for the i-th team. Let the y throughout the algorithm indicate the maximal value of argument at which the values have been output at the moment. All the functions $\varphi_{s^i_j}$ output $\langle k, l, s^k_l\rangle$ at point $(k-1)\cdot n + l - 1$ for $1 \le k \le 2n$, $1 \le l \le n$, and $\langle\rangle$ at further points, until n hypotheses $h_1, \ldots, h_n$ are produced in T on $\varphi^{[x]}_{s^1_1}$. Then let $x_0 \leftarrow \max(x, y) + 1$, output $\langle\rangle$ up to $x_0 - 1$, and go to stage 1.

- *Stage 1.*
Execute new(s^i_{n+1}) for $1 \le i \le 2n-2$ and new(s^{2n+1}_j) for $1 \le j \le n$. The functions $\varphi_{s^i_j}$ with $i \in \{1, \ldots, 2n+1\} - \{2n-1\}$, $1 \le j \le n-1$, $\varphi_{s^i_{n+1}}$ with $1 \le i \le 2n-2$, $\varphi_{s^{2n}_n}$ and $\varphi_{s^{2n+1}_n}$ output the value $\langle i, n+1, s^i_{n+1}\rangle$ at point $x_0 + i - 1$ for $1 \le i \le 2n-2$, the value $\langle 2n+1, j, s^{2n+1}_j\rangle$ at point $x_0 + 2n - 3 + j$ for $1 \le j \le n$, and $\langle\rangle$ at further points while in this stage.
Suppose hypothesis h_{n+1} is output in T on $\varphi^{[x]}_{s^1_1}$ for some x. Then the functions $\varphi_{s^i_n}$ with $1 \le i \le 2n-1$ and $\varphi_{s^{2n-1}_j}$ with $1 \le j \le n-1$ output the values listed above in this stage, let $x_0 \leftarrow \max(x, y) + 1$, all the introduced functions output $\langle\rangle$ up to $x_0 - 1$, and go to stage 2.
Suppose all the functions $\varphi_{h_1}, \ldots, \varphi_{h_n}$ output $\langle\rangle$ at $x_0 + 3n - 2$. Then execute new(s^i_j) for $1 \le i \le 2n-2$, $n+1 \le j \le 2n-1$, new(s^{2n+1}_j) for $1 \le j \le n$. The functions $\varphi_{s^i_j}$ with $1 \le i \le 2n-2$, $n \le j \le 2n-1$, $\varphi_{s^{2n-1}_j}$ and $\varphi_{s^{2n+1}_j}$ with $1 \le j \le n$ output the value $\langle i, j, s^i_j\rangle$ at point $x_0 + (i-1)\cdot(n-1) + j - n - 1$ for $1 \le i \le 2n-2$, $n+1 \le j \le 2n-1$, and the value $\langle 2n+1, j, s^{2n+1}_j\rangle$ at point $x_0 + (2n-2)(n-1) + j - 1$ for $1 \le j \le n$, and $\langle 0\rangle$ at all the further points.

- *Stage m $(2 \le m \le n-1)$.*
Execute new(s^i_{n+m}) for $1 \le i \le 2n-2m$ and new(s^i_{n+m-1}) for $2n-2m+3 \le i \le 2n+1$. The functions $\varphi_{s^i_j}$ with $i \in \{1, \ldots, 2n+1\} - \{2n-2m+1\}$, $1 \le j \le n-1$, $\varphi_{s^{2n-2m+2}_n}$, $\varphi_{s^i_{n+m}}$ with $1 \le i \le 2n-2m$, and $\varphi_{s^i_{n+m-1}}$ with $2n-2m+3 \le i \le 2n+1$ output the value $\langle i, n+m, s^i_{n+m}\rangle$ at point $x_0 + i - 1$ for $1 \le i \le 2n-2m$, the value $\langle i, n+m-1, s^i_{n+m-1}\rangle$ at point $x_0 + i - 3$ for $2n-2m+3 \le i \le 2n+1$, and $\langle\rangle$ at further points while in this stage.
Suppose hypothesis h_{n+m} is output in T on $\varphi^{[x]}_{s^1_1}$ for some x. Then the functions $\varphi_{s^i_n}$ with $1 \le i \le 2n+1$, $i \ne 2n-2m+2$, $\varphi_{s^i_j}$ with $1 \le i \le 2n+1$, $n+1 \le j \le n+m-2$, $\varphi_{s^i_{n+m-1}}$ with $1 \le i \le 2n-2m+2$, and $\varphi_{s^{2n-2m+1}_j}$ with $1 \le j \le n-1$ output the values listed above in this stage, let $x_0 \leftarrow \max(x, y) + 1$, all the introduced functions output $\langle\rangle$ up to $x_0 - 1$, and go to stage $m+1$.
Suppose n of the functions $\varphi_{h_1}, \ldots, \varphi_{h_{n+m-1}}$ output $\langle\rangle$ at $x_0 + 2n - 1$. Then execute new(s^i_j) for $1 \le i \le 2n-2m$, $n+m \le j \le 2n-1$, new(s^i_j) for $2n-2m+3 \le i \le 2n+1$, $n+m-1 \le j \le 2n-1$. The functions

$\varphi_{s_j^i}$ with $i \in \{1, \ldots, 2n+1\} - \{2n-2m+1, 2n-2m+2\}$, $n \leq j \leq 2n-1$, and $\varphi_{s_j^{2n-2m+1}}$ with $1 \leq j \leq n$ output the value $\langle i, j, s_j^i \rangle$ at point $x_0+(i-1)(n-m)+j-n-m$ for $1 \leq i \leq 2n-2m$, $n+m \leq j \leq 2n-1$, and the value $\langle i, j, s_j^i \rangle$ at point $x_0+(2n-2m)(n-m)+(i-1)(n-m+1)+j-n-m+1$ for $2n-2m+3 \leq i \leq 2n+1$, $n+m-1 \leq j \leq 2n-1$, and $\langle 0 \rangle$ at all the further points.

- *Stage n.*
  Execute new(s_{2n-1}^i) for $3 \leq i \leq 2n+1$ and new(t_{2n-1}^i) for $3 \leq i \leq 2n+1$. The functions $\varphi_{s_j^i}$ with $1 \leq i \leq 2n+1$, $1 \leq j \leq n-1$, $\varphi_{s_{2n-1}^1}$, and $\varphi_{s_{2n-1}^i}$ with $3 \leq i \leq 2n+1$ output the value $\langle i, 2n-1, s_{2n-1}^i \rangle$ at point x_0+i-3 for $3 \leq i \leq 2n+1$, and $\langle\rangle$ at all the further points.
  The functions $\varphi_{s_j^i}$ with $1 \leq i \leq 2n+1$, $n \leq j \leq 2n-2$, $\varphi_{s_{2n-1}^2}$, and $\varphi_{t_{2n-1}^i}$ with $3 \leq i \leq 2n+1$ output the value $\langle i, 2n-1, t_{2n-1}^i \rangle$ at point x_0+i-3 for $3 \leq i \leq 2n+1$, and $\langle 0 \rangle$ at all the further points.

At stage m, $1 \leq m \leq n-1$, if at least n of the hypotheses h_i output the supposedly correct value $\langle\rangle$ at some fixed point, we ensure that they have an anomaly at this point. If no more than $n-1$ of these hypotheses output $\langle\rangle$ at this point, then no more than $n-1$ of them are correct hypotheses, and the team T must issue another hypothesis. In stage n T has already issued the $2n-1$ allowed hypotheses, and at least n of them are incorrect for one of the alternatives represented by the functions $\varphi_{s_1^1}$ and $\varphi_{s_n^1}$. So in all cases at least n hypotheses issued by T are incorrect. □

4.5.3 Finiteness of cdeg for Team Identification Types

From the previous sections we see that in all considered cases, when the **Ex** identification type is modified by bounds on mindchanges and teams, the cdeg turns out to be finite. Only some number of allowed anomalies introduces infinite cdeg values. After inspecting the diagonalization proofs for the identification types with allowed anomalies, it seems intuitively that the reason for infinite cdeg values is intransitivity of the basic relation $=^a$ in the case of a anomalies, $0 < a < *$.

Now, suppose we discard the types with allowed anomalies due to this property of theirs. Which other criterions must an identification type obey to have a finite cdeg? Can we prove the finiteness of cdeg for some large class of identification types? The first question currently seems to be too complex and too general to answer. As for the second question, we have this complex hierarchy of $[m, n]$**Fin** identification types for which we were unable to find exact cdeg values. Maybe we can at least prove that they are finite?

Well, we can prove such result if we introduce an additional constraint on the strategies, similarly as it was done in [1] to prove the estimation of the complexity of the probability hierarchy for the $[m,n]$**Fin** identification types. This constraint requires that *all* hypotheses output by the strategy on *any* input are Gödel numbers of total recursive functions. Such identification types are called *Popperian.* If we additionally allow no mindchanges, we obtain the Popperian **Fin** or, as we shall denote it, **PFin** identification type.

The next theorem was proved in colaboration with A. Ambainis.

Theorem 32 $(\forall m,n \in \mathbb{N} \mid m \leq n)[\mathrm{cdeg}([m,n]\mathbf{PFin})$ *is finite*$]$.

Proof is by induction.

Base case. If $m = n$, then $[m,n]\mathbf{PFin} = \mathbf{PFin}$, and $\mathrm{cdeg}([m,n]\mathbf{PFin}) = 4$, and there is a simulation algorithm proving it.

Inductive case. Assume that $\mathrm{cdeg}([m',n']\mathbf{PFin})$ is finite for all $n' < n$ and all $m' \leq n'$, and there is a simulation algorithm proving it. We are going to prove that $\mathrm{cdeg}([m,n]\mathbf{PFin})$ is finite for any $m < n$.

Let $c_{[i,j]}$ be the smallest number for which there is a simulation algorithm proving the $(c_{[i,j]}+1)$-closedness of $[i,j]\mathbf{PFin}$ (i. e. a simulation of $[c_{[i,j]}, c_{[i,j]}+1][i,j]\mathbf{PFin}$ by $[i,j]\mathbf{PFin}$), where $i \leq j$, and define $c_{[i,j]} = 0$ for $i > j$. Define the sequence $\{a_i\}$ as follows.

$$a_1 = 2c_{[m,n-m]},\ a_j = \max\left(c_{[m,n-m+j]}, 2c_{[m,n-m]} + \textstyle\sum_{i=1}^{j-1} \max(a_i, a_{j-i})\right),$$
$$\text{for } j = 2, \ldots, m-1, \text{ and}$$
$$a_m = 2c_{[m,n-m]} + \textstyle\sum_{i=1}^{m-1} \max(a_i, a_{m-i}).$$

We claim that $\mathrm{cdeg}([m,n]\mathbf{PFin}) \leq a_m + 2$.

Consider the following algorithm for simulating a multiteam $F = \{F_1, F_2, \ldots, F_{a_m+2}\}$ consisting of $a_m + 2$ $[m,n]\mathbf{PFin}$-teams by a single $[m,n]\mathbf{PFin}$-team G.

G reads input and waits until at least m strategies output hypotheses in at least $a_m + 1$ teams. Then, m strategies from G output hypotheses based on the first m hypotheses in these teams.

All remaining strategies in G continue reading input and simulating F and its hypotheses. Hypotheses already output by G simulate hypotheses of F as well as the remaining strategies of F on the initial segments which the hypotheses of G follow. Only a certain amount of steps is performed in simulating the strategies of F at each new value of the input function by the hypotheses of G to ensure that these hypotheses are total.

If all the hypotheses of F have the same next value $f(n)$, then the hypotheses of G have the same value $f(n)$. Otherwise, we say that the hypotheses of F *split.* It is enough to define how to simulate splits into two groups because a split into more groups is equivalent to a sequence of several splits into two groups.

Suppose a split occurs.

If one group (i. e. hypotheses with one next value) contains hypotheses output by less than $a_m + 1 - c_{[m,n-m]}$ teams, no G's hypothesis follows this group (i. e.

has this next value). If the next value of this group is correct, then the remaining strategies in G apply the simulation algorithm for $[m, n-m]$**PFin** to a multiteam consisting of those F_i which do not have any hypothesis following this group. There are at least $c_{[m,n-m]}+1$ such F_i. In each of them m hypotheses follow other next values and $n-m$ strategies remain. Hence, these F_i form a multiteam of at least $c_{[m,n-m]}+1$ $[m, n-m]$**PFin**-teams. By the inductive assumption, this multiteam can be simulated by a single $[m, n-m]$**PFin**-team.

The first m hypotheses of G proceed supposing that this value is incorrect. If the same condition (less than $a_m+1-c_{[m,n-m]}$ teams) applies also to the second group, then these m hypotheses can output any values; the simulation is done by the remaining $n-m$ strategies of G as described above. Otherwise, all the m hypotheses follow the second group. If the next value of this group is also incorrect, then, again, there are at least $c_{[m,n-m]}+1$ teams in F in which at least m hypotheses follow incorrect values, and the simulation is done by the remaining $n-m$ strategies.

Supposing that the next value of the second group is correct the m hypotheses of G follow this group until the next split occurs. Then we apply the same argument.

So, the simulation will be successful if at each split at least one of the groups is not followed by any strategy from at least $c_{[m,n-m]}+1$ teams of F.

Now, suppose at some split there are no $c_{[m,n-m]}+1$ teams in F in which all the initial m hypotheses follow one of the groups. This we shall call *an essential split.* Let b_j be the number of teams in which j hypotheses follow the first value and $m-j$ hypotheses follow the second value (if this is not the first split, we can add the values which split off previously to one of the two groups; remember that no hypothesis from the first m hypotheses of G needs to actually follow these values). According to our assumption, $b_0 \leq c_{[m,n-m]}$ and $b_m \leq c_{[m,n-m]}$. Since $\sum_{j=0}^{m} b_j = a_m+1$, at least one of the following inequalities hold: $b_j > \max(a_j, a_{m-j})$ for $j = 1, \ldots, m-1$. Then the m hypotheses of G select one such j, select the corresponding b_j teams discarding the others, and j of the hypotheses follow the first value, and the $m-j$ remaining follow the second value.

The j hypotheses following the first value perform the simulation algorithm described above for the $b_j \geq a_j+1$ teams and their j hypotheses until either another essential split occurs, or until they calculate that at least $m-j$ of the remaining strategies in all of these teams but one output hypotheses following this group. Similarly the $m-j$ hypotheses following the second value simulate the $b_j \geq a_{m-j}+1$ teams and their $m-j$ hypotheses.

If another essential split occurs, say, in the first group, then denoting by b'_i the number of teams in which i hypotheses follow the first value and $j-i$ hypotheses follow the second value (then $\sum_{i=0}^{j} b'_i = b_j > a_j+1$) we again obtain for some i $(1 \leq i \leq j-1)$: $b'_i > \max(a_i, a_j - i)$, and continue the simulation for those b'_i teams.

Now, suppose that after some essential split, when j hypotheses of G follow at least $a_j + 1$ teams of F and their j hypotheses, the second possibility occurs, i. e. in at least a_j of these teams $m - j$ of the remaining $n - m$ strategies output new hypotheses following this group, and the j hypotheses of G calculate that. Then there are in total m hypotheses in each of these teams of F following this group and known to both the j hypotheses and the remaining $n - m$ strategies of G. So, if this group follows the actual values of the input function, then $m - j$ of the remaining strategies of G output new hypotheses which join with the j initial hypotheses to simulate the m hypotheses in each of the a_j teams of F. Since $a_j \geq c_{[m,n-m+j]}$, we can apply inductively the simulation algorithm for $[m, n - m + j]$**PFin**. □

Chapter 5

Identifying Languages

In this chapter we investigate the closedness degrees of the language learning classes $\mathbf{TxtEx}_b^a$ — the analogues of $\mathbf{Ex}_b^a$. For those interested in other language learning types we recommend the monograph [25].

Many results are similar to the case of function learning.

Theorem 33

$$(\forall b \in \mathbb{N})(\forall a, a' \in \mathbb{N} \cup \{*\} \mid a' \geq 2^{b+1}a)[\mathrm{csdeg}(\mathbf{TxtEx}_b^a, \mathbf{TxtEx}_b^{a'}) \leq 2^{b+2}].$$

Proof. It is sufficient to prove that $\mathbf{TxtEx}_b^a$ is 2^{b+2}-closed in $\mathbf{TxtEx}_b^{a'}$.

Let $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_{2^{b+2}} \subseteq \mathcal{E}$ be such families of languages that all the unions of $2^{b+2} - 1$ out of them are $\mathbf{TxtEx}_b^a$-identifiable. Let $F_1, F_2, \ldots, F_{2^{b+2}}$ be the strategies that identify these unions. Now we construct a strategy F.

The strategy F redirects its input to the strategies F_i until $2^{b+2}-1$ of them output a hypothesis. Such an event happens because every language $L \in \bigcup_{j=1} 2^{b+2}\mathcal{L}_j$ belongs to $2^{b+2} - 1$ of the unions of $2^{b+2} - 1$ families, thus at most one of the strategies F_i does not identify L.

Then F outputs h_0 that is based on these hypotheses. In further F outputs h_i, $1 \leq i \leq b$, iff it has output h_{i-1} and at least $2^{b+2-i} - 1$ of the strategies, on whose hypotheses was based h_{i-1}, output a new hypothesis. h_i is based on these hypotheses $h_i^1, \ldots, h_i^{2^{b+2-i}-1}$ in the following way: $x \in W_{h_i}$ iff at least 2^{b+1-i} of $W_{h_i^j}$, $1 \leq j \leq 2^{b+2-i} - 1$, contain x. h_0 is defined similarly.

It is easy to see that h_i can be a wrong $\mathbf{TxtEx}_b^*$-hypothesis only if at least $2^{b+1-i} - 1$ of the strategies, on whose hypotheses h_i was based, output a new hypothesis. But in this case h_{i+1} is output. h_b, if output, is always a correct $\mathbf{TxtEx}_b^*$-hypothesis, since it is based on the last allowed hypotheses of 3 strategies, at least 2 of which identify the language. So the last hypothesis h_{i_0} output by F is based on $2^{b+2-i_0} - 1$ hypotheses, 2^{b+1-i_0} of which are right $\mathbf{TxtEx}_b^a$-hypotheses. h_{i_0} can have an anomaly only for the values at which at least one of these right hypotheses have an anomaly, that is at no more than $2^{b+1}a$ points. □

Theorem 34 $(\forall b \in \mathbb{N})[\mathrm{csdeg}(\mathbf{TxtEx}_b, \mathbf{TxtEx}_b^*) > 2^{b+2} - 1]$.

Proof. We prove that $\mathbf{TxtEx}_b$ is not $(2^{b+2}-1)$-closed in $\mathbf{TxtEx}_b^*$. We shall use the same kinds of instructions as in the previous chapter in the case of $\mathbf{Ex}_b^a$-identification, and define the class of j-instructors $I_j^{\mathbf{TxtEx}_b^a}$ in a similar way. We define the language classes $\mathcal{L}_i = \bigcap_{j \neq i} I_j^{\mathbf{TxtEx}_b}$, where $i, j \in [1, 2^{b+2}-1]$. Then $\bigcup_{i \neq j} \mathcal{L}_i \subseteq I_j^{\mathbf{TxtEx}_b} \in \mathbf{TxtEx}_b$.

We shall prove that $\mathcal{L} = \bigcup_{j=1}^{2^{b+2}-1} \mathcal{L}_j \notin \mathbf{TxtEx}_b^*$. We apply diagonalization over the strategies F and the multiple recursion theorem to construct functions φ_{n_i} (and thus the sets W_{n_i}) that use F and the Gödel numbers of themselves. The algorithm for φ_{n_i} is as follows.

- *Stage 0.*
  Let $k \leftarrow 2^{b+2}-2$. Put $\langle 1,1,1,n_1\rangle, \ldots, \langle 1,k,1,n_k\rangle$ in all k languages W_{n_1}, …, W_{n_k}. Let $r \leftarrow 0$, $s \leftarrow k$. Feed larger and larger initial segments of a text for the defined part of W_{n_1} to the strategy F. If F outputs a hypothesis on a segment σ_0, go to stage 1.

- *Stage i $(1 \leq i \leq b+1)$.*
  Let $k \leftarrow k/2 - 1$.
  Put $\langle 1, r+k+1+j, i+1, n_{s+j}\rangle$ for $1 \leq j \leq k$, and
  $\langle 1, j, i, n_{s+k+j}\rangle$ for $1 \leq j \leq r$, and
  $\langle 1, r+2k+2+j, i, n_{s+r+k+j}\rangle$ for $1 \leq j \leq 2^{b+2}-r-2k-3$ in $W_{n_{r+1}}, \ldots,$ $W_{n_{r+k+1}}$.
  Make the languages $W_{n_{s+j}}$, $1 \leq j \leq 2^{b+2}-k-3$ to be equal to the defined part of $W_{n_{r+1}}$.
  Let $t \leftarrow s + 2^{b+2} - k - 3$.
  Put $\langle 1, r+j, i+1, n_{t+j}\rangle$ for $1 \leq j \leq k$, and
  $\langle 1, j, i, n_{t+k+j}\rangle$ for $1 \leq j \leq r$, and
  $\langle 1, r+2k+2+j, i, n_{t+r+k+j}\rangle$ for $1 \leq j \leq 2^{b+2}-r-2k-3$ in $W_{n_{r+k+2}}, \ldots,$ $W_{n_{r+2k+2}}$.
  Put $\langle 0, w\rangle$ in these $k+1$ languages for larger and larger values of w while in this stage. Make the languages $W_{n_{t+j}}$, $1 \leq j \leq 2^{b+2}-k-3$ to be equal to the defined part of $W_{n_{r+k+2}}$.
  The following applies only to the case $i \leq b$. Take larger and larger extensions of σ_{i-1} that give texts for the languages $W_{n_{r+1}}$ and $W_{n_{r+k+2}}$ and give them as an input to F. If F makes a mindchange on the text for $W_{n_{r+1}}$, let $r \leftarrow s$. If F makes a mindchange on the text for $W_{n_{r+k+2}}$, let $r \leftarrow t$. In both cases let σ_i be the segment on which the new hypothesis is output, $s \leftarrow t + 2^{b+2} - k - 3$, and go to stage $i+1$.

$k = 2^{b+2-i} - 2$ is the number of $(i+1)$-th hypotheses proposed in the instructions which are put in the languages $W_{n_{r+1}}, \ldots, W_{n_{r+2k+2}}$ at the i-th stage. s is the number of indices n_j already used at the beginning of stage, while t is the corresponding number at the middle of the stage. r is used for the indices of the languages on whose text F makes a new hypothesis.

Suppose the process reached the stage i_0 and remained there. If $i_0 = 0$, F has not issued any hypothesis on a text for $W_{n_1} \in \mathcal{L}_{2^{b+2}-1}$. If $i_0 > 0$, the last hypothesis issued by F is invalid for at least one of the languages $W_{n_{r+1}} \in \mathcal{L}_{r+2k+2}$ and $W_{n_{r+k+2}} \in \mathcal{L}_{r+k+1}$, because they differ in infinitely many values of kind $\langle 0, w\rangle$. (Note that at the stage $b+1$ the strategy F has already made b mindchanges.) So F does not $\mathbf{TxtEx}_b^*$-identify $\mathcal{L}$. $\square$

Corollary 5.1 $(\forall b \in \mathbb{N})[\mathrm{cdeg}(\mathbf{TxtEx}_b) = 2^{b+2}]$.

Corollary 5.2 $(\forall b \in \mathbb{N})[\mathrm{cdeg}(\mathbf{TxtEx}_b^*) = 2^{b+2}]$.

The next theorem is rather surprising. Recall that in the identification of total recursive functions we have: $\mathrm{csdeg}(\mathbf{Ex}_b^a, \mathbf{Ex}_{b'}^a) = 2$ for sufficiently large b' (Theorem 14).

Theorem 35 $(\forall a \in \mathbb{N} \mid a \geq 1)[\mathrm{csdeg}(\mathbf{TxtEx}_0^a, \mathbf{TxtEx}^a) = \infty]$.

Proof. Let $k \in \mathbb{N}$, $k > 1$. We prove that $\mathbf{TxtEx}_0^a$ is not k-closed in $\mathbf{TxtEx}^a$. We define the language classes $\mathcal{L}_i = \bigcap_{j\neq i} I_j^{\mathbf{TxtEx}_0^a}$, where $i, j \in [1, k]$. Then $\bigcup_{i\neq j} \mathcal{L}_i \subseteq I_j^{\mathbf{TxtEx}_0^a} \in \mathbf{TxtEx}_0^a$.

We shall prove that $\bigcup_{j=1}^k \mathcal{L}_j \notin \mathbf{TxtEx}^a$. We apply diagonalization over the strategies F and the multiple recursion theorem. The algorithm for φ_{n_i} is as follows.

- *Stage 0.*
  Put $\langle 1, j, 1, n_j\rangle$, $1 \leq j \leq k$, in $W_{n_1}, \ldots, W_{n_k}$. Let $w \leftarrow 0$. Simulate F on some text for W_{n_1}. If F outputs a hypothesis h_0 on some initial segment σ_0 of the text, then go to stage 1.

- *Stage r ($r \geq 1$).*
  Let L_r denote the set of elements put in W_{n_1} before the start of stage r.

  - *Substage 0.*
    Put $\langle 0, i\rangle$, $w \leq i \leq w + a - 1$, in $W_{n_1}, \ldots, W_{n_{k-1}}$.
    Put $\langle 0, w + a\rangle$ in $W_{n_1}, \ldots, W_{n_{k-2}}$.
    Simulate F on such extensions of σ_{r-1} that give texts for all the languages $L_r \cup P$, where P is a non-empty subset of $\{\langle 0, i\rangle \mid w \leq i \leq w + a\}$. Simultaneously compute $\varphi_{h_{r-1}}(\langle 0, i\rangle)$ for $w \leq i \leq w + a$.
    Suppose F outputs a new hypothesis $h_r \neq h_{r-1}$ on a segment $\sigma_r \supset \sigma_{r-1}$. Then for $w \leq i \leq w + a$, add $\langle 0, i\rangle$ to W_{n_k}, add $\langle 0, w + a\rangle$ to $W_{n_{k-1}}$, let $w \leftarrow w + a + 1$ and go to stage $r + 1$.
    Suppose $\varphi_{h_{r-1}}(\langle 0, i\rangle)\downarrow$ for all $i \in \{w, \ldots, w + a\}$. Then go to substage 1.

- *Substage s $(1 \leq s \leq k-3)$.*
  Put $\langle 0, w+a+s-2\rangle$ in $W_{n_{k-s+1}}, \ldots, W_{n_k}$.
  Put $\langle 0, w+a+s\rangle$ in $W_{n_1}, \ldots, W_{n_{k-s-2}}$.
  Simulate F on such extensions of σ_{r-1} that give a text for $L_r \cup \{\langle 0,i\rangle \mid (w+a-1 \leq i \leq w+a+s-2) \vee i = w+a+s\}$. Simultaneously compute $\varphi_{h_{r-1}}(\langle 0, w+a+s\rangle)$.

  Suppose F outputs a new hypothesis $h_r \neq h_{r-1}$ on a segment $\sigma_r \supset \sigma_{r-1}$. Then for $w \leq i \leq w+a+s$, add $\langle 0,i\rangle$ to $W_{n_1}, \ldots, W_{n_k}$ (if necessary: all these values already have been added to some of these languages), let $w \leftarrow w+a+s+1$ and go to stage $r+1$.

  Suppose $\varphi_{h_{r-1}}(\langle 0, w+a+s\rangle)\downarrow$. Then go to substage $s+1$.
- *Substage $k-2$.*
  Simulate F on such extensions of σ_{r-1} that give a text for $L_r \cup \{\langle 0,i\rangle \mid w+a-1 \leq i \leq w+a+k-5\}$.

  Suppose F outputs a new hypothesis $h_r \neq h_{r-1}$ on a segment $\sigma_r \supset \sigma_{r-1}$. Then for $w \leq i \leq w+a+k-3$, add $\langle 0,i\rangle$ to $W_{n_1}, \ldots, W_{n_k}$ (if necessary), let $w \leftarrow w+a+k-2$ and go to stage $r+1$.

End of stage r.

Each of the stages deals with one hypothesis output by F. The language(s) on which F is simulated is/are chosen so that the current hypothesis has $a+1$ anomalies on it/them. There are two ways F can deal with this problem. First, it can change the current hypothesis. In this case all the differences between the current versions of languages W_{n_i} are cleared, and the algorithm goes to the next stage dealing with the new hypothesis. Second, the current hypothesis function can output a new value, so decreasing the number of anomalies. Then the algorithm goes to the next substage ensuring again $a+1$ anomalies. At substage $k-2$ the current hypothesis function has no more such possibility.

So, either F makes infinitely many mindchanges, or its last hypothesis has at least $a+1$ anomalies. □

Corollary 5.3 $(\forall a \in \mathbb{N} \mid a \geq 1)(\forall b \in \mathbb{N} \cup \{*\})[\mathrm{cdeg}(\mathbf{TxtEx}_b^a) = \infty]$.

The next three theorems are obtained from results in team learning (Theorems 17 and 20 in [24]) by applying Corollaries 3.3 and 3.4 to them.

Theorem 36 $\mathrm{csdeg}(\mathbf{TxtEx}, \mathbf{TxtEx}^*) > 3$.

Theorem 37 $\mathrm{cdeg}(\mathbf{TxtEx}) = 4$.

Theorem 38 $\mathrm{cdeg}(\mathbf{TxtEx}^*) = 4$.

Chapter 6

Conclusion

Tables 6.1 and 6.2 summarize the closedness degrees of the classes $\mathbf{Ex}_b^a$ and $\mathbf{TxtEx}_b^a$. The entries marked with asterisks were proved previously by other researchers (often in a different form, for instance as a result in team learning). Please, see references in the main body of this work.

Table 6.3 illustrates the csdeg$(\mathbf{Ex}_b, \mathbf{Ex}_d)$ values (for the formula of D_n and the inclusion of the anomalies see the exact formulations of theorems in Section 4.4).

Also, we obtained some results for team identification types: cdeg$([1,n]\mathbf{Ex}) = n+2$ (due partly to [3]), cdeg$([n, 2n-1]\mathbf{Ex}_0) = 2n+2$ and cdeg$([1,2]\mathbf{Ex}_0) = 9$.

An interesting question is: How does the hierarchy of success ratios k/l for the classes $[k,l]\mathbf{Ex}_b^a$ (or the probability hierarchy) look like in the cases when the closedness degree of $\mathbf{Ex}_b^a$ is ∞? Intuitively, it seems that two different success ratios (or two different probabilities) should yield two different learning powers. Is it really so?

A survey of the proofs shows that the problem considered in this work is related to the following model. We have n streams of data on some object (in this work a function or a language). Some of them are totally incorrect, some

Table 6.1: The closedness degrees of $\mathbf{Ex}_b^a$

$\mathbf{Ex}_b^a$ $\overset{a\rightarrow}{b\downarrow}$	0	1	2	...	*
0	4^*	9	∞	∞	4
1	8^*	51	∞	∞	8
2	16^*	303	∞	∞	16
...	...	...	∞	∞	...
n	2^{n+2*}	$\frac{7\cdot 6^{n+1}+3}{5}$	∞	∞	2^{n+2}
...	...	...	∞	∞	...
*	3^*	3^*	3^*	3^*	3^*

Table 6.2: The closedness degrees of $\mathbf{TxtEx}_b^a$

$\mathbf{TxtEx}_{b\downarrow}^{a\rightarrow}$	0	1	...	$*$
0	4^*	∞	∞	4
1	8	∞	∞	8
...	...	∞	∞	...
n	2^{n+2}	∞	∞	2^{n+2}
...	...	∞	∞	...
$*$	4^*	∞	∞	4^*

Table 6.3: The csdeg($\mathbf{Ex}_b, \mathbf{Ex}_d$) values

$b\downarrow \backslash d \rightarrow$	0	1	2	3	...	n	$n+1$	...	$2n-1$	...
0	4^*	3	2	2	...	2	2	...	2	...
1	∞	8^*	5	4	...	2	2	...	2	...
2	∞	∞	16^*	8	...	...	...	...	...	...
3	∞	∞	∞	32^*	...	...	...	...	...	...
...	∞	∞	∞	∞	...	...	...	...	...	...
n	∞	∞	∞	∞	∞	2^{n+2*}	$\leq D_{n+2}+1$	...	$\leq 2n+6$	...
...	∞	∞	∞	∞	∞	∞	...	...	...	...

are partially incorrect (in the form of some anomalies), and it is unknown which are which. How many of these streams must be correct (or partially correct) to outweigh the incorrect data so that it is possible to restore the information about the object with some given precision? In the cases considered in this work an incorrect stream can always be outweighed, given enough correct information streams, in the worst case at some expense in the precision of the restored information (additional anomalies). Is this so for all more or less natural identification types?

Bibliography

[1] A. Ambainis. Probabilistic and team PFIN-type learning: general properties. *Proceedings of the Ninth Conference on Computational Learning Theory,* pp. 157–168, ACM, 1996.

[2] A. Ambainis, K. Apsītis, R. Freivalds, W. Gasarch, C. H. Smith. Team learning as a game. In M. Li, A. Maruoka, editors, *Algorithmic Learning Theory. Lecture Notes in Artificial Intelligence,* vol. 1316, pp. 2–17. Springer-Verlag, 1997.

[3] K. Apsītis. Topological considerations in composing teams of learning machines. In K. Jantke, S. Lange, editors, *Algorithmic Learning for Knowledge Based Systems. Lecture Notes in Artificial Intelligence,* vol. 961, pp. 146–154. Springer-Verlag, 1995.

[4] K. Apsītis, R. Freivalds, M. Kriķis, R. Simanovskis, J. Smotrovs. Unions of identifiable classes of total recursive functions. In K. Jantke, editor, *Analogical and Inductive Inference. Lecture Notes in Artificial Intelligence,* vol. 642, pp. 99–107. Springer-Verlag, 1992.

[5] K. Apsītis, R. Freivalds, R. Simanovskis, J. Smotrovs. Unions of identifiable families of languages. In L. Miclet, C. de la Higuera, editors, *Grammatical Inference: Learning Syntax from Sentences. Lecture Notes in Artificial Intelligence,* vol. 1147, pp. 48–58. Springer-Verlag, 1996.

[6] K. Apsītis, R. Freivalds, R. Simanovskis, J. Smotrovs. Closedness properties in EX-identification of recursive functions. In M. M. Richter, C. H. Smith, R. Wiehagen, T. Zeugmann, editors, *Algorithmic Learning Theory. Lecture Notes in Artificial Intelligence,* vol. 1501, pp. 46–60. Springer-Verlag, 1998.

[7] K. Apsītis, R. Freivalds, R. Simanovskis, J. Smotrovs. Closedness properties in Ex-identification. To be included in the special issue of the *Journal of Theoretical Computer Science* devoted to Machine Learning, 1999.

[8] J. Bārzdiņš. Две теоремы о предельном синтезе. In J. Bārzdiņš, editor, *Теория алгоритмов и программ,* vol. 1, pp. 82–88. Latvian State University, Rīga, 1974.

[9] J. Bārzdiņš and R. Freivalds. On the prediction of general recursive functions. Soviet Math. Dokl., vol. 13, pp. 1224–1228, 1972.

[10] L. Blum and M. Blum. Toward a mathematical theory of inductive inference. *Information and Control,* vol. 28, pp. 125–155, 1975.

[11] J. Case and C. Lynes. Machine inductive inference and language identification. In M. Nielsen and E. M. Schmidt, editors, *Proceedings of the 9th International Colloquium on Automata, Languages and Programming. Lecture Notes in Computer Science,* vol. 140, pp. 107–115, Springer-Verlag, 1982.

[12] J. Case and C. Smith. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science,* vol. 25 (2), pp. 193–220, 1983.

[13] R. Daley, L. Pitt, M. Velauthapillai, and T. Will. Relations between probabilistic and team one-shot learners. In M. Warmuth and L. Valiant, editors, *Proceedings of the 1991 Workshop on Computational Learning Theory,* pp. 228–239, Morgan Kaufmann Publishers, 1991.

[14] R. P. Daley and C. H. Smith. On the complexity of inductive inference. *Information and Control,* vol. 69, pp. 12–40, 1986.

[15] R. Freivalds. Functions computable in the limit by probabilistic machines. *Lecture Notes in Computer Science,* vol. 28, pp. 77–87, 1975.

[16] R. Freivalds. Minimal Gödel numbers and their identification in the limit. *Proceedings of the 4th Symposium on Mathematical Foundations of Computer Science. Lecture Notes in Computer Science,* vol. 32, pp. 219–225, Springer-Verlag, 1975.

[17] R. Freivalds. Inductive inference of recursive functions: qualitative theory. *Lecture Notes in Computer Science,* vol. 502, pp. 77–110. Springer-Verlag, 1991.

[18] R. Freivalds, M. Karpinski, and C. H. Smith. Co-learning of total recursive functions. *Proceedings of the 7th Annual ACM Workshop on Computational Learning Theory,* pp. 190–197, ACM Press, New York, 1994.

[19] R. Freivalds, C. Smith and M. Velauthapillai. Trade-offs amongst parameters effecting the inductive inferribility of classes of recursive functions. *Information and Computation,* vol. 82 (3), pp. 323–349, 1989.

[20] R. Freivalds and R. Wiehagen. Inductive inference with additional information. *Elektronische Informationsverabeitung und Kybernetik,* vol. 15 (4), pp. 179–184, 1979.

[21] E. M. Gold. Limiting recursion. *Journal of Symbolic Logic,* vol. 30, pp. 28–48, 1965.

[22] E. M. Gold. Language identification in the limit. *Information and Control,* vol. 10, pp. 447–474, 1967.

[23] S. Jain and A. Sharma. Finite learning by a team. In M. Fulk and J. Case, editors, *Proceedings of the Third Annual Workshop on Computational Learning Theory,* pp. 163–177, Morgan Kaufmann Publishers, 1990.

[24] S. Jain and A. Sharma. On identification by teams and probabilistic machines. In K. P. Jantke and S. Lange, editors, *Algorithmic Learning for Knowledge-Based Systems. Lecture Notes in Artificial Intelligence,* vol. 961, pp. 108–145, Springer-Verlag, 1995.

[25] D. N. Osherson, M. Stob, S. Weinstein. *Systems That Learn.* The MIT Press, 1986.

[26] D. Osherson, M. Stob, and S. Weinstein. Aggregating inductive expertise. *Information and Control,* vol. 70, pp. 69–95, 1986.

[27] D. Osherson and S. Weinstein. Criteria of language learning. *Information and Control,* vol. 52, pp. 123–138, 1982.

[28] L. Pitt. Probabilistic inductive inference. *Journal of the ACM,* vol. 36 (2), pp. 383–433, 1989.

[29] L. Pitt and C. Smith. Probability and plurality for aggregations of learning machines. *Information and Computation,* vol. 77, pp. 77–92, 1988.

[30] H. Rogers, Jr. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, New York, 1967.

[31] C. Smith. The power of pluralism for automatic program synthesis. *Journal of the ACM,* vol. 29, pp. 1144–1165, 1982.

[32] C. Smith. Three decades of team learning. In S. Arikawa and K. P. Jantke, editors, *Algorithmic Learning Theory. Lecture Notes in Artificial Intelligence,* vol. 872, pp. 211–228, Springer-Verlag, 1994.

[33] J. Smotrovs. *Identification of Class Unions. Master thesis,* University of Latvia, Rīga, 1996.

[34] J. Smotrovs. Closedness properties in team learning of recursive functions. In S. Ben-David, editor, *Proceedings of the Third European Conference on Computational Learning Theory. Lecture Notes in Artificial Intelligence,* vol. 1208, pp. 79–93. Springer-Verlag, 1997.

[35] R. Smullyan. *Theory of Formal Systems. Annals of Mathematical Studies,* vol. 47, Princeton, 1961.

[36] M. Velauthapillai. Inductive inference with a bounded number of mind changes. In R. Rivest, D. Haussler, and M. Warmuth, editors, *Proceedings of the 1989 Workshop on Computational Learning Theory,* pp. 200–213, Morgan Kaufmann, 1989.

[37] R. Wiehagen. Limes-Erkennung rekursiver Funktionen durch spezielle Strategien. *Journal of Information Processing and Cybernetics,* vol. 12, pp. 93–99, 1976.