Learning without Coding

Samuel E. Moelius III[1] and Sandra Zilles[2]

[1] IDA Center for Computing Sciences
17100 Science Drive, Bowie, MD 20715-4300
semoeli@super.org
[2] Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada S4S 0A2
zilles@cs.uregina.ca

June 30, 2010

Abstract. Iterative learning is a model of language learning from positive data, due to Wiehagen. When compared to a learner in Gold's original model of language learning from positive data, an iterative learner can be thought of as *memory-limited*. However, an iterative learner can memorize *some* input elements by *coding* them into the syntax of its hypotheses. A main concern of this paper is: to what extent are such coding tricks *necessary*?

One means of preventing *some* such coding tricks is to require that the hypothesis space used be free of redundancy, i.e., that it be 1-1. In this context, we make the following contributions. By extending a result of Lange & Zeugmann, we show that many interesting and non-trivial classes of languages can be iteratively identified using a Friedberg numbering as the hypothesis space. (Recall that a Friedberg numbering is a 1-1 effective numbering of all computably enumerable sets.) An example of such a class is the pattern languages over an arbitrary alphabet. On the other hand, we show that there exists a class of languages that can*not* be iteratively identified using any 1-1 effective numbering as the hypothesis space.

We also consider an iterative-like learning model in which the computational component of the learner is modeled as an *enumeration operator*, as opposed to a partial computable function. In this new model, there are no hypotheses, and, thus, no syntax in which the learner can encode what elements it has or has not yet seen. We show that there exists a class of languages that *can* be identified under this new model, but that can*not* be iteratively identified. On the other hand, we show that there exists a class of languages that can*not* be identified under this new model, but that *can* be iteratively identified using a Friedberg numbering as the hypothesis space.

Keywords: coding tricks, inductive inference, iterative learning.

1 Introduction

Iterative learning (**It**-learning, Definition 1(a)) is a model of language learning from positive data, due to Wiehagen [Wie76]. Like many models based on positive data, the **It**-learning model involves a learner that is repeatedly fed elements drawn from $\{\#\}$ and from some unknown target language $L \subseteq \mathbb{N}$, where $\mathbb{N}$ is the set of natural numbers, $\{0, 1, 2, ...\}$.[3] After being fed each such element, the learner outputs a hypothesis (provided that the learner does not diverge). The learner is said to *identify* the target language L iff there is some point from whence on the learner outputs only one hypothesis, and that hypothesis corresponds to L. Furthermore, the learner is said to identify a *class* of languages $\mathcal{L}$ iff the learner identifies *each* $L \in \mathcal{L}$ when fed the elements of L (and possibly $\#$).

In the **It**-learning model, the learner itself is modeled as a triple.

- The first element of the triple is a two-place partial computable function, whose arguments are, respectively, the learner's most recently output hypothesis, and the next input element.
- The second element of the triple is a preliminary hypothesis, i.e., the hypothesis output by the learner before being fed any input.
- The third element of the triple is a *hypothesis space*. The *hypothesis space* determines the language that corresponds to each of the learner's hypotheses. Formally, a hypothesis space is a numbering $(X_j)_{j\in\mathbb{N}}$ of some collection of subsets of $\mathbb{N}$, and that is *effective* in the sense that the two-place predicate $\lambda j, x \,.\, [x \in X_j]$ is partial computable.[4]

It-learning is a special case of Gold's original model of language learning from positive data [Gol67]. In Gold's original model, the learner is provided access to all previously seen input elements, in addition to the next input element. In this sense, a learner in Gold's model can be thought of as *memorizing* all previously seen input elements. When compared to learners in Gold's model, iterative learners are restricted in terms of the classes of languages that they can identify.[5] In this sense, the *memory-limited* aspect of iterative learners is a *true* restriction, and *not* a mere superficial difference in definitions.

This does not however mean that iterative learners are *memory-less*. In particular, an iterative learner can memorize *some* input elements by employing *coding tricks*, which we define (informally) as follows.

- A *coding trick* is any use by an iterative learner of the syntax of a hypothesis to determine what elements that learner has or has not yet seen.

[3] The symbol '#' is pronounced "pause". The inclusion of # in the model allows the target language L to be empty, i.e., in such a case, the learner is repeatedly fed #.

[4] *Not-necessarily-effective* hypothesis spaces have also been considered [dBY10]. However, such hypothesis spaces are not needed herein. For the remainder, we use the terms *hypothesis space* and *effective numbering* interchangeably.

[5] Many variants of the **It**-learning model have been considered, and have also been shown to be restricted in this sense [LZ96,CCJS07,JLMZ10].

The following is an example. Suppose that an iterative learner $(M, p, (X_j)_{j\in\mathbb{N}})$ identifies a class of languages $\mathcal{L}$. Further suppose that one desires a learner that identifies the class $\mathcal{L}'$, where

$$\mathcal{L}' = \mathcal{L} \cup \{L \cup \{0\} \mid L \in \mathcal{L}\}. \tag{1}$$

Such a learner $(M', p', (Y_k)_{k\in\mathbb{N}})$ may be obtained as follows. Let $(Y_k)_{k\in\mathbb{N}}$ be such that, for each j:

$$Y_{2j} = X_j; \qquad Y_{2j+1} = X_j \cup \{0\}. \tag{2}$$

Then, let M' be such that, for each $x \in (\mathbb{N} \cup \{\#\}) - \{0\}$:

$$\begin{aligned} M'(2j, \quad x) &= 2M(j,x); & M'(2j, \quad 0) &= 2M(j,0)+1; \\ M'(2j+1, x) &= 2M(j,x)+1; & M'(2j+1, 0) &= 2M(j,0)+1. \end{aligned} \tag{3}$$

It is easily seen that $(M', 2p, (Y_k)_{k\in\mathbb{N}})$ iteratively identifies $\mathcal{L}'$. Intuitively, M' simulates M, while using the least-significant-bit of each hypothesis to *encode* whether or not M' has seen a 0. (Note the switch from even to odd hypotheses in the upper-right of (3).) Further note that, if $\mathcal{L}$ already contains languages for which 0 is a member, then there is *redundancy* in the hypothesis space $(Y_k)_{k\in\mathbb{N}}$. In particular, if $0 \in X_j$, then $Y_{2j} = Y_{2j+1}$. For such hypotheses, the least-significant bit affects *only* their syntax, and *not* their semantics.

This example demonstrates how coding tricks can at least *facilitate* the identification of a class of languages. A main concern of this paper is: to what extent are such coding tricks *necessary*?

One approach to preventing *some* such coding tricks is to require that the hypothesis space be free of redundancy, i.e., that it be 1-1. One means of doing this is to require that the hypothesis space be a *Friedberg numbering* [Fri58,Kum90]. A *Friedberg numbering* is a 1-1 effective numbering of all computably enumerable (ce) subsets of $\mathbb{N}$. The use of such numberings as hypothesis spaces was considered by Jain & Stephan [JS08].[6] They observed, for example, that $\mathcal{F}in$, the collection of all finite subsets of $\mathbb{N}$, can*not* be iteratively identified using a Friedberg numbering as the hypothesis space [JS08, Remark 28]. For the remainder, to **FrIt**-*identify* a class of languages $\mathcal{L}$ shall mean to iteratively identify $\mathcal{L}$ using a Friedberg numbering as the hypothesis space (see Definition 1(b)).

Our first main result is to show that, despite this observation of Jain & Stephan, many interesting and non-trivial classes can be **FrIt**-identified. More specifically, we extend a result of Lange & Zeugmann [LZ96, Theorem 12] by showing that, for each class $\mathcal{L}$, if there exists a single hypothesis space witnessing that $\mathcal{L}$ is both uniformly decidable and computably finitely thick, then $\mathcal{L}$ can be **FrIt**-identified (Theorem 6). By comparison, Lange & Zeugmann showed that such a class can be **It**-identified.

We delay the definitions of the terms *uniformly decidable* and *computably finitely thick* to Section 3. In the meantime, however, we mention one significant

[6] Freivalds, et al. [FKW82] considered the use of Friedberg numberings as hypothesis spaces in the context of *function* learning.

application of our result. A *pattern language* [Ang80] is a type of language with applications to molecular biology (see, e.g., [SSS$^+$94]). Furthermore, the pattern languages naturally form classes that are **It**-identifiable by Lange & Zeugmann's result,[7] and, thus, are **FrIt**-identifiable, by ours.

We briefly recall the definition of a pattern language. Suppose that Σ is an alphabet, i.e., a non-empty, finite set of symbols. A *pattern* over Σ is a finite string whose symbols are drawn from Σ, and from some infinite collection of variables. The *language determined by* a pattern p (over Σ) is the set of all strings that result by substituting some non-empty string (over Σ) for each variable in p. A *pattern language* over Σ is any language determined by a pattern over Σ. $\mathcal{Pat}^{\Sigma}$ denotes the collection of all pattern languages over Σ.

For example, suppose that $\Sigma = \{\mathtt{0}, \mathtt{1}\}$, and that p is the patten $x\mathtt{0}x\mathtt{1}y$, where x and y are variables. Then, $\mathcal{Pat}^{\Sigma}$ includes the language determined by p, which, in turn, includes the following strings. (To lessen the burden upon the reader, we have underlined in each string a $\mathtt{0}$ and $\mathtt{1}$ that may be regarded as part of the original pattern.)

$\mathtt{00\underline{0}00\underline{1}0}$	$\mathtt{0\underline{0}0\underline{1}010}$	$\mathtt{0\underline{0}0\underline{1}110}$	$\mathtt{10\underline{0}10\underline{1}0}$	$\mathtt{1\underline{0}1\underline{1}010}$	$\mathtt{1\underline{0}1\underline{1}110}$
$\mathtt{00\underline{0}00\underline{1}1}$	$\mathtt{0\underline{0}0\underline{1}011}$	$\mathtt{0\underline{0}0\underline{1}111}$	$\mathtt{10\underline{0}10\underline{1}1}$	$\mathtt{1\underline{0}1\underline{1}011}$	$\mathtt{1\underline{0}1\underline{1}111}$
$\mathtt{0\underline{0}0\underline{1}000}$	$\mathtt{0\underline{0}0\underline{1}100}$	$\mathtt{01\underline{0}01\underline{1}0}$	$\mathtt{1\underline{0}1\underline{1}000}$	$\mathtt{1\underline{0}1\underline{1}100}$	$\mathtt{11\underline{0}11\underline{1}0}$
$\mathtt{0\underline{0}0\underline{1}001}$	$\mathtt{0\underline{0}0\underline{1}101}$	$\mathtt{01\underline{0}01\underline{1}1}$	$\mathtt{1\underline{0}1\underline{1}001}$	$\mathtt{1\underline{0}1\underline{1}101}$	$\mathtt{11\underline{0}11\underline{1}1}$

On the other hand, the language determined by p includes no other strings of length 7.

As the reader may have already noticed, if one's intent is simply to eliminate redundancy in the hypothesis space, then to require that the hypothesis space be a Friedberg numbering is really overkill. That is because to require that the hypothesis space be a Friedberg numbering is to require that it be free of redundancy *and* that it represent all of the ce sets.

Thus, we consider a milder variant of **FrIt**-learning, which we call *injective iterative learning* (**InjIt**-learning, Definition 1(c)). In this variant, the hypothesis space is required to be free of redundancy (i.e., be 1-1), but need not represent all of the ce sets.[8] Clearly, for each class $\mathcal{L}$, if $\mathcal{L}$ can be **FrIt**-identified, then $\mathcal{L}$ can be **InjIt**-identified. On the other hand, $\mathcal{Fin}$ can be **InjIt**-identified, but, as per Jain & Stephan's observation mentioned above, $\mathcal{Fin}$ can*not* be **FrIt**-identified.

Going further, if one's intent is to *prevent coding tricks*, then to require that the hypothesis space be free of redundancy may still be overkill. In particular, one might allow that there be redundancy in the hypothesis space, but require that the learner not benefit from this redundancy. This idea is captured in our next model, which is called *extensional iterative learning* (**ExtIt**-learning, Definition 1(d)).

[7] The pattern languages were first shown to be **It**-identifiable by Lange & Wiehagen [LW91].

[8] The use of 1-1 hypothesis spaces was also considered in [BBCJS10] in the context of learning certain *specific* classes of languages.

For a learner to **ExtIt**-identify a class of languages, it is required that, when presented with equivalent hypotheses and identical input elements, the learner must produce equivalent hypotheses. More formally: suppose that $\mathcal{L}$ is a class of languages, that σ_0 and σ_1 are two non-empty sequences of elements drawn from $\{\#\}$ and from two (possibly distinct) languages in $\mathcal{L}$, and that the following conditions are satisfied.

– When fed all but the last elements of σ_0 and σ_1, the learner outputs hypotheses for the same language (though those hypotheses may differ syntactically).
– The last elements of σ_0 and σ_1 are identical.

Then, for the learner to **ExtIt**-identify $\mathcal{L}$, it is required that:

– When fed *all* of σ_0 and σ_1, the learner outputs hypotheses for the same language (though those hypotheses may differ syntactically).

Clearly, if a learner identifies a class of languages using a 1-1 hypothesis space, then that learner satisfies the just above requirement. Thus, every class of languages that can be **InjIt**-identified can be **ExtIt**-identified. On the other hand, we show that there exists a class of languages that *can* be **ExtIt**-identified, but that can*not* be **InjIt**-identified (Theorem 11).

Before introducing our final model, let us recall the definition of an *enumeration operator* [Rog67, §9.7]. For now, we focus on enumeration operators of a particular type. A more general definition is given in Section 2.1.

Let $\mathcal{P}(\mathbb{N})$ be the powerset of $\mathbb{N}$, i.e., the collection of all subsets of $\mathbb{N}$. Let $\langle \cdot, \cdot \rangle$ be any pairing function, i.e., a computable, 1-1, onto function of type $\mathbb{N}^2 \to \mathbb{N}$ [Rog67, page 64]. Let $\hat{\#} = 0$, and, for each $x \in \mathbb{N}$, let $\hat{x} = x + 1$. Let $(D_j)_{j \in \mathbb{N}}$ be any canonical enumeration of $\mathcal{Fin}$.

An *enumeration operator* of type $\mathcal{P}(\mathbb{N}) \times (\mathbb{N} \cup \{\#\}) \to \mathcal{P}(\mathbb{N})$ is a mapping that is algorithmic in the following precise sense. To each enumeration operator Θ (of the given type), there corresponds a ce set H, such that, for each $X \subseteq \mathbb{N}$ and $x \in \mathbb{N} \cup \{\#\}$,

$$\Theta(X, x) = \big\{ y \mid \big\langle j, \langle \hat{x}, y \rangle \big\rangle \in H \ \wedge \ D_j \subseteq X \big\}. \tag{4}$$

Thus, given an *enumeration of* X, and given x, one can enumerate $\Theta(X, x)$ in the following manner.

– Enumerate H. For each element of the form $\big\langle j, \langle \hat{x}, y \rangle \big\rangle \in H$, if ever the finite set D_j appears in the enumeration of X, then list y into $\Theta(X, x)$.

Enumeration operators exhibit certain notable properties, including *monotonicity*. Intuitively, this means that an enumeration operator can tell from its set argument X what elements are in X, but it cannot tell from X what elements are in the *complement of* X. More is said about the properties of enumeration operators in Section 2.1.

The final model that we consider is called *iterative learning by enumeration operator* (**EOIt**-learning, Definition 1(e)). As the name suggests, the computational component of the learner is modeled as an enumeration operator, as opposed to a partial computable function. Specifically, the learner is modeled as a *pair*, where:

- The first element of the pair is an enumeration operator of type $\mathcal{P}(\mathbb{N}) \times (\mathbb{N} \cup \{\#\}) \rightarrow \mathcal{P}(\mathbb{N})$, whose arguments are, respectively, the learner's most recently output *language*, and the next input element.
- The second element of the pair is the learner's preliminarily output *language*, i.e., the language output by the learner before being fed any input. (We require that this preliminary language be ce.)

Thus, there are no hypotheses in this model. Since there are no hypotheses, there is no *syntax* in which the learner can encode what elements it has or has not yet seen.

The expulsion of hypotheses from the model has an additional consequence, and that is that the success criterion has to be adjusted. Specifically, we say that a learner in this model *identifies* a language L iff when fed the elements of L (and possibly $\#$), there is some point from whence on the learner outputs only the *language* L. The success criterion for identifying a *class* of languages is adjusted similarly. This more liberal approach to language identification, in some sense, gives an advantage to learners in this model. In particular, there exists a class of languages that *can* be **EOIt**-identified, but that can*not* be **It**-identified (Corollary 15).

Interestingly, there also exists a class of languages that can*not* be **EOIt**-identified, but that *can* be **FrIt**-identified (Theorem 20). To help to see why, consider the following two scenarios. First, suppose that $(\mathcal{M}, X)$ is a learner in the enumeration operator model, and that Y is its most recently output language. Then, since $\mathcal{M}$ is an enumeration operator, $\mathcal{M}$ can tell from Y what elements are in Y, but it cannot tell from Y what elements are in the *complement of* Y. Next, consider the analogous situation for a conventional iterative learner. That is, suppose that $(M, p, (X_j)_{j\in\mathbb{N}})$ is such a learner, and that j is its most recently output hypothesis. Then, in many cases, M *can* tell from j what elements are in the complement of X_j. In this sense, one could say that a hypothesis implicitly encodes *negative* information about the language that it represents. (In fact, this phenomenon can clearly be seen in the proof of Theorem 20 below.)

A question to then ask is: is this a *coding trick*, i.e., is it the case that *every* learner that operates on hypotheses (as opposed to languages) is employing coding tricks? At present, we do not see a clear answer to this question. Thus, we leave it as a subject for further study.

The main points of the preceding paragraphs are summarized in Figure 1. The remainder of this paper is organized as follows. Section 2 covers preliminaries. Section 3 presents our results concerning uniformly decidable and computably finitely thick classes of languages. Section 4 presents our results concerning Friedberg, injective, and extensional iterative learning (**FrIt**, **InjIt**, and **ExtIt**-learning, respectively). Section 5 presents our results concerning iterative learning by enumeration operator (**EOIt**-learning).

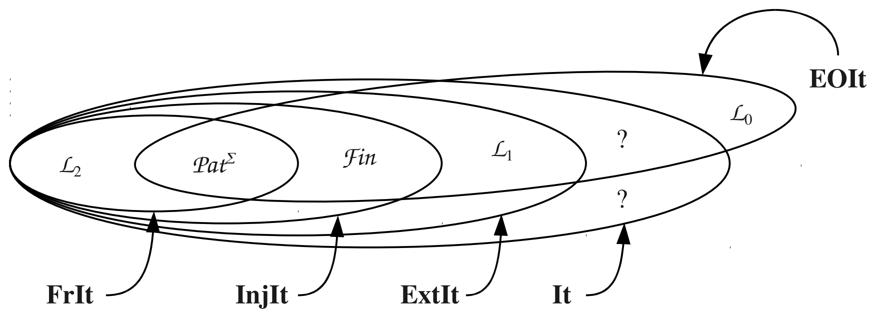

Fig. 1. A summary of main results and open problems. $\mathcal{P}at^{\Sigma}$ is the collection of all pattern languages over Σ, where Σ is an arbitrary alphabet. $\mathcal{F}in$ is the collection of all finite subsets of $\mathbb{N}$. The classes $\mathcal{L}_0$, $\mathcal{L}_1$, and $\mathcal{L}_2$ are defined in the proofs of Theorems 9, 11, and 20, respectively. The existence of the class $\mathcal{L}_3$ was shown by Jain (see Theorem 12).

2 Preliminaries

Computability-theoretic concepts not covered below are treated in [Rog67].

$\mathbb{N}$ denotes the set of natural numbers, $\{0, 1, 2, \ldots\}$. Lowercase math-italic letters (e.g., a, j, x), with or without decorations, range over elements of $\mathbb{N}$, unless stated otherwise. Uppercase italicized letters (e.g., A, J, X), with or without decorations, range over subsets of $\mathbb{N}$, unless stated otherwise. For each non-empty X, $\min X$ denotes the minimum element of X. $\min \emptyset \stackrel{\text{def}}{=} \infty$. For each non-empty, finite X, $\max X$ denotes the maximum element of X. $\max \emptyset \stackrel{\text{def}}{=} -1$. For each X and Y, $X \triangle Y$ denotes the symmetric difference of X and Y, i.e., $\{X - Y\} \cup \{Y - X\}$.

$\mathcal{P}(\mathbb{N})$ denotes the powerset of $\mathbb{N}$, i.e., the collection of all subsets of $\mathbb{N}$. $\mathcal{P}(\mathbb{N})^m$ denotes the collection of all tuples of length m whose elements are drawn from $\mathcal{P}(\mathbb{N})$. Uppercase calligraphic letters (e.g., $\mathcal{L}$, $\mathcal{X}$), with or without decorations, range over subsets of $\mathcal{P}(\mathbb{N})$, unless stated otherwise. $\mathcal{F}in$ denotes the collection of all finite subsets of $\mathbb{N}$. $(D_j)_{j \in \mathbb{N}}$ denotes a canonical enumeration of $\mathcal{F}in$.

$\langle \cdot, \cdot \rangle$ denotes any fixed pairing function, i.e., a computable, 1-1, onto function of type $\mathbb{N}^2 \to \mathbb{N}$ [Rog67, page 64]. For each x, $\langle x \rangle \stackrel{\text{def}}{=} x$. For each $x_0, ..., x_{n-1}$, where $n > 2$, $\langle x_0, ..., x_{n-1} \rangle \stackrel{\text{def}}{=} \big\langle x_0, \langle x_1, ..., x_{n-1} \rangle \big\rangle$.

$\mathbb{N}_\# \stackrel{\text{def}}{=} \mathbb{N} \cup \{\#\}$. A *text* is a total function of type $\mathbb{N} \to \mathbb{N}_\#$. For each text t and $i \in \mathbb{N}$, $t[i]$ denotes the initial segment of t of length i. For each text t, $\mathrm{content}(t) \stackrel{\text{def}}{=} \{t(i) \mid i \in \mathbb{N}\} - \{\#\}$. For each text t and $L \subseteq \mathbb{N}$, t is a text *for* L $\stackrel{\text{def}}{\Leftrightarrow}$ $\mathrm{content}(t) = L$.

Seq denotes the set of all initial segments of texts. Lowercase Greek letters (e.g., ρ, σ, τ), with or without decorations, range over elements of Seq, unless stated otherwise. λ denotes the empty initial segment (equivalently, the everywhere divergent function). For each σ, $|\sigma|$ denotes the length of σ (equivalently, the size of the domain of σ). For each σ and $i \leq |\sigma|$, $\sigma[i]$ denotes the initial segment of σ of length i. For each σ, $\mathrm{content}(\sigma) \stackrel{\text{def}}{=} \{\sigma(i) \mid i < |\sigma|\} - \{\#\}$. For each σ and τ, $\sigma \cdot \tau$ denotes the concatenation of σ and τ. For each $\sigma \in \mathrm{Seq} - \{\lambda\}$:

$$\sigma^- \overset{\text{def}}{=} \sigma[|\sigma| - 1]; \tag{5}$$

$$\mathrm{last}(\sigma) \overset{\text{def}}{=} \sigma(|\sigma| - 1). \tag{6}$$

For each L and $\mathcal{L}$, $\mathrm{Txt}(L)$, $\mathrm{Txt}(\mathcal{L})$, $\mathrm{Seq}(L)$, and $\mathrm{Seq}(\mathcal{L})$ are defined as follows.

$$\mathrm{Txt}(L) = \{\, t \mid t \text{ is a text for } L\}. \tag{7}$$

$$\mathrm{Txt}(\mathcal{L}) = \{\, t \mid (\exists L \in \mathcal{L})[t \in \mathrm{Txt}(L)]\}. \tag{8}$$

$$\mathrm{Seq}(L) = \{\sigma \mid \mathrm{content}(\sigma) \subseteq L\}. \tag{9}$$

$$\mathrm{Seq}(\mathcal{L}) = \{\sigma \mid (\exists L \in \mathcal{L})[\sigma \in \mathrm{Seq}(L)]\}. \tag{10}$$

For each one-argument partial function ψ and $x \in \mathbb{N}$, $\psi(x)\downarrow$ denotes that $\psi(x)$ converges; $\psi(x)\uparrow$ denotes that $\psi(x)$ diverges. We use $\uparrow$ to denote the value of a divergent computation.

For each $\mathcal{X}$, a *numbering of* $\mathcal{X}$ is an onto function of type $\mathbb{N} \to \mathcal{X}$. A numbering $(X_j)_{j\in\mathbb{N}}$ is *effective* $\overset{\text{def}}{\Leftrightarrow}$ the predicate $\lambda j, x \centerdot [x \in X_j]$ is partial computable. $\mathcal{EN}$ denotes the collection of all effective numberings.

$\mathcal{CE}$ denotes the collection of all computably enumerable (ce) subsets of $\mathbb{N}$. For each m and n, $\mathcal{PC}_{m,n}$ denotes the collection of partial computable functions mapping $\mathbb{N}^m \times \mathbb{N}^n_\#$ to $\mathbb{N}$. We shall be concerned primarily with $\mathcal{PC}_{1,0}$ and $\mathcal{PC}_{1,1}$. $(\varphi_p)_{p\in\mathbb{N}}$ denotes any fixed, acceptable numbering of $\mathcal{PC}_{1,0}$. For each i, $W_i \overset{\text{def}}{=} \{x \mid \varphi_i(x)\downarrow\}$. Thus, $(W_i)_{i\in\mathbb{N}}$ is an effective numbering of $\mathcal{CE}$.

For each $M \in \mathcal{PC}_{1,1}$ and p, the partial function M^*_p is such that, for each $\sigma \in \mathrm{Seq}$ and $x \in \mathbb{N}_\#$:

$$M^*_p(\lambda) = p; \tag{11}$$

$$M^*_p(\sigma \cdot x) = \begin{cases} M\big(M^*_p(\sigma), x\big), & \text{if } M^*_p(\sigma)\downarrow; \\ \uparrow, & \text{otherwise.} \end{cases} \tag{12}$$

2.1 Enumeration Operators

An *enumeration operator* is a mapping of type $\mathcal{P}(\mathbb{N})^m \times \mathbb{N}^n_\# \to \mathcal{P}(\mathbb{N})$, for some m and n, and that is algorithmic in the following precise sense. To each enumeration operator $\Theta : \mathcal{P}(\mathbb{N})^m \times \mathbb{N}^n_\# \to \mathcal{P}(\mathbb{N})$, there corresponds a ce set H, such that, for each $X_0, ..., X_{m-1}$ and $x_0, ..., x_{n-1}$,

$$\begin{aligned} &\Theta(X_0, ..., X_{m-1}, x_0, ..., x_{n-1}) \\ &\quad = \{y \mid \langle j_0, ..., j_{m-1}, x_0, ..., x_{n-1}, y\rangle \in H \ \wedge\ (\forall i < m)[D_{j_i} \subseteq X_i]\}. \end{aligned} \tag{13}$$

A strategy for enumerating $\Theta(X_0, ..., X_{m-1}, x_0, ..., x_{n-1})$, given $X_0, ..., X_{m-1}$ and $x_0, ..., x_{n-1}$, can easily be generalized from that given for enumeration operators of type $\mathcal{P}(\mathbb{N}) \times \mathbb{N}_\# \to \mathcal{P}(\mathbb{N})$ in Section 1.

For each $m, n \in \mathbb{N}$, $\mathcal{EO}_{m,n}$ denotes the collection of all enumeration operators of type $\mathcal{P}(\mathbb{N})^m \times \mathbb{N}^n_\# \to \mathcal{P}(\mathbb{N})$. We shall be concerned primarily with $\mathcal{EO}_{1,0}$ and $\mathcal{EO}_{1,1}$.

Enumeration operators exhibit monotonicity and continuity properties [Rog67, Theorem 9-XXI], described below for $\mathcal{EO}_{1,1}$.

– *Monotonicity*: for each $\mathcal{M} \in \mathcal{EO}_{1,1}$, $X, Y \subseteq \mathbb{N}$, and $x \in \mathbb{N}_{\#}$,

$$X \subseteq Y \;\Rightarrow\; \mathcal{M}(X, x) \subseteq \mathcal{M}(Y, x). \tag{14}$$

– *Continuity*: for each $\mathcal{M} \in \mathcal{EO}_{1,1}$, $X \subseteq \mathbb{N}$, $x \in \mathbb{N}_{\#}$, and $y \in \mathbb{N}$,

$$y \in \mathcal{M}(X, x) \;\Rightarrow\; (\exists A \in \mathcal{Fin})[A \subseteq X \;\wedge\; y \in \mathcal{M}(A, x)]. \tag{15}$$

For each $\mathcal{M} \in \mathcal{EO}_{1,1}$ and X, the function $\mathcal{M}^*_X : \mathrm{Seq} \to \mathcal{P}(\mathbb{N})$ is such that, for each $\sigma \in \mathrm{Seq}$ and $x \in \mathbb{N}_{\#}$:

$$\mathcal{M}^*_X(\lambda) = X; \tag{16}$$

$$\mathcal{M}^*_X(\sigma \cdot x) = \mathcal{M}\big(\mathcal{M}^*_X(\sigma), x\big). \tag{17}$$

2.2 Iterative and Iterative-like Learning Models

The following are the formal definitions of the learning models described in Section 1. The symbols **Fr**, **Inj**, **Ext**, and **EO** are mnemonic for *Friedberg*, *injective*, *extensional*, and *enumeration operator*, respectively.

Definition 1 For each $\mathcal{L}$, (a)-(e) below. In parts (a)-(d), $(M, p, (X_j)_{j\in\mathbb{N}}) \in \mathcal{PC}_{1,1} \times \mathbb{N} \times \mathcal{EN}$. In part (e), $(\mathcal{M}, X) \in \mathcal{EO}_{1,1} \times \mathcal{CE}$.

(a) **(Wiehagen [Wie76])** $(M, p, (X_j)_{j\in\mathbb{N}})$ **It**-*identifies* $\mathcal{L}$ $\Leftrightarrow$ for each $t \in \mathrm{Txt}(\mathcal{L})$, there exists $i_0 \in \mathbb{N}$ such that $X_{M^*_p(t[i_0])} = \mathrm{content}(t)$, and $(\forall i \geq i_0)\big[M^*_p(t[i]) = M^*_p(t[i_0])\big]$.
(b) **(Jain & Stephan [JS08])** $(M, p, (X_j)_{j\in\mathbb{N}})$ **FrIt**-*identifies* $\mathcal{L}$ $\Leftrightarrow$ $(M, p, (X_j)_{j\in\mathbb{N}})$ **It**-identifies $\mathcal{L}$, and $(X_j)_{j\in\mathbb{N}}$ is a Friedberg numbering.
(c) $(M, p, (X_j)_{j\in\mathbb{N}})$ **InjIt**-*identifies* $\mathcal{L}$ $\Leftrightarrow$ $(M, p, (X_j)_{j\in\mathbb{N}})$ **It**-identifies $\mathcal{L}$, and $(X_j)_{j\in\mathbb{N}}$ is 1-1.
(d) $(M, p, (X_j)_{j\in\mathbb{N}})$ **ExtIt**-*identifies* $\mathcal{L}$ $\Leftrightarrow$ $(M, p, (X_j)_{j\in\mathbb{N}})$ **It**-identifies $\mathcal{L}$, and, for each $\sigma_0, \sigma_1 \in \mathrm{Seq}(\mathcal{L}) - \{\lambda\}$,

$$[X_{M^*_p(\sigma_0^-)} = X_{M^*_p(\sigma_1^-)} \;\wedge\; \mathrm{last}(\sigma_0) = \mathrm{last}(\sigma_1)] \;\Rightarrow\; X_{M^*_p(\sigma_0)} = X_{M^*_p(\sigma_1)}. \tag{18}$$

(e) $(\mathcal{M}, X)$ **EOIt**-*identifies* $\mathcal{L}$ $\Leftrightarrow$ for each $t \in \mathrm{Txt}(\mathcal{L})$, there exists $i_0 \in \mathbb{N}$ such that $(\forall i \geq i_0)\big[\mathcal{M}^*_X(t[i]) = \mathrm{content}(t)\big]$.

Definition 2 Let **It** be as follows.

$$\mathbf{It} = \big\{\mathcal{L} \mid \big(\exists (M, p, (X_j)_{j\in\mathbb{N}}) \in \mathcal{PC}_{1,1} \times \mathbb{N} \times \mathcal{EN}\big)[(M, p, (X_j)_{j\in\mathbb{N}}) \text{ } \mathbf{It}\text{-identifies } \mathcal{L}]\big\}.$$

Let **FrIt**, **InjIt**, **ExtIt**, and **EOIt** be defined similarly.

3 Uniform Decidability and Computable Finite Thickness

In this section, we extend a result of Lange & Zeugmann by showing that, for each class $\mathcal{L}$, if there exists a single hypothesis space witnessing that $\mathcal{L}$ is both uniformly decidable and computably finitely thick, then $\mathcal{L}$ can be **FrIt**-identified (Theorem 6). We also show that there exists a class of languages that *is* uniformly decidable and computably finitely thick, but that is *not* in **It**, let alone **FrIt** (Theorem 9). Thus, one could not arrive at the conclusion of the just mentioned Theorem 6 if one were to merely require that: there exists a uniformly decidable effective numbering of $\mathcal{L}$, and a possibly *distinct* computably finitely thick effective numbering of $\mathcal{L}$.

The following are the formal definitions of the terms *uniformly decidable* and *computably finitely thick*. For additional background, see [LZZ08].

Definition 3

(a) An effective numbering $(X_j)_{j\in\mathbb{N}}$ is *uniformly decidable* $\Leftrightarrow$ the predicate $\lambda j, x\,.\,[x \in X_j]$ is decidable.
(b) A class of languages $\mathcal{L}$ is *uniformly decidable* $\Leftrightarrow$ there exists a uniformly decidable effective numbering of $\mathcal{L}$.
(c) An effective numbering $(X_j)_{j\in\mathbb{N}}$ is *computably finitely thick* $\Leftrightarrow$ there exists a computable function $f : \mathbb{N} \to \mathbb{N}$ such that, for each x,

$$\{X_j \mid j \in D_{f(x)}\} = \{L \mid x \in L \ \wedge \ (\exists j)[X_j = L]\}. \tag{19}$$

(d) **(Lange & Zeugmann [LZ96, Definition 9])** A class of languages $\mathcal{L}$ is *computably finitely thick* $\Leftrightarrow$ there exists a computably finitely thick effective numbering of $\mathcal{L}$.

N.B. In part (c) just above, the function f need *not* satisfy $D_{f(x)} = \{j \mid x \in X_j\}$. However, see Lemma 7 below.

Example 4

(a) $\mathcal{F}in$ *is* uniformly decidable, but is *not* computably finitely thick.
(b) $\mathcal{CE}$ is *neither* uniformly decidable *nor* computably finitely thick.
(c) The class $\big\{\{e\}, \{e\} \cup (W_e + e + 1) \mid e \in \mathbb{N}\big\}$ is *not* uniformly decidable, but *is* computably finitely thick.[9]
(d) The class $\{\mathbb{N} + e \mid e \in \mathbb{N}\}$ is both uniformly decidable and computably finitely thick. Moreover, there exists a single effective numbering witnessing both properties simultaneously.
(e) Let $\mathcal{L}$ be as follows.

$$\mathcal{L} = \big\{\{e\} \mid e \in \mathbb{N}\big\} \cup \big\{\{e, \varphi_e(0) + e + 1\} \mid e \in \mathbb{N} \ \wedge \ \varphi_e(0){\downarrow}\big\}. \tag{20}$$

[9] The classes given in parts (c) and (e) of Example 4 can be shown to be computably finitely thick using a technique similar to that used in the proof of Theorem 9 below (see Figure 3(b), specifically).

Then, $\mathcal{L}$ is both uniformly decidable and computably finitely thick,[10] but there is *no* effective numbering of $\mathcal{L}$ witnessing both properties simultaneously. In fact, no such numbering exists for any class containing $\mathcal{L}$.

The following result, due to Lange & Zeugmann, gives a sufficient condition for a class of languages to be **It**-identifiable.

Theorem 5 (Lange & Zeugmann [LZ96, Theorem 12]). For each $\mathcal{L}$, if there exists an effective numbering of $\mathcal{L}$ that is both uniformly decidable and computably finitely thick, then $\mathcal{L} \in \mathbf{It}$.[11]

The following result strengthens Theorem 5 (Lange & Zeugmann) just above.

Theorem 6. For each $\mathcal{L}$, if there exists an effective numbering of $\mathcal{L}$ that is both uniformly decidable and computably finitely thick, then $\mathcal{L} \in \mathbf{FrIt}$.

The proof of Theorem 6 relies on the following lemma.

Lemma 7 For each $\mathcal{L}$, if there exists an effective numbering of $\mathcal{L}$ that is both uniformly decidable and computably finitely thick, then there exists an effective numbering $(X'_j)_{j\in\mathbb{N}}$ of $\mathcal{L}$ satisfying (i) and (ii) below.

(i) $(X'_j)_{j\in\mathbb{N}}$ is uniformly decidable.
(ii) $(X'_j)_{j\in\mathbb{N}}$ satisfies the following *strong* form of computable finite thickness. There exists a computable function $f' : \mathbb{N} \to \mathbb{N}$ such that, for each x,

$$D_{f'(x)} = \{j \mid x \in X'_j\}. \tag{21}$$

Proof. Suppose that $\mathcal{L}$ satisfies the conditions of the lemma, as witnessed by $(X_j)_{j\in\mathbb{N}}$ and $f : \mathbb{N} \to \mathbb{N}$. The proof is straightforward in the case when $\mathcal{L}$ is finite. So, suppose that $\mathcal{L}$ is infinite. Construct $(X'_j)_{j\in\mathbb{N}}$ and $f' : \mathbb{N} \to \mathbb{N}$ by executing stages $s = 0, 1, ...$ successively as follows.

- Stage $s = 0$. If $\emptyset \in \mathcal{L}$, then list $\emptyset$ into $(X'_j)_{j\in\mathbb{N}}$.
- Stage $s = 2x + 1$. For each $j \in D_{f(x)}$ such that $\min X_j = x$, list X_j into $(X'_j)_{j\in\mathbb{N}}$.
- Stage $s = 2x + 2$. If no sets have yet been listed into $(X'_j)_{j\in\mathbb{N}}$, then let $j_{\max} = -1$; otherwise, let $j_{\max}$ be the largest index into $(X'_j)_{j\in\mathbb{N}}$ used thus far. Set $D_{f'(x)} = \{j \leq j_{\max} \mid x \in X'_j\}$.

Clearly, $(X'_j)_{j\in\mathbb{N}}$ is a numbering of $\mathcal{L}$. Furthermore, it is straightforward to show that $(X'_j)_{j\in\mathbb{N}}$ satisfies (i) in the statement of the lemma. To show that $(X'_j)_{j\in\mathbb{N}}$ satisfies (ii): let x be fixed. Note that any set listed into $(X'_j)_{j\in\mathbb{N}}$ subsequent to stage $2x + 2$ will have a minimum element larger than x. Thus, if $j_{\max}$ is as in stage $2x + 2$, then

$$(\forall j)[x \in X'_j \;\Rightarrow\; j \leq j_{\max}]. \tag{22}$$

[10] See footnote 9.

[11] In [LZ96], Theorem 12 is not stated exactly as Theorem 5 is stated here. However, based on the proof of this result, we believe that what is stated here is what is meant.

Clearly, then, f' satisfies (21). $\square$ (**Lemma 7**)

Proof of Theorem 6. Suppose that $\mathcal{L}$ satisfies the conditions of the theorem, as witnessed by $(X_j)_{j\in\mathbb{N}}$. Without loss of generality, suppose that $(X_j)_{j\in\mathbb{N}}$ satisfies the strong form of computable finite thickness of Lemma 7(ii). Let $f : \mathbb{N} \to \mathbb{N}$ be the function witnessing this strong from of computable finite thickness.

For each x and J, say that x *narrows* $J \Leftrightarrow$ by letting $J' = \{j \in J \mid x \in X_j\}$,

$$\emptyset \neq J' \subset J \;\wedge\; \{w \in (\textstyle\bigcap_{j\in J'} X_j) \mid w < x\} = \{w \in (\textstyle\bigcap_{j\in J} X_j) \mid w < x\}. \tag{23}$$

Let $(Y_k)_{k\in\mathbb{N}}$ be any Friedberg numbering.

An effective numbering $(Z_\ell)_{\ell\in\mathbb{N}}$ is constructed in Figure 2. The construction makes use of two procedures: survey and $\mathsf{descend}$. The procedure survey takes one argument: an element of $\mathbb{N}$. The procedure $\mathsf{descend}$ takes two arguments: a finite subset of $\mathbb{N}$, and an element of $\mathbb{N}$.

In conjunction with $(Z_\ell)_{\ell\in\mathbb{N}}$, a partial computable function ζ from $\mathcal{Fin}$ to $\mathbb{N}$ is constructed. It is clear from the construction that ζ is 1-1, i.e., $(\forall J, J' \in \mathcal{Fin})[\zeta(J)\!\downarrow = \zeta(J') \;\Rightarrow\; J = J']$.

The construction of $(Z_\ell)_{\ell\in\mathbb{N}}$ is also annotated with several facts. For the purpose of these facts, and for the claims below, the following notation is adopted, for each S, J, and x. (Where relevant, let J' and A be defined in the obvious way.)

- S is *listed* 0-*indirectly* by $\mathsf{descend}(J, x) \Leftrightarrow$
  - cond. (i) applies in $\mathsf{descend}(J, x)$, and S is of the form $A \cup (Y_k + x + 1)$;
  - cond. (ii) applies in $\mathsf{descend}(J, x)$, and S is of the form $A \cup \{x\} \cup (Y_k + x + 1)$; or
  - cond. (iii) applies in $\mathsf{descend}(J, x)$, and $S = (\bigcap_{j\in J'} X_j)$.
- S is *listed* $(n + 1)$-*indirectly* by $\mathsf{descend}(J, x) \Leftrightarrow$
  - cond. (i) or (ii) applies in $\mathsf{descend}(J, x)$, and S is listed n-indirectly by $\mathsf{descend}(J, x + 1)$; or
  - cond. (iii) applies in $\mathsf{descend}(J, x)$, and S is listed n-indirectly by either $\mathsf{descend}(J, x + 1)$ or $\mathsf{descend}(J', x + 1)$.
- S is *listed directly* by $\mathsf{descend}(J, x) \Leftrightarrow S$ is listed 0-indirectly by $\mathsf{descend}(J, x)$.
- S is *listed indirectly* by $\mathsf{descend}(J, x) \Leftrightarrow$ there exists n such that S is listed $(n + 1)$-indirectly by $\mathsf{descend}(J, x)$.
- S is *listed* by $\mathsf{descend}(J, x) \Leftrightarrow S$ is listed directly or indirectly by $\mathsf{descend}(J, x)$.
- S is *listed directly* by $\mathsf{survey}(x) \Leftrightarrow$
  - cond. (a) applies in $\mathsf{survey}(x)$, and S is of the form $\{x\} \cup (Y_k + x + 1)$; or
  - cond. (b) applies in $\mathsf{survey}(x)$, and $S = (\bigcap_{j\in D_{f(x)}} X_j)$.
- S is *listed indirectly* by $\mathsf{survey}(x) \Leftrightarrow$ cond. (b) applies in $\mathsf{survey}(x)$, and S is listed by $\mathsf{descend}(D_{f(x)}, x + 1)$.
- S is *listed* by $\mathsf{survey}(x) \Leftrightarrow S$ is listed directly or indirectly by $\mathsf{survey}(x)$.

List $\emptyset$ into $(Z_\ell)_{\ell\in\mathbb{N}}$ exactly once. Then, for each x, run $\mathsf{survey}(x)$.

$\mathsf{survey}(x)$: Act according the following conditions.

- Cond. (a) $[D_{f(x)} = \emptyset \;\vee\; \min(\bigcap_{j\in D_{f(x)}} X_j) \neq x]$. For each k, list $\{x\} \cup (Y_k + x + 1)$ into $(Z_\ell)_{\ell\in\mathbb{N}}$ exactly once.

  (Fact: For each set S listed, $\min S = x$.)
- Cond. (b) $[D_{f(x)} \neq \emptyset \;\wedge\; \min(\bigcap_{j\in D_{f(x)}} X_j) = x]$. List $(\bigcap_{j\in D_{f(x)}} X_j)$ into $(Z_\ell)_{\ell\in\mathbb{N}}$ exactly once, set $\zeta(D_{f(x)})$ to the index used to list this set, and run $\mathsf{descend}(D_{f(x)}, x+1)$.

  (Fact: For each set S listed, $\min S = x$.)

$\mathsf{descend}(J, x)$: Let J', x_0, and A be such that:

$$J' = \{j \in J \mid x \in X_j\}; \quad x_0 = \min(\textstyle\bigcap_{j\in J} X_j); \quad A = \{w \in (\bigcap_{j\in J} X_j) \mid w < x\}.$$

Act according to the following conditions.

- Cond. (i) $\big[J' = J\big]$. For each k, list $A \cup (Y_k + x + 1)$ into $(Z_\ell)_{\ell\in\mathbb{N}}$ exactly once, and run $\mathsf{descend}(J, x+1)$.

  (Fact: For each set S listed, $\{w \in S \mid w < x\} = A$. For each set S listed directly, $x \in (\bigcap_{j\in J} X_j) - S$.)
- Cond. (ii) $\big[x \leq x_0 \;\vee\; [J' \subset J \;\wedge\; x \text{ does } \textit{not} \text{ narrow } J]\big]$. For each k, list $A \cup \{x\} \cup (Y_k + x + 1)$ into $(Z_\ell)_{\ell\in\mathbb{N}}$ exactly once, and run $\mathsf{descend}(J, x+1)$.

  (Fact: For each set S listed, $\{w \in S \mid w < x\} = A$. For each set S listed directly, $x \in S - (\bigcap_{j\in J} X_j)$.)
- Cond. (iii) $\big[x > x_0 \;\wedge\; x \text{ narrows } J\big]$. List $(\bigcap_{j\in J'} X_j)$ into $(Z_\ell)_{\ell\in\mathbb{N}}$ exactly once, set $\zeta(J')$ to the index used to list this set, and run both $\mathsf{descend}(J, x+1)$ and $\mathsf{descend}(J', x+1)$.

  (Fact: For each set S listed, $\{w \in S \mid w < x\} = A$. For each set S listed directly (i.e., $S = (\bigcap_{j\in J'} X_j)$), $x \in S - (\bigcap_{j\in J} X_j)$.)

Fig. 2. The construction of $(Z_\ell)_{\ell\in\mathbb{N}}$ in the proof of Theorem 6.

The facts annotating the construction of $(Z_\ell)_{\ell\in\mathbb{N}}$ are shown by a straightforward induction, the details of which are omitted.

That $(Z_\ell)_{\ell\in\mathbb{N}}$ is a Friedberg numbering is established by Claims 6.3 and 6.5 below.

Claim 6.1. (a)-(e) below.

(a) For each x, $\emptyset$ is *not* listed by $\mathsf{survey}(x)$.
(b) For each x_0 and x_1, if $x_0 \neq x_1$, then

$$\{S \mid S \text{ is listed by } \mathsf{survey}(x_0)\} \cap \{S \mid S \text{ is listed by } \mathsf{survey}(x_1)\} = \emptyset. \quad (24)$$

(c) Suppose that x is such that cond. (b) applies in $\mathsf{survey}(x)$. Then, $(\bigcap_{j\in D_{f(x)}} X_j)$ is *not* listed by $\mathsf{descend}(D_{f(x)}, x+1)$.

(d) Suppose that J and x are such that cond. (i) or (ii) applies in $\mathsf{descend}(J,x)$. Then, for each set S listed directly by $\mathsf{descend}(J,x)$, S is *not* listed by $\mathsf{descend}(J,x+1)$.
(e) Suppose that J and x are such that cond. (iii) applies in $\mathsf{descend}(J,x)$. Let $J' = \{j \in J \mid x \in X_j\}$. Then, (i)-(iii) below.
  (i) $(\bigcap_{j\in J'} X_j)$ is *not* listed by $\mathsf{descend}(J,x+1)$.
  (ii) $(\bigcap_{j\in J'} X_j)$ is *not* listed by $\mathsf{descend}(J',x+1)$.
  (iii) $\{S \mid S \text{ is listed by } \mathsf{descend}(J,x+1)\}$
  $\cap\,\{S \mid S \text{ is listed by } \mathsf{descend}(J',x+1)\} = \emptyset$.

Proof of Claim. Straightforward, given the facts annotating the construction of $(Z_\ell)_{\ell\in\mathbb{N}}$. □ (**Claim 6.1**)

Claim 6.2. For each n, S, J, and x, if S is listed n-indirectly by $\mathsf{descend}(J,x)$, then S is listed *at most once* by $\mathsf{descend}(J,x)$.

Proof of Claim. The proof is by induction on n. For the case when $n=0$, suppose that S, J, and x are such that S is listed directly by $\mathsf{descend}(J,x)$. Then, it follows from parts (d), (e)(i), and (e)(ii) of Claim 6.1 that S is listed at most once by $\mathsf{descend}(J,x)$.

Next, suppose inductively that the claim holds for n, and that S, J, and x are such that S is listed $(n+1)$-indirectly by $\mathsf{descend}(J,x)$. Thus, cond. (iii) applies in $\mathsf{descend}(J,x)$, and S is listed n-indirectly by either $\mathsf{descend}(J,x+1)$ or $\mathsf{descend}(J',x+1)$. By the induction hypothesis, S is listed at most once by $\mathsf{descend}(J,x+1)$, and at most once by $\mathsf{descend}(J',x+1)$. Clearly, then, by Claim 6.1(e)(iii), S is listed at most once by $\mathsf{descend}(J,x)$. □ (**Claim 6.2**)

Claim 6.3. $(Z_\ell)_{\ell\in\mathbb{N}}$ is 1-1.

Proof of Claim. To show the claim, it suffices to show that, for each $S \in \mathcal{CE}$, S is listed *at most once* during the construction of $(Z_\ell)_{\ell\in\mathbb{N}}$. So, let $S \in \mathcal{CE}$ be fixed. If $S=\emptyset$, then it follows from Claim 6.1(a) that S is listed at most once. So, suppose that $S \neq \emptyset$. By Claim 6.1(b), it suffices to show that, for each x, S is listed at most once by $\mathsf{survey}(x)$. So, suppose that x is such that S is listed *at least once* by $\mathsf{survey}(x)$. If cond. (a) applies in $\mathsf{survey}(x)$, then S list clearly listed at most once by $\mathsf{survey}(x)$. So, suppose cond. (b) applies in $\mathsf{survey}(x)$. If S is listed directly by $\mathsf{survey}(x)$, then it follows from Claim 6.1(c) that S is listed at most once by $\mathsf{survey}(x)$. So, suppose that S is listed indirectly by $\mathsf{survey}(x)$, i.e., S is listed by $\mathsf{descend}(D_{f(x)},x+1)$. Let n be such that S is listed n-indirectly $\mathsf{descend}(D_{f(x)},x+1)$. Then, by Claim 6.2, S is listed at most once by $\mathsf{descend}(D_{f(x)},x+1)$. □ (**Claim 6.3**)

Claim 6.4. Let $S \in \mathcal{CE} - \{\emptyset\}$ be fixed, let $x_0 = \min S$, and let $J_0 = D_{f(x_0)}$. Suppose that $[J_0 \neq \emptyset \ \wedge\ \min(\bigcap_{j\in J_0} X_j) = x_0]$. Then, there exists an n and two sequences, $x_1,...,x_n$ and $J_1,...,J_n$, satisfying (a)-(g) below.

(a) For each $i<n$, $x_{i+1} > x_i$.
(b) For each $i<n$, x_{i+1} narrows J_i.
(c) For each $i<n$, $J_{i+1} = \{j \in J_i \mid x_{i+1} \in X_j\}$.

(d) For each $i \le n$, $\{w \in (\bigcap_{j\in J_i} X_j) \mid w \le x_i\} = \{w \in S \mid w \le x_i\}$.
(e) For each $i \le n$, $(\bigcap_{j\in J_i} X_j)$ is listed into $(Z_\ell)_{\ell\in\mathbb{N}}$.
(f) For each $i \le n$, $\mathsf{descend}(J_i, x_i + 1)$ is run.
(g) One of (i)-(iii) below holds.
  (i) $(\bigcap_{j\in J_n} X_j) = S$.
  (ii) There exists $x' > x_n$ such that $x' \in (\bigcap_{j\in J_n} X_j) - S$.
  (iii) There exists $x' > x_n$ such that $x' \in S - (\bigcap_{j\in J_n} X_j)$ and x' does *not* narrow J_n.

Proof of Claim. Suppose that S, x_0, and J_0 are as stated. We show that:

- parts (a)-(f) hold for 0;
- if parts (a)-(f) hold for n, and part (g) *fails* for n, then parts (a)-(f) hold for $n + 1$.

By part (b), this process cannot continue indefinitely, i.e., there must exist *some* n for which parts (a)-(g) hold.

For the case when $n = 0$, clearly, parts (a)-(d) hold. Furthermore, when $\mathsf{survey}(x_0)$ is run, $(\bigcap_{j\in J_0} X_j)$ is listed into $(Z_\ell)_{\ell\in\mathbb{N}}$, and $\mathsf{descend}(J_0, x_0 + 1)$ is run.

Next, suppose inductively that parts (a)-(f) hold for n, and that part (g) *fails* for n, i.e., ($\bar{\text{i}}$)-($\overline{\text{iii}}$) below.

($\bar{\text{i}}$) $(\bigcap_{j\in J_n} X_j) \neq S$.
($\overline{\text{ii}}$) For each $x' > x_n$, if $x' \in (\bigcap_{j\in J_n} X_j) - S$, then $x' \in S$.
($\overline{\text{iii}}$) For each $x' > x_n$, if $x' \in S - (\bigcap_{j\in J_n} X_j)$, then x' narrows J_n.

By ($\bar{\text{i}}$), there exists a *least* $x_{n+1} > x_n$ such that $x_{n+1} \in (\bigcap_{j\in J_n} X_j) \bigtriangleup S$. If $x_{n+1} \in (\bigcap_{j\in J_n} X_j) - S$, then, by ($\overline{\text{ii}}$), $x_{n+1} \in S$ — a contradiction. Thus, it must be the case that $x_{n+1} \in S - (\bigcap_{j\in J_n} X_j)$. By the choice of x_{n+1}, $x_{n+1} > x_n$. By ($\overline{\text{iii}}$), x_{n+1} narrows J_n. Let $J_{n+1} = \{j \in J_n \mid x_{n+1} \in X_j\}$.

To show part (d) for $n + 1$: by part (d) for n,

$$\{w \in (\textstyle\bigcap_{j\in J_n} X_j) \mid w \le x_n\} = \{w \in S \mid w \le x_n\}. \tag{25}$$

Since x_{n+1} is least such that $x_{n+1} > x_n$ and $x_{n+1} \in (\bigcap_{j\in J_n} X_j) \bigtriangleup S$,

$$\{w \in (\textstyle\bigcap_{j\in J_n} X_j) \mid x_n < w < x_{n+1}\} = \{w \in S \mid x_n < w < x_{n+1}\}. \tag{26}$$

Since x_{n+1} narrows J_n,

$$\{w \in (\textstyle\bigcap_{j\in J_{n+1}} X_j) \mid w < x_{n+1}\} = \{w \in (\textstyle\bigcap_{j\in J_n} X_j) \mid w < x_{n+1}\}. \tag{27}$$

Clearly, x_{n+1} is in both $(\bigcap_{j\in J_{n+1}} X_j)$ and S. Combining this fact with (25)-(27) yields:

$$\{w \in (\textstyle\bigcap_{j\in J_{n+1}} X_j) \mid w \le x_{n+1}\} = \{w \in S \mid w \le x_{n+1}\}. \tag{28}$$

To show parts (e) and (f) for $n+1$: by part (f) for n, $\mathsf{descend}(J, x_n+1)$ is run. Clearly, then, $\mathsf{descend}(J, x_{n+1})$ is run, and cond. (iii) applies in $\mathsf{descend}(J_n, x_{n+1})$.

Thus, $(\bigcap_{j\in J_{n+1}} X_j)$ is listed into $(Z_\ell)_{\ell\in\mathbb{N}}$, and $\mathsf{descend}(J_{n+1}, x_{n+1}+1)$ is run. $\square$ (**Claim 6.4**)

Claim 6.5. $(Z_\ell)_{\ell\in\mathbb{N}}$ is a numbering of $\mathcal{CE}$.

Proof of Claim. Let $S \in \mathcal{CE}$ be fixed. If $S \in \emptyset$, then S is listed into $(Z_\ell)_{\ell\in\mathbb{N}}$ at the beginning of the construction of $(Z_\ell)_{\ell\in\mathbb{N}}$. So suppose that $S \neq \emptyset$. Let $x_0 = \min S$ and let $J_0 = D_{f(x_0)}$. If $[J_0 = \emptyset \ \vee \ \min(\bigcap_{j\in J_0} X_j) \neq x_0]$, then S is clearly listed into $(Z_\ell)_{\ell\in\mathbb{N}}$ when $\mathsf{survey}(x_0)$ is run. So suppose that $[J_0 \neq \emptyset \ \wedge \ \min(\bigcap_{j\in J_0} X_j) = x_0]$. Then, there exist n, $x_1, ..., x_n$, and $J_1, ..., J_n$ as in Claim 6.4 for S. Consider the following cases based on part (g) of Claim 6.4.

Case [part (g)(i) of Claim 6.4 holds for S]. Then, by part (e) for n of Claim 6.4, S is listed into $(Z_\ell)_{\ell\in\mathbb{N}}$.

Case [part (g)(ii) of Claim 6.4 holds for S]. Let x' be as asserted to exist by part (g)(ii) of Claim 6.4 for S. By part (f) for n, $\mathsf{descend}(J, x_n + 1)$ is run. Clearly, then, $\mathsf{descend}(J, x')$ is run, cond. (i) applies in $\mathsf{descend}(J_n, x')$, and S is listed into $(Z_\ell)_{\ell\in\mathbb{N}}$.

Case [part (g)(iii) of Claim 6.4 holds for S]. Let x' be as asserted to exist by part (g)(iii) of Claim 6.4 for S. By part (f) for n, $\mathsf{descend}(J, x_n + 1)$ is run. Clearly, then, $\mathsf{descend}(J, x')$ is run, cond. (ii) applies in $\mathsf{descend}(J_n, x')$, and S is listed into $(Z_\ell)_{\ell\in\mathbb{N}}$. $\square$ (**Claim 6.5**)

To complete the proof of the theorem, it suffices to show that $\mathcal{L}$ can be **It**-identified using $(Z_\ell)_{\ell\in\mathbb{N}}$ as the hypothesis space. For ease of presentation, suppose that $Z_0 = \emptyset$. Let $M \in \mathcal{PC}_{1,1}$ be such that, for each $\ell > 0$ and x:

$$M(0, \#) = 0; \tag{29}$$

$$M(0, x) = \begin{cases} \zeta(D_{f(x)}), \text{ if } \zeta(D_{f(x)})\downarrow; \\ \uparrow, \qquad\quad \text{otherwise;} \end{cases} \tag{30}$$

$$M(\ell, \#) = \ell; \tag{31}$$

$$M(\ell, x) = \begin{cases} \zeta(J'), \text{ where } J \text{ and } J' \text{ are such that } \zeta(J) = \ell \text{ and} \\ \qquad\quad J' = \{j \in J \mid x \in X_j\}, \text{ if such a } J \text{ exists} \\ \qquad\quad \text{and } \zeta(J')\downarrow; \\ \uparrow, \quad \text{otherwise.} \end{cases} \tag{32}$$

Clearly, $(M, 0, (Z_\ell)_{\ell\in\mathbb{N}})$ identifies $\emptyset$. Furthermore, it is straightforward to show that, for each τ,

$$(\forall \sigma \subseteq \tau)[\text{content}(\sigma) \neq \emptyset \ \Rightarrow \ \zeta(\{j \mid \text{content}(\sigma) \subseteq X_j\})\downarrow] \ \Rightarrow \\ M_0^*(\tau) = \zeta(\{j \mid \text{content}(\tau) \subseteq X_j\}), \tag{33}$$

and that, for each finite J,

$$\zeta(J)\downarrow \ \Rightarrow \ Z_{\zeta(J)} = (\textstyle\bigcap_{j\in J} X_j). \tag{34}$$

Thus, if it can be shown that

$$\big(\forall \sigma \in \text{Seq}(\mathcal{L})\big)[\text{content}(\sigma) \neq \emptyset \ \Rightarrow \ \zeta(\{j \mid \text{content}(\sigma) \subseteq X_j\})\downarrow], \tag{35}$$

then the fact that $(M, 0, (Z_\ell)_{\ell\in\mathbb{N}})$ exhibits the correct behavior follows easily. Claim 6.8 below establishes (35).

Claim 6.6. Suppose that J is finite and non-empty. Let $x_0 = \min(\bigcap_{j\in J} X_j)$. Then, (a) and (b) below.

(a) $D_{f(x_0)} \supseteq J$.
(b) $\min(\bigcap_{j\in D_{f(x_0)}} X_j) = x_0$.

Proof of Claim. Suppose that J and x_0 are as stated. Since $x_0 \in (\bigcap_{j\in J} X_j)$, it is immediate that $J \subseteq \{j \mid x_0 \in X_j\}$. Furthermore, since f witnesses that $(X_j)_{j\in\mathbb{N}}$ satisfies the strong form of computable finite thickness of Lemma 7(ii), $D_{f(x_0)} = \{j \mid x_0 \in X_j\}$. Thus, $D_{f(x_0)} \supseteq J$.

To show part (b): clearly, $x_0 \in (\bigcap_{j\in D_{f(x_0)}} X_j)$. Furthermore, by part (a), $(\bigcap_{j\in D_{f(x_0)}} X_j) \subseteq (\bigcap_{j\in J} X_j)$. Thus, since there is nothing smaller than x_0 in $(\bigcap_{j\in J} X_j)$, there can be nothing smaller than x_0 in $(\bigcap_{j\in D_{f(x_0)}} X_j)$. □ (**Claim 6.6**)

Claim 6.7. Suppose that J is finite and non-empty. let $x_0 = (\min \bigcap_{j\in J} X_j)$, and let $J_0 = D_{f(x_0)}$. Then, there exists an n and two sequences, $x_1, ..., x_n$ and $J_1, ..., J_n$, satisfying (a)-(h) below.

(a) For each $i < n$, $x_{i+1} > x_i$.
(b) For each $i < n$, x_{i+1} narrows J_i.
(c) For each $i < n$, $J_{i+1} = \{j \in J_i \mid x_{i+1} \in X_j\}$.
(d) For each $i \le n$, $\{w \in (\bigcap_{j\in J_i} X_j) \mid w \le x_i\} = \{w \in (\bigcap_{j\in J} X_j) \mid w \le x_i\}$.
(e) For each $i \le n$, $J_i \supseteq J$.
(f) For each $i \le n$, $\zeta(J_i)\!\downarrow$.
(g) For each $i \le n$, $\mathsf{descend}(J_i, x_i + 1)$ is run.
(h) There is *no* $x' > x_n$ such that x' narrows J_n and $\{j \in J_n \mid x' \in X_j\} \supseteq J$.

Proof of Claim. Suppose that J, x_0, and J_0 are as stated. Much like in the proof of Claim 6.4, we show that:

– parts (a)-(g) hold for 0;
– if parts (a)-(g) hold for n, and part (h) *fails* for n, then parts (a)-(g) hold for $n + 1$.

Again, by part (b), this process cannot continue indefinitely, i.e., there must exist *some* n for which parts (a)-(h) hold.

For the case when $n = 0$, clearly, parts (a)-(d) hold. Note that, by Claim 6.6,

$$J_0 = D_{f(x_0)} \supseteq J \ (\neq \emptyset) \ \wedge \ \min(\bigcap_{j\in D_{f(x_0)}} X_j) = x_0. \tag{36}$$

It follows that $\zeta(J_0)$ is set when $\mathsf{survey}(x_0)$ is run, and that $\mathsf{descend}(J_0, x_0 + 1)$ is run.

Next, suppose inductively that parts (a)-(g) hold for n, and that part (h) *fails* for n. Let x_{n+1} be *least* such that $x_{n+1} > x_n$, x_{n+1} narrows J_n, and $\{j \in J_n \mid x_{n+1} \in X_j\} \supseteq J$. Let $J_{n+1} = \{j \in J_n \mid x_{n+1} \in X_j\}$.

To show part (d) for $n+1$: by way of contradiction, let x' be *least* such that $x' \in (\bigcap_{j\in J_{n+1}} X_j) \triangle (\bigcap_{j\in J} X_j)$. By part (d) for n,

$$\{w \in (\textstyle\bigcap_{j\in J_n} X_j) \mid w \le x_n\} = \{w \in (\bigcap_{j\in J} X_j) \mid w \le x_n\}. \tag{37}$$

Since x_{n+1} narrows J_n,

$$\{w \in (\textstyle\bigcap_{j\in J_{n+1}} X_j) \mid w < x_{n+1}\} = \{w \in (\bigcap_{j\in J_n} X_j) \mid w < x_{n+1}\}. \tag{38}$$

Combining (37), (38), and the fact that $x_n < x_{n+1}$ yields:

$$\{w \in (\textstyle\bigcap_{j\in J_{n+1}} X_j) \mid w \le x_n\} = \{w \in (\bigcap_{j\in J} X_j) \mid w \le x_n\}. \tag{39}$$

Thus, it must be the case that $x' > x_n$. Since $J_{n+1} \supseteq J$, $(\bigcap_{j\in J_{n+1}} X_j) \subseteq (\bigcap_{j\in J} X_j)$. Thus, $x' \in (\bigcap_{j\in J} X_j) - (\bigcap_{j\in J_{n+1}} X_j)$. Clearly, $x_{n+1} \in (\bigcap_{j\in J_{n+1}} X_j)$. Thus, it must be the case that $x' < x_{n+1}$. Given these conditions, it is easily seen that x' narrows J_n. Furthermore, since $J_n \supseteq J$,

$$\{j \in J_n \mid x' \in X_j\} \supseteq \{j \in J \mid x' \in X_j\} = J. \tag{40}$$

But since $x' < x_{n+1}$, the existence of x' contradicts the minimality of x_{n+1}.

To show parts (f) and (g) for $n+1$: by part (g) for n, $\mathsf{descend}(J_n, x_n+1)$ is run. Clearly, then, $\mathsf{descend}(J, x_{n+1})$ is run, and cond. (iii) applies in $\mathsf{descend}(J_n, x_{n+1})$. Thus, $\zeta(J_{n+1})$ is set, and, and $\mathsf{descend}(J_{n+1}, x_{n+1}+1)$ is run. $\square$ (**Claim 6.7**)

Claim 6.8. Suppose that $\sigma \in \mathrm{Seq}(\mathcal{L})$ is such that $\mathrm{content}(\sigma) \neq \emptyset$. Let $J = \{j \mid \mathrm{content}(\sigma) \subseteq X_j\}$. Then, $\zeta(J)\!\downarrow$.

Proof of Claim. Suppose that σ and J are as stated. Since $\sigma \in \mathrm{Seq}(\mathcal{L})$, $J \neq \emptyset$. Thus, there exist n, $x_1, ..., x_n$, and $J_1, ..., J_n$ as in Claim 6.7 for J. By part (f) for n, $\zeta(J_n)\!\downarrow$. Thus, to show the claim, it suffices to show that $J_n = J$.

By part (e) for n of Claim 6.7, $J_n \supseteq J$. So, by way of contradiction, suppose that $J_n \supset J$. Thus, there exists $j \in J_n$ such that $\mathrm{content}(\sigma) \not\subseteq X_j$. It follows that $(\bigcap_{j\in J_n} X_j) \subset (\bigcap_{j\in J} X_j)$. Let x' be *least* such that $x' \in (\bigcap_{j\in J} X_j) - (\bigcap_{j\in J_n} X_j)$. Thus,

$$\{w \in (\textstyle\bigcap_{j\in J_n} X_j) \mid w < x'\} = \{w \in (\bigcap_{j\in J} X_j) \mid w < x'\}. \tag{41}$$

By part (d) for n of Claim 6.7,

$$\{w \in (\textstyle\bigcap_{j\in J_n} X_j) \mid w \le x_n\} = \{w \in (\bigcap_{j\in J} X_j) \mid w \le x_n\}. \tag{42}$$

Thus, it must be the case that $x' > x_n$. Clearly, x' narrows J_n. Furthermore, since $J_n \supseteq J$,

$$\{j \in J_n \mid x' \in X_j\} \supseteq \{j \in J \mid x' \in X_j\} = J. \tag{43}$$

But then the existence of x' contradicts part (h) for n of Claim 6.7.

$\square$ (**Claim 6.8**)

$\square$ (**Theorem 6**)

Recall that $\mathcal{P}at^{\Sigma}$ is the collection of all pattern languages over Σ, where Σ is an arbitrary alphabet. It is straightforward to show that, for each Σ, there exists an effective numbering of $\mathcal{P}at^{\Sigma}$ that is both uniformly decidable and computably finitely thick. Thus, one has the following corollary of Theorem 6.

(a) For each i, execute stage 0 below.

- Stage 0. For each i, include $(\mathbb{N}+i)$ and $\{i\}$ in $\mathcal{L}_0$. Go to stage 1.
- Stage 1. Let (M,p) be the ith pair in $((M,p)_i)_{i\in\mathbb{N}}$. Search for a $k \geq i$ such that

$$M_p^*\big((i\cdot\cdots\cdot k)\cdot(k+1)\big)\downarrow \;=\; M_p^*\big((i\cdot\cdots\cdot k)\cdot k\big) \;=\; M_p^*(i\cdot\cdots\cdot k).$$

If such a k is found, then include $\{i,...,k\}$ and $\{i,...,k+1\}$ in $\mathcal{L}_0$, and terminate the construction (for i). If *no* such k is found, then search indefinitely.

(b) For each i, execute stage 0 below.

- Stage 0. Set $X_{\mathrm{start}(i)} = \mathbb{N}+i$, and, for each $j \in \{\mathrm{start}(i)+1,...,\mathrm{start}(i+1)-1\}$, set $X_j = \{i\}$. Go to stage 1.
- Stage 1. In a dovetailing manner, monitor and act according to the following conditions.
  - Cond. [in the construction of $\mathcal{L}_0$ above, a k is found for i]. Set $X_{\mathrm{start}(i)+2} = \{i,...,k\}$ and $X_{\mathrm{start}(i)+3} = \{i,...,k+1\}$.
  - Cond. [$i \in X_j,\;$ where $j < \mathrm{start}(i)$]. Set $X_{\mathrm{start}(i)+j+4} = X_j$.

Fig. 3. (a) The construction of $\mathcal{L}_0$ in the proof of Theorem 9. **(b)** The construction of $(X_j)_{j\in\mathbb{N}}$ in the proof of Theorem 9. The function start is defined in (44).

Corollary 8 (of Theorem 6) For each alphabet Σ, $\mathcal{P}at^{\Sigma}$ is **FrIt**-identifiable.

The proof of Theorem 9 below exhibits a class of languages $\mathcal{L}_0$ that is uniformly decidable and computably finitely thick, but $\mathcal{L}_0 \notin \mathbf{It}$. Thus, one could not arrive at the conclusion of Theorem 6 if one were to merely require that: there exists a uniformly decidable effective numbering of $\mathcal{L}$, and a possibly *distinct* computably finitely thick effective numbering of $\mathcal{L}$.

Theorem 9. There exists a class of languages $\mathcal{L}_0$ that is uniformly decidable and computably finitely thick, but $\mathcal{L}_0 \notin \mathbf{It}$.

Proof. Let $((M,p)_i)_{i\in\mathbb{N}}$ be an algorithmic enumeration of all pairs of type $\mathcal{PC}_{1,1} \times \mathbb{N}$. Let start : $\mathbb{N} \to \mathbb{N}$ be such that, for each i,

$$\mathrm{start}(i) = 2^{i+1} - 4. \tag{44}$$

Note that, for each i,

$$\mathrm{start}(i+1) - \mathrm{start}(i) = \mathrm{start}(i) + 4. \tag{45}$$

The class $\mathcal{L}_0$ is constructed in Figure 3(a). An effective numbering $(X_j)_{j\in\mathbb{N}}$, which is easily seen to be of $\mathcal{L}_0$, is constructed in Figure 3(b). This effective numbering $(X_j)_{j\in\mathbb{N}}$ is used to show that $\mathcal{L}_0$ is computably finitely thick. It is straightforward to construct an effective numbering witnessing that $\mathcal{L}_0$ is uniformly decidable.

The following are easily verifiable from the construction of $(X_j)_{j\in\mathbb{N}}$.

– For each $L \in \mathcal{L}_0$ and $i \in L$, there exists $j < \text{start}(i) + 4$ such that $X_j = L$.
– For each i and $j < \text{start}(i)$, if $i \in X_j$, then there exists $j' \in \{\text{start}(i) + 4, ..., \text{start}(i+1) - 1\}$ such that $X_{j'} = X_j$.
– For each $j \in \{\text{start}(i), ..., \text{start}(i+1) - 1\}$, $i \in X_j$.

Given these facts, if one lets $f : \mathbb{N} \to \mathbb{N}$ be such that, for each i,

$$D_{f(i)} = \{\text{start}(i), ..., \text{start}(i+1) - 1\}, \tag{46}$$

then f clearly witnesses that $(X_j)_{j\in\mathbb{N}}$ is computably finitely thick.

It remains to show that $\mathcal{L}_0 \notin \textbf{It}$. By way of contradiction, suppose otherwise, as witnessed by $(M, p, (X_j)_{j\in\mathbb{N}})$, where (M, p) is the ith pair in $((M, p)_i)_{i\in\mathbb{N}}$. Then, since $(\mathbb{N} + i) \in \mathcal{L}_0$, there exists a *least* $k_0 \geq i$ such that

$$(\forall k \geq k_0)\big[M_p^*\big((i \cdot \cdots \cdot k_0) \cdot k\big)\!\downarrow = M_p^*(i \cdot \cdots \cdot k_0)\big]. \tag{47}$$

It follows that some $k_1 \geq k_0$ is discovered in stage 1 of the construction of $\mathcal{L}_0$ (for i), and that $\{i, ..., k_1\}$ and $\{i, ..., k_1 + 1\}$ are in $\mathcal{L}_0$. Note that

$$t_0 = (i \cdot \cdots \cdot k_1) \cdot k_1^\infty; \tag{48}$$
$$t_1 = (i \cdot \cdots \cdot k_1) \cdot (k_1 + 1)^\infty \tag{49}$$

are, respectively, texts for $\{i, ..., k_1\}$ and $\{i, ..., k_1 + 1\}$. But, by (47), $(M, p, (X_j)_{j\in\mathbb{N}})$ can*not* distinguish the languages contained in these texts — a contradiction. □ (**Theorem 9**)

4 Friedberg, Injective, and Extensional Iterative Learning

This section examines the Friedberg, injective, and extensional iterative learning models (**FrIt**, **InjIt**, and **ExtIt**, respectively). Recall that, for each $(M, p, (X_j)_{j\in\mathbb{N}}) \in \mathcal{PC}_{1,1} \times \mathbb{N} \times \mathcal{EN}$ and $\mathcal{L}$:

– $(M, p, (X_j)_{j\in\mathbb{N}})$ **FrIt**-identifies $\mathcal{L}$ $\Leftrightarrow$ $(M, p, (X_j)_{j\in\mathbb{N}})$ **It**-identifies $\mathcal{L}$, and $(X_j)_{j\in\mathbb{N}}$ is a Friedberg numbering;
– $(M, p, (X_j)_{j\in\mathbb{N}})$ **InjIt**-identifies $\mathcal{L}$ $\Leftrightarrow$ $(M, p, (X_j)_{j\in\mathbb{N}})$ **It**-identifies $\mathcal{L}$, and $(X_j)_{j\in\mathbb{N}}$ is 1-1;
– $(M, p, (X_j)_{j\in\mathbb{N}})$ **ExtIt**-identifies $\mathcal{L}$ $\Leftrightarrow$ $(M, p, (X_j)_{j\in\mathbb{N}})$ **It**-identifies $\mathcal{L}$, and, for each $\sigma_0, \sigma_1 \in \text{Seq}(\mathcal{L}) - \{\lambda\}$,

$$[X_{M_p^*(\sigma_0^-)} = X_{M_p^*(\sigma_1^-)} \wedge \text{last}(\sigma_0) = \text{last}(\sigma_1)] \;\Rightarrow\; X_{M_p^*(\sigma_0)} = X_{M_p^*(\sigma_1)}. \tag{50}$$

In terms of the classes of languages learnable by these models and by **It**, they are clearly related as follows.

$$\textbf{FrIt} \subseteq \textbf{InjIt} \subseteq \textbf{ExtIt} \subseteq \textbf{It}. \tag{51}$$

In this section we establish that **InjIt** $\not\subseteq$ **FrIt** (Proposition 10), and that **ExtIt** $\not\subseteq$ **InjIt** (Theorem 11). The fact that **It** $\not\subseteq$ **ExtIt** is due to Jain (Theorem 12).

Proposition 10 just below establishes that **InjIt** $\not\subseteq$ **FrIt**.

Proposition 10 $\mathbf{InjIt} \not\subseteq \mathbf{FrIt}$.

Proof. Recall that $\mathcal{F}in$ is the collection of all finite subsets of $\mathbb{N}$. Jain & Stephan observed that $\mathcal{F}in \notin \mathbf{FrIt}$ [JS08, Remark 28]. However, it is easily seen that $\mathcal{F}in \in \mathbf{InjIt}$. $\square$ (**Proposition 10**)

Theorem 11 just below establishes that $\mathbf{ExtIt} \not\subseteq \mathbf{InjIt}$.

Theorem 11. $\mathbf{ExtIt} \not\subseteq \mathbf{InjIt}$.

Proof. Let $\mathcal{L}_1$ be as follows.

$$\begin{aligned}\mathcal{L}_1 = \quad & \big\{2\mathbb{N}\big\} \\ \cup & \big\{\{0,2,...,2e\} \cup \{2e+1\} \cup 2W_e, \\ & \quad \{0,2,...,2e\} \cup \{2e+1\} \cup 2X \quad \mid e \in \mathbb{N} \ \wedge \ \text{(a)-(c) below}\big\}. \\ & \quad \text{(a) } (\forall e' \in X)[W_{e'} = X]. \\ & \quad \text{(b) If } W_e = X, \text{ then } W_e \text{ and } X \text{ are finite.} \\ & \quad \text{(c) If } W_e \neq X, \text{ then } 2W_e \subseteq \{0,2,...,2e\} \text{ and } X \text{ is infinite.}\end{aligned} \tag{52}$$

First, we show that $\mathcal{L}_1 \in \mathbf{ExtIt}$. Let $f : \mathbb{N}^2 \to \mathbb{N}$ be a computable, 1-1 function such that, for each e and e',

$$W_{f(0,\ 0)} = 2\mathbb{N}; \tag{53}$$
$$W_{f(e+1,\ 0)} = \{0,2,...,2e\} \cup \{2e+1\} \cup 2W_e; \tag{54}$$
$$W_{f(e+1,\ e'+1)} = \{0,2,...,2e\} \cup \{2e+1\} \cup 2W_{e'}. \tag{55}$$

(It does not matter what $W_{f(\cdot,\cdot)}$ is for the remaining pairs.) For each $e, e' \in \mathbb{N}$ and $x \in \mathbb{N}_{\#}$, let $\tilde{M}$ be as follows. We use "unchanged" as a synonym for $\tilde{M}$'s first argument.

$$\tilde{M}\big(f(0,\ 0),x\big) = \begin{cases} f(e+1,0), & \text{where } x = 2e+1; \\ \text{unchanged}, & \text{otherwise;}\end{cases} \tag{56}$$
$$\tilde{M}\big(f(e+1,0),x\big) = \begin{cases} f(e+1,e'+1), & \text{where } x = 2e' \\ & \quad \text{and } 2e' > 2e+1; \\ \text{unchanged}, & \text{otherwise;}\end{cases} \tag{57}$$
$$\tilde{M}\big(f(e+1,e'+1),x\big) = \text{unchanged}. \tag{58}$$

It is straightforward to show that $(\tilde{M}, f(0,0))$ **It**-identifies $\mathcal{L}_1$. That $(\tilde{M}, f(0,0))$ **ExtIt**-identifies $\mathcal{L}_1$ follows from Claim 11.1 just below.

Claim 11.1. $(\tilde{M}, f(0,0))$ is extensional with respect to $\mathcal{L}_1$, in the sense of Definition 1(d).

Proof of Claim. Suppose that $\sigma_0, \sigma_1 \in \mathrm{Seq}(\mathcal{L}) - \{\lambda\}$ are such that

$$W_{\tilde{M}_p^*(\sigma_0^-)} = W_{\tilde{M}_p^*(\sigma_1^-)} \ \wedge \ \mathrm{last}(\sigma_0) = \mathrm{last}(\sigma_1). \tag{59}$$

It must be shown that $W_{\tilde{M}_p^*(\sigma_0)} = W_{\tilde{M}_p^*(\sigma_1)}$. If $\tilde{M}_p^*(\sigma_0^-) = \tilde{M}_p^*(\sigma_1^-)$, then $W_{\tilde{M}_p^*(\sigma_0)} = W_{\tilde{M}_p^*(\sigma_1)}$ follows immediately. So, suppose that $\tilde{M}_p^*(\sigma_0^-) \neq \tilde{M}_p^*(\sigma_1^-)$. Clearly, the only way that this can occur is if, for some e and e' with $2e' > 2e+1$,

$$\tilde{M}_p^*(\sigma_0^-) = f(e+1, e'+1) \;\wedge\; \tilde{M}_p^*(\sigma_1^-) = f(e+1, 0), \tag{60}$$

or (60) with σ_0 and σ_1 reversed. If σ_0 and σ_1 are reversed, then the proof is symmetric. So, suppose (60). By (60) and the first conjunct of (59),

$$\{0, 2, ..., 2e\} \cup \{2e+1\} \cup 2W_{e'} = \{0, 2, ..., 2e\} \cup \{2e+1\} \cup 2W_e. \tag{61}$$

Furthermore, it follows from the first conjunct of (60) that

$$\{2e+1, 2e'\} \subseteq \mathrm{content}(\sigma_0^-). \tag{62}$$

Let $L \in \mathcal{L}$ be such that $\sigma_0 \in \mathrm{Seq}(L)$. Given (61) and (62), a straightforward analysis of (52) reveals that L must be a language for which $W_e = X$. Thus,

$$L = \{0, 2, ..., 2e\} \cup \{2e+1\} \cup 2W_e, \tag{63}$$

and

$$(\forall e'' \in W_e)[W_{e''} = W_e]. \tag{64}$$

Note that, regardless of $\mathrm{last}(\sigma_0)$, $\tilde{M}_p^*(\sigma_0) = \tilde{M}_p^*(\sigma_0^-)$. Thus, to complete the proof of the claim, it suffices to show that $W_{\tilde{M}_p^*(\sigma_1)} = W_{\tilde{M}_p^*(\sigma_1^-)}$. If $\tilde{M}_p^*(\sigma_1) = \tilde{M}_p^*(\sigma_1^-)$, then $W_{\tilde{M}_p^*(\sigma_1)} = W_{\tilde{M}_p^*(\sigma_1^-)}$ follows immediately. So, suppose that $\tilde{M}_p^*(\sigma_1) \neq \tilde{M}_p^*(\sigma_1^-)$. Then, it must be the case that $\mathrm{last}(\sigma_1) = 2e''$, for some $2e'' > 2e+1$. Thus,

$$\tilde{M}_p^*(\sigma_1) = f(e+1, e''+1). \tag{65}$$

It follows from the second conjunct of (60) that

$$2e+1 \in \mathrm{content}(\sigma_1^-). \tag{66}$$

Thus, $\sigma_1 \in \mathrm{Seq}(L)$, and, by (63), $e'' \in W_e$. Furthermore, by (64),

$$W_{e''} = W_e. \tag{67}$$

Finally,

$$\begin{aligned}
W_{\tilde{M}_p^*(\sigma_1)} &= W_{f(e+1, e''+1)} && \{\text{by } (65)\} \\
&= \{0, 2, ..., 2e\} \cup \{2e+1\} \cup 2W_{e''} && \{\text{by } (55)\} \\
&= \{0, 2, ..., 2e\} \cup \{2e+1\} \cup 2W_e && \{\text{by } (67)\} \\
&= W_{f(e+1, 0)} && \{\text{by } (54)\} \\
&= W_{\tilde{M}_p^*(\sigma_1^-)} && \{\text{by } (60)\}.
\end{aligned}$$

□ (**Claim 11.1**)

– STAGE 0. Search for an $m \geq 1$ such that

$$M_p^*\big((0 \cdot 2 \cdot \dots \cdot 2e_0) \cdot (2e_0+1)^{m+1}\big)\downarrow = M_p^*\big((0 \cdot 2 \cdot \dots \cdot 2e_0) \cdot (2e_0+1)^{m}\big).$$

If such an m is found, then set $\sigma_1 = (0 \cdot 2 \cdot \dots \cdot 2e_0) \cdot (2e_0+1)^m$, and go to stage 1. If *no* such m is found, then search indefinitely.

– STAGE 1. For larger and larger values of n, make it the case that

$$W_{e_1} = \dots = W_{e_n} = \{e_1, ..., e_n\}.$$

Simultaneously, search for an $i \in \{1, ..., n\}$ such that

$$M_p^*(\sigma_1 \cdot 2e_i)\downarrow \neq M_p^*(\sigma_1).$$

If such i and n are found, then make it the case that

$$W_{e_0} = \{e_1, ..., e_n\},$$

and terminate the construction. If *no* such i and n are found, then search indefinitely, while making it the case that $(\forall i \geq 1)[W_{e_i} = \{e_i \mid i \geq 1\}]$.

Fig. 4. The construction of $(e_i)_{i \in \mathbb{N}}$ in the proof of Theorem 11.

It remains to show that $\mathcal{L}_1 \notin \mathbf{InjIt}$. By way of contradiction, suppose otherwise, as witnessed by $(M, p, (X_j)_{j\in\mathbb{N}}) \in \mathcal{PC}_{1,1} \times \mathbb{N} \times \mathcal{EN}$. Then, there exists a k_0 such that

$$(\forall e \geq k_0)[M_p^*(0 \cdot 2 \cdot \dots \cdot 2k_0 \cdot 2e)\downarrow = M_p^*(0 \cdot 2 \cdot \dots \cdot 2k_0)]. \tag{68}$$

By Case's 1-1 Operator Recursion Theorem [Cas74,Cas94],[12] there exists a computably enumerable sequence of pairwise-distinct φ-programs $(e_i)_{i\in\mathbb{N}}$ such that

$$(\forall i)[e_i \geq k_0], \tag{69}$$

and such that the behavior of $(e_i)_{i\in\mathbb{N}}$ is as in Figure 4.

Claim 11.2. Stage 0 is exited.

Proof of Claim. By way of contradiction, suppose that stage 0 is *not* exited. Then, $W_{e_0} = \emptyset$. Let L be as follows.

$$L = \{0, 2, ..., 2e_0\} \cup \{2e_0 + 1\}. \tag{70}$$

Note that L is a language in $\mathcal{L}_1$. Furthermore,

$$t = (0 \cdot 2 \cdot \dots \cdot 2e_0) \cdot (2e_0+1)^\infty \tag{71}$$

[12] Intuitively, the 1-1 Operator Recursion Theorem allows one to construct a computably enumerable sequence of pairwise-distinct φ-programs $(e_i)_{i\in\mathbb{N}}$ such that each program e_i *knows* all programs in the sequence and its own index i.

is a text for L. But, since stage 0 is *not* exited, $(M, p, (X_j)_{j\in\mathbb{N}})$ does *not* converge to a single hypothesis on this text — a contradiction. $\square$ (**Claim 11.2**)

For the remainder of the proof of the theorem, let m_0 be the m discovered in stage 0. By Claim 11.2, such an m_0 exists. Let σ_1 be as it would be set in stage 0, i.e.,

$$\sigma_1 = (0 \cdot 2 \cdot \cdots \cdot 2e_0) \cdot (2e_0 + 1)^{m_0}. \tag{72}$$

Consider the following cases.

Case [stage 1 is *not* exited]. Then, $W_{e_0} = \emptyset$ and $(\forall i \geq 1)[W_{e_i} = \{e_i \mid i \geq 1\}]$. Furthermore, for each $i \geq 1$,

$$[M_p^*(\sigma_1 \cdot 2e_i) = M_p^*(\sigma_1)] \ \vee \ [M_p^*(\sigma_1 \cdot 2e_i)\uparrow]. \tag{73}$$

Let L_0 and L_1 be as follows.

$$L_0 = \{0, 2, ..., 2e_0\} \cup \{2e_0 + 1\}. \tag{74}$$

$$L_1 = \{0, 2, ..., 2e_0\} \cup \{2e_0 + 1\} \cup \{2e_i \mid i \geq 1\}. \tag{75}$$

Note that each of L_0 and L_1 is a language in $\mathcal{L}_1$. Furthermore,

$$t_0 = (0 \cdot 2 \cdot \cdots \cdot 2e_0) \cdot (2e_0 + 1)^{\infty}; \tag{76}$$

$$t_1 = (0 \cdot 2 \cdot \cdots \cdot 2e_0) \cdot (2e_0 + 1)^{m_0} \cdot 2e_1 \cdot 2e_2 \cdot \cdots \tag{77}$$

are, respectively, texts for L_0 and L_1. But, by (73), either $(M, p, (X_j)_{j\in\mathbb{N}})$ cannot distinguish L_0 and L_1 on these texts, or $(M, p, (X_j)_{j\in\mathbb{N}})$ diverges on t_1. Either way, $(M, p, (X_j)_{j\in\mathbb{N}})$ does *not* **InjIt**-identify $\mathcal{L}_1$ — a contradiction.

Case [stage 1 *is* exited]. Then, for some n_0, $(\forall i \leq n_0)[W_{e_i} = \{e_1, ..., e_{n_0}\}]$. Furthermore, for some $i_0 \in \{1, ..., n_0\}$,

$$M_p^*(\sigma_1 \cdot 2e_{i_0})\downarrow \neq M_p^*(\sigma_1). \tag{78}$$

Let L be as follows.

$$L = \{0, 2, ..., 2e_0\} \cup \{2e_0 + 1\} \cup \{e_1, ..., e_{n_0}\}. \tag{79}$$

Note that L is a language in $\mathcal{L}_1$. Furthermore,

$$t_0 = (0 \cdot 2 \cdot \cdots \cdot 2e_0) \cdot (2e_1 \cdot 2e_2 \cdot \cdots \cdot 2e_{n_0}) \cdot (2e_0 + 1)^{\infty}; \tag{80}$$

$$t_1 = (0 \cdot 2 \cdot \cdots \cdot 2e_0) \cdot (2e_1 \cdot 2e_2 \cdot \cdots \cdot 2e_{n_0}) \cdot (2e_0 + 1)^{m_0} \cdot (2e_{i_0})^{\infty} \tag{81}$$

are each texts for L. Let j_0 be such that

$$M_p^*\big((0 \cdot 2 \cdot \cdots \cdot 2e_0) \cdot (2e_0 + 1)^{m_0}\big) = j_0. \tag{82}$$

Note that, for each $m \geq m_0$,

$$\begin{aligned}
& M_p^*\big((0 \cdot 2 \cdot \cdots \cdot 2e_0) \cdot (2e_1 \cdot 2e_2 \cdot \cdots \cdot 2e_{n_0}) \cdot (2e_0 + 1)^{m}\big) \\
&= M_p^*\big((0 \cdot 2 \cdot \cdots \cdot 2e_0) \cdot (2e_0 + 1)^{m}\big) && \{\text{by (68) and (69)}\} \\
&= M_p^*\big((0 \cdot 2 \cdot \cdots \cdot 2e_0) \cdot (2e_0 + 1)^{m_0}\big) && \{\text{by the choice of } m_0\} \\
&= j_0 && \{\text{by (82)}\}.
\end{aligned}$$

Thus, on t_0, $(M, p, (X_j)_{j\in\mathbb{N}})$ converges to j_0. Let j_1 be the hypothesis to which $(M, p, (X_j)_{j\in\mathbb{N}})$ converges on t_1. Note that

$$\begin{array}{ll} M(j_0, 2e_{i_0}) & \\ = M\Big(M^*_p\big((0\cdot 2\cdot\dots\cdot 2e_0)\cdot(2e_0+1)^{m_0}\big), 2e_{i_0}\Big) & \{\text{by (82)}\} \\ = M^*_p\big((0\cdot 2\cdot\dots\cdot 2e_0)\cdot(2e_0+1)^{m_0}\cdot 2e_{i_0}\big) & \{\text{immediate}\} \\ {\downarrow}\neq M^*_p\big((0\cdot 2\cdot\dots\cdot 2e_0)\cdot(2e_0+1)^{m_0}\big) & \{\text{by (78)}\} \\ = j_0 & \{\text{by (82)}\}. \end{array}$$

Thus, it must be the case that $j_0 \neq j_1$. But then this contradicts the fact that $(M, p, (X_j)_{j\in\mathbb{N}})$ **InjIt**-identifies L. □ (**Theorem 11**)

As mentioned above, the fact that **It** $\not\subseteq$ **ExtIt** is due to Jain.

Theorem 12 (Jain [Jai10]). It $\not\subseteq$ ExtIt.

We conclude this section with the following remark.

Remark 13 The fact **It** $\not\subseteq$ **InjIt** (as opposed to **It** $\not\subseteq$ **ExtIt** or **ExtIt** $\not\subseteq$ **InjIt**) can be shown directly using either of the next two pre-existing results.

- There exists a class of languages that *can* be **It**-identified, but that can*not* be so identified *order-independently* (in the sense of [BB75,Ful90]).[13]
- There exists a class of languages that *can* be **It**-identified, but that can*not* be so identified *strongly non-U-shapedly* [CK10, Theorem 5.4] (see also [Bei84,Wie91,CM08]).

5 Iterative Learning by Enumeration Operator

This section examines the iterative learning by enumeration operator model (**EOIt**). Recall that **EOIt** is similar to **It**, except that the computational component of the learner is modeled as an enumeration operator, as opposed to a partial computable function. Our main results of this section are the following.

- Every computably finitely thick class of languages can be **EOIt**-identified (Theorem 14). (Recall that computable finite thickness was covered in Definition 3.)
- **EOIt** $\not\subseteq$ **It** (Corollary 15).
- **FrIt** $\not\subseteq$ **EOIt** (Theorem 20).

This section also includes other results (Propositions 16 and 17) whose main purpose is to fulfill the diagram in Figure 1. An open problem that remains is whether **It** $\cap$ **EOIt** $\subseteq$ **ExtIt**, i.e., whether every class of languages that can be **It**-identified *and* **EOIt**-identified can be **ExtIt**-identified (Problem 21).

Theorem 14 just below is our first main result of this section.

Theorem 14. Suppose that $\mathcal{L}$ is computably finitely thick. Then, $\mathcal{L} \in$ **EOIt**.

[13] An anonymous referee attributes this result to Liepe & Wiehagen.

Proof. Suppose that $\mathcal{L}$ is computably finitely thick. Let $\psi : \mathcal{Fin} \to \mathcal{CE}$ be such that, for each $A \in \mathcal{Fin} - \{\emptyset\}$:

$$\psi(\emptyset) = \emptyset; \tag{83}$$
$$\psi(A) = \bigcap\{L \in \mathcal{L} \mid A \subseteq L\}. \tag{84}$$

Claim 14.1.

(a) For each finite A, $A \subseteq \psi(A)$.
(b) For each $L \in \mathcal{L}$, and each finite $A \subseteq L$, $\psi(A) \subseteq L$.

Proof of Claim. Straightforward. $\square$ (**Claim 14.1**)

Let $\Psi : \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ be such that, for each $X \subseteq \mathbb{N}$,

$$\Psi(X) = \bigcup\{\psi(A) \mid A \text{ is finite } \wedge\ A \subseteq X\}. \tag{85}$$

Let $\mathcal{M} : \mathcal{P}(\mathbb{N}) \times \mathbb{N}_{\#} \to \mathcal{P}(\mathbb{N})$ be such that, for each $X \subseteq \mathbb{N}$ and x:

$$\mathcal{M}(X, \#) = X; \tag{86}$$
$$\mathcal{M}(X, x) = \Psi(X \cup \{x\}). \tag{87}$$

Claim 14.2.

(a) $\mathcal{M} \in \mathcal{EO}_{1,1}$.
(b) For each σ, $\Psi\big(\text{content}(\sigma)\big) \subseteq \mathcal{M}^*_{\emptyset}(\sigma)$.
(c) For each $L \in \mathcal{L}$, and each $\sigma \in \text{Seq}(L)$, $\mathcal{M}^*_{\emptyset}(\sigma) \subseteq L$.

Proof of Claim. Part (a) follows from the fact that $\mathcal{L}$ is computably finitely thick. (In particular, the fact that $\mathcal{L}$ is computably finitely thick implies that ψ is algorithmic.) Part (b) is shown by a straightforward induction using essentially Claim 14.1(a). For part (c): let $L \in \mathcal{L}$ be fixed. Note that $\mathcal{M}^*_{\emptyset}(\lambda) = \emptyset \subseteq L$. So, suppose inductively that $\sigma \neq \lambda$, and that $\mathcal{M}^*_{\emptyset}(\sigma^-) \subseteq L$. If $\text{last}(\sigma) = \#$, then $\mathcal{M}^*_{\emptyset}(\sigma) = \mathcal{M}^*_{\emptyset}(\sigma^-) \subseteq L$. So, suppose that $\text{last}(\sigma) \neq \#$. Let $x = \text{last}(\sigma)$. Then,

$$\begin{aligned}
&\mathcal{M}^*_{\emptyset}(\sigma) \\
&= \mathcal{M}\big(\mathcal{M}^*_{\emptyset}(\sigma^-), x\big) && \{\text{immediate}\} \\
&= \Psi\big(\mathcal{M}^*_{\emptyset}(\sigma^-) \cup \{x\}\big) && \{\text{by (87)}\} \\
&= \bigcup\big\{\psi(A) \mid A \text{ is finite } \wedge\ A \subseteq \big(\mathcal{M}^*_{\emptyset}(\sigma^-) \cup \{x\}\big)\big\} && \{\text{by (85)}\} \\
&\subseteq \bigcup\big\{\psi(A) \mid A \text{ is finite } \wedge\ A \subseteq L \cup \{x\}\big\} && \{\text{by the i.h.}\} \\
&= \bigcup\big\{\psi(A) \mid A \text{ is finite } \wedge\ A \subseteq L\big\} && \{\text{because } x \in L\} \\
&\subseteq L && \{\text{by Claim 14.1(b)}\}.
\end{aligned}$$

$\square$ (**Claim 14.2**)

To show that $(\mathcal{M}, \emptyset)$ witnesses $\mathcal{L} \in$ **EOIt**: let $L \in \mathcal{L}$ be fixed. Clearly, if $L = \emptyset$, then $(\mathcal{M}, \emptyset)$ identifies L. So, suppose that $L \neq \emptyset$. Let $x_0 \in L$ be fixed.

Since $\mathcal{L}$ is computably finitely thick, there exist only finitely many $L' \in \mathcal{L}$ such that $x_0 \in L'$. Thus, there exists a finite set $A_0 \supseteq \{x_0\}$ such that

$$A_0 \subseteq L \ \wedge\ (\forall L' \in \mathcal{L})[A_0 \subseteq L' \ \Rightarrow\ L \subseteq L']. \tag{88}$$

By (88), it is clearly the case that

$$\psi(A_0) = \bigcap\{L' \in \mathcal{L} \mid A_0 \subseteq L'\} \supseteq L. \tag{89}$$

Let t be any text for L. Let i_0 be such that $A_0 \subseteq \mathrm{content}(t[i_0])$. By Claim 14.2(c), for each i, $\mathcal{M}^*_\emptyset(t[i]) \subseteq L$. Furthermore,

$$\begin{aligned}
\mathcal{M}^*_\emptyset(t[i]) &\supseteq \Psi\big(\mathrm{content}(t[i])\big) && \{\text{by Claim 14.2(b)}\}\\
&\supseteq \Psi\big(\mathrm{content}(t[i_0])\big) && \{\text{by the monotonicity of } \Psi\}\\
&\supseteq \Psi(A_0) && \{\text{by the monotonicity of } \Psi\}\\
&\supseteq \psi(A_0) && \{\text{by (85)}\}\\
&\supseteq L && \{\text{by (89)}\}.
\end{aligned}$$

□ (**Theorem 14**)

Recall that the proof of Theorem 9 exhibited a computably finitely thick class of languages $\mathcal{L}_0 \notin \mathbf{It}$. By Theorem 14, $\mathcal{L}_0 \in \mathbf{EOIt}$. Thus, one has the following.

Corollary 15 (of Theorems 9 and 14) EOIt $\not\subseteq$ It.

Recall that $\mathcal{F}in$ is the collection of all finite subsets of $\mathbb{N}$. Proposition 16 just below establishes that $\mathcal{F}in \in \mathbf{EOIt}$.

Proposition 16 $\mathcal{F}in \in \mathbf{EOIt}$.

Proof. Straightforward. □ (**Proposition 16**)

Recall that the class $\mathcal{L}_1$ from the proof Theorem 11 was used to show that **ExtIt** $\not\subseteq$ **InjIt**. Proposition 17 just below establishes that $\mathcal{L}_1 \in \mathbf{EOIt}$.

Proposition 17 Let $\mathcal{L}_1$ be as in the proof of Theorem 11. Then, $\mathcal{L}_1 \in \mathbf{EOIt}$.

Proof. Let $\Psi : \mathcal{P}(\mathbb{N}) \to \mathcal{P}(\mathbb{N})$ be such that, for each $X \subseteq \mathbb{N}$,

$$\begin{aligned}
\Psi(X) = X &\cup \bigcup\{2W_e \mid 2e+1 \in X\}\\
&\cup \bigcup\{2W_{e'} \mid 2e' \in X \ \wedge\ (\exists e)[2e+1 \in X \ \wedge\ 2e+1 < 2e']\}.
\end{aligned} \tag{90}$$

Clearly, $\Psi \in \mathcal{EO}_{1,0}$. Let $\mathcal{M} : \mathcal{P}(\mathbb{N}) \times \mathbb{N}_\# \to \mathcal{P}(\mathbb{N})$ be such that, for each $X \subseteq \mathbb{N}$ and $e \in \mathbb{N}$,

$$\mathcal{M}(X, \#) = X; \tag{91}$$

$$\mathcal{M}(X, 2e) = \Psi(X \cup \{2e\}); \tag{92}$$

$$\mathcal{M}(X, 2e+1) = \Psi(\{0, 2, ..., 2e\} \cup \{2e+1\}) \cup \begin{cases} X, & \text{if } 2e+1 \in X;\\ \emptyset, & \text{otherwise.}\end{cases} \tag{93}$$

Clearly, $\mathcal{M} \in \mathcal{EO}_{1,1}$. Furthermore, it is straightforward to show that $(\mathcal{M}, 2\mathbb{N})$ **EOIt**-identifies $\mathcal{L}_1$. □ (**Proposition 17**)

The proof of Theorem 20 below exhibits a class $\mathcal{L}_2 \in \mathbf{FrIt} - \mathbf{EOIt}$. The proof of Theorem 20 makes use of the next two lemmas.

Lemma 18 There exists a Friedberg numbering $(X_j)_{j\in\mathbb{N}}$ satisfying

$$(\forall j)[1 \le |D_j| \le 3 \ \Rightarrow\ X_j = D_j]. \tag{94}$$

Proof. Straightforward. □ (**Lemma 18**)

Lemma 19 For each $(\mathcal{M}, X) \in \mathcal{EO}_{1,1} \times \mathcal{CE}$, and each $\rho, \sigma, \tau \in \mathrm{Seq}$,

$$\mathcal{M}^*_X(\rho) \subseteq \mathcal{M}^*_X(\sigma) \ \Rightarrow\ \mathcal{M}^*_X(\rho \cdot \tau) \subseteq \mathcal{M}^*_X(\sigma \cdot \tau). \tag{95}$$

Proof. A straightforward induction using essentially the monotonicity of $\mathcal{M}$.
□ (**Lemma 19**)

Theorem 20. FrIt $\not\subseteq$ **EOIt**.

Proof. Let $(X_j)_{j\in\mathbb{N}}$ be a Friedberg numbering as asserted to exist by Lemma 18, and let $(Y_k)_{k\in\mathbb{N}}$ be any Friedberg numbering satisfying: $Y_0 = \emptyset$. Let $(Z_\ell)_{\ell\in\mathbb{N}}$ be such that, for each j and k, $Z_{\langle j,k\rangle} = (2X_j) \cup (2Y_k + 1)$. It is straightforward to show that $(Z_\ell)_{\ell\in\mathbb{N}}$ is a Friedberg numbering. Let $\mathcal{L}_2$ be the following class of languages.

$$\mathcal{L}_2 = \{Z_{\langle j, \max D_j\rangle} \mid 1 \le |D_j| \le 3\}. \tag{96}$$

Note that, for each j such that $1 \le |D_j| \le 3$,

$$\begin{aligned} Z_{\langle j, \max D_j\rangle} &= (2X_j) \cup (2Y_{\max D_j} + 1) \\ &= (2D_j) \cup (2Y_{\max D_j} + 1). \end{aligned} \tag{97}$$

It is straightforward to show that $\mathcal{L}_2 \in$ **FrIt** (e.g., using $(Z_\ell)_{\ell\in\mathbb{N}}$ as the hypothesis space). It remains to show that $\mathcal{L}_2 \notin$ **EOIt**. By way of contradiction, suppose otherwise, as witnessed by $(\mathcal{M}, L_0) \in \mathcal{EO}_{1,1} \times \mathcal{CE}$. Recall that $Y_0 = \emptyset$. Let $k_0 = 0$, and let $L_0 = \{2k_0\}$. Note that $L_0 \in \mathcal{L}_2$. It follows that there exists an m_0 such that

$$\mathcal{M}^*_{L_0}\big((2k_0)^{m_0}\big) = L_0. \tag{98}$$

Let k_1 be such that $Y_{k_1} = \mathbb{N}$, and let $L_1 = \{2k_0, 2k_1\} \cup (2\mathbb{N} + 1)$. Note that $L_1 \in \mathcal{L}_2$. It follows that there exists an n_0 such that

$$(\forall n \ge n_0)\big[\mathcal{M}^*_{L_0}\big((2k_0)^{m_0} \cdot 2k_1 \cdot 1 \cdot 3 \cdot \cdots \cdot 2n+1\big) = L_1\big]. \tag{99}$$

Let k_2 and n_1 be such that (a)-(c) below.

(a) $k_1 < k_2$.
(b) $n_0 \le n_1$.
(c) $Y_{k_2} = \{0, ..., n_1\}$.

Clearly, such k_2 and n_1 exist. Let $L_2 = \{2k_0, 2k_1, 2k_2\} \cup \{1, 3, ..., 2n_1 + 1\}$. Note that $L_2 \in \mathcal{L}_2$, and that $L_1 \not\subseteq L_2$. Let $\sigma = 2k_1 \cdot 1 \cdot 3 \cdot \cdots \cdot 2n_1 + 1$, let t be any text for L_2, and let t' be such that

$$t' = \sigma \cdot t(0) \cdot \sigma \cdot t(1) \cdot \cdots . \tag{100}$$

Note that t' is a text for L_2. It follows that there exists an i_0 such that

$$(\forall i \geq i_0)[\mathcal{M}^*_{L_0}(t[i]) = L_2]. \tag{101}$$

Without loss of generality, suppose that i_0 is divisible by $|\sigma| + 1$. By (98) and (101),

$$\mathcal{M}^*_{L_0}\big((2k_0)^{m_0}\big) = L_0 \subseteq L_2 = \mathcal{M}^*_{L_0}\big(t[i_0]\big). \tag{102}$$

Furthermore, by (99) and the fact that $n_0 \leq n_1$,

$$L_1 \subseteq \mathcal{M}^*_{L_0}\big((2k_0)^{m_0} \cdot \sigma\big). \tag{103}$$

From (102), (103), and Lemma 19, it follows that

$$L_1 \subseteq \mathcal{M}^*_{L_0}\big(t[i_0] \cdot \sigma\big) = \mathcal{M}^*_{L_0}\big(t[i_0 + |\sigma|]\big). \tag{104}$$

But since $L_1 \not\subseteq L_2$, (101) and (104) are contradictory. □ (**Theorem 20**)

As mentioned above, the following problem remains open.

Problem 21 Is it the case that **It** $\cap$ **EOIt** $\subseteq$ **ExtIt**?

References

[Ang80] D. Angluin. Finding patterns common to a set of strings. *Journal of Computer and System Sciences*, 21(1):46–62, 1980.

[BB75] L. Blum and M. Blum. Toward a mathematical theory of inductive inference. *Information and Control*, 28(2):125–155, 1975.

[BBCJS10] L. Becerra-Bonache, J. Case, S. Jain, and F. Stephan. Iterative learning of simple external contextual languages. *Theoretical Computer Science*, 411(29–30):2741–2756, 2010.

[Bei84] H-R. Beick. *Induktive Inferenz mit höchster Konvergenzgeschwindigkeit*. PhD thesis, Sektion Mathematik, Humboldt-Universität Berlin, 1984.

[Cas74] J. Case. Periodicity in generations of automata. *Mathematical Systems Theory*, 8(1):15–32, 1974.

[Cas94] J. Case. Infinitary self-reference in learning theory. *Journal of Experimental and Theoretical Artificial Intelligence*, 6(1):3–16, 1994.

[CCJS07] L. Carlucci, J. Case, S. Jain, and F. Stephan. Results on memory-limited U-shaped learning. *Information and Computation*, 205(10):1551–1573, 2007.

[CK10] J. Case and T. Kötzing. Strongly non-U-shaped learning results by general techniques. In *Proc. of the 23rd Conference on Learning Theory*, pages 181–193. Omnipress, 2010.

[CM08] J. Case and S. Moelius. Optimal language learning. In *Proc. of the 19th Annual Conference on Algorithmic Learning Theory (ALT'2008)*, volume 5254 of *Lecture Notes in Artificial Intelligence*, pages 419–433. Springer, 2008.

[dBY10] M. de Brecht and A. Yamamoto. Topological properties of concept spaces (full version). *Information and Computation*, 208(4):327–340, 2010.

[FKW82] R. Freivalds, E. Kinber, and R. Wiehagen. Inductive inference and computable one-one numberings. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 28(27):463–479, 1982.

[Fri58] R. Friedberg. Three theorems on recursive enumeration. I. Decomposition. II. Maximal set. III. Enumeration without duplication. *Journal of Symbolic Logic*, 23(3):309–316, 1958.

[Ful90] M. Fulk. Prudence and other conditions on formal language learning. *Information and Computation*, 85(1):1–11, 1990.

[Gol67] E. Mark Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.

[Jai10] S. Jain. Private communcation, 2010.

[JLMZ10] S. Jain, S. Lange, S. Moelius, and S. Zilles. Incremental learning with temporary memory. *Theoretical Computer Science*, 411(29–30):2757–2772, 2010.

[JS08] S. Jain and F. Stephan. Learning in Friedberg numberings. *Information and Computation*, 206(6):776–790, 2008.

[Kum90] M. Kummer. An easy priority-free proof of a theorem of Friedberg. *Theoretical Computer Science*, 74(2):249–251, 1990.

[LW91] S. Lange and R. Wiehagen. Polynomial time inference of arbitrary pattern languages. *New Generation Computing*, 8(4):361–370, 1991.

[LZ96] S. Lange and T. Zeugmann. Incremental learning from positive data. *Journal of Computer and System Sciences*, 53(1):88–103, 1996.

[LZZ08] S. Lange, T. Zeugmann, and S. Zilles. Learning indexed families of recursive languages from positive data: A survey. *Theoretical Computer Science*, 397(1–3):194–232, 2008.

[Rog67] H. Rogers. *Theory of Recursive Functions and Effective Computability*. McGraw Hill, 1967. Reprinted, MIT Press, 1987.

[SSS+94] S. Shimozono, A. Shinohara, T. Shinohara, S. Miyano, S. Kuhara, and S. Arikawa. Knowledge acquisition from amino acid sequences by machine learning system BONSAI. *Transactions of the Information Processing Society of Japan*, 35(10):2009–2018, 1994.

[Wie76] R. Wiehagen. Limes-Erkennung rekursiver Funktionen durch spezielle Strategien. *Elektronische Informationsverarbeitung und Kybernetik*, 12(1/2):93–99, 1976.

[Wie91] R. Wiehagen. A thesis in inductive inference. In *Proc. of the 1st International Workshop on Nonmonotonic and Inductive Logic (1990)*, volume 543 of *Lecture Notes in Artificial Intelligence*, pages 184–207. Springer, 1991.