On Learning and Co-learning of Minimal Programs

Sanjay Jain
Department of Information Systems and Computer Science
National University of Singapore
Singapore
sanjay@iscs.nus.sg

Efim Kinber
Department of Computer Science
Sacred Heart University
5151 Park Avenue
Fairfield, CT 06432-1000
kinber@shu.sacredheart.edu

Rolf Wiehagen
Fachbereich Informatik
Universität Kaiserslautern
P. O. Box 3049
D-67653 Kaiserslautern, Germany
wiehagen@informatik.uni-kl.de

Abstract

Freivalds, Karpinski and Smith [8] explored a special type of learning in the limit: identification of an unknown concept (function) by eliminating (erasing) all but one possible hypothesis (this type of learning is called co-learning). The motivation behind learning by erasing lies in the process of human and automated computer learning: often we can discard incorrect solutions much easier than to come up with the correct one. In Gödel numberings any learnable family can be learned by an erasing strategy. In this paper we concentrate on co-learning minimal programs. We show that co-learning of minimal programs, as originally defined is significantly weaker than learning minimal programs in Gödel numberings. In order to enhance the learning power
of erasing strategies, we generalize the concept of co-learning in two directions. Firstly, we consider learning by erasing all programs, but a few correct ones. Secondly, we consider learning by erasing some incorrect programs only, including all programs that are shorter than the minimal one. We show that each of these types of co-learning minimal programs is considerably more powerful than erasing all but the minimal program. We also exhibit various relationships and differences between these types of co-learning minimal programs. In particular, we explore these types in Kolmogorov numberings that can be viewed as “natural” Gödel numberings of partial recursive functions.

1 Introduction

In [8] Freivalds, Karpinski and Smith explored a special type of learning in the limit: identification of an unknown concept (function) by eliminating (erasing) all but one possible hypothesis (CoAll in our notation). The motivation behind learning by erasing lies in the process of human and automated computer learning: often we can discard incorrect solutions much easier than to come up with the correct one. However, if all incorrect solutions have been successfully discarded, one can easily solve the problem just picking any conjecture that has been left. The version of learning recursive functions by erasing (referred thereafter as co-learning) suggested in [8] (erasing all but one conjecture) turned out to be significantly more restrictive than learning in the limit for special computable non-Gödel numberings of partial recursive functions. However, in Gödel numberings any learnable family can be learned by an erasing strategy (co-learned). One can consider co-learning in a broader sense, when some or all correct programs of the unknown function are not being erased [13], however, in the light of abovementioned observation, it does not increase the power of co-learning.

In this paper we concentrate on co-learning minimal programs and learning minimal programs by strategies that result from co-learning. We show that co-learning of minimal programs, as originally defined in [8], is significantly weaker than learning minimal programs even in Gödel numberings. In order to enhance the learning power of erasing strategies, we generalize the concept of co-learning in two directions. Firstly, we allow the learning strategy to possibly not cancel out more than one of the programs for the input function, but we still require the strategy to cancel all incorrect programs of that function (CoSuper in our notation). Secondly, we observe learning by erasing some incorrect programs only, including all programs that are
shorter than the minimal correct one (CoSub in our notation). We show that each of those types of co-learning minimal programs is considerably more powerful than erasing all but the minimal program. Then we exhibit various relationships and differences between these two types of co-learning minimal programs and give some examples of classes co-learnable within each of those paradigms. In particular, we explore co-learning of both types in Kolmogorov numberings that can be viewed as “natural” Gödel numberings of partial recursive functions.

Co-learning of minimal programs naturally suggests a special strategy of learning minimal programs: each new hypothesis is “longer” than the prior one. In contrast, one can also consider learning minimal programs by strategies of the opposite type: each new hypothesis is “shorter” than the prior one. We show that these both types of learning minimal programs are weaker than learning minimal programs in the general case. We also give the complete picture of relationships between the abovementioned types of learning minimal programs.

The power of co-learning can be enhanced if the learner can make several attempts to apply the learning strategy; the same concerns learning by special strategies. For example, the strategy that always chooses greater conjectures, may do this for a while, and then start it all over again; it can perform such “trials” a finite number of times. If the number of the trials is not limited, then any learning strategy can be arranged in this fashion. However, if the number of trials is uniformly bounded, we can show that increasing the number of trials enhances the power of learning minimal numbers by strategies that increase their conjectures, as well as by those that decrease the conjectures.

Co-learning criteria considered in this paper have also been considered by [13] in the context of indexed families of recursive languages.

2 Preliminaries

2.1 Notation

\(\mathbb{N}\) denotes the set of natural numbers. \(\emptyset, \in, \subseteq, \supseteq, \leq, \geq\) respectively, denote empty set, element of, proper subset, subset, proper superset, superset. \(\max(\cdot), \min(\cdot)\), denote maximum and minimum of a set, where by convention \(\max(\emptyset) = 0\) and \(\min(\emptyset) = \infty\). Cardinality of a set \(S\) is denoted by \(\text{card}(S)\). \(\overline{S}\) denotes the complement of the set \(S\). \(\langle \cdot, \cdot \rangle\) stands for an arbitrary 1–1, computable, encoding of all pairs of natural numbers onto \(\mathbb{N}\).
The quantifiers $\forall$ and $\exists$ denote, for all but finitely many and there exist infinitely many, respectively.

$\mathcal{R}$ denotes the set of total recursive functions of one argument. For $n \geq 2$, $\mathcal{R}^n$ denotes the set of total recursive functions of $n$ arguments. $\mathcal{P}$ denotes the set of partial recursive functions of one argument. For $n \geq 2$, $\mathcal{P}^n$ denotes the set of partial recursive functions of $n$ arguments. A numbering $\psi$ is a (possibly partial) computable function from $\mathbb{N}^2$ to $\mathbb{N}$, i.e., $\psi \in \mathcal{P}^2$. For a numbering $\psi$, $\psi_i$ denotes the function $\lambda x. \psi(i, x)$. $\mathcal{R}_\psi = \{ \psi_i \mid i \in \mathbb{N} \land \psi_i \in \mathcal{R} \}$. $\text{Progs}_\psi(f)$ denotes the set of $\psi$-programs for $f$, i.e., $\text{Progs}_\psi(f) = \{ i \mid \psi_i = f \}$. $\text{MinProgs}_\psi(f) = \min(\text{Progs}_\psi(f))$. A class $\mathcal{C}$ of recursive functions is said to be r.e. if $\mathcal{C} = \mathcal{R}_\psi$ for some $\psi \in \mathcal{R}^2$, i.e., if $\mathcal{C}$ can be recursively enumerated by some total recursive numbering.

A numbering $\psi$ is called acceptable [15] iff for every numbering $\eta$, there exists a recursive function $h$ such that $(\forall i)[\eta_i = \psi_{h(i)}]$. Acceptable numberings are also called Gödel numberings. A numbering $\psi$ is a Kolmogorov numbering iff for every numbering $\eta$, there exists a recursive function $h$ and a constant $c$ such that $(\forall i)[\eta_i = \psi_{h(i)} \land h(i) \leq \max(c * i, c)]$. A numbering $\psi$ is called 1–1 iff, for all $i \neq j$, $\psi_i \neq \psi_j$.

$\varphi$ denotes a standard acceptable (Gödel) numbering, $\Phi$ denotes a Blum complexity measure [2] for $\varphi$. We use $\downarrow$ and $\uparrow$ to denote that a computation converges or diverges. Thus, $\varphi_i(x) = \uparrow$ denotes that $\varphi_i(x)$ is undefined.

### 2.2 Criteria of Inference

We direct the reader to [10, 3, 1, 14, 12] for a general background in the area of learning theory. Here we briefly consider the notions needed for the paper.

For any recursive function $f$ and any $n \in \mathbb{N}$, let $f[n]$ denote the initial segment $\langle f(0), f(1), \ldots, f(n-1) \rangle$ (or, more formally, the code of the tuple $\langle f(0), f(1), \ldots, f(n-1) \rangle$ in some fixed one-one computable numbering of all tuples of numbers in $\mathbb{N}$). Let $\text{SEG} = \{ f[n] \mid f \in \mathcal{R} \land n \in \mathbb{N} \}$. $\Lambda$ denotes the empty segment, $f[0]$. A learning machine is an algorithmic mapping from $\text{SEG}$ to $\mathbb{N} \cup \{ ? \}$. Intuitively, we will interpret the output of a learning machine as a program. $\Lambda$ then represents the case where the machine outputs "no conjecture." We let $\mathcal{M}$, with or without, decorations range over learning machines.

In the following definitions let $\psi$ denote any numbering and $f$ a recursive function.
Definition 1 [10] M $\text{Ex}_\psi$-identifies $f$ (written $f \in \text{Ex}_\psi(M)$) iff $(\exists i \mid \psi_i = f)(\forall n)[M(f[n]) = i]$.  
$\text{Ex}_\psi = \{C \mid (\exists M)[C \subseteq \text{Ex}_\psi(M)]\}$. 

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

Definition 2 [4] M $\text{MinEx}_\psi$-identifies $f$ (written $f \in \text{MinEx}_\psi(M)$) iff $f \in R_\psi$ and $(\forall n)[M(f[n]) = \text{MinProg}_\psi(f)]$.  
$\text{MinEx}_\psi = \{C \mid (\exists M)[C \subseteq \text{MinEx}_\psi(M)]\}$. 

Unlike $\text{Ex}$ inference, the class $\text{MinEx}_\psi$ depends on the Gödel numbering $\psi$ [4].

Definition 3 [10] M $\text{Fin}_\psi$-identifies $f$ (written $f \in \text{Fin}_\psi(M)$) iff $(\exists i \mid \psi_i = f)(\forall m < n)[M(f[m]) = ?] \land (\forall m \geq n)[M(f[m]) = i]$.  
$\text{Fin}_\psi = \{C \mid (\exists M)[C \subseteq \text{Fin}_\psi(M)]\}$. 

The class $\text{Fin}_\psi$ is same for all Gödel numberings $\psi$. Thus we often refer to $\text{Fin}_\varphi$ as just $\text{Fin}$.

Definition 4 [4] M $\text{MinFin}_\psi$-identifies $f$ (written $f \in \text{MinFin}_\psi(M)$) iff $f \in R_\psi$ and $(\exists n)[(\forall m < n)[M(f[m]) = ?] \land (\forall m \geq n)[M(f[m]) = \text{MinProg}_\psi(f)]]$.  
$\text{MinFin}_\psi = \{C \mid (\exists M)[C \subseteq \text{MinFin}_\psi(M)]\}$. 

The class $\text{MinFin}_\psi$ depends on the Gödel numbering $\psi$ [4].

3 Definitions of Co-learning

We will consider three versions of co-learnability in this paper. Similar to the definitions of $\text{Ex}$ and $\text{Fin}$, we will first define the general version of each type of co-learnability and then the minimal version. Again in the definitions below let $\psi$ denote any numbering and $f$ any recursive function.

The first version of co-learning was defined in [8]. In this type of co-learning, $M$ needs to cancel out all the programs, except exactly one program, which must be a program for the function being learned.
Definition 5 Let ProgSet(M, f) = \{M(f[n]) \mid n \in N \land M(f[n]) \neq \phi\}.
Let ProgSet(M, f[m]) = \{M(f[n]) \mid n \leq m \land M(f[n]) \neq \phi\}.

We say that M co-converges on f [9] to i iff i = \min(N - \text{ProgSet}(M, f)).
We say that M co-converges on f iff there exists an i such that M co-
converges on f to i.

Definition 6 [8] M CoAll\(_\psi\)-identifies f (written f \in CoAll\(_\psi\)(M)) iff (\exists i \mid \psi_i = f)[\text{ProgSet}(M, f) = N - \{i\}]

CoAll\(_\psi\) = \{C \mid (\exists M)[C \subseteq \text{CoAll}\(_\psi\)(M)]\}.

Definition 7 M MinCoAll\(_\psi\)-identifies f (written f \in MinCoAll\(_\psi\)(M))
iff f \in R\(_\psi\) and ProgSet(M, f) = N - \{MinProg\(_\psi\)(f)\}.

MinCoAll\(_\psi\) = \{C \mid (\exists M)[C \subseteq \text{MinCoAll}\(_\psi\)(M)]\}.

We now consider our second type of co-learning. As a generalization to
CoAll\(_\psi\), in this version we allow the machine to possibly not cancel out
more than one of the programs for the input function, but we still require
the machine to cancel all incorrect programs of that function. Hence the
machine must cancel a superset of the set of all incorrect programs.

Definition 8 M CoSuper\(_\psi\)-identifies f (written f \in CoSuper\(_\psi\)(M)) iff
f \in R\(_\psi\), Prog\(_\psi\)(f) \subseteq \text{ProgSet}(M, f) and M co-converges on f to a program
in Prog\(_\psi\)(f).

CoSuper\(_\psi\) = \{C \mid (\exists M)[C \subseteq \text{CoSuper}\(_\psi\)(M)]\}.

Definition 9 M MinCoSuper\(_\psi\)-identifies f (written f \in MinCoSuper\(_\psi\)(M))
iff f \in R\(_\psi\), Prog\(_\psi\)(f) \subseteq \text{ProgSet}(M, f) and M co-converges on f to MinProg\(_\psi\)(f).

MinCoSuper\(_\psi\) = \{C \mid (\exists M)[C \subseteq \text{MinCoSuper}\(_\psi\)(M)]\}.

We now consider our third type of co-learnability. In this version, we
require that all the programs cancelled by the machine should actually be
incorrect ones. Hence the machine may cancel only a subset of the set of all
incorrect programs.

Definition 10 M CoSub\(_\psi\)-identifies f (written f \in CoSub\(_\psi\)(M)) iff f \in
R\(_\psi\), ProgSet(M, f) \subseteq Prog\(_\psi\)(f) and M co-converges on f to a program in
Prog\(_\psi\)(f).

CoSub\(_\psi\) = \{C \mid (\exists M)[C \subseteq \text{CoSub}\(_\psi\)(M)]\}.
Note that the types of co-learning functions from Definitions 8 and 10 above are also considered for co-learning indexed families of recursive languages in [13].

**Definition 11** M MinCoSub$_\psi$-identifies $f$ (written $f \in$ MinCoSub$_\psi$(M)) iff $f \in R_\psi$, ProgSet(M, $f$) $\subseteq$ Progs$_\psi$(f) and M co-converges on $f$ to MinProg$_\psi$(f).

$$\text{MinCoSub}_\psi = \{C \mid (\exists M)[C \subseteq \text{MinCoSub}_\psi(M)]\}.$$ 

Co-learning of the type MinCoSub$_\psi$ suggests a natural strategy of 
learning minimal programs: cross out, in increasing order, all the indices smaller than the minimal one for the given function. We will observe below in Proposition 2 that this type of learning minimal programs and MinCoSub$_\psi$ are of the same power.

**Definition 12** M MinIncEx$_\psi$-identifies $f$ (written $f \in$ MinIncEx$_\psi$(M)) iff $(\forall m, n \mid m \leq n \land M(f[m]) \neq \bot \lor M(f[n]) \neq \bot)[M(f[m]) \leq M(f[n])]$ and M MinInc$_\psi$-identifies $f$.

$$\text{MinIncEx}_\psi = \{C \mid (\exists M)[C \subseteq \text{MinIncEx}_\psi(M)]\}.$$ 

As a dual to MinIncEx$_\psi$ criterion we consider the case where the machine is required to output its conjectures in a decreasing order.

**Definition 13** M MinDecEx$_\psi$-identifies $f$ (written $f \in$ MinDecEx$_\psi$(M)) iff $(\forall m, n \mid m \leq n \land M(f[m]) \neq \bot \lor M(f[n]) \neq \bot)[M(f[m]) \geq M(f[n])]$ and M MinDec$_\psi$-identifies $f$.

$$\text{MinDecEx}_\psi = \{C \mid (\exists M)[C \subseteq \text{MinDecEx}_\psi(M)]\}.$$ 

It can easily be shown that there exists an r.e. sequence of machines $M_0, M_1, \ldots$, such that, for any criterion $I_\psi$ of inference discussed in this paper, $C \in I_\psi \Rightarrow (\exists i)[C \subseteq I_\psi(M_i)]$. Intuitively, this enumeration of machines allows us to restrict our attention to just these machines in the diagonalizations.

### 4 Results

Clearly, the minimal version of each of the criteria defined above can only be restrictive, thus:
Proposition 1 \(\text{MinFin}_\psi \subseteq \text{Fin}_\psi\).
MinEx\(_\psi\) \(\subseteq\) Ex\(_\psi\).
MinCoSub\(_\psi\) \(\subseteq\) CoSub\(_\psi\).
MinCoSuper\(_\psi\) \(\subseteq\) CoSuper\(_\psi\).

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

Proposition 2 \(\text{MinIncEx}_\psi = \text{MinCoSub}_\psi = \text{CoSub}_\psi\).
MinIncEx\(_\psi\) \(\cap\) MinDecEx\(_\psi\) = MinFin\(_\psi\).

Proposition 3 \((\forall\psi)[\text{MinFin}_\psi \subseteq \text{MinCoAll}_\psi \subseteq \text{MinCoSuper}_\psi \subseteq \text{MinCoSub}_\psi \subseteq \text{MinEx}_\psi]\).

\((\forall\psi)[\text{MinFin}_\psi \subseteq \text{MinDecEx}_\psi \subseteq \text{MinEx}_\psi]\).

**Proof.** We show MinCoSuper\(_\psi\) \(\subseteq\) MinCoSub\(_\psi\). All other inclusions follow from the definitions.

Suppose \(\mathcal{M}\) is given. Let \(\mathcal{M}'\) be such that, for all \(f\), ProgSet\((\mathcal{M}', f) = \{i \mid (\forall j \leq i)[j \in \text{ProgSet}(\mathcal{M}, f)]\}. It is easy to construct \(\mathcal{M}'\) as above. It is now easy to see that MinCoSuper\(_\psi\)(\(\mathcal{M}\)) \(\subseteq\) MinCoSub\(_\psi\)(\(\mathcal{M}'\)).

Proposition 4 \((\forall \text{Gödel Numbering }\psi)[\text{CoAll}_\psi = \text{CoSuper}_\psi = \text{Ex}_\psi]\).

**Proof.** Clearly, for all numberings \(\psi\), CoAll\(_\psi\) \(\subseteq\) CoSuper\(_\psi\) \(\subseteq\) Ex\(_\psi\). For Gödel numberings \(\psi\), Ex\(_\psi\) \(\subseteq\) CoAll\(_\psi\) ([8]). Proposition follows.

Since we are mainly interested in Gödel numberings and Kolmogorov numberings, due to the above proposition, we will mostly be interested only in minimal identification version of co-learning criteria.

### 4.1 Kolmogorov Numberings

In this section we show that the different minimal identification criteria considered in this paper are separated in every Kolmogorov numbering (except for the case MinCoSuper\(_\psi\) vs MinCoAll\(_\psi\); we do not know at present whether MinCoSuper\(_\psi\) and MinCoAll\(_\psi\) are separated in every Kolmogorov numbering \(\psi\)).

Let \(\mathcal{H} = \{h_j \mid j \in \mathbb{N}\}\), where \(h_j\) is defined as follows.

\[
h_j(x) = \begin{cases} 1, & \text{if } x = j; \\ 0, & \text{otherwise}. \end{cases}
\]
Theorem 1 [5, 6] (∀ Kolmogorov Numbering ψ)(∃C ⊆ H | card(C) = ∞)[C ∈ MinFinψ].

Note that MinFinψ ⊆ MinIncExψ. It is easy to verify the following proposition.

Proposition 5 (∀ Gōdel Numbering ψ)(∀C ∈ MinIncExψ)(∀f ∈ R)[C ∪ {f} ∈ MinIncExψ].

Note that the above proposition does not hold for MinDecEx replacing MinIncEx. Let ZERO be the everywhere 0 function. Note that any machine which MinDecExψ-identifies ZERO can MinDecExψ-identify only finitely many functions in H. Thus:

Proposition 6 (∀ψ)(∀C ⊆ H | card(C) = ∞)[C ∪ {ZERO} ∉ MinDecExψ].

As a corollary to Theorem 1 and Propositions 5 and 6 we immediately have:

Theorem 2
(∀ Kolmogorov Numbering ψ)[MinIncExψ − MinDecExψ ≠ ∅].

Corollary 1 (∀ Kolmogorov Numbering ψ)[MinDecExψ ⊆ MinExψ].

Freivalds’ proof of Theorem 1 essentially also shows:

Theorem 3 (∀ Kolmogorov Numbering ψ)(∃ε > 0)(∃C ∈ MinFinψ)
(∃ r)[card({i ≤ r | hi ∈ C}) ≥ εr].

Theorem 4 (∀ Kolmogorov Numbering ψ)(∀C ⊆ H | card(C) = ∞)[C ∪ {ZERO} ∈ MinCoAllψ].

Proof. Let M, ε > 0, C′ ⊆ H, be such that, C′ ⊆ MinCoAllψ(M) and

(∃ r)[card({i ≤ r | hi ∈ C′}) ≥ εr]

Note that there exists such M, ε, C′ by Theorem 3 and Proposition 3.

Let M′ be defined as follows. Suppose z = MinProgψ(ZERO).

ProgSet(M′, ZERO[n]) = {x | x ≤ n/2 ∧ x ≠ z}

ProgSet(M′, hj) = ProgSet(M, hj) ∪ ProgSet(M′, hj[j])
Note that such an $M'$ can be easily constructed.

Note that $M'$ MinCoAll$_{\psi}$-identifies ZERO. It may however not MinCoAll$_{\psi}$-identify all the functions MinCoAll$_{\psi}$-identified by $M$ (due to the extra programs output by $M'$ on $h_j$).

Let $C = \{ h_j | h_j \in C' \land \text{MinProg}_{\psi}(h_j) > \epsilon * j/2 \}$.

It is easy to verify that $M'$ MinCoAll$_{\psi}$-identifies each function in $C$. Moreover, $\text{card}(C) = \infty$, since it contains at least $(\epsilon * j/2) * r$ functions in $\{h_1, \ldots, h_r\}$ for infinitely many $r$. \hfill \rule{2mm}{2mm}

As a corollary, we have:

**Theorem 5**

$(\forall \text{Kolmogorov Numbering } \psi) [\text{MinCoAll}_{\psi} - \text{MinDecEx}_{\psi} \neq \emptyset]$.

**Theorem 6**

$(\forall \text{Kolmogorov Numbering } \psi) [\text{MinDecEx}_{\psi} - \text{MinIncEx}_{\psi} \neq \emptyset]$.

**Proof.** Suppose a Kolmogorov numbering $\psi$ is given. We will construct a numbering $\tau$.

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

- $l_0^i = 1$.
- $l_{i+1}^j = u_i^j + 1$, $l_{i+1}^{j+1} = u_i^j + 1$.
- $u_i^j = l_i^j * i * 3 + 1$.

Let $I_i^j = \{ p | l_i^j \leq p \leq u_i^j \}$.

Let $I_i = \{ p | (\exists j \leq i)(p \in I_i^j) \}$.

Each $\tau_k$ will either be a total recursive function or the everywhere undefined function.

We will now define the functions $\tau_k$, for $k \in I_i$ (such a construction is carried out for each $i$ separately). Intuitively, this will give us a collection of functions $S_i$, such that none of the machines $M_0, M_1, \ldots, M_{i-1}$, will MinIncEx$_{\psi}$-identify any of the functions in $S_i$.

**Definition of $\tau_k$, for $k \in I_i$.**

Let $\sigma_i^0(0) = i$ ($\sigma_i^0$ is of length 1). Go to stage 0.

Begin stage $s$.  

10
1. Let $j = i - s$.
2. Let $m = \lfloor \sigma_i^s \rfloor$ (i.e., $m$ is the least element not in the domain of $\sigma_i^s$).
3. For all $p \in I_i^j$, define $\tau_p$ as follows
   \[
   \tau_p(x) = \begin{cases} 
   \sigma_s(x), & \text{if } x < m; \\
   p, & \text{if } x = m; \\
   0, & \text{otherwise}.
   \end{cases}
   \]
4. Search for $p \in I_i^j$ and $y > m$, such that, \( \text{card}(\{r < i \mid (\exists q \geq i \ast l_i^j)[q \in \text{ProgSet}(M_r, \tau_p[y])]\}) > s. \)
   (* Intuitively if $(\exists q \geq i \ast l_i^j)[q \in \text{ProgSet}(M_r, \tau_p[y])]$ then $M_r$ has output a “large” program, and thus would become useless *)
5. If and when such $p, y$ are discovered, let $\sigma_i^{s+1} = \tau_p[y]$ and go to stage $s+1$.

End stage $s$.

End of Definition of $\tau_k$, for $k \in I_i$.

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

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

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

For all $r < i$, either $\text{ProgSet}(M_r, \tau_p)$ contains a program $> i \ast u_i^{j_i} \geq i \ast p > c \ast p$ (M$_r$ was diagonalized against in the previous stages), or $\text{ProgSet}(M_r, \tau_p)$ contains only programs less than $i \ast l_i^{j_i}$ (M$_r$ could not be diagonalized against in the last stage).

Thus $M_r$, $r < i$, can MinIncEx$\psi$-identify only those $\tau_p$ in $\{\tau_p \mid l_i^{j_i} \leq p \leq u_i^{j_i}\}$, whose minimal $\psi$ programs are $< i \ast l_i^{j_i}$. Let
\[
S_i = \{\tau_p \mid l_i^{j_i} \leq p \leq u_i^{j_i} \land \text{MinProg}_\psi(\tau_p) \geq i \ast l_i^{j_i}\}
\]
It immediately follows that no machine $M_r$ can \textbf{MinIncEx}_\psi-identify any function in $S_i$, for $i > \max(\{r, c\})$. We will construct our diagonalizing class as an infinite subset of $\bigcup_{i>c} S_i$, using a trick used by Freivalds [5, 6]. For $i > c$, let

$$S_i' = \{\tau_p \mid \tau_p \in S_i \land \text{card}(\{q \leq c * u_{i,j}^i \mid \tau_p[\sigma_{i,j}^i] + 1 \leq \psi_q\}) \leq 4c\}$$

Note that $\text{card}(S_i') \geq [u_{i,j}^i - i * l_{i,j}^i] - u_{i,j}^i/4 - i * l_{i,j}^i$ (the $u_{i,j}^i/4$ term is for functions spoiled due to them having small $< i * l_{i,j}^i$ minimal programs in $\psi$; the $u_{i,j}^i$ term is due to the functions having more than $4c$ programs $\leq c u_{i,j}^i$ in the numbering $\psi$). Thus $S_i'$ is non-empty (for $i > c$).

Let $C = \bigcup_{i>c} S_i'$. We now construct $M'_1, M'_2, \ldots, M'_c$, such that at least one of these machines \textbf{MinDecEx}_\psi-identifies an infinite subset of $C$.

The idea of the construction of $M'_i$ is as follows. Suppose the input function is $f \in S_i \cap C$ ($i$ can be determined from $f(0)$). Thus $f$ must be the same as $\tau_p$, for some $p \in I_i^j$. Note that, for $p \in I_i^j$, $j \geq j_i$, $\text{card}(\{x > 0 \mid \tau_p(x) \neq 0\}) = j - j_i + 1$. Moreover, if $\tau_p = f$, then $\text{MiniProg}_\psi(f)$ must lie in the interval $[i * l_{i,j}^i, c u_{i,j}^i]$ (note that $c u_{i,j}^i \leq i * u_{i,j}^i < i * l_{i,j}^i + 1$). This is what our construction uses. Note that $\sigma_{i,j}^i$, if defined, can be effectively determined. Let $n_{i,j}^i = |\sigma_{i,j}^i| + 1$ (note that $n_{i,j}^i - 1$ determines the point at which the functions in $\{\tau_p \mid p \in I_i^j\}$ differ).

Let $X_i^j = \{q \mid i * l_{i,j}^i \leq q \leq c * u_{i,j}^i \land f[q] \subseteq \psi_q\}$.

$M'_i$ is defined as follows. $M'_i(f[n]) = \neg$, for $n \leq 1$. For $n > 1$, let $j = i - \text{card}(\{x \mid 0 < x < n \land f(x) \neq 0\}) + 1$; $M'_i(f[n])$ is then the $r$-th element, if any, in a standard 1-1 enumeration of $X_i^j$.

It is easy to note that the conjectures of $M'_i$ are monotonically decreasing (since $c u_{i,j}^i \leq i * u_{i,j}^{i+1}$—recall that $i > c$, and the $j$'s as used in the definition of $M'_i$ are monotonically decreasing). Moreover, at least one of the machines $M'_r$, $1 \leq r \leq 4c$, \textbf{MinEx}_\psi-identifies $f$ (since $\text{card}(X_{i,j}^i) \leq 4c$, for $f \in C$, and $X_{i,j}^i$ contains a minimal program for $f$.) Thus for every function $f \in C$, at least one of the machines $M'_1, \ldots, M'_c$ \textbf{MinDecEx}_\psi-identifies $f$. Since $C$ is infinite, there exists a machine which \textbf{MinDecEx}_\psi-identifies an infinite subset of $C$. Since no infinite subset of $C$ is in \textbf{MinIncEx}_\psi, the theorem follows.

\textbf{Theorem 7}

$(\forall \text{ Kolmogorov Numbering } \psi)[\text{MinIncEx}_\psi - \text{MinCoSuper}_\psi \neq \emptyset]$.
PROOF. The idea of the proof is similar to that of the proof of Theorem 6 though there are subtle differences.

Suppose a Kolmogorov numbering $\psi$ is given. We will construct a numbering $\tau$.

For each $i \in N$, $j \leq i$, we will define, $l^i_j$, $u^i_j$.

\begin{align*}
l^0_j &= 1, \\
I^{0}_{i+1} &= u^i_1 + 1, \\
u^i_j &= (l^i_j)^2(3i + 1)^{5i + j}.
\end{align*}

Let $I^i_j = \{ p | l^i_j \leq p \leq u^i_j \}$.

Let $I_i = \{ p | (\exists j \leq i)[p \in I^i_j] \}$.

Each $\tau_k$ will either be a total recursive function or the everywhere undefined function.

We will now define the functions $\tau_k$, for $k \in I_i$ (such a construction is carried out for each $i$ separately). Intuitively, this will give us a collection of functions $\mathcal{S}_i$, such that none of the machines $M_0, M_1, \ldots, M_{i-1}$, will $\text{MinCoSuper}_\psi$-identify any of the functions in $\mathcal{S}_i$.

Definition of $\tau_k$, for $k \in I_i$.

Let $\sigma^0_i(0) = i$ ($\sigma^0_i$ is of length 1). Go to stage 0.

Begin stage $s$.

1. Let $j = s$.
2. Let $m = |\sigma^s_i|$ (i.e. $m$ is the least element not in the domain of $\sigma^s_i$).
3. For all $p \in I^s_i$, define $\tau_p$ as follows

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

4. Search for $p \in I^s_i$ and $y > m$, such that, $\text{card}(\{ r < i \mid \text{card}(\{ q \leq i \ast u^i_1 \mid q \notin \text{ProgSet}(M_r, \tau_p[y]) \}) \leq \sqrt{i \ast u^i_1} \}) > j$.

(* Intuitively $\text{card}(\{ q \leq i \ast u^i_1 \mid q \notin \text{ProgSet}(M_r, \tau_p[y]) \}) \leq \sqrt{i \ast u^i_1}$ means that $M_r$ can $\text{MinCoSuper}_\psi$-identify only “few” relevant functions *)

5. If and when such $p, y$ are discovered, let $\sigma^{s+1}_i = \tau_p[y]$ and go to stage $s + 1$.

End stage $s$.

End of Definition of $\tau_k$, for $k \in I_i$. 

13
First note that there cannot be infinitely many stages. In fact, if the
correction reaches stage $i$, then the search at step 4 cannot succeed. Let
$s_i$ denote the last stage that is executed, and let $j_i = s_i$.

It is easy to observe that each $\tau_i$ is either a total function or the every-
where undefined function (for $p \in I_i^i$, $\tau_p$ is total iff $j \leq j_i$.) Moreover, total
functions in the numbering $\tau$ are pairwise different.

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

For all $r < i$, either
(a) there exists an $S \subseteq \{x | x \leq i * u_i^i\}$, such that $\text{card}(S) < \sqrt{i * u_i^i}$ and, for all $p \in I_i^i$, $N - \text{ProgSet}(M_r, \tau_p) \subseteq S$ or
(b) for each $p \in I_i^i$, $\text{ProgSet}(M_r, \tau_p)$ does not contain at least
$\sqrt{i * u_i^i}$ programs $\leq i * u_i^i$.

In either case $M_r$, $r < i$, can $\text{MinCoSuper}_{\psi}$-identify at most $\sqrt{i * u_i^i}$
of the functions in $\{\tau_p | l_i^i \leq p \leq u_i^i\}$ (in case (a), there are only $\sqrt{i * u_i^i}$
possibilities for $M_r$ to co-converge to; in case (b) there can be at most $\sqrt{i * u_i^i}$
distinct programs for which $M_r$ cancels out all the incorrect programs $\leq i * u_i^i$).

Let

$S_i = \{\tau_p | l_i^i \leq p \leq u_i^i \land (\forall r < i)[\tau_p \notin \text{MinCoSuper}_{\psi}(M_r)]\}$.

It immediately follows that no machine $M_r$ can $\text{MinCoSuper}_{\psi}$-identify
any function in $S_i$, for $i > \max\{\{r, c\}\}$.

We will construct our diagonalizing class as an infinite subset of $\bigcup_{i > c} S_i$, using
a trick used by Freivalds [5, 6]. For $i > c$, let

$S'_i = \{\tau_p | \tau_p \in S_i \land \text{MinProg}_{\psi}(\tau_p) \geq i * l_i^i \land \text{card}(\{q \leq c * u_i^i | \tau_p[|\sigma_q^i| + 1] \subseteq \psi_q\}) \leq 4c\}$.

Note that, for large enough $i$, $\text{card}(S'_i) \geq (u_i^i - l_i^i) - u_i^i/4 - i * l_i^i - i *$
$\sqrt{i * u_i^i} > 0$, $i * l_i^i$ term is for functions spoiled due to them having small
($< i * l_i^i$) minimal program in $\psi$; the $u_i^i/4$ term is due to the functions
having more than $4c$ programs $\leq c \cdot u^j_1$ in the numbering $\psi$; $i \cdot \sqrt{i \cdot u^j_1}$ term is for the functions which are **MinCoSuper**$\psi$-identified by some $M_r$, $r < i$. Thus $S_i$ is non-empty (for large enough $i > c$).

Let $C = \bigcup_{i > c} S_i$. We now construct $M'_1, M'_2, \ldots, M'_{4c}$, such that at least one of these machines **MinIncEx**$\psi$-identifies an infinite subset of $C$.

The idea of the construction of $M'_i$ is as follows. Suppose the input function is $f \in S_i \cap C$ ($i$ can be determined from $f(0)$). Thus $f$ must be the same as $\tau_p$, for some $p \in I^j_i$. Note that, for $p \in I^j_i$, $j \leq j_i$, card($\{x > 0 \mid \tau_p(x) \neq 0\}) = j + 1$. Moreover, for $f \in C \cap S_i$, MinProg$\psi(f)$ must lie in the interval $[i \cdot l_i^j, c \cdot u^j_1]$ (note that $c \cdot u^j_1 \leq i \cdot l_i^j < i \cdot l_i^{j+1}$). This is what our construction uses.

Note that $\sigma_i^j$, if defined, can be effectively determined. Let $n_i^j = |\sigma_i^j| + 1$ (note that $n_i^j - 1$ determines the point at which the functions in $\{\tau_p \mid p \in I^j_i\}$ differ).

Let $X_i^j = \{q \mid i \cdot l_i^j \leq q \leq c \cdot u^j_1 \land f[n^j_i] \subseteq \psi_q\}$.

$M'_r$ is defined as follows. $M'_r(f[n]) = q$, for $n \leq 1$. For $n > 1$, let $j = \text{card}(\{x \mid 0 < x < n \land f(x) \neq 0\}) + 1$; $M'_r(f[n])$ is then the $r$-th element, if any, in a standard 1-1 enumeration of $X_i^j$.

It is easy to note that the conjectures of $M'_r$ are monotonically increasing (since $c \cdot u^j_1 \leq i \cdot u^j_1$)—recall that $i > c$, and the $j$'s as used in the definition of $M'_r$ are monotonically increasing). Moreover, at least one of the machines $M'_r$, $1 \leq r \leq 4c$, **MinEx**$\psi$-identifies $f$ (since $\text{card}(X_i^j) \leq 4c$, for $f \in C$, and $X_i^j$ contains a minimal program for $f$.) Thus for every function $f \in C$, at least one of the machines $M'_1, \ldots, M'_{4c}$ **MinIncEx**$\psi$-identifies $f$. Since $C$ is infinite, there exists a machine which **MinIncEx**$\psi$-identifies an infinite subset of $C$. Since no infinite subset of $C$ is in **MinCoSuper**$\psi$, the theorem follows.

**Theorem 8**

\(\exists\text{ Kolmogorov Numbering } \psi)[\text{MinCoSuper}_\psi – \text{MinCoAll}_\psi \neq \emptyset].\)

The above theorem follows from Theorem 24. We are not sure at this point whether the above theorem holds for all Kolmogorov numberings. However, we conjecture that it does.

Our main aim in this section was to separate all the minimal identification criteria in each Kolmogorov numbering. As an aside we note that a variant of the proof of Theorem 1 in [11] can be used to show that \((\exists\text{ an infinite class } C \subseteq H)(\forall \text{Kolmogorov Numbering } \psi)[C \in \text{MinFin}_\psi] \).
Thus Theorem 2 can be strengthened to:

**Theorem 9**

\(\exists \mathcal{C}) (\forall \text{ Kolmogorov Numbering } \psi)[\mathcal{C} \in \text{MinIncEx}_\psi - \text{MinDecEx}_\psi].\)

Thus the *same* class could be used for diagonalization for all Kolmogorov numberings. Such a result can also be obtained for Theorem 5. However, we do not yet know whether we could use the same class for other diagonalizations in this section.

We can generalize \text{MinIncEx}_\psi as follows. Let us say that the machine \(M\) \(\text{MinIncEx}_\psi^c\)-identifies \(f\) for a constant \(c \in N\) if it starts to work on \(f\) as \(\text{MinIncEx}_\psi\)-strategy at most \(c\) times (formal details of the definitions and proofs omitted). In the same way one can extend \text{MinDecEx}_\psi to \text{MinDecEx}_\psi^c. We can exhibit hierarchies of those classes, showing that, for each \(c\), \(\text{MinIncEx}_\psi^c \subset \text{MinIncEx}_\psi^{c+1}\) and \(\text{MinDecEx}_\psi^c \subset \text{MinDecEx}_\psi^{c+1}\). Accordingly, results of Theorems 2 and 6 can be extended to classes \text{MinIncEx}_\psi^c and \text{MinDecEx}_\psi^c.

### 4.2 Gödel Numberings

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

**Theorem 10** \(\exists \text{ Gödel Numbering } \psi)[\text{MinFin}_\psi = \text{MinCoAll}_\psi = \text{MinCoSuper}_\psi = \text{MinIncEx}_\psi = \text{MinDecEx}_\psi = \text{MinEx}_\psi].\)

**Proof.** Taking a Gödel numbering \(\psi\) such that \text{MinEx}_\psi contains only finite classes of functions ([4]) gives the theorem. \[\square\]

In fact, using the above theorem, for every “reasonable” relationship between the minimal criteria considered in this paper, we can construct a Gödel numbering in which this relationship does hold.

The essential idea is to interleave the needed diagonalizations in Theorems 12, 13, 14, 16, 17, with the numbering generated in Theorem 10 ([7] used a similar trick to generate Gödel numberings for any reasonable relationship between \text{MinFin}_\psi, \text{MinCoAll}_\psi, \text{MinEx}_\psi). We omit the details.

**Theorem 11** Suppose \(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6 \in \{\subset, =\}\), such that \(\alpha_5\) and \(\alpha_6\) are both \(\subset\) iff \(\alpha_1, \alpha_2, \alpha_3, \alpha_4\) are all \(\subset\). Then there exists a Gödel numbering \(\psi\) such that
(a) \( \text{MinFin}_\psi \alpha_1, \text{MinCoAll}_\psi \alpha_2, \text{MinCoSuper}_\psi \alpha_3, \text{MinIncEx}_\psi \alpha_4 \) \( \text{MinEx}_\psi \), and
(b) \( \text{MinFin}_\psi \alpha_5, \text{MinDecEx}_\psi \alpha_6, \text{MinEx}_\psi \).

4.3 Non-Gödel Numberings

In this section we prove the theorems on non-Gödel numberings needed to prove Theorem 11. In addition we prove that every \textbf{Ex}-identifiable class is \textbf{MinCoAll}_\psi-identifiable in some (non-Gödel) numbering \( \psi \).

\textbf{Theorem 12} (\( \exists \psi \)) \( [\text{card}(\mathcal{R}_\psi) = \infty \land \mathcal{R}_\psi \in \text{MinIncEx}_\psi \land (\forall S \subseteq \mathcal{R}_\psi) [\text{card}(S) = \infty \Rightarrow S \notin \text{MinDecEx}_\psi \cup \text{CoSuper}_\psi]] \).

\textbf{Proof.} We exploit the fact that for \textbf{CoSuper}_\psi one needs to cancel out all the incorrect programs.

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

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

We will make sure that (a), (b) and (c) are satisfied.

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

(b) For \( j < i \), \( f_i \notin \text{MinDecEx}_\psi(M_j) \cup \text{CoSuper}_\psi(M_j) \).

(c) \( C \in \text{MinIncEx}_\psi \).

This will prove the theorem.

Fix \( i \). We now define \( \psi_j \), for \( j \in S_i \).

\textbf{Definition of} \( \psi_j \), for \( j \in S_i \).

Let \( \text{CancelF}_i^0 = \text{CancelD}_i^0 = \emptyset \).

Go to stage 0.

\textbf{Begin stage} \( s \)

1. Let \( p(i, s) = h(i) + 1 + \text{card}(\text{CancelF}_i^s) + \text{card}(\text{CancelD}_i^s) \).

2. For all \( x \leq s \), let \( \psi_{p(i, s)}(x) = f_i(x) \).

   (Intuitively, we want to make \( \psi_{p(i, \infty)} = f_i \).

3. Let \( \text{CancelD}_i^{s+1} = \text{CancelD}_i^s \cup \{ j < i \mid p(i, s) \in \text{ProgSet}(M_j, f_i[s]) \} \).

4. Let \( \text{CancelF}_i^{s+1} = \text{CancelF}_i^s \cup \{ j < i \mid \{ l \leq h(i + 1) \mid l \neq p(i, s) \} \subseteq \text{ProgSet}(M_j, f_i[s]) \} \).

5. Go to stage \( s + 1 \).
End stage $s$.

End of Definition of $\psi_j$, for $j \in S_i$.

It is easy to verify that $\text{CancelD}^\psi_i$, $\text{CancelF}^\psi_i$ are monotonically non-decreasing (with respect to $\leq$) in $s$. Let $\text{CancelD}^\psi_i = \lim_{s \to \infty} \text{CancelD}^\psi_i$ and $\text{CancelF}^\psi_i = \lim_{s \to \infty} \text{CancelF}^\psi_i$. Note that $p(i, s)$ is monotonically non-decreasing in $s$. Let $p(i, \infty) = \lim_{s \to \infty} p(i, s)$. Note that $h(i) < p(i, s) \leq h(i) + 1 + 2i \leq h(i + 1)$. Also, $\psi_{p(i, \infty)} = f_i$, and for all $j \in S_i - \{p(i, \infty)\}$, $\psi_j$ is a non-total function. Thus (a) is satisfied.

Consider any $j < i$. If $j \in \text{CancelD}^\psi_i$, then $\text{ProgSet}(M_j, f_i)$ contains a program $< p(i, s) \leq p(i, \infty)$. If $j \notin \text{CancelD}^\psi_i$, then $\text{ProgSet}(M_j, f_i)$ does not contain $p(i, \infty)$. In either case $\psi_{p(i, \infty)} = f_i \notin \text{MinDecEx}_\psi(M_j)$.

Similarly, if $j \in \text{CancelF}^\psi_i$, then $\text{ProgSet}(M_j, f_i)$ contains $p(i, \infty)$. If $j \notin \text{CancelF}^\psi_i$, then $\{i \leq h(i + 1) \mid i \neq \infty\} \notin \text{ProgSet}(M_j, f_i)$. In either case $\psi_{p(i, \infty)} = f_i \notin \text{CoSuper}_\psi(M_j)$. Thus (b) is satisfied.

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

\textbf{Theorem 13} \hspace{1em} (\exists \psi)[\text{card}(\mathcal{R}_\psi) = \infty \land \mathcal{R}_\psi \in \text{MinCoSuper}_\psi \land (\forall S \subseteq \mathcal{R}_\psi)[\text{card}(S) = \infty \Rightarrow S \notin \text{MinCoAll}_\psi \cup \text{MinDecEx}_\psi]].

\textbf{Proof.} The proof of this theorem is similar to the proof of Theorem 12.

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

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

We will make sure that (a), (b) and (c) are satisfied.

(a) For each $i$, there exists a $j \in S_i = \{p \mid h(i) < p \leq h(i + 1)\}$, such that (a.1) $(\forall j' \mid j < j' \leq h(i + 1)) [\psi_{j'} = f_i]$ and (a.2) $(\forall j' \mid h(i) < j' < j) [\psi_{j'} \notin \mathcal{R}]$. This will make $\mathcal{R}_\psi = \mathcal{C}$.

(b) For $j < i$, $f_i \notin \text{MinDecEx}_\psi(M_j) \cup \text{MinCoAll}_\psi(M_j)$.

(c) $\mathcal{C} \in \text{MinCoSuper}_\psi$.

This will prove the theorem.

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

\textbf{Definition of $\psi_j$, for $j \in S_i$.}

Let $\text{CancelC}^0_i = \text{CancelD}^0_i = \emptyset$.

Go to stage 0.

Begin stage $s$.

18
1. Let \( p(i, s) = h(i) + 1 + \text{card}(\text{CancelC}_i^s) + \text{card}(\text{CancelD}_i^s) \).
2. For all \( j' \) such that \( p(i, s) \leq j' \leq h(i+1) \), for all \( x \leq s \), let \( \psi_{j'}(x) = f_i(x) \).
   (Intuitively, we want to make \( p(i, \infty) \) as \( j \) in clause (a)).
3. Let \( \text{CancelD}_{i+1} = \text{CancelD}_i^s \cup \{ j < i \mid p(i, s) \in \text{ProgSet}(M_j, f_i[s]) \} \).
4. Let \( \text{CancelC}_{i+1} = \text{CancelC}_i^s \cup \{ j < i \mid \{ l \leq h(i+1) \mid l \neq p(i, s) \} \subseteq \text{ProgSet}(M_j, f_i) \} \).
5. Go to stage \( s + 1 \).
End stage \( s \).
End of Definition of \( \psi_j \), for \( j \in S_i \).

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

Consider any \( j < i \). If \( j \in \text{CancelD}_i^s \) then \( \text{ProgSet}(M_j, f_i) \) contains a program \( < p(i, s) \leq p(i, \infty) \). If \( j \notin \text{CancelD}_i^\infty \), then \( \text{ProgSet}(M_j, f_i) \) does not contain \( p(i, \infty) \). In either case \( \psi_{p(i, \infty)} = f_i \notin \text{MinDecEx}_\psi(M_j) \). Similarly, if \( j \in \text{CancelC}_i^s \) then \( \text{ProgSet}(M_j, f_i) \) contains \( p(i, \infty) \). If \( j \notin \text{CancelC}_i^\infty \), then \( \{ l \leq h(i+1) \mid l \neq p(i, \infty) \} \notin \text{ProgSet}(M_j, f_i) \). In either case \( \psi_{p(i, \infty)} = f_i \notin \text{MinCoAll}_\psi(M_j) \). Thus (b) is satisfied.

To show that \( \mathcal{C} \in \text{MinCoSuper}_\psi \), note that \( p(i, s) \) is a monotonically non-decreasing function of \( s \), which converges to the minimal \( \psi \)-program for \( f_i \). Moreover, for all \( j' \), such that \( p(i, \infty) \leq j' \leq h(i+1) \), \( \psi_{j'} = f_i \). Thus \( \mathcal{C} \in \text{MinCoSuper}_\psi \). Thus (c) is satisfied.

**Theorem 14** \((\exists \psi)[\text{card}(\mathcal{R}_\psi) = \infty \land \mathcal{R}_\psi \in \text{MinDecEx}_\psi \land (\forall S \subseteq \mathcal{R}_\psi)[\text{card}(S) = \infty \Rightarrow S \notin \text{MinIncEx}_\psi]\].

**Proof.** The proof of this theorem is similar to the proof of Theorem 12, with some minor changes.

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

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

We will make sure that (a), (b) and (c) are satisfied.

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

(b) For \( j < i \), \( f_i \not\in \text{MinIncEx}_\psi(M_j) \).

c) \( C \in \text{MinDecEx}_\psi \).

This will prove the theorem.

Fix \( i \). We now define \( \psi_j \), for \( j \in S_i \).

Definition of \( \psi_j \), for \( j \in S_i \).

Let \( \text{Cancell}_i^0 = \emptyset \).

Go to stage 0.

Begin stage \( s \)

1. Let \( p(i, s) = h(i + 1) - \text{card}(\text{Cancell}_i^s) \).
2. For all \( x \leq s \), let \( \psi_{p(i, s)}(x) = f_i(x) \).
   (Intuitively, we want to make \( \psi_{p(i, \infty)} = f_i \)).
3. Let \( \text{Cancell}_{i+1}^s = \text{Cancell}_i^s \cup \{ j < i \mid p(i, s) \in \text{ProgSet}(M_j, f_i[s]) \} \).
4. Go to stage \( s + 1 \).

End stage \( s \).

End of Definition of \( \psi_j \), for \( j \in S_i \).

It is easy to verify that \( \text{Cancell}_i^s \) is monotonically nondecreasing (with respect to \( \subseteq \)) in \( s \). Let \( \text{Cancell}_i^\infty = \lim_{s \to \infty} \text{Cancell}_i^s \). Note that \( p(i, s) \) is monotonically nonincreasing in \( s \). Let \( p(i, \infty) = \lim_{s \to \infty} p(i, s) \). Note that \( h(i + 1) \geq p(i, s) \geq h(i + 1) - 2i > h(i) \). Also, \( \psi_{p(i, \infty)} = f_i \), and for all \( j \in S_i - \{ p(i, \infty) \} \), \( \psi_j \) is a non-total function. Thus (a) is satisfied.

Consider any \( j < i \). If \( j \in \text{Cancell}_i^s \) then \( \text{ProgSet}(M_j, f_i) \) contains a program \( > p(i, s) \geq p(i, \infty) \). If \( j \not\in \text{Cancell}_i^\infty \), then \( \text{ProgSet}(M_j, f_i) \) does not contain \( p(i, \infty) \). In either case \( \psi_{p(i, \infty)} = f_i \not\in \text{MinIncEx}_\psi(M_j) \). Thus (b) is satisfied.

To show that \( C \in \text{MinDecEx}_\psi \), note that \( p(i, s) \) is a monotonically nonincreasing function of \( s \), which converges to the minimal \( \psi \)-program for \( f_i \).

Thus \( C \in \text{MinDecEx}_\psi \). Thus (c) is satisfied.

\[ \textbf{Theorem 15} \ (\forall \psi \in \mathcal{R}^2)[\mathcal{R}_\psi \in \text{MinCoSuper}_\psi]. \]

\[ \textbf{Proof}. \] Let \( M \) be such that, for all \( f \), \( \text{ProgSet}(M, f) = \{ i \mid (\exists x)[\psi_i(x) \not= f_i(x)] \} \). Note that it is easy to construct such a machine. It is easy to verify that \( \mathcal{R}_\psi \subseteq \text{MinCoSuper}_\psi(M) \). 

---

20
Theorem 16 (∃ψ)[\text{card}(\mathcal{R}_\psi) = \infty \land \mathcal{R}_\psi \in \text{MinCoAll}_\psi \land (\forall S \subseteq \mathcal{R}_\psi)[\text{card}(S) = \infty \Rightarrow S \notin \text{MinDecEx}_\psi]]

Proof. We will construct a numbering \( \psi, \psi \in \mathcal{R}_\psi \), such that
(a) \( \psi \) is 1–1, and
(b) no infinite subset of \( \mathcal{R}_\psi \) is in \( \text{MinDecEx}_\psi \).

This would prove the theorem (since for any 1–1 total numbering \( \psi, \mathcal{R}_\psi \in \text{MinCoSuper}_\psi = \text{MinCoAll}_\psi \)).

If there exists a \( \sigma \), such that \( \mathcal{M}_0(\sigma) \neq ? \), then let \( \sigma_0 \) be the least such \( \sigma \); otherwise, let \( \sigma_0 = \Lambda \). For \( i > 0 \), if there exists a \( \sigma \supseteq \sigma_{i-1} \), such that \( \mathcal{M}_i(\sigma) \neq ? \), then let \( \sigma_i \) be the least such \( \sigma \); otherwise, let \( \sigma_i = \sigma_{i-1} \). Note that \( \sigma_i \) can be determined effectively in the limit. Let \( \tau_i^j \), be such that
(i) \( \tau_i^j \) can be determined effectively from \( i \) and \( j \),
(ii) \( \tau_i^j \subseteq \tau_{i+1}^j \), and
(iii) \( \lim_{j \rightarrow \infty} \tau_i^j = \sigma_i \).

Note that there exist such \( \tau_i^j \).
Let \( \psi_i \) be defined as follows.

\[
\psi_i(x) = \begin{cases} 
\tau_i^j(x), & \text{if } x < |\tau_i^j|; \\
\hat{i}, & \text{otherwise.}
\end{cases}
\]

It is easy to verify that each \( \psi_i \in \mathcal{R}_\psi \), and \( \psi_i \)'s are pairwise different. Thus (a) is satisfied. Note that for each \( i \), for all but finitely many \( j \), \( \sigma_i \subseteq \tau_i^j \).

Thus, for all \( i \),
(c) for all but finitely many \( j \), \( \sigma_i \subseteq \psi_j \),
(d) either \( \mathcal{M}_i(\sigma_i) \neq ? \), or (\( \forall \sigma \supseteq \sigma_i \)[\( \mathcal{M}_i(\sigma) = ? \)]).

It follows immediately that \( \mathcal{M}_i \) can \( \text{MinDecEx}_\psi \)-identify at most finitely many \( \psi_j \). Thus (b) is satisfied.

Theorem 17 (∃ψ)[\text{card}(\mathcal{R}_\psi) = \infty \land \mathcal{R}_\psi \in \text{MinEx}_\psi \land (\forall S \subseteq \mathcal{R}_\psi)[\text{card}(S) = \infty \Rightarrow S \notin \text{MinDecEx}_\psi \cup \text{MinIncEx}_\psi]]

Proof. This is a somewhat more complicated modification of the proof of Theorem 12.

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

Let \( C = \{ f_i \mid i \in N \} \). Let \( h(0) = 0 \). Let \( h(i + 1) = h(i) + 2^{2i+2} - 1 \).

We will make sure that (a), (b) and (c) are satisfied.

(a) For each \( i \), exactly one of the programs in \( S_i = \{ p \mid h(i) < p \leq h(i+1) \} \) will compute \( f_i \). All the other programs in \( S_i \) will compute non-total functions. This will make \( \mathcal{R}_\psi = C \).
(b) For \( j < i, f_i \not\in \text{MinDecEx}_\psi(M_j) \cup \text{MinIncEx}_\psi(M_j) \).

(c) \( C \in \text{MinEx}_\psi \).

This will prove the theorem.

Fix \( i \). We now define \( \psi_j \), for \( j \in S_i \).

Definition of \( \psi_j \), for \( j \in S_i \).

Let \( \text{Cancell}^0_i = \text{CancelD}^0_i = \emptyset \).

\( l(i, 0) = h(i) + 1 \). \( u(i, 0) = h(i + 1) \).

Go to stage 0.

Begin stage \( s \)

1. Let \( p(i, s) = \frac{l(i, s) + u(i, s)}{2} \).

2. For all \( x \leq s \), let \( \psi_{p(i, s)}(x) = f_i(x) \).

   (Intuitively, we want to make \( \psi_{p(i, \infty)} = f_i \).

3. \textbf{if} there exists a \( j < i \), such that \( j \not\in \text{CancelD}^s_i \) and \( p(i, s) \in \text{ProgSet}(M_j, f_i[s]) \)

   \textbf{then} let \( l(i, s) = p(i, s) + 1 \) and \( \text{CancelD}^{s+1}_i = \text{CancelD}^s_i \cup \{ j < i \mid p(i, s) \in \text{ProgSet}(M_j, f_i[s]) \} \).

4. \textbf{elseif} there exists a \( j < i \), such that \( j \not\in \text{Cancell}^s_i \) and \( p(i, s) \in \text{ProgSet}(M_j, f_i[s]) \)

   \textbf{then} let \( u(i, s) = p(i, s) - 1 \) and \( \text{Cancell}^{s+1}_i = \text{Cancell}^s_i \cup \{ j < i \mid p(i, s) \in \text{ProgSet}(M_j, f_i[s]) \} \).

\textbf{endif}

5. Go to stage \( s + 1 \).

End stage \( s \).

End of Definition of \( \psi_j \), for \( j \in S_i \).

It is easy to verify that \( \text{CancelD}^s_i \), \( \text{Cancell}^s_i \) are monotonically nondecreasing (with respect to \( \subseteq \)) in \( s \). Let \( \text{CancelD}^\infty_i = \lim_{s \to \infty} \text{CancelD}^s_i \) and \( \text{Cancell}^\infty_i = \lim_{s \to \infty} \text{Cancell}^s_i \). Also it is easy to verify that \( l^s_i \) is monotonically nondecreasing and \( u^s_i \) is monotonically nonincreasing in \( s \). Let \( u^\infty_i = \lim_{s \to \infty} u^s_i \), and \( l^\infty_i = \lim_{s \to \infty} l^s_i \). Let \( p(i, \infty) = \lim_{s \to \infty} p(i, s) = \frac{l^\infty_i + u^\infty_i}{2} \).

Note that \( h(i) \leq l^s_i < p(i, s) \leq u^s_i \leq h(i + 1) \). Also, \( \psi_{p(i, \infty)} = f_i \), and for all \( j \in S_i \setminus \{ p(i, \infty) \} \), \( \psi_j \) is a non-total function. Thus (a) is satisfied.

Consider any \( j < i \). If \( j \in \text{CancelD}^s_i \) then \( \text{ProgSet}(M_j, f_i) \) contains a program \( < l(i, s) \leq l(i, \infty) < p(i, \infty) \). If \( j \not\in \text{CancelD}^\infty_i \), then \( \text{ProgSet}(M_j, f_i) \) does not contain \( p(i, \infty) \). In either case \( \psi_{p(i, \infty)} = f_i \not\in \text{MinDecEx}_\psi(M_j) \).
Similarly, if \( j \in \text{Cancell}^i \) then \( \text{ProgSet}(M_j, f_i) \) contains a program \( u(i, s) \geq u(i, \infty) > p(i, \infty) \). If \( j \notin \text{Cancell}^i \), then \( \text{ProgSet}(M_j, f_i) \) does not contain \( p(i, \infty) \). In either case \( \psi_{p(i, \infty)} = f_i \notin \text{MinIncEx}_\psi(M_j) \). Thus (b) is satisfied.

To show that \( \mathcal{C} \in \text{MinEx}_\psi \), note that \( p(i, s) \) converges to the minimal \( \psi \)-program for \( f_i \). Thus \( \mathcal{C} \in \text{MinEx}_\psi \). Thus (c) is satisfied.

The following theorem says that, for suitable non-Gödel numberings, the minimality criteria discussed in the paper are not so weak.

**Theorem 18** \((\forall \mathcal{C} \in \text{Ex})(\exists \ l-1 \ numbering \ \psi)[\mathcal{C} \in \text{MinCoAll}_\psi].\)

**Proof.** We use the following result from [16].

**Lemma 1** \( \mathcal{C} \in \text{Ex} \) iff there exists a numbering \( \psi \) such that

(i) \( \mathcal{C} \subseteq \mathcal{R}_\psi \), and

(ii) there exists a \( d \in \mathbb{R}^2 \) such that, for any \( i \neq j \), \( \psi_i[d(i, j)] \neq \psi_j[d(i, j)] \).

Now, suppose \( \mathcal{C} \in \text{Ex} \). Let \( \psi, d \) be as given in the lemma. First note that \( \psi \) contains exactly one program for any function in \( \mathcal{C} \). Thus showing that \( \mathcal{C} \in \text{CoAll}_\psi \) suffices. Let \( M \) be defined such that:

\[ \text{ProgSet}(M, f) = \{ j \mid (\exists i \neq j) [f[d(i, j)] \subseteq \psi_i] \}. \]

It is easy to verify that \( M \) \text{MinCoAll}_\psi-identifies \( \mathcal{C} \).

Note that the above theorem indirectly characterizes \( \bigcup_{\psi \in \mathbb{R}^2} \text{I}_\psi \), where \( \text{I} \) is any of the (min) co-learnability criteria discussed in this paper (since they turn out to be equivalent to \( \text{Ex} \)).

**Theorem 19**

\((\exists \psi)[\text{CoSub}_\psi \neq \emptyset \land \text{CoSuper}_\psi \neq \emptyset \land \text{CoSub}_\psi \neq \emptyset].\)

**Proof.** Let \( \eta \) be a numbering such that \( \text{CoSub}_\eta \neq \emptyset \). Let \( \beta \) be a numbering such that \( \text{CoSuper}_\beta \neq \emptyset \). Define \( \psi \) as follows. \( \psi_{2i}(0) = 1 \). \( \psi_{2i+1}(0) = 2 \). For all \( x \), \( \psi_{2i}(x + 1) = \eta_i(x) \). For all \( x \), \( \psi_{2i+1}(x + 1) = \beta_i(x) \). It is easy to verify that \( \psi \) witnesses the theorem.

**Theorem 20** \((\exists \tau \in \mathbb{R}^2)[\text{CoAll}_\tau \subset \text{MinCoSuper}_\tau].\)

23
Proof. By Theorem 15 for any $\tau \in \mathcal{R}^2$, $\mathcal{R}_\tau \in \text{MinCoSuper}_\tau$. So we just need to construct a $\tau \in \mathcal{R}^2$ such that $\mathcal{R}_\tau \not\in \text{CoAll}_\tau$. Define $\tau$ as follows.

Let $f_i(0) = i$, and, for $x > 0$, $f_i(x) = 0$. $\tau_{2i}(0) = \tau_{2i+1}(0) = i$. For $x > 0$, $\tau_{2i}(x), \tau_{2i+1}(x)$ are defined as follows.

$$
\tau_{2i}(x) = \begin{cases} 
0, & \text{if } 2i, 2i + 1 \not\in \text{ProgSet}(M_i, f_i[x]) \\
0, & \text{if } 2i \in \text{ProgSet}(M_i, f_i[x]) \\
1, & \text{if } 2i + 1 \in \text{ProgSet}(M_i, f_i[x]) \text{ and } 2i \not\in \text{ProgSet}(M_i, f_i[x]) 
\end{cases}
$$

$$
\tau_{2i+1}(x) = \begin{cases} 
0, & \text{if } 2i, 2i + 1 \not\in \text{ProgSet}(M_i, f_i[x]) \\
0, & \text{if } 2i + 1 \in \text{ProgSet}(M_i, f_i[x]) \\
1, & \text{if } 2i \in \text{ProgSet}(M_i, f_i[x]) \text{ and } 2i + 1 \not\in \text{ProgSet}(M_i, f_i[x]) 
\end{cases}
$$

It is easy to see that at least one of $\tau_{2i}, \tau_{2i+1}$ is $f_i$. Moreover, by construction, $M_i$ does not $\text{CoAll}_{\tau}$-identify $f_i$. \hfill \Box

4.4 Miscellaneous

**Theorem 21** ($\forall$ infinite r.e. $\mathcal{U}$) ($\exists$ infinite $\mathcal{V} \subseteq \mathcal{U}$) ($\exists$ Kolmogorov Numbering $\psi$) [$\mathcal{V} \in \text{MinCoAll}_{\psi}$].

**Proof.** Let $\eta \in \mathcal{R}^2$ be a 1-1 numbering such that $\mathcal{U} = \{\eta_i \mid i \in \mathbb{N}\}$. Note that there exists such a numbering. Let $M_{\text{MinCoAll}_{\psi}}$-identify $\mathcal{U}$ (note that there exists such a machine due to Theorem 15 and $\eta$ being a 1-1 numbering). Let $\beta$ be a Kolmogorov numbering. Define $\psi$ as follows. $\psi_{3i} = \beta_i$, $\psi_{3i+1} = \eta_{2i}$ and $\psi_{3i+2} = \eta_{2i+1}$. Let $M'$ be such that $\text{ProgSet}(M', f) = \{3i \mid i \in \mathbb{N}\} \cup \{3i+1 \mid 2i \in \text{ProgSet}(M, f)\} \cup \{3i+2 \mid 2i+1 \in \text{ProgSet}(M, f)\}$.

Let $\mathcal{V} = \{\eta_{2i} \mid (\forall j \leq i)[\beta_j \neq \eta_{2i}]\} \cup \{\eta_{2i+1} \mid (\forall j \leq i)[\beta_j \neq \eta_{2i+1}]\}$. It is easy to verify that $\mathcal{V}$ is infinite and $\mathcal{V} \subseteq \text{MinCoAll}_{\psi}(M')$. \hfill \Box

**Theorem 22** ($\exists \mathcal{C}$) ($\exists$ Kolmogorov Numbering $\psi$)$[\mathcal{C} \in \text{MinDecEx}_{\psi} \wedge (\forall \text{Gödel Numbering } \eta)\mathcal{C} \not\in \text{MinIncEx}_{\eta}]$.**

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

Without loss of generality suppose that $\varphi$ is a Kolmogorov numbering. Let $\psi_{2i+1} = \varphi_i$. Note that this makes $\psi$ a Kolmogorov numbering.

24
For all $x$, let $\psi_{2x}(x) = \varphi_i(0)$.

Consider the following machine $M$. $M(f[0]) = \varphi_i(0)$.

$$M(f[n]) = \begin{cases} 
2j, & \text{if } (\exists j' \leq n)[\Phi_{j'}(0) \leq n \land \varphi_{j'}(0) = f(0)] \\
? & \text{and } j = \min\{j' \leq n \mid \Phi_{j'}(0) \leq n \land \varphi_{j'}(0) = f(0)\}.
\end{cases}$$

It is easy to verify that $C \subseteq \text{MinDecEx}_\psi(M)$.

Let $\eta$ be any Gödel numbering and $M$ be any learning machine. We now show that $C \not\subseteq \text{MinIncEx}_\eta(M)$. This would prove the theorem.

By implicit use of Kleene recursion theorem there exists an $e$ such that $\eta_e$ may be defined as follows. Let $f_e$ denote the function, $f_e(x) = e$, for all $x$.

Definition of $\eta_e$.

1. Search for $e, n \in N$, such that $M(f_e[n]) > e$.
2. If and when such $e, n$ are found, let $\eta_e = f_e$.

End of Definition of $\eta_e$.

Note that if step 1 does not succeed then $M$ can $\text{MinIncEx}$-identify only finitely many functions in $C$. On the other hand, if step 1 search succeeds then clearly, $\text{MinProg}_\eta(f_e) \leq e$. However, since $M$ on $f_e$ outputs a program larger than $e$, $M$ cannot $\text{MinIncEx}_\eta$-identify $f_e$.

It follows that $C \not\subseteq \text{MinIncEx}_\eta(M)$.

The following theorem follows from the proof of Theorem 5.

**Theorem 23** $(\forall \text{ Kolmogorov Numbering } \psi)(\exists C)[C \in \text{MinCoAll}_\psi \land (\forall \text{ Gödel Numbering } \eta)[C \not\subseteq \text{MinDecEx}_\eta]]$.

**Theorem 24** $(\exists \text{ Kolmogorov Numbering } \psi)(\exists C)[C \in \text{MinCoSuper}_\psi \land (\forall \text{ Gödel Numbering } \eta)[C \not\subseteq \text{MinCoAll}_\eta]]$.

**Proof.** Without loss of generality suppose that $\varphi$ is a Kolmogorov numbering. We will construct a Kolmogorov numbering $\psi$ and class $C$ witnessing the theorem.

Let $\psi_{2i} = \varphi_i$. Note that this makes $\psi$ a Kolmogorov numbering. Define $h$ as follows.

$$h(0) = 1. \ h(i + 1) = 3(h(i) + 1).$$

25
Note that for any Gödel numbering \( \eta \), there must be a recursive function \( \varphi_j \) witnessing the reduction from \( \psi \) to \( \eta \). Thus we will try to diagonalize against all pairs of machines, \( M_i \), and potential reduction functions, \( \varphi_j \).

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

Let \( C = \{ \psi_p \mid p \in \bigcup_k S_k \land p = \text{MinProg}_\psi(\psi_p) \} \).

Note that totality of functions computed by \( \psi_p, p \in \bigcup_k S_k \), immediately implies that \( C \in \text{MinCoSuper}_\psi \). For each \( k = \langle i, j \rangle \), we will construct the functions computed by \( \psi_p, p \in S_k \), in such a way that, if \( \varphi_j \) is an increasing function witnessing a reduction from \( \psi \) to Gödel numbering \( \eta \), then, for at least one \( p \in S_k \), \( \psi_p \in C - \text{MinCoAll}_\eta(M_i) \). This would prove that \( C \notin \text{MinCoAll}_\eta \), for any Gödel numbering \( \eta \).

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

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

It is easy to verify that all functions computed by programs in \( \bigcup_k S_k \) are total. Suppose \( \varphi_j \) witnesses the reduction between \( \psi \) and \( \eta \). Then \( M_i \) does not \( \text{MinCoAll}_\psi \)-identify at least one of \( \psi_{\eta + w} \) and \( \psi_{\eta + w + 3} \). Moreover, for each \( k \), there exists an \( l, h(k) < l \leq h(k + 1) \), and a \( w \in \{1, 2, 3\} \), such that for all \( x \leq h(k + 1) \), \( \varphi_x(0) \neq 7l + w \). (This is so since the number of such pairs \( l, w \) is \( 3(h(k + 1) - h(k)) > h(k + 1) \)). It immediately follows that there exists a \( p \in S_k \), such that \( \psi_p \in C \), and \( M_i \) does not \( \text{MinCoAll}_\psi \)-identify \( \psi_p \).

It is open at present whether \( (\exists \text{ Kolmogorov Numbering } \psi)(\exists C)[C \in \text{MinIncEx}_\psi \land (\forall \text{ Gödel Numbering } \eta)[C \notin \text{MinCoSuper}_\eta]] \).
5 A characterization of $\text{MinIncEx}_\psi$

In this section we provide a characterization of $\text{MinIncEx}_\psi$ (and thus of $\text{MinCoSub}_\psi$ and $\text{CoSub}_\psi$).

**Theorem 25** $C \in \text{MinIncEx}_\psi$ iff there exists a $P \in \mathcal{P}$ such that the following properties are satisfied.

1. $C \subseteq \mathcal{R}_\psi$;
2. For all $f \in C$, $P(\text{MinProg}_\psi(f)) \downarrow$;
3. For all $i \in N$, $P(i) \downarrow$ implies (3a) and (3b):
   3a) For all $x < P(i)$, $\psi_i(x) \downarrow$.
   3b) For all $j < i$ with $\psi_j \in C$, $\psi_j[P(i)] \neq \psi_i[P(i)]$.

**Proof.** ($\Rightarrow$) Suppose $C \in \text{MinIncEx}_\psi$ as witnessed by $M$. (1) must clearly hold. Let $P$ be defined as follows:

$$P(i) = \min\{x \mid (\exists y < x)[\psi(y) \downarrow] \land M(\psi[y]) = i\}$$

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

($\Leftarrow$) We use the fact that $\text{CoSub}_\psi = \text{MinIncEx}_\psi$. Suppose (1) holds and $P$ is such that (2) and (3) hold. Then consider a machine $M$ such that

$$\text{ProgSet}(M, f) = \{j \mid (\exists i > j)[P(i) \downarrow = n \land f[n] = \psi_i[n]]\}$$

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

It would be interesting to get a similar characterization of $\text{MinDecEx}_\psi$ and other criteria of inference considered in this paper.

**References**


27


