U-Shaped, Iterative, and Iterative-with-Counter Learning*

John Case and Samuel E. Moelius III

August 31, 2007

Abstract

This paper solves an important problem left open in the literature by showing that *U-shapes* are *un*necessary in *iterative learning*. A *U-shape* occurs when a learner first *learns*, then *unlearns*, and, finally, *relearns*, some target concept. *Iterative learning* is a Gold-style learning model in which each of a learner's output conjectures depends *only* upon the learner's most recent conjecture and input element. Previous results had shown, for example, that U-shapes are *un*necessary for explanatory learning, but *are* necessary for behaviorally correct learning.

Work on the aforementioned problem led to the consideration of an iterative-like learning model, in which each of a learner's conjectures may, *in addition*, depend upon the number of elements so far presented to the learner. Learners in this new model are strictly more powerful than traditional iterative learners, yet not as powerful as full explanatory learners. Can any class of languages learnable in this new model be learned without U-shapes? For now, *this* problem is left open.

1 Introduction

1.1 U-Shapes

A *U-shape* occurs when a learner first *learns*, then *unlearns*, and, finally, *relearns*, some target concept. This phenomenon has been observed, for example, in children learning the use of regular and irregular verbs, e.g., a child first correctly learns that the past tense of "speak" is "spoke"; then, the child overregularizes and *in*correctly uses "speaked"; finally, the child returns to correctly using "spoke" [MPU+92, PM91, TA02].

Important questions regarding U-shapes are the following. Are U-shapes an *un*necessary accident of human evolution, *or*, are there classes of tasks that can be learned *with* U-shapes, but *not* otherwise? That is, are there classes of tasks that are learnable *only* by returning to some *abandoned* correct behavior?

*This paper is an expanded version of [CM07].

There have been mathematical attempts to answer these questions in the context of Gold-style language learning [Gol67, JORS99].[1] Models of Gold-style language learning differ from one another in ways described hereafter, however, the following is common to all. Infinitely often, a *learner* is fed successively longer finite, initial sequences of an *infinite* sequence of numbers and, possibly, pauses (#). The set of all such numbers represents a *language*, and, the infinite sequence, itself, is called a *text* for the language. For each finite, initial sequence of a text, the learner either: outputs a *conjecture*, or diverges (e.g., goes into an infinite loop). A conjecture may be either: a grammar (possibly for the language represented by the text), or a '?'.

One way in which Gold models differ from one another is in the criteria used to judge the *success* of a learner. Examples of models with differing criteria are *explanatory learning* (**Ex**-learning) [Gol67, JORS99][2] and *behaviorally correct learning* (**Bc**-learning) [CL82, JORS99]. In both models, for a learner to be successful, all but finitely many of the learner's conjectures must correctly (semantically) identify the input language. However, **Ex**-learning has the additional requirement that a learner converge *syntactically* to a single conjecture.

In Gold-style learning, a U-shape is formalized as: outputting a semantically correct conjecture, then outputting a semantically incorrect conjecture, and, finally, returning to a semantically correct conjecture [CCJS05, CCJS06, BCM$^+$07]. As it turns out, U-shapes are *un*necessary for **Ex**-learning, i.e., every class of languages that can be **Ex**-learned can be **Ex**-learned with*out* U-shapes [BCM$^+$07, Theorem 20]. On the other hand, U-shapes *are* necessary for **Bc**-learning, i.e., there are classes of languages that *can* be **Bc**-learned *with* U-shapes, but *not* with*out* [FJO94, proof of Theorem 4]. Thus, in at least some contexts, this *seemingly* inefficient behavior can actually increase one's learning power.[3]

1.2 Iterative Learning

For both **Ex**-learning and **Bc**-learning, a learner is free to base a conjecture upon *every* element presented to the learner up to that point. Thus, in a sense, an **Ex**-learner or **Bc**-learner can *remember* every element presented to it. One could argue that such an ability is beyond that possessed by (most) humans. This calls into question the applicability of **Ex**-learning and **Bc**-learning to modeling human learning. That is, it would seem that any model of human learning should be *memory limited* in some respect.

Iterative learning (**It**-learning) [Wie76, LZ96a, CJLZ99] is a straightforward variation of the **Ex**-learning model that *is* memory limited.[4] In this model, each of a learner's conjectures can depend *only* upon the learner's most recent conjecture and input element. An **It**-learner can remember elements fed to it by

[1] In this paper, we focus exclusively on language learning, as opposed to, say, function learning [JORS99].

[2] **Ex**-learning is the model that was actually studied by Gold [Gol67].

[3] There exist Gold models that lie strictly between **Ex** and **Bc** [Cas99]. For nearly every such model considered, U-shapes are necessary [CCJS05].

[4] Other memory limited models are considered in [OSW86, FJO94, CJLZ99, CCJS06].

coding them into its conjectures. However, like an **Ex**-learner, an **It**-learner is required to converge syntactically to a single conjecture. Thus, on any given text, an **It**-learner can perform such a coding-trick for only finitely many elements.

There have been previous attempts to determine whether U-shapes are necessary in **It**-learning [CCJS06, Jai06]. The memory limited aspect of **It**-learning makes it more nearly applicable than **Ex**-learning or **Bc**-learning to modeling human learning.

Herein (Theorem 2 in Section 3), we solve this important open problem by showing that U-shapes are *un*necessary in **It**-learning, i.e., any class of languages that can be **It**-learned can be **It**-learned with*out* U-shapes.

1.3 Other Restricted Forms of Learning

Two other restricted forms of learning that have been well studied are *set-driven learning* (**SDEx**-learning) and *partly set-driven learning* (**PSDEx**-learning) [WC80, SR84, Ful90, LZ96b].[5] The **SDEx**-learning model requires that a learner output syntactically identical conjectures when fed two different initial sequences with the same content, i.e., listing the same set of numbers. So, for example, when forming a conjecture, an **SDEx**-learner can*not* consider the number of elements so far presented to it, or the order in which those elements were presented. The **PSDEx**-learning model is similar, except that a learner is required to output identical conjectures when fed initial sequences with the same content *and length*. Thus, when forming a conjecture, a **PSDEx**-learner can*not* consider the order in which elements were presented to it, but *can* consider the number of such elements.

SDEx-learners and **It**-learners are alike in that neither can consider the number of elements so far presented to it when forming a conjecture. Furthermore, **PSDEx**-learners are like **SDEx**-learners with *just this one* restriction lifted. Herein, we consider a similar counterpart to **It**-learners. That is, we consider a model in which each of a learner's output conjectures can depend *only* upon the learner's most recent conjecture, the most recent input element, *and a counter* indicating the number of elements so far presented to the learner. For example, if the learner were fed

$$0 \diamond 1 \diamond \# \diamond \# \diamond 0,$$

then the value of the counter would be 5. Thus, a repetition or pause is treated just like any other element in determining the value of the counter. We call this model *iterative-with-counter* learning (**ItCtr**-learning). In Section 4, we show that **ItCtr**-learning and **SDEx**-learning are incomparable (Theorems 3 and 4), i.e., for each, there is a class of languages learnable by that one, but *not* the other. It follows that **ItCtr**-learning is strictly more powerful than **It**-learning, yet not as powerful as full **Ex**-learning.

In an early attempt at showing that U-shapes are unnecessary in **It**-learning, we obtained the partial result that U-shapes are unnecessary in **It**-learning of

[5] **PSDEx**-learning is also called *rearrangement independent learning* in the literature (e.g., [LZ96b]).

classes of *infinite* languages. Independently, Sanjay Jain obtained the same (partial) result [Jai06]. Thus, we hypothesize: learning without U-shapes is easier when the learner has access to some source of *infinitude*, e.g., the cardinality of the input language. This belief is what led us to consider the **ItCtr**-learning model, as every learner in this model has access to a source of infinitude, i.e., the counter, even when fed a text for a *finite* language.

Assuming our above hypothesis is correct, it should be easy to show that U-shapes are unnecessary in **ItCtr**-learning. Unfortunately, however, this problem has turned out to be more difficult than we had anticipated. So, for now, it is left open.

1.4 Organization

The remainder of this paper is organized as follows. Section 2, just below, gives notation and preliminaries. Section 3 proves our main result, namely, that U-shapes are *un*necessary in **It**-learning. Section 4 explores **ItCtr**-learning, and, restates, formally, the problem that this paper leaves open.

2 Notation and Preliminaries

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

$\mathbb{N}$ denotes the set of natural numbers, $\{0, 1, 2, \ldots\}$. $\mathbb{N}_? \stackrel{\text{def}}{=} \mathbb{N} \cup \{?\}$. $\mathbb{N}_\# \stackrel{\text{def}}{=} \mathbb{N} \cup \{\#\}$. Lowercase Roman letters, with or without decorations, range over elements of $\mathbb{N}$, unless stated otherwise. A and L, with or without decorations, range over subsets of $\mathbb{N}$. $\mathcal{L}$ ranges over collections of subsets of $\mathbb{N}$. For all finite, nonempty A, $\max A$ denotes the maximum element of A. $\max \emptyset \stackrel{\text{def}}{=} -1$.

ψ ranges over one-argument partial functions. For all ψ and x, $\psi(x)\downarrow$ denotes that $\psi(x)$ converges; $\psi(x)\uparrow$ denotes that $\psi(x)$ diverges.[6] For all ψ, $\mathrm{dom}(\psi) \stackrel{\text{def}}{=} \{x : \psi(x)\downarrow\}$ and $\mathrm{rng}(\psi) \stackrel{\text{def}}{=} \{y : (\exists x)[\psi(x) = y]\}$. We use $\uparrow$ to denote the value of a divergent computation. λ denotes the empty function.

$\varphi_0, \varphi_1, \ldots$ denotes any fixed, acceptable numbering of all one-argument partial computable functions [Rog67]. Φ denotes a fixed Blum complexity measure for φ [Blu67]. For all p, $W_p \stackrel{\text{def}}{=} \mathrm{dom}(\varphi_p)$. Thus, for all p, W_p is the pth recursively enumerable set [Rog67]. $W_\uparrow \stackrel{\text{def}}{=} \emptyset$.

$\mathbb{N}_\#^*$ denotes the set of all finite initial segments of total functions of type $\mathbb{N} \to \mathbb{N}_\#$. $\mathbb{N}_\#^{\leq\omega}$ denotes the set of *all* (finite and infinite) initial segments of total functions of type $\mathbb{N} \to \mathbb{N}_\#$. α, β, ϱ, σ, and τ, with or without decorations, range over elements of $\mathbb{N}_\#^*$. T, with or without decorations, ranges over total functions of type $\mathbb{N} \to \mathbb{N}_\#$.

For all $f \in \mathbb{N}_\#^{\leq\omega}$, $\mathrm{content}(f) \stackrel{\text{def}}{=} \mathrm{rng}(f) - \{\#\}$. For all T and L, *T is a text for L* $\stackrel{\text{def}}{\Leftrightarrow} \mathrm{content}(T) = L$. For all σ, $|\sigma|$ (pronounced: the *length* of σ) $\stackrel{\text{def}}{=} |\mathrm{dom}(\sigma)|$.

[6] For all one-argument partial functions ψ and x, $\psi(x)$ *converges* iff there exists y such that $\psi(x) = y$; $\psi(x)$ *diverges* iff there is *no* y such that $\psi(x) = y$. If ψ is partial computable, and x is such that $\psi(x)$ diverges, then one can imagine that a program associated with ψ *goes into an infinite loop* on input x.

For all $f \in \mathbb{N}_{\#}^{\leq\omega}$, and all σ, n, and i, (1) and (2) below.

$$f[n](i) \stackrel{\text{def}}{=} \begin{cases} f(i), & \text{if } i < n; \\ \uparrow, & \text{otherwise.} \end{cases} \tag{1}$$

$$(\sigma \diamond f)(i) \stackrel{\text{def}}{=} \begin{cases} \sigma(i), & \text{if } i < |\sigma|; \\ f(i - |\sigma|), & \text{otherwise.} \end{cases} \tag{2}$$

M, with or without decorations, ranges over partial computable functions of type $\mathbb{N}_{\#}^{*} \to \mathbb{N}_{?}$.[7]

The following are the Gold-style learning models considered in this paper.

Definition 1 For all **M** and L, (a)-(e) below.

(a) **M Ex**-*identifies* L $\Leftrightarrow$ for all texts T for L, there exist i and p such that $(\forall j \geq i)\big[\mathbf{M}(T[j]) = p\big]$ and $W_p = L$.

(b) **M SDEx**-*identifies* L $\Leftrightarrow$ **M Ex**-identifies L, *and*, for all ϱ and σ, if $\text{content}(\varrho) = \text{content}(\sigma)$, then $\mathbf{M}(\varrho) = \mathbf{M}(\sigma)$.

(c) **M PSDEx**-*identifies* L $\Leftrightarrow$ **M Ex**-identifies L, *and*, for all ϱ and σ, if $|\varrho| = |\sigma|$ and $\text{content}(\varrho) = \text{content}(\sigma)$, then $\mathbf{M}(\varrho) = \mathbf{M}(\sigma)$.

(d) **M It**-*identifies* L $\Leftrightarrow$ **M Ex**-identifies L, *and*, for all ϱ, σ, and τ such that $\text{content}(\varrho) \cup \text{content}(\sigma) \cup \text{content}(\tau) \subseteq L$, (i) and (ii) below.

  (i) $\mathbf{M}(\varrho)\downarrow$.

  (ii) $\mathbf{M}(\varrho) = \mathbf{M}(\sigma) \;\Rightarrow\; \mathbf{M}(\varrho \diamond \tau) = \mathbf{M}(\sigma \diamond \tau)$.

(e) **M ItCtr**-*identifies* L $\Leftrightarrow$ **M Ex**-identifies L, *and*, for all ϱ, σ, and τ such that $\text{content}(\varrho) \cup \text{content}(\sigma) \cup \text{content}(\tau) \subseteq L$, (i) and (ii) below.

  (i) $\mathbf{M}(\varrho)\downarrow$.

  (ii) $[|\varrho| = |\sigma| \;\wedge\; \mathbf{M}(\varrho) = \mathbf{M}(\sigma)] \;\Rightarrow\; \mathbf{M}(\varrho \diamond \tau) = \mathbf{M}(\sigma \diamond \tau)$.

Ex, **SD**, **PSD**, **It**, and **ItCtr** are mnemonic for *explanatory*, *set-driven*, *partly set-driven*, *iterative*, and *iterative-with-counter*, respectively.

Definition 2 For all $\mathcal{I} \in \{\mathbf{Ex}, \mathbf{SDEx}, \mathbf{PSDEx}, \mathbf{It}, \mathbf{ItCtr}\}$, (a) and (b) below.

(a) For all **M**, $\mathcal{I}(\mathbf{M}) = \{L : \mathbf{M}\ \mathcal{I}\text{-identifies } L\}$.

(b) $\mathcal{I} = \{\mathcal{L} : (\exists \mathbf{M})[\mathcal{L} \subseteq \mathcal{I}(\mathbf{M})]\}$.

In some parts of the literature (e.g., [CCJS06]), an iterative learner is given a different formulation from that of Definition 1(d). Specifically, in [CCJS06], such a learner is defined as: a pair consisting of a partial computable function of type $\mathbb{N}_{?} \times \mathbb{N}_{\#} \to \mathbb{N}_{?}$ and an initial conjecture. The two arguments of the partial computable function represent, respectively, the most recent conjecture (where, initially, this is the second element of the pair) and the most recent

[7]Such an **M** is often called an *inductive inference machine* [JORS99].

input element. The *equivalence* of the formulation of Definition 1(d) and that of [CCJS06] is given by the next two propositions. Furthermore, these propositions provide algorithmic translations between the two formulations that *preserve non-U-shapedness*.

We prefer our formulation of Definition 1(d) to that of [CCJS06] since it makes the partial function type of the learner match that of other models, e.g., **Ex**. We further note an interesting similarity between iterative learners of either kind and an *automaton* with a potentially infinite set of states, corresponding to the learner's conjectures. It was thinking of iterative learners in this way, and the Myhill-Nerode Theorem [DSW94], that led us to formulate iterative learners as in Definition 1(d).

Proposition 1 For all $\mathbf{M}$, there exists a partial computable function M of type $\mathbb{N}_? \times \mathbb{N}_\# \to \mathbb{N}_?$ and $p_0 \in \mathbb{N}$ satisfying the following. Let $L \in \mathbf{It}(\mathbf{M})$ be fixed, and let T be a text for L. For all i, let $p_{i+1} = M\big(p_i, T(i)\big)$. Then, for all i, (a)-(c) below.

(a) $M\big(p_i, T(i)\big)\downarrow$.

(b) If $\mathbf{M}(T[i]) \in \mathbb{N}$, then $W_{p_i} = W_{\mathbf{M}(T[i])}$.

(c) If $\mathbf{M}(T[i+1]) = \mathbf{M}(T[i])$, then $p_{i+1} = p_i$.

Proof. Let $f : \mathbb{N}_\#^* \to \mathbb{N}$ be a 1-1, computable function such that, for all σ,

$$W_{f(\sigma)} = \begin{cases} W_{\mathbf{M}(\sigma)}, & \text{if } \mathbf{M}(\sigma) \in \mathbb{N}; \\ \emptyset, & \text{otherwise.} \end{cases} \tag{3}$$

Clearly, such an f exists. For all σ and $x \in \mathbb{N}_\#$, let

$$M\big(f(\sigma), x\big) = \begin{cases} \uparrow, & \text{if } (*)\ \mathbf{M}(\sigma)\uparrow\ \vee\ \mathbf{M}(\sigma \diamond x)\uparrow; \\ f(\sigma \diamond x), & \text{if } \neg(*)\ \wedge\ \mathbf{M}(\sigma \diamond x) \neq \mathbf{M}(\sigma); \\ f(\sigma), & \text{otherwise.} \end{cases} \tag{4}$$

Let $L \in \mathbf{It}(\mathbf{M})$ be fixed, and let T be a text for L. Let $p_0 = f(\lambda)$. For all i, let $p_{i+1} = M\big(p_i, T(i)\big)$. The remainder of the proof is to show that, for all i, (i)-(iii) below.

(i) $M\big(p_i, T(i)\big)\downarrow$.

(ii) There exists σ such that $p_i = f(\sigma)$ and $\mathbf{M}(T[i]) = \mathbf{M}(\sigma)$.

(iii) If $\mathbf{M}(T[i+1]) = \mathbf{M}(T[i])$, then $p_{i+1} = p_i$.

(a)-(c) in statement of the proposition follow easily from (i)-(iii), just above, and from the definition of f. It is straightforward to show that (i)-(iii) hold in the case when $i = 0$. So, suppose, inductively, that (i)-(iii) hold for i. Let σ be such that $p_i = f(\sigma)$ and $\mathbf{M}(T[i]) = \mathbf{M}(\sigma)$.

To show that (i) holds for $i + 1$: Since T is a text for a language in $\mathbf{It}(\mathbf{M})$, $\mathbf{M}(T[i])\downarrow$ and $\mathbf{M}(T[i+1])\downarrow$. From the fact that $\mathbf{M}(T[i]) = \mathbf{M}(\sigma)$, it follows that

$\mathbf{M}(T[i+1]) = \mathbf{M}\big(\sigma \diamond T(i)\big)$. Thus, $\mathbf{M}(\sigma)\!\downarrow$ and $\mathbf{M}\big(\sigma \diamond T(i)\big)\!\downarrow$, and, therefore, $M\big(p_i, T(i)\big)\!\downarrow$.

To show that (ii) holds for $i+1$: By reasoning from the just previous paragraph, $\mathbf{M}(\sigma)\!\downarrow$ and $\mathbf{M}\big(\sigma\diamond T(i)\big)\!\downarrow$. If $\mathbf{M}\big(\sigma\diamond T(i)\big) \neq \mathbf{M}(\sigma)$, then $p_{i+1} = f\big(\sigma\diamond T(i)\big)$, and, as already mentioned, $\mathbf{M}(T[i+1]) = \mathbf{M}\big(\sigma \diamond T(i)\big)$. On the other hand, if $\mathbf{M}\big(\sigma\diamond T(i)\big) = \mathbf{M}(\sigma)$, then $p_{i+1} = f(\sigma)$, and $\mathbf{M}(T[i+1]) = \mathbf{M}\big(\sigma\diamond T(i)\big) = \mathbf{M}(\sigma)$.

To show that (iii) holds for $i+1$: Suppose that $\mathbf{M}(T[i+1]) = \mathbf{M}(T[i])$. Since $\mathbf{M}(T[i]) = \mathbf{M}(\sigma)$, $\mathbf{M}\big(\sigma \diamond T(i)\big) = \mathbf{M}(\sigma)$. Thus, $p_{i+1} = f(\sigma) = p_i$. □ (*Proposition 1*)

Proposition 2 Let M be a partial computable function of type $\mathbb{N}_? \times \mathbb{N}_\# \to \mathbb{N}_?$, and let $p_0 \in \mathbb{N}_?$ be fixed. For all texts T, let $p_0^T = p_0$, and, for all i, let $p_{i+1}^T = M\big(p_i^T, T(i+1)\big)$. Then, there exists $\mathbf{M}$ satisfying the following. Suppose that L is such that, for all texts T for L, (a) and (b) below.

(a) $(\forall i)\big[M\big(p_i^T, T(i)\big)\!\downarrow\big]$.

(b) There exists i such that $(\forall j \geq i)[p_j^T = p_i^T \in \mathbb{N}]$ and $W_{p_i^T} = L$.

Then, $L \in \mathbf{It}(\mathbf{M})$. Moreover, for all texts T for L, and all i, if $p_i^T \in \mathbb{N}$, then $W_{\mathbf{M}(T[i])} = W_{p_i^T}$.

Proof. Let $\mathbf{M}$ be such that $\mathbf{M}(\lambda) = p_0$, and, for all ϱ, and all $x \in \mathbb{N}_\#$,

$$\mathbf{M}(\varrho \diamond x) = M\big(\mathbf{M}(\varrho), x\big). \tag{5}$$

It is easy to verify that $\mathbf{M}$ has the desired properties. □ (*Proposition 2*)

Definition 3 For all $\mathcal{I} \in \{\mathbf{Ex}, \mathbf{SDEx}, \mathbf{PSDEx}, \mathbf{It}, \mathbf{ItCtr}\}$, (a) and (b) below.

(a) For all $\mathbf{M}$, L, and texts T for L, $\mathbf{M}$ *exhibits a U-shape on* $T \Leftrightarrow$ there exist i, j, and k such that $i < j < k$, $\{\mathbf{M}(T[i]), \mathbf{M}(T[j]), \mathbf{M}(T[k])\} \subset \mathbb{N}$, and

$$W_{\mathbf{M}(T[i])} = L \ \wedge\ W_{\mathbf{M}(T[j])} \neq L \ \wedge\ W_{\mathbf{M}(T[k])} = L. \tag{6}$$

(b) $\mathbf{NU}\mathcal{I} = \big\{\mathcal{L} : (\exists \mathbf{M})\big[\mathcal{L} \subseteq \mathcal{I}(\mathbf{M}) \wedge (\forall L \in \mathcal{L})[\mathbf{M} \text{ does } \textit{not} \text{ exhibit a U-shape on any text for } L]\big]\big\}$.

NU is mnemonic for *non-U-shaped*. Clearly, for all $\mathcal{I}$ as above, $\mathbf{NU}\mathcal{I} \subseteq \mathcal{I}$.

3 It = NUIt

In this section, we prove our main result (Theorem 2), namely, that U-shapes are *un*necessary in **It**-learning. The proof of this theorem is rather involved, however, the intuitive discussion in the next two paragraphs may be helpful.

The proof is a simulation argument. That is, given a class of languages $\mathcal{L} \in \mathbf{It}$, we fix a learner $\mathbf{M}$ such that $\mathcal{L} \subseteq \mathbf{It}(\mathbf{M})$. Then, we construct $\mathbf{M}'$ so that

$\mathcal{L} \subseteq \mathbf{It}(\mathbf{M}')$ and $\mathbf{M}'$ does *not* exhibit a U-shape on any text for a language in $\mathcal{L}$. $\mathbf{M}'$ operates, in part, by simulating $\mathbf{M}$, but *not* necessarily on the same inputs fed to $\mathbf{M}'$.

Suppose $L \in \mathbf{It}(\mathbf{M})$. For $\mathbf{M}$ to exhibit a U-shape on some text T for L, there must be some prefix σ of T and $w \in L \cup \{\#\}$ such that

$$W_{\mathbf{M}(\sigma)} = L \ \wedge\ W_{\mathbf{M}(\sigma \diamond w)} \neq L. \tag{7}$$

Thus, for such a pair (σ, w), we have

$$\mathbf{M}(\sigma \diamond w)\downarrow \neq \mathbf{M}(\sigma)\downarrow \ \wedge\ [w \neq \# \ \Rightarrow\ w \in W_{\mathbf{M}(\sigma)}]. \tag{8}$$

The set of all pairs, (σ, w), as in (8), is recursively enumerable. Thus, $\mathbf{M}'$ can, in the limit, discover the existence of each such pair (σ, w). Furthermore, these pairs serve as indicators that *perhaps* $\mathbf{M}$ is in the midst of a U-shape. By detecting these pairs, $\mathbf{M}'$ can avoid blindly following $\mathbf{M}$ into a U-shape.

Definition 4, below, introduces a notion that we call *canniness*. Intuitively, an $\mathbf{It}$-learner that is canny does not change its mind excessively, and, therefore, is much easier to reason about. Theorem 1, below, shows that, for any $\mathcal{L} \in \mathbf{It}$, there exists a canny learner that $\mathbf{It}$-identifies every language in $\mathcal{L}$. This fact is used in the proof of Theorem 2.

Definition 4 For all $\mathbf{M}$, $\mathbf{M}$ is *canny* $\Leftrightarrow$ for all σ, (a)-(c) below.

(a) $\mathbf{M}(\sigma)\downarrow \ \Rightarrow\ \mathbf{M}(\sigma) \in \mathbb{N}$, i.e., $\mathbf{M}$ never outputs ?.

(b) $\mathbf{M}(\sigma \diamond \#) = \mathbf{M}(\sigma)$.

(c) For all $x \in \mathbb{N}$, if $\mathbf{M}(\sigma \diamond x) \neq \mathbf{M}(\sigma)$, then, for all $\tau \supseteq \sigma \diamond x$, $\mathbf{M}(\tau \diamond x) = \mathbf{M}(\tau)$.

Theorem 1 For all $\mathcal{L} \in \mathbf{It}$, there exists $\mathbf{M}'$ such that $\mathcal{L} \subseteq \mathbf{It}(\mathbf{M}')$ and $\mathbf{M}'$ is canny.

Proof. Let $\mathcal{L} \in \mathbf{It}$ be fixed. Let $\mathbf{M}$ be such that $\mathcal{L} \subseteq \mathbf{It}(\mathbf{M})$. Let $f : \mathbb{N}_\#^* \to \mathbb{N}$ be a 1-1, computable function such that, for all σ,

$$W_{f(\sigma)} = \begin{cases} W_{\mathbf{M}(\sigma \diamond \#^m)}, & \text{where } m \text{ is } \textit{least} \text{ such that} \\ & \quad \mathbf{M}(\sigma \diamond \#^{m+1}) = \mathbf{M}(\sigma \diamond \#^m) \in \mathbb{N}, \\ & \quad \text{if such an } m \text{ exists;} \\ \emptyset, & \text{otherwise.} \end{cases} \tag{9}$$

Clearly, such an f exists. $\mathbf{M}'$ is such that $\mathbf{M}'(\lambda) = f(\lambda)$, and, for all ϱ and σ, and all $x \in \mathbb{N}_\#$, $[\mathbf{M}'(\varrho)\uparrow \ \Rightarrow\ \mathbf{M}'(\varrho \diamond x)\uparrow]$ and $\mathbf{M}'(\varrho) = f(\sigma) \ \Rightarrow\ \mathbf{M}'(\varrho \diamond x) =$

$$\begin{cases} \uparrow, & \text{if } (*)\ \mathbf{M}(\sigma)\uparrow \ \vee\ \mathbf{M}(\sigma \diamond \#)\uparrow \ \vee\ \mathbf{M}(\sigma \diamond \# \diamond x)\uparrow; \\ f(\sigma \diamond \# \diamond x), & \text{if } \neg(*) \ \wedge\ x \notin \text{content}(\sigma) \cup \{\#\} \\ & \quad \wedge \big[\mathbf{M}(\sigma \diamond \#) \neq \mathbf{M}(\sigma) \ \vee\ \mathbf{M}(\sigma \diamond \# \diamond x) \neq \mathbf{M}(\sigma)\big]; \\ f(\sigma), & \text{otherwise.} \end{cases} \tag{10}$$

Clearly, $\mathbf{M}'$ satisfies (a) and (b) in Definition 4. That $\mathbf{M}'$ satisfies (c) in Definition 4 is demonstrated by Claim 2 below. That $\mathcal{L} \subseteq \mathbf{It}(\mathbf{M}')$ is demonstrated by Claims 11 and 13 below.

Claim 1. Let T be any text. Then, (a) and (b) below.

(a) For all i, j, σ, and τ, if $\mathbf{M}'(T[i]) = f(\sigma)$, $\mathbf{M}'(T[j]) = f(\tau)$, and $i \leq j$, then $\sigma \subseteq \tau$.

(b) For all i and σ, if $\mathbf{M}'(T[i]) = f(\sigma)$, then $\text{content}(\sigma) \subseteq \text{content}(T[i])$.

Proof of Claim. Easily verifiable from the definition of $\mathbf{M}'$. □ (*Claim 1*)

Claim 2. Let T be any text. For all i, if $T(i) \in \mathbb{N}$ and $\mathbf{M}'(T[i+1]) \neq \mathbf{M}'(T[i])$, then, for all $j > i$ such that $T(j) = T(i)$, $\mathbf{M}'(T[j+1]) = \mathbf{M}'(T[j])$.

Proof of Claim. Let T be any text, and let i be such that $T(i) \in \mathbb{N}$ and $\mathbf{M}'(T[i+1]) \neq \mathbf{M}'(T[i])$. If $\mathbf{M}'(T[i+1])\uparrow$, then, clearly, $(\forall j > i)[\mathbf{M}'(T[j])\uparrow]$, and the claim is satisfied. So, suppose that $\mathbf{M}'(T[i+1])\downarrow$. Let σ be such that $\mathbf{M}'(T[i]) = f(\sigma)$. Clearly, by the definition of $\mathbf{M}'$, $\mathbf{M}'(T[i+1]) = f\big(\sigma \diamond \# \diamond T(i)\big)$. Let j be such that $j > i$ and $T(j) = T(i)$. If $\mathbf{M}'(T[j])\uparrow$, then, similarly, the claim is satisfied. So, let τ be such that $\mathbf{M}'(T[j]) = f(\tau)$. By Claim 1(a), $\sigma \diamond \# \diamond T(i) \subseteq \tau$, and, thus, $T(j) = T(i) \in \text{content}(\tau)$. Clearly, then, by the definition of $\mathbf{M}'$, $\mathbf{M}'(T[j+1]) = f(\tau)$. □ (*Claim 2*)

Let $L \in \mathcal{L}$ be fixed, and let T be a fixed text for L.

Claim 3. For all i, $\mathbf{M}'(T[i])\downarrow$.

Proof of Claim. It follows from Claim 1(b) that condition $(*)$ *never* applies as $\mathbf{M}'$ is fed T, and, thus, for all i, $\mathbf{M}'(T[i])\downarrow$. □ (*Claim 3*)

For all i, let σ_i be such that

$$\mathbf{M}'(T[i]) = f(\sigma_i). \tag{11}$$

By Claim 3, such σ_i exist. Let $k_0 = 0$, and let

$$\{k_1 < k_2 < \cdots\} = \big\{k > 0 : T(k-1) \notin \text{content}(\sigma_{k-1}) \cup \{\#\}\big\}. \tag{12}$$

Let $\eta \in \mathbb{N} \cup \{\omega\}$ be the order type [Rog67, Sie65, KM67] of (12). Thus, η is equal to the largest subscript occurring on the left hand side of (12), if (12) is finite; $\eta = \omega$, if (12) is infinite. Let $k_{1+\eta} = \omega$. (Recall: $1 + \omega = \omega$.)

Claim 4. $(\forall i < 1+\eta)(\forall \ell)[k_i \leq \ell < k_{i+1} \;\Rightarrow\; \sigma_\ell = \sigma_{k_i}]$.

Proof of Claim. By way of contradiction, suppose that $i < 1 + \eta$, and let ℓ be *least* such that $k_i \leq \ell < k_{i+1}$ and $\sigma_\ell \neq \sigma_{k_i}$. Clearly, $\ell > k_i$. Furthermore, by the definition of $\mathbf{M}'$, it must be the case that $T(\ell - 1) \notin \text{content}(\sigma_{\ell-1}) \cup \{\#\}$. If $k_{i+1} < \omega$, then $k_i - 1 < \ell - 1 < k_{i+1} - 1$, which contradicts the choice of k_{i+1}. On the other hand, if $k_{i+1} = \omega$, then $k_\eta - 1 = k_i - 1 < \ell - 1$, which contradicts the choice of η. □ (*Claim 4*)

Let T' be such that, for all i,

$$T'(2i) \;=\; \#; \tag{13}$$

$$T'(2i+1) \;=\; \begin{cases} T(k_{i+1} - 1), & \text{if } i < \eta; \\ \#, & \text{otherwise.} \end{cases} \tag{14}$$

Claim 5. T' is a text for L.

Proof of Claim. Clearly, $\text{content}(T') \subseteq \text{content}(T)$. So, suppose, by way of contradiction, that $\text{content}(T') \subset \text{content}(T)$. Let ℓ be *least* such that $T(\ell) \notin \text{content}(T') \cup \{\#\}$. Since ℓ is least such, clearly, $T(\ell) \notin \text{content}(T[\ell])$, and, by Claim 1(b), $T(\ell) \notin \text{content}(\sigma_\ell)$. Thus, there must exist $i < \eta$ such that $k_{i+1}-1 = \ell$. But then, clearly, $T'(2i+1) = T(k_{i+1}-1) = T(\ell)$ — a contradiction. □ (*Claim 5*)

Claim 6. For all i, $\mathbf{M}'(T'[i])\downarrow$.

Proof of Claim. The reasoning is, essentially, the same as that of Claim 3. □ (*Claim 6*)

For all i, let σ'_i be such that

$$\mathbf{M}'(T'[i]) = f(\sigma'_i). \tag{15}$$

By Claim 6, such σ'_i exist.

Claim 7. For all i, $\sigma'_{2i+1} = \sigma'_{2i}$.

Proof of Claim. Immediate by the definition of $\mathbf{M}'$ and the fact that, for all i, $T'(2i) = \#$. □ (*Claim 7*)

Claim 8.

(a) $(\forall i < 1 + \eta)[\sigma'_{2i} = \sigma_{k_i}]$.

(b) $(\forall i < \eta)[T'(2i + 1) \notin \text{content}(\sigma'_{2i+1}) \cup \{\#\}]$.

Proof of Claim. The proof is by simultaneous induction. Clearly, (a) holds in the case when $i = 0$, since $\sigma_{k_0} = \sigma_0 = \lambda$. So, suppose, inductively, that (a) holds for i, i.e., $\sigma'_{2i} = \sigma_{k_i}$. If $i < \eta$, then to see that (b) holds for i:

$$\begin{array}{rcll} T'(2i+1) & = & T(k_{i+1}-1) & \{\text{by the definition of } T'\} \\ & \notin & \text{content}(\sigma_{k_{i+1}-1}) \cup \{\#\} & \{\text{by the choice of } k_{i+1}\} \\ & = & \text{content}(\sigma_{k_i}) \cup \{\#\} & \{\text{by Claim 4}\} \\ & = & \text{content}(\sigma'_{2i}) \cup \{\#\} & \{\text{by (a) for } i\} \\ & = & \text{content}(\sigma'_{2i+1}) \cup \{\#\} & \{\text{by Claim 7}\}. \end{array}$$

If $i + 1 < 1 + \eta$, then to see that (a) holds for $i + 1$, consider the following cases.

CASE $\mathbf{M}(\sigma_{k_{i+1}-1} \diamond \#) \neq \mathbf{M}(\sigma_{k_{i+1}-1}) \ \vee\ \mathbf{M}\big(\sigma_{k_{i+1}-1} \diamond \# \diamond T(k_{i+1}-1)\big) \neq \mathbf{M}(\sigma_{k_{i+1}-1})$. By Claim 4, this case is equivalent to

$$\mathbf{M}(\sigma_{k_i} \diamond \#) \neq \mathbf{M}(\sigma_{k_i}) \ \vee\ \mathbf{M}\big(\sigma_{k_i} \diamond \# \diamond T(k_{i+1}-1)\big) \neq \mathbf{M}(\sigma_{k_i}); \tag{16}$$

by (a) for i and the definition of T', it is equivalent to

$$\mathbf{M}(\sigma'_{2i} \diamond \#) \neq \mathbf{M}(\sigma'_{2i}) \ \vee\ \mathbf{M}\big(\sigma'_{2i} \diamond \# \diamond T'(2i+1)\big) \neq \mathbf{M}(\sigma'_{2i}); \tag{17}$$

and by Claim 7, it is equivalent to

$$\mathbf{M}(\sigma'_{2i+1} \diamond \#) \neq \mathbf{M}(\sigma'_{2i+1}) \ \vee\ \mathbf{M}\big(\sigma'_{2i+1} \diamond \# \diamond T'(2i+1)\big) \neq \mathbf{M}(\sigma'_{2i+1}). \tag{18}$$

Thus,

$$\begin{array}{lll} & \sigma'_{2i+2} & \\ = & \sigma'_{2i+1} \diamond \# \diamond T'(2i+1) & \{\text{by the definition of } \mathbf{M}', \text{(b) for } i, \\ & & \quad \text{and (18)}\} \\ = & \sigma'_{2i} \diamond \# \diamond T'(2i+1) & \{\text{by Claim 7}\} \\ = & \sigma_{k_i} \diamond \# \diamond T(k_{i+1}-1) & \{\text{by (a) for } i \text{ and the definition of } T'\} \\ = & \sigma_{k_{i+1}-1} \diamond \# \diamond T(k_{i+1}-1) & \{\text{by Claim 4}\} \\ = & \sigma_{k_{i+1}} & \{\text{by the def. of } \mathbf{M}' \text{ and the case}\}. \end{array}$$

CASE $\mathbf{M}(\sigma_{k_{i+1}-1} \diamond \#) = \mathbf{M}(\sigma_{k_{i+1}-1}) \ \wedge \ \mathbf{M}\big(\sigma_{k_{i+1}-1} \diamond \# \diamond T(k_{i+1}-1)\big) = \mathbf{M}(\sigma_{k_{i+1}-1})$. By Claim 4, this case is equivalent to

$$\mathbf{M}(\sigma_{k_i} \diamond \#) = \mathbf{M}(\sigma_{k_i}) \ \wedge \ \mathbf{M}\big(\sigma_{k_i} \diamond \# \diamond T(k_{i+1}-1)\big) = \mathbf{M}(\sigma_{k_i}); \tag{19}$$

by the (a) for i and the definition of T', it is equivalent to

$$\mathbf{M}(\sigma'_{2i} \diamond \#) = \mathbf{M}(\sigma'_{2i}) \ \wedge \ \mathbf{M}\big(\sigma'_{2i} \diamond \# \diamond T'(2i+1)\big) = \mathbf{M}(\sigma'_{2i}); \tag{20}$$

and by Claim 7, it is equivalent to

$$\mathbf{M}(\sigma'_{2i+1} \diamond \#) = \mathbf{M}(\sigma'_{2i+1}) \ \wedge \ \mathbf{M}\big(\sigma'_{2i+1} \diamond \# \diamond T'(2i+1)\big) = \mathbf{M}(\sigma'_{2i+1}). \tag{21}$$

Thus,

$$\begin{array}{llll} \sigma'_{2i+2} & = & \sigma'_{2i+1} & \{\text{by the definition of } \mathbf{M}' \text{ and (21)}\} \\ & = & \sigma'_{2i} & \{\text{by Claim 7}\} \\ & = & \sigma_{k_i} & \{\text{by (a) for } i\} \\ & = & \sigma_{k_{i+1}-1} & \{\text{by Claim 4}\} \\ & = & \sigma_{k_{i+1}} & \{\text{by the definition of } \mathbf{M}' \text{ and the case}\}. \end{array}$$

□ (*Claim 8*)

Claim 9. For all $i < 1+\eta$, $\mathbf{M}(T'[2i]) = \mathbf{M}(\sigma'_{2i})$.

Proof of Claim. Clearly, the claim holds in the case when $i = 0$. So, let i be such that $i+1 < 1+\eta$ and suppose, inductively, that $\mathbf{M}(T'[2i]) = \mathbf{M}(\sigma'_{2i})$. Consider the following cases.

CASE $\mathbf{M}(\sigma'_{2i+1} \diamond \#) \neq \mathbf{M}(\sigma'_{2i+1}) \ \vee \ \mathbf{M}\big(\sigma'_{2i+1} \diamond \# \diamond T'(2i+1)\big) \neq \mathbf{M}(\sigma'_{2i+1})$. Then,

$$\begin{array}{lll} & \mathbf{M}(T'[2i+2]) & \\ = & \mathbf{M}\big(T'[2i] \diamond \# \diamond T'(2i+1)\big) & \{\text{by the definition of } T'\} \\ = & \mathbf{M}\big(\sigma'_{2i} \diamond \# \diamond T'(2i+1)\big) & \{\text{by the induction hypothesis}\} \\ = & \mathbf{M}\big(\sigma'_{2i+1} \diamond \# \diamond T'(2i+1)\big) & \{\text{by Claim 7}\} \\ = & \mathbf{M}(\sigma'_{2i+2}) & \{\text{by the definition of } \mathbf{M}', \\ & & \quad \text{Claim 8(b), and the case}\}. \end{array}$$

CASE $\mathbf{M}(\sigma'_{2i+1} \diamond \#) = \mathbf{M}(\sigma'_{2i+1}) \ \wedge\ \mathbf{M}\big(\sigma'_{2i+1} \diamond \# \diamond T'(2i+1)\big) = \mathbf{M}(\sigma'_{2i+1})$. Then,

$$\begin{array}{lll} & \mathbf{M}(T'[2i+2]) & \\ = & \mathbf{M}\big(T'[2i] \diamond \# \diamond T'(2i+1)\big) & \{\text{by the definition of } T'\} \\ = & \mathbf{M}\big(\sigma'_{2i} \diamond \# \diamond T'(2i+1)\big) & \{\text{by the induction hypothesis}\} \\ = & \mathbf{M}\big(\sigma'_{2i+1} \diamond \# \diamond T'(2i+1)\big) & \{\text{by Claim 7}\} \\ = & \mathbf{M}(\sigma'_{2i+1}) & \{\text{by the case}\} \\ = & \mathbf{M}(\sigma'_{2i+2}) & \{\text{by the def. of } \mathbf{M}' \text{ and the case}\}. \end{array}$$

□ (*Claim 9*)

Claim 10. If $\eta < \omega$, then, for all m,

$$\mathbf{M}(\sigma_{k_\eta} \diamond \#^{m+1}) \neq \mathbf{M}(\sigma_{k_\eta} \diamond \#^{m}) \ \Leftrightarrow\ \mathbf{M}(T'[2\eta+m+1]) \neq \mathbf{M}(T'[2\eta+m]). \quad (22)$$

Proof of Claim. Suppose that $\eta < \omega$. Then, for all m,

$$\begin{array}{lll} & \mathbf{M}(\sigma_{k_\eta} \diamond \#^{m+1}) \neq \mathbf{M}(\sigma_{k_\eta} \diamond \#^{m}) & \\ \Leftrightarrow & \mathbf{M}(\sigma'_{2\eta} \diamond \#^{m+1}) \neq \mathbf{M}(\sigma'_{2\eta} \diamond \#^{m}) & \{\text{by Claim 8(a)}\} \\ \Leftrightarrow & \mathbf{M}(T'[2\eta] \diamond \#^{m+1}) \neq \mathbf{M}(T'[2\eta] \diamond \#^{m}) & \{\text{by Claim 9}\} \\ \Leftrightarrow & \mathbf{M}(T'[2\eta+m+1]) \neq \mathbf{M}(T'[2\eta+m]) & \{\text{by the definition of } T'\}. \end{array}$$

□ (*Claim 10*)

Claim 11. If $\eta < \omega$, then $\mathbf{M}'$ **It**-identifies L from T.

Proof of Claim. Suppose that $\eta < \omega$. By Claim 4, for all $\ell \geq k_\eta$, $\sigma_\ell = \sigma_{k_\eta}$.

Let m be *least* such that $\mathbf{M}$ converges to $\mathbf{M}(T'[2\eta+m])$ on T'. Thus, $\mathbf{M}(T'[2\eta+m]) \in \mathbb{N}$ and $W_{\mathbf{M}(T'[2\eta+m])} = L$. By Claim 10, m is least such that $\mathbf{M}(\sigma_{k_\eta} \diamond \#^{m+1}) = \mathbf{M}(\sigma_{k_\eta} \diamond \#^{m})$. Thus, $W_{f(\sigma_{k_\eta})} = W_{\mathbf{M}(\sigma_{k_\eta} \diamond \#^m)}$. Furthermore,

$$\begin{array}{lll} & W_{\mathbf{M}(\sigma_{k_\eta} \diamond \#^m)} & \\ = & W_{\mathbf{M}(\sigma'_{2\eta} \diamond \#^m)} & \{\text{by Claim 8(a)}\} \\ = & W_{\mathbf{M}(T'[2\eta] \diamond \#^m)} & \{\text{by Claim 9}\} \\ = & W_{\mathbf{M}(T'[2\eta+m])} & \{\text{by the definition of } T'\} \\ = & L & \{\text{by the choice of } m\}. \end{array}$$

□ (*Claim 11*)

Claim 12. For all $i < \eta$,

$$\sigma_{k_{i+1}} \neq \sigma_{k_i} \ \Leftrightarrow\ \big[\mathbf{M}(T'[2i+1]) \neq \mathbf{M}(T'[2i]) \ \vee\ \mathbf{M}(T'[2i+2]) \neq \mathbf{M}(T'[2i])\big]. \quad (23)$$

Proof of Claim. For all $i < \eta$,

$$
\begin{array}{lll}
& \sigma_{k_{i+1}} \neq \sigma_{k_i} & \\
\Leftrightarrow & \sigma_{k_{i+1}} \neq \sigma_{k_{i+1}-1} & \{\text{by Claim 4}\} \\
\Leftrightarrow & \mathbf{M}(\sigma_{k_{i+1}-1} \diamond \#) \neq \mathbf{M}(\sigma_{k_{i+1}-1}) & \\
& \quad \vee\ \mathbf{M}\big(\sigma_{k_{i+1}-1} \diamond \# \diamond T(k_{i+1}-1)\big) & \{\text{by the def. of } \mathbf{M}' \text{ and} \\
& \qquad \neq \mathbf{M}(\sigma_{k_{i+1}-1}) & \quad \text{the choice of } k_{i+1}\} \\
\Leftrightarrow & \mathbf{M}(\sigma_{k_i} \diamond \#) \neq \mathbf{M}(\sigma_{k_i}) & \\
& \quad \vee\ \mathbf{M}\big(\sigma_{k_i} \diamond \# \diamond T(k_{i+1}-1)\big) \neq \mathbf{M}(\sigma_{k_i}) & \{\text{by Claim 4}\} \\
\Leftrightarrow & \mathbf{M}(\sigma'_{2i} \diamond \#) \neq \mathbf{M}(\sigma'_{2i}) & \\
& \quad \vee\ \mathbf{M}\big(\sigma'_{2i} \diamond \# \diamond T(k_{i+1}-1)\big) \neq \mathbf{M}(\sigma'_{2i}) & \{\text{by Claim 8(a)}\} \\
\Leftrightarrow & \mathbf{M}(T'[2i] \diamond \#) \neq \mathbf{M}(T'[2i]) & \\
& \quad \vee\ \mathbf{M}\big(T'[2i] \diamond \# \diamond T(k_{i+1}-1)\big) \neq \mathbf{M}(T'[2i]) & \{\text{by Claim 9}\} \\
\Leftrightarrow & \mathbf{M}(T'[2i+1]) \neq \mathbf{M}(T'[2i]) & \\
& \quad \vee\ \mathbf{M}(T'[2i+2]) \neq \mathbf{M}(T'[2i]) & \{\text{by the def. of } T'\}.
\end{array}
$$

□ (*Claim 12*)

Claim 13. If $\eta = \omega$, then $\mathbf{M}'$ **It**-identifies L from T.

Proof of Claim. Suppose that $\eta = \omega$. Let i be such that $\mathbf{M}$ converges to $\mathbf{M}(T'[2i])$ on T'. Thus, $\mathbf{M}(T'[2i]) \in \mathbb{N}$ and $W_{\mathbf{M}(T'[2i])} = L$. By Claim 12, for all $j \geq i$, $\sigma_{k_{j+1}} = \sigma_{k_j}$. It then follows from Claim 4 that, for all $\ell \geq k_i$, $\sigma_\ell = \sigma_{k_i}$.

Consider the behavior of $\mathbf{M}'$ on $T[k_{i+1}]$. By the choice of k_{i+1}, $T(k_{i+1}-1) \notin \text{content}(\sigma_{k_{i+1}-1}) \cup \{\#\}$. Thus, since $\sigma_{k_{i+1}} = \sigma_{k_i} = \sigma_{k_{i+1}-1}$, it must be the case that

$$
\begin{aligned}
& \mathbf{M}(\sigma_{k_{i+1}-1} \diamond \#) = \mathbf{M}(\sigma_{k_{i+1}-1}) \\
& \quad \wedge\ \mathbf{M}\big(\sigma_{k_{i+1}-1} \diamond \# \diamond T(k_{i+1}-1)\big) = \mathbf{M}(\sigma_{k_{i+1}-1}).
\end{aligned} \tag{24}
$$

Furthermore,

$$
\begin{array}{lll}
& W_{f(\sigma_{k_i})} & \\
= & W_{f(\sigma_{k_{i+1}-1})} & \{\text{by Claim 4}\} \\
= & W_{\mathbf{M}(\sigma_{k_{i+1}-1})} & \{\text{by the definition of } f \text{ and (24)}\} \\
= & W_{\mathbf{M}(\sigma_{k_i})} & \{\text{by Claim 4}\} \\
= & W_{\mathbf{M}(\sigma'_{2i})} & \{\text{by Claim 8(a)}\} \\
= & W_{\mathbf{M}(T'[2i])} & \{\text{by Claim 9}\} \\
= & L & \{\text{by the choice of } i\}.
\end{array}
$$

□ (*Claim 13*)

□ (*Theorem 1*)

Definition 5, just below, introduces notation used in the proof of Theorem 2.

Definition 5 For all $\mathbf{M}$ and σ, (a)-(d) below.

(a) $C_{\mathbf{M}}(\sigma) = \{x \in \mathbb{N}_\# : \mathbf{M}(\sigma \diamond x)\downarrow = \mathbf{M}(\sigma)\downarrow\}$.

(b) $B_{\mathbf{M}}(\sigma) = \{x \in \mathbb{N}_\# : \mathbf{M}(\sigma \diamond x)\downarrow \neq \mathbf{M}(\sigma)\downarrow\}$.

(c) $B^{\cap}_{\mathbf{M}}(\sigma) = \bigcap_{0 \leq i \leq |\sigma|} B_{\mathbf{M}}(\sigma[i])$.

(d) $CB_{\mathbf{M}}(\sigma) = \big(\bigcup_{0 \leq i < |\sigma|} C_{\mathbf{M}}(\sigma[i])\big) \cap B_{\mathbf{M}}(\sigma)$.

C is mnemonic for *cycle*. B is mnemonic for *branch*.

Lemmas 1-3, just below, are used in the proof of Theorem 2.

Lemma 1 Suppose that $\mathbf{M}$ and $\mathcal{L}$ are such that $\mathcal{L} \subseteq \mathbf{It}(\mathbf{M})$ and $\mathbf{M}$ is canny. Suppose that L and σ are such that $L \in \mathcal{L}$ and $\mathrm{content}(\sigma) \subseteq L$. Suppose, finally, that $L \cap B^{\cap}_{\mathbf{M}}(\sigma) = \emptyset$ and that $L \cap CB_{\mathbf{M}}(\sigma)$ is finite. Then, $W_{\mathbf{M}(\sigma)} = L$.

Proof (Sketch). Suppose the hypotheses. Let $A = L \cap CB_{\mathbf{M}}(\sigma)$. Clearly,

$$(\forall x \in A)(\exists \varrho \subset \sigma)[x \in C_{\mathbf{M}}(\varrho)]. \tag{25}$$

Furthermore, since $L \cap B^{\cap}_{\mathbf{M}}(\sigma) = \emptyset$,

$$L - A \subseteq C_{\mathbf{M}}(\sigma). \tag{26}$$

Consider a text T for L described, informally, as follows. T looks, initially, like σ with the elements of A interspersed. The elements of A are positioned in T in such a way that $\mathbf{M}$ does *not* make a mind-change when encountering these elements. The ϱ in (25) make this possible. Beyond this initial sequence resembling σ, T consists of the elements of $L - A$ and, possibly, pauses (#), in any order. Clearly, by (26) and the fact the $\mathbf{M}$ is canny, $\mathbf{M}$ converges to $\mathbf{M}(\sigma)$ on such a text T. Thus, it must be the case that $W_{\mathbf{M}(\sigma)} = L$. $\approx\Box$ (*Lemma 1*)

Lemma 2 Suppose that $\mathbf{M}$, $\mathcal{L}$, L, and σ are as in Lemma 1. Suppose, *in addition*, that L *is finite*. Then, for all τ such that $[\sigma \subseteq \tau \ \wedge\ \mathrm{content}(\tau) \subseteq L]$, $W_{\mathbf{M}(\tau)} = L$.

Proof. Suppose the hypotheses, and let τ be such that $\sigma \subseteq \tau$ and $\mathrm{content}(\tau) \subseteq L$. Since $L \cap B^{\cap}_{\mathbf{M}}(\sigma) = \emptyset$ and $\sigma \subseteq \tau$, clearly, $L \cap B^{\cap}_{\mathbf{M}}(\tau) = \emptyset$. Furthermore, since L is finite, $L \cap CB_{\mathbf{M}}(\tau)$ is finite. Thus, by Lemma 1, $W_{\mathbf{M}(\tau)} = L$. $\Box$ (*Lemma 2*)

Lemma 3 Suppose that $\mathbf{M}$ and $\mathcal{L}$ are such that $\mathcal{L} \subseteq \mathbf{It}(\mathbf{M})$. Suppose that L and σ are such that $L \in \mathcal{L}$ and $\mathrm{content}(\sigma) \subseteq L$. Suppose, finally, that $L \cap B_{\mathbf{M}}(\sigma)$ is infinite. Then, for all texts T for L, and all i, there exists $j \geq i$ such that $T(j) \in B_{\mathbf{M}}(\sigma)$.

Proof. Suppose the hypotheses. By way of contradiction, let T and i be such that, for all $j \geq i$, $T(j) \notin B_{\mathbf{M}}(\sigma)$. Then it must be the case that $L \cap B_{\mathbf{M}}(\sigma) \subseteq \{T(0), ..., T(i-1)\} \cap B_{\mathbf{M}}(\sigma)$. But since $L \cap B_{\mathbf{M}}(\sigma)$ is infinite and $\{T(0), ..., T(i-1)\} \cap B_{\mathbf{M}}(\sigma)$ is finite, this is a contradiction. $\Box$ (*Lemma 3*)

Theorem 2 **It** = **NUIt**.

Proof. Clearly, **NUIt** $\subseteq$ **It**. Thus, it suffices to show that **It** $\subseteq$ **NUIt**. Let $\mathcal{L} \in \mathbf{It}$ be fixed. Let $\mathbf{M}$ be such that $\mathcal{L} \subseteq \mathbf{It}(\mathbf{M})$. Without loss of generality, assume that $\mathbf{M}$ is canny. Let $p_{\mathbf{M}}$ be such that

$$\varphi_{p_{\mathbf{M}}} = \mathbf{M}. \tag{27}$$

Let $e : \mathbb{N}_{\#}^{*} \times \mathbb{N} \to \mathbb{N}$ be a partial computable function such that, for all σ, (a)-(c) below.

(a) $\mathrm{dom}\big(e(\sigma,\cdot)\big)$ is an initial segment of $\mathbb{N}$.

(b) $e(\sigma,\cdot)$ is 1-1.

(c) $\mathrm{rng}\big(e(\sigma,\cdot)\big) = W_{\mathbf{M}(\sigma)}$.

Clearly, such an e exists. Let $f : \mathbb{N}_{\#}^{*} \times \mathbb{N} \times \mathbb{N}_{\#}^{*} \to \mathbb{N}$ be a 1-1, computable function such that, for all σ, m, α, and q, if $f(\sigma, m, \alpha) = q$, then W_q is the least fixpoint of the following recursive definition.[8]

> Stage $s \geq 0$. If $e(\sigma, s){\downarrow}$, then let $x = e(\sigma, s)$, and let $A = W_q^s \cup \{x\}$. If *each* of (a)-(d) below is satisfied, then set $W_q^{s+1} = A$ and proceed to stage $s+1$; otherwise, go into an infinite loop thereby making W_q finite.
>
> (a) $e(\sigma, s){\downarrow}$.
>
> (b) $\mathbf{M}(\sigma \diamond x){\downarrow}$.
>
> (c) $x \in C_{\mathbf{M}}(\sigma) \cup CB_{\mathbf{M}}(\sigma)$.
>
> (d) $(\forall w \in A)[w \in CB_{\mathbf{M}}(\sigma) \Rightarrow w \leq m]$
> $\vee\ (\forall \tau)\big[[\sigma \subset \tau \ \wedge\ \mathrm{content}(\tau) \subseteq A \ \wedge\ |\tau| \leq |A|] \Rightarrow A \subseteq W_{f(\tau,0,\lambda)}\big]$.

Clearly, such an f exists.

Claim 1.

(a) For all σ, m, and α, $W_{f(\sigma,m,\alpha)} \subseteq W_{\mathbf{M}(\sigma)}$.

(b) For all σ, m, n, and α, if $m \leq n$, then $W_{f(\sigma,m,\alpha)} \subseteq W_{f(\sigma,n,\alpha)}$.

(c) For all σ, m, α, and β, $W_{f(\sigma,m,\alpha)} = W_{f(\sigma,m,\beta)}$.

Proof of Claim. Easily verifiable from the definition of f. $\square$ (*Claim 1*)

Let P be such that, for all σ and m, and all $x \in \mathbb{N}_{\#}$, $P(\sigma, m, x) \Leftrightarrow x \neq \#$ and

$$(\exists w)[\Phi_{\mathbf{M}(\sigma)}(w) \leq x \ \wedge\ \Phi_{p_{\mathbf{M}}}(\sigma \diamond w) \leq x \ \wedge\ w \in CB_{\mathbf{M}}(\sigma) \ \wedge\ m < w \leq x]. \tag{28}$$

[8] The requirement that the programs produced by f witness least fixpoints of this definition is, in fact, not necessary for the proof of the theorem. The requirement is here simply to make the definition of $W_{f(\cdot,\cdot,\cdot)}$ unambiguous.

Note that P is a computable predicate. Let $\mathbf{M}'$ be such that $\mathbf{M}'(\lambda) = f(\lambda, 0, \lambda)$, and, for all ϱ, σ, m, and α, and all $x \in \mathbb{N} \cup \{\#\}$, if $\mathbf{M}'(\varrho)\uparrow$, then $\mathbf{M}'(\varrho \diamond x)\uparrow$; furthermore, if $\mathbf{M}'(\varrho) = f(\sigma, m, \alpha)$, then $\mathbf{M}'(\varrho \diamond x)$ is:

$$\begin{array}{llll}
\uparrow, & & & \text{if (i) } \mathbf{M}(\sigma)\uparrow \vee \mathbf{M}(\sigma \diamond x)\uparrow \vee \mathbf{M}(\sigma \diamond \alpha)\uparrow \vee \mathbf{M}(\sigma \diamond \alpha \diamond x)\uparrow; \\
f(\sigma \diamond \alpha \diamond x, & 0, & \lambda \quad), & \text{if (ii) } \neg\text{(i)} \ \wedge \ \big[x \in B^{\cap}_{\mathbf{M}}(\sigma) \ \vee \ [x \in CB_{\mathbf{M}}(\sigma) \ \wedge \ x > m]\big]; \\
f(\sigma, & m, & \alpha \diamond x), & \text{if (iii) } \neg\text{(i)} \ \wedge \ x \in CB_{\mathbf{M}}(\sigma \diamond \alpha) \ \wedge \ x \leq m; \\
f(\sigma, & x, & \lambda \quad), & \text{if (iv) } \neg\text{(i)} \ \wedge \ x \in C_{\mathbf{M}}(\sigma \diamond \alpha) \ \wedge \ P(\sigma, m, x) \ \wedge \ \alpha = \lambda; \\
f(\sigma \diamond \alpha, & 0, & \lambda \quad), & \text{if (v) } \neg\text{(i)} \ \wedge \ x \in C_{\mathbf{M}}(\sigma \diamond \alpha) \ \wedge \ P(\sigma, m, x) \ \wedge \ \alpha \neq \lambda; \\
f(\sigma, & m, & \alpha \quad), & \text{if (vi) } \neg\text{(i)} \ \wedge \ x \in C_{\mathbf{M}}(\sigma \diamond \alpha) \ \wedge \ \neg P(\sigma, m, x).
\end{array}$$

Let $L \in \mathcal{L}$ be fixed, and let T be a fixed text for L.

Claim 2. For all i, $\mathbf{M}'(T[i])\downarrow$.

Proof of Claim. Clearly, for all i, σ, m, and α, if $\mathbf{M}'(T[i]) = f(\sigma, m, \alpha)$, then $\text{content}(\sigma) \cup \text{content}(\alpha) \subseteq \text{content}(T[i]) \subseteq L$. It follows that condition (i) *never* applies as $\mathbf{M}'$ is fed T, and, thus, for all i, $\mathbf{M}'(T[i])\downarrow$. □ (*Claim 2*)

For all i, let σ_i, m_i, and α_i be such that

$$\mathbf{M}'(T[i]) = f(\sigma_i, m_i, \alpha_i). \tag{29}$$

By Claim 2, such σ_i, m_i, and α_i exist.

Claim 3. For all i, (a)-(e) below.

(a) $\sigma_i \diamond \alpha_i \subseteq \sigma_{i+1} \diamond \alpha_{i+1} \subseteq \sigma_i \diamond \alpha_i \diamond T(i)$.

(b) If $T(i) \in B_{\mathbf{M}}(\sigma_i \diamond \alpha_i)$, then $\sigma_{i+1} \diamond \alpha_{i+1} = \sigma_i \diamond \alpha_i \diamond T(i)$.

(c) If $T(i) \in B^{\cap}_{\mathbf{M}}(\sigma_i)$, then $\sigma_{i+1} = \sigma_i \diamond \alpha_i \diamond T(i)$.

(d) If $\sigma_i = \sigma_{i+1}$, then $m_i \leq m_{i+1}$.

(e) $\mathbf{M}(T[i])\downarrow = \mathbf{M}(\sigma_i \diamond \alpha_i)\downarrow$.

Proof of Claim. (a)-(d) are easily verifiable from the definition of $\mathbf{M}'$. (e) follows from (a) and (b). □ (*Claim 3*)

Claim 4. There exists i such that, for all $j \geq i$, condition (vi) applies in calculating $\mathbf{M}'(T[j+1])$.

Proof of Claim. Suppose, by way of contradiction, that one or more of conditions (i)-(v) applies infinitely often as $\mathbf{M}'$ is fed T. By Claim 2, condition (i) *never* applies as $\mathbf{M}'$ is fed T. Also, note that, for all i, if condition (v) applies in calculating $\mathbf{M}'(T[i+1])$, then $\alpha_i \neq \lambda$ and $\alpha_{i+1} = \lambda$. Furthermore, for all i, if $\alpha_i = \lambda$ and $\alpha_{i+1} \neq \lambda$, then condition (iii) applies in calculating $\mathbf{M}'(T[i+1])$. Thus, if condition (v) applies infinitely often, then it must also be the case that condition (iii) applies infinitely often. Therefore, it suffices to consider the following cases.

> CASE condition (iii) applies infinitely often. Then, for infinitely many i, $T(i) \in B_{\mathbf{M}}(\sigma_i \diamond \alpha_i)$. Furthermore, by Claim 3(e), for infinitely many i, $T(i) \in B_{\mathbf{M}}(T[i])$. Thus, $\mathbf{M}$ does *not* converge on T — a contradiction.

Case condition (ii) applies infinitely often, but condition (iii) applies only finitely often. Let i be such that, for all $j \geq i$, condition (iii) does *not* apply in calculating $\mathbf{M}'(T[j+1])$. Let j be such that $j \geq i$ and $\alpha_j = \lambda$. Since condition (ii) applies infinitely often, such a j must exist.

Clearly, by the definition of $\mathbf{M}'$,

$$(\forall k \geq j)[\alpha_k = \lambda]. \tag{30}$$

Since condition (ii) applies infinitely often, for infinitely many $k \geq j$,

$$\begin{array}{rcll} T(k) & \in & B_{\mathbf{M}}(\sigma_k) & \\ & = & B_{\mathbf{M}}(\sigma_k \diamond \alpha_k) & \{\text{by (30)}\}, \\ & = & B_{\mathbf{M}}(T[k]) & \{\text{by Claim 3(e)}\}. \end{array}$$

Thus, $\mathbf{M}$ does *not* converge on T — a contradiction.

Case condition (iv) applies infinitely often, but conditions (ii) and (iii) apply only finitely often. Let i be such that, for all $j \geq i$, neither condition (ii) nor (iii) applies in calculating $\mathbf{M}'(T[j+1])$. Let j be such that $j \geq i$ and $\alpha_j = \lambda$. Since condition (iv) applies infinitely often, such a j must exist.

Clearly, by the definition of $\mathbf{M}'$,

$$(\forall k \geq j)[\sigma_k = \sigma_j \ \wedge \ \alpha_k = \lambda]. \tag{31}$$

Furthermore, for all $k \geq j$,

$$\begin{array}{rcll} \mathbf{M}(T[k]) & = & \mathbf{M}(\sigma_k \diamond \alpha_k) & \{\text{by Claim 3(e)}\} \\ & = & \mathbf{M}(\sigma_j) & \{\text{by (31)}\}. \end{array}$$

Thus, $\mathbf{M}$ converges to $\mathbf{M}(\sigma_j)$ on T, and, therefore, $W_{\mathbf{M}(\sigma_j)} = L$. Since condition (iv) applies infinitely often, it must be the case that $W_{\mathbf{M}(\sigma_j)} \cap CB_{\mathbf{M}}(\sigma_j)$ is infinite. Thus, $L \cap CB_{\mathbf{M}}(\sigma_j)$ is infinite. By Lemma 3, there exists $k \geq j$ such that $T(k) \in B_{\mathbf{M}}(\sigma_j)$. Thus, there exists $k \geq j$ such that $T(k) \in B_{\mathbf{M}}(\sigma_k \diamond \alpha_k)$. But then, clearly, condition (ii) or (iii) applies in calculating $\mathbf{M}'(T[k+1])$ — a contradiction.

□ (*Claim 4*)

Henceforth, let k_1 be *least* such that

$$(\forall i \geq k_1)[\text{condition (vi) applies in calculating } \mathbf{M}'(T[i+1])]. \tag{32}$$

By Claim 4, such a k_1 exists.

Claim 5. For all $i \geq k_1$, (a)-(g) below.

(a) $\sigma_i = \sigma_{k_1}$.

(b) $m_i = m_{k_1}$.

(c) $\alpha_i = \alpha_{k_1}$.

(d) $T(i) \in C_{\mathbf{M}}(\sigma_{k_1}) \cup CB_{\mathbf{M}}(\sigma_{k_1})$.

(e) $T(i) \in CB_{\mathbf{M}}(\sigma_{k_1}) \Rightarrow T(i) \leq m_{k_1}$.

(f) $\neg P\big(\sigma_{k_1}, m_{k_1}, T(i)\big)$.

(g) $\mathbf{M'}(T[i]) = \mathbf{M'}(T[k_1])$.

Proof of Claim. (a)-(f) follow from the definition of $\mathbf{M'}$ and the choice of k_1. (g) follows from (a)-(c). □ (*Claim 5*)

Claim 6. $L \cap B^{\cap}_{\mathbf{M}}(\sigma_{k_1}) = \emptyset$.

Proof of Claim. By way of contradiction, let x be such that $x \in L \cap B^{\cap}_{\mathbf{M}}(\sigma_{k_1})$. By Claim 5(d), there exists $i < k_1$ such that $T(i) = x$. Clearly, $x \in B^{\cap}_{\mathbf{M}}(\sigma_i)$. Thus, by Claim 3(c), it must be the case that $x \in \text{content}(\sigma_{k_1})$. But this contradicts the assumption that $\mathbf{M}$ is canny. □ (*Claim 6*)

Henceforth, let k_0 be *least* such that

$$L \cap B^{\cap}_{\mathbf{M}}(\sigma_{k_0}) = \emptyset. \tag{33}$$

By Claim 6, such a k_0 exists.

Claim 7. For all $i < k_0$, $L \not\subseteq W_{\mathbf{M'}(T[i])}$.

Proof of Claim. Let i be such that $i < k_0$. By the choice of k_0, there exists x such that $x \in L \cap B^{\cap}_{\mathbf{M}}(\sigma_i)$. Since $x \in B^{\cap}_{\mathbf{M}}(\sigma_i)$, clearly, by the definition of f, $x \notin W_{\mathbf{M'}(T[i])}$. □ (*Claim 7*)

Claim 8. If L is finite, then, for all σ' such that $[\sigma_{k_0} \subseteq \sigma' \ \wedge \ \text{content}(\sigma') \subseteq L]$, (a) and (b) below.

(a) $W_{\mathbf{M}(\sigma')} = L$.

(b) $W_{\mathbf{M}(\sigma')} \cap B^{\cap}_{\mathbf{M}}(\sigma') = \emptyset$.

Proof of Claim. (a) is immediate by Lemma 2. (b) follows from (a) and the choice of k_0. □ (*Claim 8*)

Let Q be such that, for all σ', $Q(\sigma') \Leftrightarrow$ for all τ,

$$[\sigma' \subset \tau \wedge \text{content}(\tau) \subseteq W_{\mathbf{M}(\sigma')} \wedge |\tau| \leq |W_{\mathbf{M}(\sigma')}|] \Rightarrow W_{\mathbf{M}(\sigma')} \subseteq W_{f(\tau,0,\lambda)}. \tag{34}$$

Claim 9. If L is finite, then, for all σ' such that $[\sigma_{k_0} \subseteq \sigma' \ \wedge \ \text{content}(\sigma') \subseteq L \wedge Q(\sigma')]$, $L \subseteq W_{f(\sigma',0,\lambda)}$.

Proof of Claim. Suppose that L is finite. Let σ' be such that $\sigma_{k_0} \subseteq \sigma'$, $\text{content}(\sigma') \subseteq L$, and $Q(\sigma')$. By Claim 8(a), $W_{\mathbf{M}(\sigma')} = L$. Consider the calculation of $f(\sigma', 0, \lambda)$. Clearly, if it can be shown that, for each stage s in which $e(\sigma', s)\!\downarrow$, conditions (b)-(d) are satisfied, then $L \subseteq W_{f(\sigma',0,\lambda)}$.

Let s be such that $e(\sigma', s)\!\downarrow$. Let x and A be as in stage s of the calculation of $f(\sigma', 0, \lambda)$. Since $x \in W_{\mathbf{M}(\sigma')} = L$, clearly, $\mathbf{M}(\sigma' \diamond x)\!\downarrow$. Furthermore, by

Claim 8(b), $W_{\mathbf{M}(\sigma')} \cap B^{\cap}_{\mathbf{M}}(\sigma') = \emptyset$. Thus, since $x \in W_{\mathbf{M}(\sigma')}$, $x \in C_{\mathbf{M}}(\sigma') \cup CB_{\mathbf{M}}(\sigma')$. Finally, since $Q(\sigma')$ and $A \subseteq W_{\mathbf{M}(\sigma')}$,

$$(\forall \tau)\big[[\sigma' \subset \tau \ \wedge \ \text{content}(\tau) \subseteq A \ \wedge \ |\tau| \leq |A|] \ \Rightarrow \ A \subseteq W_{f(\tau,0,\lambda)}\big]. \tag{35}$$

□ (*Claim 9*)

Claim 10. If L is finite, then, for all σ' such that $[\sigma_{k_0} \subseteq \sigma' \ \wedge \ \text{content}(\sigma') \subseteq L]$, $Q(\sigma')$.

Proof of Claim. Suppose that L is finite. Let σ' be such that $\sigma_{k_0} \subseteq \sigma'$ and $\text{content}(\sigma') \subseteq L$. By Claim 8(a), $W_{\mathbf{M}(\sigma')} = L$. Thus, if $|\sigma'| \geq |L|$, then $Q(\sigma')$ holds vacuously. So, suppose, inductively, that

$$(\forall \sigma'')\big[[\sigma_{k_0} \subseteq \sigma'' \ \wedge \ \text{content}(\sigma'') \subseteq L \ \wedge \ |\sigma'| < |\sigma''|] \ \Rightarrow \ Q(\sigma'')\big]. \tag{36}$$

Let τ be such that $\sigma' \subset \tau$ and $\text{content}(\tau) \subseteq W_{\mathbf{M}(\sigma')}$. Clearly, $\sigma_{k_0} \subseteq \tau$, $\text{content}(\tau) \subseteq L$, and $|\sigma'| < |\tau|$. Thus, by (36), $Q(\tau)$. Furthermore,

$$\begin{array}{rcll} W_{f(\tau,0,\lambda)} & \supseteq & L & \{\text{by Claim 9}\} \\ & = & W_{\mathbf{M}(\sigma')} & \{\text{by Claim 8(a)}\}. \end{array}$$

□ (*Claim 10*)

Claim 11. If L is finite, then, for all σ' such that $[\sigma_{k_0} \subseteq \sigma' \ \wedge \ \text{content}(\sigma') \subseteq L]$, $L \subseteq W_{f(\sigma',0,\lambda)}$.

Proof of Claim. Immediate by Claims 9 and 10. □ (*Claim 11*)

Claim 12. If L is finite, then, for all $i \geq k_0$, $W_{\mathbf{M}'(T[i])} = L$.

Proof of Claim. Suppose that L is finite, and let i be such that $i \geq k_0$. Clearly, by the definition of $\mathbf{M}'$, $\sigma_{k_0} \subseteq \sigma_i$. Thus,

$$\begin{array}{rcll} L & \subseteq & W_{f(\sigma_i,0,\lambda)} & \{\text{by Claim 11}\} \\ & \subseteq & W_{\mathbf{M}'(T[i])} & \{\text{by (b) and (c) of Claim 1}\} \\ & \subseteq & W_{\mathbf{M}(\sigma_i)} & \{\text{by Claim 1(a)}\} \\ & = & L & \{\text{by Claim 8(a)}\}. \end{array}$$

□ (*Claim 12*)

Claim 13. If L is finite, then, for all i, $W_{\mathbf{M}'(T[i])} = L \ \Leftrightarrow \ i \geq k_0$.

Proof of Claim. Immediate by Claims 7 and 12. □ (*Claim 13*)

Claim 14. If L is finite, then $\mathbf{M}'$ **It**-identifies L from T, and, furthermore, $\mathbf{M}'$ does *not* exhibit a U-shape on T.

Proof of Claim. Immediate by Claims 5(g) and 13. □ (*Claim 14*)

Claim 15. For all i such that $k_0 \leq i < k_1$, if $\sigma_i \neq \sigma_{i+1}$, then there exists $w \in (L \cup W_{\mathbf{M}(\sigma_i)}) \cap CB_{\mathbf{M}}(\sigma_i)$ such that $w > m_i$.

Proof of Claim. Let i be such that $k_0 \leq i < k_1$ and $\sigma_i \neq \sigma_{i+1}$. Clearly, one of the following cases must apply.

CASE condition (ii) applies in calculating $\mathbf{M}'(T[i+1])$. Then, clearly, $T(i) \in L \cap CB_{\mathbf{M}}(\sigma_i)$ and $T(i) > m_i$.

CASE condition (v) applies in calculating $\mathbf{M}'(T[i+1])$. Then, since $P(\sigma_i, m_i, T(i))$, clearly, there exists $w \in W_{\mathbf{M}(\sigma_i)} \cap CB_{\mathbf{M}}(\sigma_i)$ such that $w > m_i$.

□ (*Claim 15*)

Claim 16. For all i such that $k_0 \leq i < k_1$, if there exists j such that $i < j \leq k_1$ and $\sigma_i \neq \sigma_j$, then there exists $w \in (L \cup W_{\mathbf{M}(\sigma_i)}) \cap CB_{\mathbf{M}}(\sigma_i)$ such that $w > m_i$.

Proof of Claim. Let i be such that $k_0 \leq i < k_1$, and let j be *least* such that $i < j \leq k_1$ and $\sigma_i \neq \sigma_j$. By Claim 15, there exists $w \in (L \cup W_{\mathbf{M}(\sigma_{j-1})}) \cap CB_{\mathbf{M}}(\sigma_{j-1}) = (L \cup W_{\mathbf{M}(\sigma_i)}) \cap CB_{\mathbf{M}}(\sigma_i)$ such that $w > m_{j-1}$. Furthermore, by Claim 3(d), $m_{j-1} \geq m_i$, and, thus, $w > m_i$. □ (*Claim 16*)

Claim 17. If L is infinite, then, for all i and j such that $k_0 \leq i < j \leq k_1$, if $L \subseteq W_{\mathbf{M}'(T[i])}$, then $W_{\mathbf{M}'(T[i])} \subseteq W_{\mathbf{M}'(T[j])}$.

Proof of Claim. By way of contradiction, suppose that L is infinite, and let i and j be such that $k_0 \leq i < j \leq k_1$, $L \subseteq W_{\mathbf{M}'(T[i])}$, and $W_{\mathbf{M}'(T[i])} \not\subseteq W_{\mathbf{M}'(T[j])}$. By Claim 1(a), $L \subseteq W_{\mathbf{M}(\sigma_i)}$. By (b) and (c) of Claim 1, it must be the case that $\sigma_i \subset \sigma_j$. Thus, by Claim 16, there exists $w \in (L \cup W_{\mathbf{M}(\sigma_i)}) \cap CB_{\mathbf{M}}(\sigma_i) = W_{\mathbf{M}(\sigma_i)} \cap CB_{\mathbf{M}}(\sigma_i)$ such that $w > m_i$.

For all s, let x^s denote the value of x during stage s of the calculation of $f(\sigma_i, m_i, \alpha_i)$, and let A^s denote the contents of the set A during stage s of the calculation of $f(\sigma_i, m_i, \alpha_i)$. Choose s such that (a)-(f) below.

(a) $\mathbf{M}(\sigma_i \diamond x^s)\downarrow$.

(b) $x^s \in C_{\mathbf{M}}(\sigma_i) \cup CB_{\mathbf{M}}(\sigma_i)$.

(c) $w \in A^s$.

(d) $\text{content}(\sigma_j) \subseteq A^s$.

(e) $|\sigma_j| \leq |A^s|$.

(f) $A^s \not\subseteq W_{\mathbf{M}'(T[j])}$.

Clearly, such an s exists. However, since $A^s \not\subseteq W_{\mathbf{M}'(T[j])}$, by (b) and (c) of Claim 1, $A^s \not\subseteq W_{f(\sigma_j,0,\lambda)}$. Thus, by the definition of f, it must be the case that $W_{\mathbf{M}'(T[i])}$ is finite. But this contradicts $L \subseteq W_{\mathbf{M}'(T[i])}$. □ (*Claim 17*)

Claim 18. $L \cap CB_{\mathbf{M}}(\sigma_{k_1})$ is finite.

Proof of Claim. By Claim 5(e), $L \cap CB_{\mathbf{M}}(\sigma_{k_1}) \subseteq \text{content}(T[k_1]) \cup \{0, ..., m_{k_1}\}$. □ (*Claim 18*)

Claim 19. $W_{\mathbf{M}(\sigma_{k_1})} = L$.

Proof of Claim. Immediate by Claims 6 and 18, and by Lemma 1. □ (*Claim 19*)

Claim 20. If L is infinite, then $\max\big(L \cap CB_{\mathbf{M}}(\sigma_{k_1})\big) \le m_{k_1}$.

Proof of Claim. By way of contradiction, suppose that L is infinite, and let x be such that $x \in L \cap CB_{\mathbf{M}}(\sigma_{k_1})$ and $x > m_{k_1}$. Choose $i \ge k_1$ such that (a)-(c) below.

(a) $\Phi_{\mathbf{M}(\sigma_{k_1})}(x) \le T(i)$.

(b) $\Phi_{p_{\mathbf{M}}}(\sigma_{k_1} \diamond x) \le T(i)$.

(c) $x \le T(i)$.

By Claim 19 and the fact that L is infinite, such an i exists. Clearly, $P\big(\sigma_{k_1}, m_{k_1}, T(i)\big)$. But this contradicts Claim 5(f). □ (*Claim 20*)

Claim 21. If L is infinite, then $W_{\mathbf{M}'(T[k_1])} = L$.

Proof of Claim. Follows from Claims 6, 19, and 20, and from the definition of f. □ (*Claim 21*)

Claim 22. If L is infinite, then there exists i such that, for all j, $W_{\mathbf{M}'(T[j])} = L \Leftrightarrow j \ge i$.

Proof of Claim. Immediate by Claims 7, 17, and 21. □ (*Claim 22*)

Claim 23. If L is infinite, then $\mathbf{M}'$ **It**-identifies L from T, and, furthermore, $\mathbf{M}'$ does *not* exhibit a U-shape on T.

Proof of Claim. Immediate by Claims 5(g) and 22. □ (*Claim 23*)

□ (*Theorem 2*)

4 Iterative-with-Counter Learning

This section explores a learning model that we call *iterative-with-counter learning* (**ItCtr**-learning) (Definition 6, below). In this model, each of a learner's output conjectures can depend *only* upon the learner's most recent conjecture, the most recent input element, *and a counter* indicating the number of elements so far presented to the learner. Theorems 3 and 4, together, show that **ItCtr**-learning and **SDEx**-learning are incomparable, i.e., for each, there is a class of languages learnable by that one, but *not* the other. It follows that **ItCtr**-learning is strictly more powerful than **It**-learning, yet not as powerful as full **Ex**-learning. Finally, Problem 1, below, restates, formally, the problem that this paper leaves open.

ItCtr-learning was introduced in Definition 1(e) in Section 2, but is repeated here for convenience.

Definition 6

(a) For all $\mathbf{M}$ and L, $\mathbf{M}$ **ItCtr**-*identifies* $L \Leftrightarrow$ $\mathbf{M}$ **Ex**-identifies L, *and*, for all ϱ, σ, and τ such that $\mathrm{content}(\varrho) \cup \mathrm{content}(\sigma) \cup \mathrm{content}(\tau) \subseteq L$, (i) and (ii) below.

(i) $\mathbf{M}(\varrho)\!\downarrow$.

(ii) $[|\varrho| = |\sigma| \ \wedge \ \mathbf{M}(\varrho) = \mathbf{M}(\sigma)] \ \Rightarrow \ \mathbf{M}(\varrho \diamond \tau) = \mathbf{M}(\sigma \diamond \tau)$.

(b) For all $\mathbf{M}$, $\mathbf{ItCtr}(\mathbf{M}) = \{L : \mathbf{M}\ \mathbf{ItCtr}\text{-identifies } L\}$.

(c) $\mathbf{ItCtr} = \{\mathcal{L} : (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{ItCtr}(\mathbf{M})]\}$.

Theorem 3 (Based on [KS95, remark on page 238]) Let $\mathcal{L}$ be such that

$$\mathcal{L} = \big\{\{0, ..., m\} : m \in \mathbb{N}\big\} \cup \big\{\mathbb{N} - \{0\}\big\}. \tag{37}$$

Then, $\mathcal{L} \in \mathbf{SDEx} - \mathbf{ItCtr}$.

Proof. It is already known that $\mathcal{L} \in \mathbf{SDEx}$ [KS95, remark on page 238]. To see that $\mathcal{L} \notin \mathbf{ItCtr}$, suppose, by way of contradiction, that $\mathbf{M}$ is such that $\mathcal{L} \subseteq \mathbf{ItCtr}(\mathbf{M})$. For all s, let σ^s be as follows.

$$\sigma^0 \quad = \quad \lambda. \tag{38}$$

$$\sigma^{s+1} \quad = \quad \begin{cases} \sigma^s \diamond \#^n \diamond (s+1), & \text{where } n \text{ is } \textit{least} \text{ such that} \\ & \quad \mathbf{M}\big(\sigma^s \diamond \#^n\big) \neq \mathbf{M}(\sigma^s), \\ & \quad \text{if such an } n \text{ exists;} \\ \sigma^s \diamond (s+1), & \text{otherwise.} \end{cases} \tag{39}$$

Let $T = \lim_{s \to \infty} \sigma^s$. Clearly, T is a text for $\mathbb{N} - \{0\}$. Furthermore, since $\mathbf{M}$ **ItCtr**-identifies $\mathbb{N} - \{0\}$, the otherwise case in (39) must hold for all but finitely many s. Thus, there must exist i such that (a) and (b) below.

(a) $T(i) \neq \#$.

(b) $\mathbf{M}(T[i] \diamond \#)\!\downarrow = \mathbf{M}(T[i])\!\downarrow = \mathbf{M}(T[i+1])\!\downarrow$.

Let $L_0 = \text{content}(T[i]) \cup \{0\}$ and $L_1 = \text{content}(T[i+1]) \cup \{0\}$. Clearly, L_0 and L_1 are in $\mathcal{L}$. Furthermore, since no element of $\mathbb{N} - \{0\}$ appears in T more than once, by (a) above,

$$L_0 \neq L_1. \tag{40}$$

For all $j \in \{0, 1\}$, let T'_j be such that, for all $k \neq i$,

$$T'_j(k) = \begin{cases} T(k), & \text{if } k < i; \\ 0, & \text{if } k = i+1; \\ \#, & \text{otherwise;} \end{cases} \tag{41}$$

and, for $k = i$,

$$T'_0(k) \quad = \quad \#; \tag{42}$$

$$T'_1(k) \quad = \quad T(i). \tag{43}$$

Clearly, T'_0 is a text for L_0 and T'_1 is a text for L_1. Note that

$$\begin{array}{rcll} \mathbf{M}(T'_0[i+1])\downarrow & = & \mathbf{M}(T'_0[i]\diamond\#)\downarrow & \{\text{by (42)}\} \\ & = & \mathbf{M}(T[i]\diamond\#)\downarrow & \{\text{by (41)}\} \\ & = & \mathbf{M}\big(T[i]\diamond T(i)\big)\downarrow & \{\text{by (b) above}\} \\ & = & \mathbf{M}\big(T'_1[i]\diamond T(i)\big)\downarrow & \{\text{by (41)}\} \\ & = & \mathbf{M}(T'_1[i+1])\downarrow & \{\text{by (43)}\}. \end{array}$$

Let $k \geq i+1$ be such that, for all $j \in \{0,1\}$ and $\ell \geq k$, $\mathbf{M}(T'_j[\ell]) = \mathbf{M}(T'_j[k]) \in \mathbb{N}$. Then, by the above, and the fact the $\mathcal{L} \subseteq \mathbf{ItCtr}(\mathbf{M})$, $\mathbf{M}(T'_0[k]) = \mathbf{M}(T'_1[k])$. But then,

$$L_0 = W_{\mathbf{M}(T'_0[k])} = W_{\mathbf{M}(T'_1[k])} = L_1, \tag{44}$$

which contradicts (40). □ (*Theorem 3*)

Theorem 4 Let $\langle\cdot,\cdot\rangle : \mathbb{N}\times\mathbb{N}\to\mathbb{N}$ be any 1-1, onto, computable function [Rog67], and let $\mathcal{L}$ be such that

$$\mathcal{L} = \big\{\{\langle e,i\rangle : i\in\mathbb{N}\} : \varphi_e(0)\uparrow\big\} \cup \big\{\{\langle e,i\rangle : i \leq \varphi_e(0)\} : \varphi_e(0)\downarrow\big\}. \tag{45}$$

Then, $\mathcal{L} \in \mathbf{ItCtr} - \mathbf{SDEx}$.

Proof. To see that $\mathcal{L} \in \mathbf{ItCtr}$, consider the following. Let $f : \mathbb{N}\to\mathbb{N}$ be a 1-1, computable function such that, for all e,

$$W_{f(2e)} = \{\langle e,i\rangle : i\in\mathbb{N}\}; \tag{46}$$

$$W_{f(2e+1)} = \begin{cases} \{\langle e,i\rangle : i\leq\varphi_e(0)\}, & \text{if } \varphi_e(0)\downarrow; \\ \emptyset, & \text{otherwise.} \end{cases} \tag{47}$$

Clearly, such an f exists. Let $\mathbf{M}$ be such that $\mathbf{M}(\lambda) = ?$, and, for all ϱ, and all $x \in \mathbb{N}_\#$,

$$\mathbf{M}(\varrho\diamond x) = \begin{cases} f(2e), & \text{where } \mathbf{M}(\varrho) = ? \text{ and } (\exists i)[x = \langle e,i\rangle], \\ & \quad \text{if such an } e \text{ exists;} \\ f(2e+1), & \text{where } \mathbf{M}(\varrho) = f(2e) \text{ and } \Phi_e(0) \leq |\varrho\diamond x|, \\ & \quad \text{if such an } e \text{ exists;} \\ \mathbf{M}(\varrho), & \text{otherwise.} \end{cases} \tag{48}$$

Clearly, $\mathcal{L} \subseteq \mathbf{ItCtr}(\mathbf{M})$.

To see that $\mathcal{L} \notin \mathbf{SDEx}$, suppose, by way of contradiction, that $\mathbf{M}'$ is such that $\mathcal{L} \subseteq \mathbf{SDEx}(\mathbf{M}')$. By Kleene's Recursion Theorem [Rog67], there exists e such that, for all x,

$$\varphi_e(x) = \begin{cases} m, & \text{where } m \text{ is } \textit{any} \text{ such that } \mathbf{M}'(\langle e,0\rangle\diamond\cdots\diamond\langle e,m\rangle) \in \mathbb{N} \\ & \quad \text{and } W_{\mathbf{M}'(\langle e,0\rangle\diamond\cdots\diamond\langle e,m\rangle)} \not\subseteq \{\langle e,0\rangle,...,\langle e,m\rangle\}, \\ & \quad \text{if such an } m \text{ exists;} \\ \uparrow, & \text{otherwise.} \end{cases} \tag{49}$$

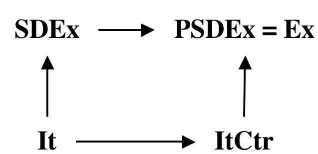
Consider the following cases.

CASE $\varphi_e(0)\uparrow$. Let $L = \{\langle e,i\rangle : i \in \mathbb{N}\}$. Clearly, $L \in \mathcal{L}$. By (49), it must be the case that, for all m, either $\mathbf{M}'(\langle e,0\rangle \diamond \cdots \diamond \langle e,m\rangle) \notin \mathbb{N}$ or $W_{\mathbf{M}'(\langle e,0\rangle\diamond\cdots\diamond\langle e,m\rangle)} \subseteq \{\langle e,0\rangle, ..., \langle e,m\rangle\}$. But then, $\mathbf{M}'$ does *not* identify L from the text $\langle e,0\rangle \diamond \langle e,1\rangle \diamond \cdots$ — a contradiction.

CASE $\varphi_e(0)\downarrow$. Let $m = \varphi_e(0)$ and $L = \{\langle e,i\rangle : i \leq m\}$. Clearly, $L \in \mathcal{L}$. By (49), it must be the case that $\mathbf{M}'(\langle e,0\rangle \diamond \cdots \diamond \langle e,m\rangle) \in \mathbb{N}$ and $W_{\mathbf{M}'(\langle e,0\rangle\diamond\cdots\diamond\langle e,m\rangle)} \not\subseteq \{\langle e,0\rangle, ..., \langle e,m\rangle\}$. Furthermore, since $\mathbf{M}'$ is set-driven, for all n, $\mathbf{M}'(\langle e,0\rangle \diamond \cdots \diamond \langle e,m\rangle \diamond \#^n) = \mathbf{M}'(\langle e,0\rangle \diamond \cdots \diamond \langle e,m\rangle)$. But then, $\mathbf{M}'$ does *not* identify L from the text $\langle e,0\rangle \diamond \cdots \diamond \langle e,m\rangle \diamond \# \diamond \# \cdots$ — a contradiction.

□ (*Theorem 4*)

Kinber, *et al.* [KS95, Theorem 7.7 and remark on page 238] showed that **It** $\subset$ **SDEx**. Schäfer-Richter [SR84] and Fulk [Ful90], independently, showed that **SDEx** $\subset$ **PSDEx** and that **PSDEx** $=$ **Ex**. Clearly, **It** $\subseteq$ **ItCtr** $\subseteq$ **Ex**. From these observations and Theorems 3 and 4, above, it follows that the *only* inclusions (represented by arrows) among **It**, **SDEx**, **ItCtr**, and **PSDEx** $=$ **Ex** are the following.



Problem 1 Is it the case that **ItCtr** $=$ **NUItCtr**?

References

[BCM+07] G. Baliga, J. Case, W. Merkle, F. Stephan, and W. Wiehagen. When unlearning helps, 2007. Submitted.

[Blu67] M. Blum. A machine independent theory of the complexity of recursive functions. *Journal of the ACM*, 14:322–336, 1967.

[Cas99] J. Case. The power of vacillation in language learning. *SIAM Journal on Computing*, 28(6):1941–1969, 1999.

[CCJS05] L. Carlucci, J. Case, S. Jain, and F. Stephan. Non U-shaped vacillatory and team learning. In *Proceedings of The 16th International Conference on Algorithmic Learning Theory (ALT'05)*, Lecture Notes in Artificial Intelligence, pages 241–255. Springer-Verlag, Berlin, 2005.

[CCJS06] L. Carlucci, J. Case, S. Jain, and F. Stephan. Memory-limited U-shaped learning. In *Proceedings of the 19th Annual Conference on Learning Theory (COLT'06)*, volume 4005 of *Lecture Notes in Artificial Intelligence*, pages 244–258. Springer-Verlag, Berlin, 2006. Journal version in proof for *Information and Computation*.

[CJLZ99] J. Case, S. Jain, S. Lange, and T. Zeugmann. Incremental concept learning for bounded data mining. *Information and Computation*, 152:74–110, 1999.

[CL82] J. Case and C. Lynes. Machine inductive inference and language identification. In M. Nielsen and E. Schmidt, editors, *Proceedings of the 9th International Colloquium on Automata, Languages and Programming*, volume 140 of *Lecture Notes in Computer Science*, pages 107–115. Springer-Verlag, Berlin, 1982.

[CM07] J. Case and S. E. Moelius. U-shaped, iterative, and iterative-with-counter learning. In *Proceedings of the 20th Annual Conference on Learning Theory (COLT'07)*, volume 4539 of *Lecture Notes in Artificial Intelligence*, pages 172–186. Springer-Verlag, Berlin, 2007.

[DSW94] M. Davis, R. Sigal, and E. Weyuker. *Computability, Complexity, and Languages*. Academic Press, second edition, 1994.

[FJO94] M. Fulk, S. Jain, and D. Osherson. Open problems in Systems That Learn. *Journal of Computer and System Sciences*, 49(3):589–604, December 1994.

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

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

[Jai06] S. Jain, 2006. Private communication.

[JORS99] S. Jain, D. Osherson, J. Royer, and A. Sharma. *Systems that Learn: An Introduction to Learning Theory.* MIT Press, Cambridge, Mass., second edition, 1999.

[KM67] K. Kuratowski and A. Mostowski. *Set Theory.* North-Holland, 1967.

[KS95] E. Kinber and F. Stephan. Language learning from texts: mind changes, limited memory, and monotonicity. *Information and Computation*, 123:224–241, 1995.

[LZ96a] S. Lange and T. Zeugmann. Incremental learning from positive data. *Journal of Computer and System Sciences*, 53:88–103, 1996.

[LZ96b] S. Lange and T. Zeugmann. Set-driven and rearrangement-independent learning of recursive languages. *Mathematical Systems Theory*, 6:599–634, 1996.

[MPU+92] G. Marcus, S. Pinker, M. Ullman, M. Hollander, T.J. Rosen, and F. Xu. *Overregularization in Language Acquisition.* Monographs of the Society for Research in Child Development, vol. 57, no. 4. University of Chicago Press, 1992. Includes commentary by H. Clahsen.

[OSW86] D. Osherson, M. Stob, and S. Weinstein. *Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists.* MIT Press, Cambridge, Mass., 1986.

[PM91] K. Plunkett and V. Marchman. U-shaped learning and frequency effects in a multilayered perceptron: implications for child language acquisition. *Cognition*, 38:43–102, 1991.

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

[Sie65] W. Sierpinski. *Cardinal and ordinal numbers.* PWN –Polish Scientific Publishers, 1965. Second revised edition.

[SR84] G. Schäfer-Richter. *Über Eingabeabhängigkeit und Komplexität von Inferenzstrategien.* PhD thesis, Rheinisch-Westfälische Technische Hochschule Aachen, Germany, 1984.

[TA02] N. A. Taatgen and J. R. Anderson. Why do children learn to say broke? A model of learning the past tense without feedback. *Cognition*, 86:123–155, 2002.

[WC80] K. Wexler and P. Culicover. *Formal Principles of Language Acquisition.* MIT Press, Cambridge, Mass, 1980.

[Wie76] R. Wiehagen. Limes-erkennung rekursiver funktionen durch spezielle strategien. *Electronische Informationverarbeitung und Kybernetik*, 12:93–99, 1976.