On Restrictions in Computational Language Learning

Master Thesis

submitted in partial fulfillment of the requirements for the degree of

Master of Science (M.Sc.)

in **Computer Science**

Friedrich Schiller University Jena
Department of Mathematics and Computer Science

Submitted by: Martin Friedrich Schirneck

born on February 21, 1990 in Gera

Supervisor: Prof. Dr. Tobias Friedrich

Advisor: Dr. Timo Kötzing

Jena, Winter Term 2014/15

Abstract

In 1990 Fulk [14] proved that partially set-drivenness (rearrangement-independence) does *not* weaken the power of unrestricted computational language learning. The question arises whether this result still holds if paired with various learning restrictions. We investigate the influence of two main categories of such restrictions, namely *content-based* and *delayable* ones. An adaption of Fulk's theorem is verified for content-based learning and some delayable restrictions regarding *U-shaped* learning. On the other hand, we give an example criterion of delayable learning—*explanatory learning from text by a strongly monotone scientist*—for which partially set-drivenness does *reduce* the learning power. Additionally, the interdependence of these restrictions with several other interaction operators and success criteria are explored.

Contents

List of Symbols

Whenever a variable ranges over a certain domain, all its variants—with or without possible decorations such as sub- or superscripts—are intended to range over the same domain if not explicitly stated otherwise.

$\mathbb{N} = \{0; 1; 2; \dots\}$	set of all natural numbers
$e; i; j; k; m; n; t; x; y$	numbers (elements of $\mathbb{N}$)
$\#$	pause symbol
$A; B$	sets (subsets of $\mathbb{N}$)
D	finite set (finite subset of $\mathbb{N}$)
$\mathcal{P}_{\text{fin}}(\mathbb{N})$	collection of all finite sets
$\mathbb{N}^*$	set of all finite sequences of natural numbers
$[n] = (0; 1; \dots; n-1)$	initial sequence of the first n natural numbers
$\sigma; \tau$	finite sequences (elements of $(\mathbb{N} \cup \{\#\})^*$)
$\text{content}(\cdot)$	content
$\sqsubseteq$	prefix
$\diamond$	concatenation
$\lvert\cdot\rvert$	cardinality; length
$\emptyset$	empty set; empty sequence
$\in$	element of
$\cap$	intersection
$\cup$	union
$\setminus$	set difference
$\times$	Cartesian product
$\subseteq$	subset
$\subset_{\text{fin}}$	finite subset
$\subsetneq$	proper subset
$\wedge$	logical and
$\vee$	logical or
$\Rightarrow$	logical consequence
$\exists$	there is at least one
$\exists^\infty$	there are infinitely many
$\forall$	for all
$\forall^\infty$	for all but finitely many

$+$	plus
$-$	minus
$\leq$	less than or equal to
$<$	strictly less than
$=$	equal to
$\neq$	unequal to
$\mathcal{E}$	collection of all r.e. sets of natural numbers
$\mathcal{L}$	class of languages (subset of $\mathcal{E}$)
L	language (element of $\mathcal{E}$)
W	effective numbering of all r.e. sets
$\mathfrak{R}$	set of all total numerical functions
$\mathfrak{P}$	set of all partial numerical functions
$p; q$	infinite sequences (elements of $\mathfrak{P}$)
$\mathrm{dom}(\cdot)$	domain
$\mathrm{range}(\cdot)$	range
$\circ$	composition
$\mapsto$	mapped to
$\rightarrow$	total mapping
$\rightsquigarrow$	partial mapping
T	text (mapping $\mathbb{N} \rightarrow \mathbb{N} \cup \{\#\}$)
$T[n]$	initial sequence of text T of length n (mapping $[n] \rightarrow \mathbb{N} \cup \{\#\}$)
$\mathbf{Txt}$	set of all texts
$\mathbf{Txt}(L)$	set of all texts for language L
$\mathcal{R}$	set of all total recursive functions
$\mathcal{P}$	set of all partial recursive functions
$f; g; h$	partial recursive functions (elements of $\mathcal{P}$)
$\langle \cdot; \cdot \rangle$	pairing function
$\langle \cdot \rangle$	coding function
pad	padding function
φ	acceptable programming system
Φ	complexity measure associated with φ
$\varphi_i(x)\downarrow$	computation converges
$\varphi_i(x)\uparrow$	computation diverges
$[I]$	collection of all I-learnable classes of languages
$\square$	end of definition
$\blacksquare$	end of theorem
qed	end of proof

1 Introduction

1.1 Inductive Inference

Inductive inference, or *empirical inquiry* [17], is the art and science of concluding general principles from observable instances. This discipline of epistemology aims to abstract from the actual given environment and the incomplete, sometimes contradictory data it provides. The ultimate goal is a deep insight into the laws defining nature. Although, as opposed to *deductive* reasoning, the truth of these insights can not be guaranteed, inductive inference is an indispensable tool of human comprehension. In fact, virtually any human knowledge has its source in abstraction from our shallow, incomplete perception of reality. We call this *learning*. Throughout the history many philosophers asked the same question:

How can human beings reach thorough understanding of their surroundings from just limited, personal contact?

Learning theory, at the boundary of philosophy, psychology, neuroscience and sociology, tries to give a scientific explanation. Unfortunately though, the vast majority of cognitive processes in human knowledge is still widely unknown. *Computational learning theory*, that is, inductive inference by algorithmic devices, adds to this discussion a somehow mechanical point of view. It is based on a model of learning that is portraying a learner as a mere apparatus transforming the data it perceives to a stream of guesses about the world we live in. This stream, in case of success, may stabilize to an explanation of the phenomena all around. Just like a child reaches a stable idea of the grammar of the English language or a scientist reaches a formula describing planetary movement over time. Computational learning theory is divided into two main branches, *algorithmic function learning* and *algorithmic language learning*. This thesis investigates a certain topic in the latter. We want to know what information is necessary to enable a machine to recognize languages, human as well as artificial ones, what set of instruments an algorithm needs to succeed in the task of language identification. As we only examine hypothetical inference machines, we have to place our reasoning on a theoretical foundation. For this, we will utilize the mathematical theory of recursive computability.

We would like to take this opportunity to express our sincere thanks to our advisor Dr. Timo Kötzing. He was the one introducing us to the theory of computation and algorithmic learning in the first place and he stood, with word and deed, at our side all through the preparation and composition of this thesis. Thank you.

We will use the remainder of this preliminary section to summarize the computational background necessary for algorithmic learning. In section 2 we establish a first criterion of formal language learning. This is then generalized to a classification scheme of computational learning due to Kötzing [20]. In the same section a seminal result of Fulk [14] regarding language learning is shown. In the last two sections we present our own research. We are investigating

rearrangement-independent learning paired with several additional restrictions. One of these restrictions, strongly monotone learning, stood out to us and is examined intensively in section 4.

1.2 Computational Preliminaries

We give a brief overview on the mathematical foundations we need to argue about language learning. As the name suggests, in order to formalize *computational learning* the theory of recursive computability can be taken to good use. It provides us with a developed framework as well as a widely accepted language and notation. We expect the reader to have some background in the field since we can only mention the upmost important definitions and theorems. For an extensive treatment of the matter we strongly recommend ROGERS [28]. Regarding recursive functions we will follow his notational conventions. A general list of symbols be found at the beginning of this work.

Let $\mathbb{N} = \{0; 1; 2; \dots\}$ be the set of all non-negative integers and let such numbers $e; i; j; k; m; n$ serve as indices, t as a measure (of time or of length) and $x; y$ as further set members or function inputs. For each at most countable set A, let A^* denote the set of all finite sequences of elements of A. For any index n, $[n] = (0; 1; 2; \cdots; n-1)$ shall be the initial sequence of the first n natural numbers. Symbols $\sigma; \tau \in \mathbb{N}^*$ stand for finite sequences of natural numbers. The number of elements of a sequence σ, its *length*, is denoted $|\sigma|$ and the set of all natural numbers appearing in σ, its *content*, is denoted $\mathrm{content}(\sigma)$. It will be useful sometimes to conceive finite sequences as partial functions with finite domain $[|\sigma|]$, so $\sigma(i)$ is referring to the i-th entry of σ. The *concatenation* of sequences $\sigma; \tau$ (in this order) is written $\sigma \diamond \tau$, multiple concatenations of the same sequence is denoted, for example, $\sigma \diamond \tau^t$. The set $\mathbb{N}^*$ is partially ordered by the prefix relation: A sequence $\tau \sqsubseteq \sigma$ is a *prefix* just in case $|\tau| \leq |\sigma|$ and $\forall i \leq |\tau| \colon \tau(i) = \sigma(i)$. Intuitively, the sequence σ "starts with" prefix τ.

Symbols $f; g; h$, with or without subscripts denote (possibly partial) numerical functions $\mathbb{N} \rightsquigarrow \mathbb{N}$. $\mathfrak{P}$ and $\mathfrak{R}$, respectively, denote the collections of all partial and total numerical functions. Most of the time the functions we speak of will be computable ones. So let $\mathcal{R}$ be the collection of all total computable (*recursive*) functions and $\mathcal{P}$ be the set of all partial recursive functions. Let $\mathrm{dom}(f)$ and $\mathrm{range}(f)$ denote the *domain* and *range* of function f, respectively. Throughout this work we fix an *acceptable programming system* φ [28], e.g. the Gödel numbering of all deterministic multi-tape Turing machines. That means, we write $\varphi_i(x)\!\downarrow$ just in case the i-th Turing machine holds on input x and $\varphi_i(x)\!\uparrow$ otherwise. Via φ natural numbers become both inputs of and (codings of) *programs* of computable functions. Note that $\{\varphi_i\}_{i \in \mathbb{N}}$ forms a numbering of all partial recursive functions, but each one of them has infinitely many φ-indices. Furthermore, we fix a complexity measure Φ associated with φ [6]. $\Phi_i(x)$ can be thought of as a clockwork counting the number of steps the i-th Turing machine takes on calculating the value $\varphi_i(x)$. It is important to note that, for each $i; x$ and t, $\Phi_i(x) \leq t$ is *decidable*, while $\varphi_i(x)\!\downarrow$ is not (Halting Problem). Slightly abusing notation, we will also write $p(x)\!\downarrow$ indicating that some partial but maybe non-recursive function $p \in \mathfrak{P}$ is defined at point x.

Let $\langle \cdot; \cdot \rangle \colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be a total recursive onto *pairing* function. By

left-association of $\langle\cdot;\cdot\rangle$ we indeed get a total recursive onto *coding* function $\langle\cdot\rangle$. It is possible to code any finite set or sequence of natural numbers and even finite collections thereof into $\mathbb{N}$ [28]. W.l.o.g. we can assume this coding to be monotone, that is, $\tau \sqsubseteq \sigma$ implies $\langle\tau\rangle \leq \langle\sigma\rangle$. To put this the other way around, via $\langle\cdot\rangle$ we can conceive any set of sequences as a subset of $\mathbb{N}$ and therefore to be well-ordered. The same goes for finite sets $D \subset_{\text{fin}} \mathbb{N}$, i.e. $D \subseteq D' \Rightarrow \langle D\rangle \leq \langle D'\rangle$. Whenever a function with arity greater than 1 is used, we tacitly consider the input to be coded, e.g. $f(x;y) = f(\langle x;y\rangle)$. Note that these pairing and coding functions are recursive isomorphisms, thus, computably invertible. For example, there is a total recursive function $\pi_1 \in \mathcal{R}$ such that $\forall x;y\colon \pi_1(\langle x;y\rangle) = x$.

There are some basic but useful theorems about recursive functions such as the *Parameter Theorem* (s-m-n Theorem), *Kleene's Recursion Theorem* (KRT), cf. Rogers [28], and the *Operator Recursion Theorem* (ORT) due to Case [7]. We will only mention them here and omit their proofs.

Theorem 1.1 (Parameter Theorem, s-m-n Theorem; cf. Rogers [28])**:**
For any partial recursive function $f \in \mathcal{P}$, there is a total recursive $s \in \mathcal{R}$ such that

$$\forall x;y\colon \varphi_{s(x)}(y) = f(x;y).$$

Moreover, s can be chosen to be one-one and strictly monotone increasing.

■

Theorem 1.2 (Kleene's Recursion Theorem, KRT; cf. Rogers [28])**:**
For any partial recursive $f \in \mathcal{P}$, there is an index e such that

$$\forall x\colon \varphi_e(x) = f(e;x).$$

■

For full understanding of the Operator Recursion Theorem one should define the notion of an *effective operator* in sufficient rigor first. However, for our purpose it is enough to think of an operator as any computable mapping $\Theta\colon \mathfrak{P} \to \mathfrak{P}$ between (partial, not necessarily recursive) numerical functions. That is, whenever a Turing machine is successively fed the graph of a function f as input and outputs the graph of another function, say g, this transformation is regarded as an application of an effective operator, $\Theta(f) = g$.

Theorem 1.3 (Operator Recursion Theorem, ORT; Case [7])**:**
For any effective operator Θ, there is a total recursive $e \in \mathcal{R}$ such that

$$\forall i;x\colon \varphi_{e(i)}(x) = \Theta(e)(i;x).$$

Moreover, e can be chosen to be one-one and strictly monotone increasing.

■

Intuitively speaking, when using multiple inputs for a computation, the Parameter Theorem allows us to code all the information stored in the first input directly to a program which then carries out this very computation on the remaining inputs.

Thus, the s-m-n Theorem is a tool for building new computational functions from the ones we already know. The two recursion theorems are even more powerful: They deploy the mentioned self-referential characteristics of computation—natural numbers both as programs and inputs of recursive functions—to gain programs that are in some sense aware of themselves and the functions they are computing. While KRT hands us a single such index applying a given computation onto *itself*, ORT even allows us to use infinitely many of them, each one equally aware of all others. Moreover, Kleene's theorem refers to codings of programs while the Operator Recursion Theorem is employed directly on the level of recursive functions. Since learners are conceived as partial recursive mappings, the extensive use of ORT is very common in computational learning theory. Most of the time the operator will only be stated implicitly and will heavily rely on the context of the computation.

2 Computational Language Learning

In this thesis we are concerned with an algorithmic model of language acquisition. It should be equally able to depict aspects of both child learning and scientific discovery. For the latter, one may have to extend one's understanding of a *language*: We have already seen that natural numbers can stand as ciphers for arbitrary objects of interest, as long as they are drawn from an at most countable universe. As a matter of fact, this includes measuring physical quantities as well. Even real numbers, members of a denumerable domain, can be approximated and eventually coded using natural numbers [17]. Once one accepts a set of natural numbers as a language, applications of language learning appear in all disciplines of science. Unfortunately though, since we limit ourself to *computational* learning, not all sets of natural numbers can serve as target languages likewise. We will approach this issue below. As we mentioned before, inductive inference in terms of computational theory provides us with an already developed collection of ideas. In this section we will outline this environment to an extent we find useful for the presentation of our own work in the following chapters.

2.1 Gold-style Explanatory Learning from Text

Following Jain et al. [17], every *learning paradigm* has to implement five essential concepts, namely:

- *a theoretically possible reality*
- *intelligible hypotheses*
- *the data available about any given reality, were it actual*
- *a scientist*
- *successful behavior by a scientist working in a given, possible reality*

To establish the foundations of computational language learning we will also draw on the great work of Gold, who, in his seminal paper *Language Identification in the Limit* [16], founded this branch of research. We will employ these frameworks to define the most common paradigm in algorithmic language learning, *Gold-style explanatory language identification from text* (**TxtGEx**-learning, cf. [20]). It will exemplify the usual techniques of defining and modifying learning paradigms. For naming and grading concrete criteria we will use the unified approach of Kötzing [20] in turn.

We are investigating language acquisition or, more precisely, learning of *formal languages*. This means any *recursively enumerable* (r.e.) set of natural numbers. A set $A \subseteq \mathbb{N}$ is called recursively enumerable if there is a partial recursive function $f \in \mathcal{P}$ satisfying that $f(x) = 1$ if and only if $x \in A$. A well-known proposition in the theory of computation states that a set is r.e. just in case it is the domain of a partial recursive function [28]. For any index i, $W_i = \mathrm{dom}(\varphi_i)$ shall denote the domain of the i-th partial recursive function. This is a numbering onto $\mathcal{E}$, the

collection of all r.e. sets, thus, $\mathcal{E} = \{W_i\}_{i\in\mathbb{N}}$. This sets up our *theoretically possible realities* and a space of *intelligible hypotheses* [17] at the same time. Recall that, from the ambiguity of system φ, every single r.e. set has infinitely many W-indices. In fact, there is even a total recursive *padding function* pad $\in \mathcal{R}$ strongly monotone increasing such that, for all i and x, $W_{\mathrm{pad}(i;x)} = W_i$ [28]. Again, pad is recursively invertible.

It remains open what *a scientist*—we prefer the term *learner*—really is and how it can witness information about its learnee. Gold's idea [16] was to present a formal language $L \in \mathcal{E}$ as a *text*, an infinite stream of positive examples of members of L.

Definition 2.1 (Gold [16])**:**

(I) A *text* is any mapping $T\colon \mathbb{N} \to \mathbb{N} \cup \{\#\}$.

(II) The *content* of a text T is the set of natural numbers appearing in T and is denoted content(T).

(III) A text T is *for* a language $L \in \mathcal{E}$ if content$(T) = L$.

(IV) The collection of all texts is denoted **Txt**. For any language $L \in \mathcal{E}$, the set of all texts for L is denoted **Txt**(L).

□

So texts for languages are (not necessarily recursive) enumerations of their elements, possibly stretched by a special symbol #, read *pause*. Such enumerations could list certain members arbitrarily often while others appear only once, as long as the whole set is shown eventually. The content of a text extends the notion of the content of a finite sequence in a natural way, ignoring pause symbols. Some further discussion revolves around pauses in texts since they do not provide any additional information about the target language. As a consequence, some authors suggested deviant definitions of texts making pauses obsolete. This problem arises particularly in the field of *function learning*, see, for example, Blum & Blum [5] or Fulk [15].

Since the essence of learning—as we see it—is to find a finite representation of an infinite object in a finite amount of time, a text cannot be presented to a learner as a whole. Instead, we use longer and longer initial sequences (elements of $(\mathbb{N} \cup \{\#\})^*$) as input. This is acknowledging the fact that a learner might need more and more information, and more and more time, in order to identify a certain language. Therefore, symbols $\sigma; \tau$ will from now on comprise sequences including pause symbols.

Definition 2.2 (Gold [16])**:**
Suppose $T \in$ **Txt** to be a text. For each n, let

$$T[n]\colon \{0; \ldots; n{-}1\} \to \mathbb{N} \cup \{\#\};\ i \mapsto T(i)$$

be the initial sequence (prefix) of T of length n.

□

The symbol #, of course, can be coded as a natural number. This allows us to conceive finite sequences $T[n]$ to be coded into $\mathbb{N}$ as well. Therefore, they are eligible inputs for recursive functions. We can now straightforwardly define learners to be computational mappings from initial sequences of texts to hypotheses for formal languages.

Definition 2.3:
A *learner* is any partial recursive mapping $(\mathbb{N} \cup \{\#\})^* \rightsquigarrow \mathbb{N}$.
By coding, the collection of all learners equals $\mathcal{P}$.

□

We still need to define what successful learning shall mean in our setting.

Definition 2.4 (Gold [16]; cf. Kötzing [20])**:**

(I) A learner $h \in \mathcal{P}$ **TxtGEx**-*identifies* a text T if

$$\exists e \, \forall^\infty n \colon h(T[n])\downarrow = e \wedge W_e = \text{content}(T).$$

In this case h is said to *converge to* e on T.

(II) A learner $h \in \mathcal{P}$ **TxtGEx**-*identifies* a language $L \in \mathcal{E}$ if it **TxtGEx**-identifies every text for L, written $L \in \mathbf{TxtGEx}(h)$.

(III) A learner $h \in \mathcal{P}$ **TxtGEx**-*identifies* a class $\mathcal{L} \subseteq \mathcal{E}$ of languages if h **TxtGEx**-identifies every element $L \in \mathcal{L}$, written $\mathcal{L} \subseteq \mathbf{TxtGEx}(h)$.
In this case $\mathcal{L}$ is said to be **TxtGEx**-*identifiable*.

(IV) $[\mathbf{TxtGEx}] := \{\mathcal{L} \subseteq \mathcal{E} \mid \exists \, h \in \mathcal{P} \colon \mathcal{L} \subseteq \mathbf{TxtGEx}(h)\}$ is the collection of all **TxtGEx**-identifiable classes of languages.

□

To converge on a text T it is necessary for a learner to be undefined on at most finitely many initial sequences $T[n]$. W.l.o.g. we even can assume **TxtGEx**-learners to be defined on *all* initial sequences of texts for languages they identify [17]. As we mentioned above, the name of the criterion derives from *Gold-style explanatory learning from text* [20]. It requires the learner to finally yield a sole explanation of the learnee (a W-index of it), the learner is given access to the whole input sequence, which was first investigated by Gold [16], and the language is presented as a text. The intuition behind Definition 2.4 is that finitely many initial guesses of a learner might as well be wrong, but in order to be successful the learner must from one point on repeatedly output the same correct hypothesis for the target language. This setting therefore was originally named *Language Identification in the Limit* [16]. Every singleton subset of $\mathcal{E}$ is trivially identifiable by a constant learner which just happen to output a correct index of the target language. In this case no real inference takes place. That is why we are interested in classes of languages learnable by a single scientist. A **TxtGEx**-learner keeps on suggesting hypotheses as it perceives more and more data. It does *not* need to be aware of its

own success, that is, its own convergence. One can even show that learners that are forced to acknowledge their own success can identify strictly less classes of languages. See Freivalds & Wiehagen [13] for an according result. This shift of attention from confirmed knowledge of success to mere successful identification is the focal point Gold added to the psychology of learning [16]. This has opened the field to formal analysis, thus, founding computational learning theory.

The next theorem states an important technical property of **TxtGEx**-learners which is used in virtually any proof regarding Gold-style explanatory learning.

Definition 2.5 (Blum & Blum [5])**:**
Suppose $L \in \mathcal{E}$ to be a language and $h \in \mathcal{P}$ a learner. A sequence σ with $\text{content}(\sigma) \subseteq L$ is said to be a *locking sequence* for h on L if, for all sequences $\tau \in (L \cup \{\#\})^*$, $h(\sigma) = h(\sigma \diamond \tau)$ and $W_{h(\sigma)} = L$.

□

That means, from the moment on a learner sees a locking sequence it outputs a *correct* hypothesis and *no* data from the target language can *ever* make it change its mind again. Of course, a locking sequence is a very handy thing to have and it would be great if they would appear quite regularly.

Theorem 2.6 (Locking Lemma; Blum & Blum [5])**:**
Suppose $h \in \mathcal{P}$ to be a learner, $L \in \mathbf{TxtGEx}(h)$ an identifiable language and σ a finite sequence with $\text{content}(\sigma) \subseteq L$. *There is an extension $\sigma \sqsubseteq \tau$ such that τ is a locking sequence for h on L.*

■

Necessarily this extension's content is in L as well. Note that this theorem solely depends on the characteristics of explanatory learning and is not affected by any further restrictions as introduced below. Blum & Blum proved that *every* compatible sequence can be extended to a locking sequence. Unfortunately though, this extension can not hoped to be computable.

2.2 Classification of Learning Criteria

We have established a first learning paradigm step by step by implementing the concepts of inductive inference stated by Jain et al. [17]. We used the modular approach of Kötzing [20] to name the criterion. Moreover, said approach is also a blueprint to design new paradigms. The main idea is to split a given setting into fragments, assessing how each one of them affects the distinct outcome. The particles gained this way naturally correspond with the concepts of Jain et al. as well.

First, we consider different classes of admissible learners. Our original choice were partially recursive numerical functions, drawn from $\mathcal{P}$. Observe that this particular option is not marked in the name of the criterion. Other possible collections of scientists may be all total recursive functions ($\mathcal{R}$) or even non-recursive total or partial functions, i.e. $\mathfrak{R}$ or $\mathfrak{P}$. One might wonder now whether it is a restriction to language learning forcing the learner to be total.

It is a well-known fact that this is *not* the case.

Theorem 2.7 (cf. Jain et al. [17])**:**
Every **TxtGEx***-learnable class of languages can be so learned by a total recursive scientist, thus,* $[\mathcal{R}\mathbf{TxtGEx}] = [\mathbf{TxtGEx}]$.

■

On the other hand, our assumption of a learner to be *recursive* is indeed a severe restriction to learning. For a discussion of that matter, see [17].

Besides different classes of scientist there may also be different ways to present a formal language to a specific learner. In this work we will only consider learning from texts, which is why we will always use the particle **Txt**. Other possibilities are learning from *informants*, cf. e.g. Gold [16] again, or from *good examples*, see Lange et al. [23]. For more ways of presentation and their respective abbreviations, see Kötzing [20].

The next particle denotes how the learner conceives the presented information, it corresponds with *the data available about any given reality* [17]. This is modeled using so-called *sequence generating operators* [20].

Definition 2.8 (Kötzing [20])**:**
Suppose $\mathcal{C} \subseteq \mathfrak{P}$ to be an admissible class of learners. A *sequence generating operator* (*interaction operator*) is any mapping $\mathcal{C} \times \mathbf{Txt} \to \mathfrak{P}$, matching pairs of learners and texts to infinite (possibly partial) sequences of hypotheses.

□

We already know an example of an interaction operator from Definition 2.4.(I), namely *Gold-style learning*:

$$\mathbf{G}\colon \mathfrak{P} \times \mathbf{Txt} \to \mathfrak{P};\ (h;T) \mapsto (n \mapsto h(T[n])).$$

This is also called *full-information learning* for obvious reasons.

Definition 2.9 (Wexler & Culicover [30], Schäfer-Richter [29]; cf. Kötzing [20])**:**
Suppose $h \in \mathcal{P}$ to be a learner, n a natural number and $T \in \mathbf{Txt}$ a text.
We define the following sequence generating operators.

(I) Partially set-driven learning: $\mathbf{Psd}(h;T)(n) = h(\mathrm{content}(T[n]);n)$;

(II) Set-driven learning: $\mathbf{Sd}(h;T)(n) = h(\mathrm{content}(T[n]))$;

(III) Iterative learning: $\mathbf{It}(h;T)(n) = \begin{cases} h(\emptyset), & \text{if } n = 0; \\ h(\mathbf{It}(h;T)(n-1);T(n-1)), & \text{otherwise.} \end{cases}$

□

The symbol $h(\emptyset)$ denotes the initial hypothesis of learner h in the absence of any input data. A *set-driven* learner can only access the content of the sequence seen so far. It can neither use the order in which the elements were presented nor their

multiplicity. A *partially set-driven* learner at least has the additional information of how many (not necessarily different) elements already have been shown. *Iterative* learning models an extreme case of memory limitation as the scientist can only depend on its last guess and the current data point. Set-drivenness and iterativeness were both introduced to inductive inference by Wexler & Culicover [30], partially set-drivenness was first investigated by Schäfer-Richter [29]. The latter paradigm is discussed in more detail in the next section.

One may like to impose further constraints on the strategy of learning. These restrictions can involve limitation of time and of memory, as we examined above, but might as well concern eligible hypotheses [17] or update constraints [22]. These constraints can be implemented as predicates on the sequences outputted by the interaction operators and the texts used to generate them.

Definition 2.10 (Kötzing [20])**:**
A *learning restriction* is any predicate $\mathfrak{P} \times \mathbf{Txt} \to \{0;1\}$.

□

For ease of notation, we sometimes identify a restriction δ with the pre-image $\delta^{-1}(1) = \{(p;T) \mid \delta(p;T) = 1\}$ and then write $\delta \Rightarrow \delta'$ for $\delta^{-1}(1) \subseteq (\delta')^{-1}(1)$. Once again Definition 2.4.(I) gives us an impression of a learning restriction. In this case it is used to formulate a success criterion, *explanatory learning*:

$$\mathbf{Ex}\colon \mathfrak{P} \times \mathbf{Txt} \to \{0;1\};\ (p;T) \mapsto (\exists e\, \forall^{\infty} n\colon p(n) = e \wedge W_e = \mathrm{content}(T)).$$

We will almost always use this criterion throughout this work. The sole exception is section 4.2 where we will take a short excursion to *behaviorally correct* inference. We tacitly tighten both success criteria by forcing the learner to be defined on all initial sequences of texts for languages it learns (for the case of **Ex**, see also above). Multiple restrictions can be used in conjunction. This appears in the name as a juxtaposition of the according particles. Learning restrictions may be applied to the general behavior of the learner or only on texts for languages it identifies. This is reflected in the notion of *global* and *class* constraints found in Jain et al. [17].

The difference of the two concepts shall be illustrated by the following example: Requiring a scientist to include in his hypotheses all the knowledge about his subject previously acquired, appears to be a quite reasonable constraint. This leads to the notion of a *consistent* learner.

Definition 2.11 (Angluin [1]; cf. Kötzing [20])**:**
Suppose $p \in \mathfrak{P}$ to be a (possible partial) sequence of hypotheses and T a text.
We define the following learning restriction, *consistency*, as

$$\mathbf{Cons}(p;T) \Leftrightarrow p \in \mathfrak{R} \wedge \forall n\colon\ \mathrm{content}(T[n]) \subseteq W_{p(n)}.$$

□

In this case we explicitly force consistent learning sequences to be total. For a discussion of *conditional consistency*—the hypotheses need to be consistent whenever they are defined—see Jain et al. [17]. The totality of successful learning sequences has some consequences for the application of this restriction.

Definition 2.12 (Kötzing [20])**:**
Suppose $h \in \mathcal{P}$ to be a learner and $\mathcal{L} \subseteq \mathcal{E}$ a class of languages.

(I) Learner h is said to **TxtGConsEx**-*identify* $\mathcal{L}$ if it **TxtGEx**-learns $\mathcal{L}$ and is consistent on all texts $T \in \mathbf{Txt}(L)$, for each $L \in \mathcal{L}$.

(II) Learner h is said to $\tau(\mathbf{Cons})$**TxtGEx**-*identify* $\mathcal{L}$ if it **TxtGEx**-learns $\mathcal{L}$ and is consistent on *all* texts $T \in \mathbf{Txt}$.

□

Globally consistent learner are necessarily total, while locally consistent ones may be undefined (or inconsistent) on input data for languages they cannot identify. Either case of consistency is known to be a severe reduction of learning power.

Theorem 2.13 (Angluin [1], Bārzdiņš [3])**:**
The following chain of learning criteria holds:

$$[\tau(\mathbf{Cons})\mathbf{TxtGEx}] \subsetneq [\mathcal{R}\mathbf{TxtGConsEx}] \subsetneq [\mathbf{TxtGConsEx}] \subsetneq [\mathbf{TxtGEx}]$$

■

Now we have all the building blocks to assemble various learning criteria, natural and rather artificial ones likewise. It comes all together in the next definition.

Definition 2.14 (Kötzing [20])**:**

(I) A *learning criterion* is a tuple $(\alpha; \mathcal{C}; \beta; \delta)$ consisting of:

- two learning restrictions, $\alpha; \delta \colon \mathfrak{P} \times \mathbf{Txt} \to \{0; 1\}$
- a class $\mathcal{C} \subseteq \mathfrak{P}$ of admissible learners
- a sequence generating operator $\beta \colon \mathcal{C} \times \mathbf{Txt} \to \mathfrak{P}$

(II) Let $I = (\alpha; \mathcal{C}; \beta; \delta)$ be a learning criterion.
A learner $h \in \mathfrak{P}$ I-learns no language at all if $h \notin \mathcal{C}$ is not admissible or there is a text $T \in \mathbf{Txt}$ such that $\alpha(\beta(h; T); T)$ does *not* hold.

Otherwise, h I-learns the set

$$I(h) = \tau(\alpha)\mathcal{C}\mathbf{Txt}\beta\delta(h) = \{L \in \mathcal{E} \mid \forall T \in \mathbf{Txt}(L) \colon \delta(\beta(h; T); T)\}.$$

(III) $[I] = [\tau(\alpha)\mathcal{C}\mathbf{Txt}\beta\delta] = \{\mathcal{L} \subseteq \mathcal{E} \mid \exists\, h \in \mathcal{C} \colon \mathcal{L} \subseteq \tau(\alpha)\mathcal{C}\mathbf{Txt}\beta\delta(h)\}$
denotes the collection of all I-learnable classes of languages.

□

In other words, an admissible learner, drawn from $\mathcal{C}$, is fed information about its learnee using the interaction operator β. To learn a certain class of languages $\mathcal{L}$ the scientist has to respect the global restriction α on arbitrary texts and additionally

the local restriction δ on texts for languages $L \in \mathcal{L}$. We use the following notational convention regarding learning criteria: For the sake of readability, default choices—like the class $\mathcal{P}$ of partial recursive learners or the restriction $\mathbf{T}$ which is always *true*—does not appear in the name of a concrete criterion. As an illustration, recall the definition of our first learning paradigm. Regarding the class of learnable languages we have $[\mathbf{TxtGEx}] = [\tau(\mathbf{T})\mathcal{P}\mathbf{TxtGEx}]$. For the same reasons, we will often write $I = \mathbf{TxtGEx}$ instead of $I = (\mathbf{T}; \mathcal{P}; \mathbf{G}; \mathbf{Ex})$ when naming the criterion itself.

2.3 Fulk Normal Form

Since, in the late 1960's, Gold [16] established formal language learning, the possibilities and limits of this setting have been explored extensively [1, 3, 5, 15, 17]. However, with the introduction of new ways to present languages to learners a new question arose:

What information does a learner need to identify a certain class of languages?

In this work we will primarily examine the paradigms of partially set-driven learning. Recall that **Psd**-learner is given the collection of all input data and additionally the number of examples (and pause symbols) shown so far. In 1990 Fulk reached a breakthrough in the theory of inductive inference as he was able to prove that partially set-drivenness *does not weaken* unrestricted Gold-style learning [14]. Meaning that every **TxtGEx**-learnable class of languages can be inferred by a partially set-driven scientist. We will show his findings in the remainder of this section. The way Fulk presented his result was to verify that a **TxtGEx**-learner can be assumed to have several additional properties easing the process of learning. This leads to the notion of a learner in *Fulk normal form*. Before we can understand Fulk's theorem we have to introduce some new concepts as well as some technical terms.

We can preorder the collection of interaction operators. This order derives from the observation that certain operators can be simulated by others.

Definition 2.15 (Case & Kötzing [9])**:**
Suppose $\beta; \beta'$ to be two sequence generating operators. We say *β-learners can be translated to β'-learners*, written $\beta \preceq \beta'$, if, for every β-learner h, there is a β'-learner h' such that

$$\forall T \in \mathbf{Txt}\colon \beta(h;T) = \beta'(h';T).$$

□

For example, set-driven learners can sure be simulated by partially set-driven scientist ignoring the additional information of the length of the input sequence. On the other hand, the latter can be emulated using full information.

Theorem 2.16 (Case & Kötzing [9])**:**
We have $\mathbf{Sd} \prec \mathbf{Psd} \prec \mathbf{G}$ *and* $\mathbf{It} \prec \mathbf{G}$.

■

The idea of translating one learner into another carries over to the level of learnable classes. This is giving us a first set of relations between learning criteria.

Theorem 2.17 (Case & Kötzing [9])**:**
Suppose $I = (\alpha; \mathcal{C}; \beta; \delta)$ *and* $I' = (\alpha'; \mathcal{C}'; \beta'; \delta')$ *to be two learning criteria such that* $\mathcal{C} \subseteq \mathcal{C}'$, $\beta \preceq \beta'$, $\alpha \Rightarrow \alpha'$ *and* $\delta \Rightarrow \delta'$. *Then we have* $[I] \subseteq [I']$.

■

In this work we will extensively use the following instance of the above theorem:

$$[\mathbf{TxtSd}\delta\mathbf{Ex}] \subseteq [\mathbf{TxtPsd}\delta\mathbf{Ex}] \subseteq [\mathbf{TxtG}\delta\mathbf{Ex}] \subseteq [\mathbf{TxtGEx}].$$

Although sequence generating operator **It** is incomparable with both **Sd** and **Psd**, there is a connection between the respective collections of identifiable classes.

Theorem 2.18 (Kinber & Stephan [19])**:**
We have $[\mathbf{TxtItEx}] \subsetneq [\mathbf{TxtSdEx}]$.

The fact that many interaction operators can be emulated by **G**-learners leads to the notion of a *starred learner* as a first step towards a normal form.

Definition 2.19 (Case & Kötzing [9])**:**
Suppose $\beta \preceq \mathbf{G}$ to be a sequence generating operator and $h \in \mathcal{P}$ a β-learner.
Let $h^* \in \mathcal{P}$ denote the **G**-learner to simulate h as given by Definition 2.15.

In particular, for every sequence σ,

(I) if h is a **Sd**-learner: $h^*(\sigma) = h(\text{content}(\sigma))$;

(II) if h is a **Psd**-learner: $h^*(\sigma) = h(\text{content}(\sigma); |\sigma|)$;

(III) if h is an **It**-learner: $h^*(\sigma) = \begin{cases} h(\emptyset), & \text{if } \sigma = \emptyset; \\ h(h^*(\sigma^-); \text{last}(\sigma)), & \text{otherwise.} \end{cases}$

□

Hereby, $\text{last}(\sigma) = \sigma(|\sigma|-1)$ denotes the last element of sequence σ or $\emptyset$ if $\sigma = \emptyset$. Accordingly, σ^- denotes the prefix of σ without its last element.

The following concepts will only regard **G**-learners. This is justified by the above definition, a β-learner is said to have a certain property just in case its starred learner has it. A learner does not need to converge to the same natural number, if any, on every text for a target language L. Even if a scientist identifies L, the order in which its elements are presented as well as the number of pauses in between may affect the particular outcome. It is desirable to disregard these differences. Meaning that a successful learning sequence shall converge to the same index for every text for a language.

Definition 2.20 (Blum & Blum [5])**:**
A learner $h \in \mathcal{P}$ is said to be *order-independent* if, for all $L \in \mathbf{TxtGEx}(h)$ and any two texts $T; T' \in \mathbf{Txt}(L)$ for L, we have

$$\lim_{n\to\infty} h(T[n]) = \lim_{n\to\infty} h(T'[n]).$$

□

Another way of abstraction aims to abstain from the particular succession in which the examples are presented to the learner.

Definition 2.21 (Blum & Blum [5])**:**
A learner $h \in \mathcal{P}$ is said to be *rearrangement-independent* if, for any two sequences $\sigma; \tau$ such that $\mathrm{content}(\sigma) = \mathrm{content}(\tau)$ and $|\sigma| = |\tau|$, we have $h(\sigma) = h(\tau)$.

□

The next theorem is evident.

Theorem 2.22:
Suppose $\mathcal{L} \subseteq \mathcal{E}$ to be class of languages. $\mathcal{L}$ can be **TxtGEx***-identified by a rearrangement-independent scientist just in case $\mathcal{L} \in [\mathbf{TxtPsdEx}]$.*

■

We will use *partially set-driven* and *rearrangement-independent* learning as synonyms in turn, unless it is necessary to explicitly distinguish between a **Psd**-learner and its starred counterpart. Now let L denote a language identifiable by some learner h. Though the Locking Lemma (Theorem 2.6) states that each initial sequence of any text for L can be extended to a locking sequence for h on L, this extension might deviate from the text it derives from. As a result, there may be texts such that *no* initial sequence serves as a locking sequence. Fixing this flaw simplifies both the process of learning itself and the structure of proofs in learning theory [9].

Definition 2.23 (Kötzing & Palenta [22])**:**
A learner $h \in \mathcal{P}$ is said to be *strongly locking* if, for all $L \in \mathbf{TxtGEx}(h)$ and any text $T \in \mathbf{Txt}(L)$ for L, there is an index n_0 such that $T[n_0]$ is a locking sequence for h on L.

□

We have now the notation to discuss FULK's seminal result.

Definition 2.24 (FULK [14])**:**
A learner $h \in \mathcal{P}$ is said to be in *Fulk normal form* if (I) to (V) hold.

(I) The learner h is order-independent.

(II) The learner h is rearrangement-independent.

(III) If h **TxtGEx**-identifies a language $L \in \mathcal{E}$ from some text for L, then h **TxtGEx**-identifies L (from any text for L).

(IV) If there is a locking sequence for h on some language $L \in \mathcal{E}$, then h **TxtGEx**-identifies L.

(V) The learner h is strongly locking.

□

Theorem 2.25 (FULK [14])**:**
For every class of languages $\mathcal{L} \in [\mathbf{TxtGEx}]$, there is a learner in Fulk normal form **TxtGEx***-identifying $\mathcal{L}$. Moreover, we can assume the learner to be total.*

Unlike with the other theorems, we will show the proof to FULK's result below. It emphasizes another major technique, besides Theorem 2.17, of linking up partially set-driven learning with its full-information equivalent.

Proof: (*Of Theorem 2.25.*) Suppose $\mathcal{L} = \mathbf{TxtGEx}(h)$ to be a class of languages w.l.o.g. identifiable by a *total* learner $h \in \mathcal{R}$ (cf. Theorem 2.7).

For any finite set $D \subset_{\text{fin}} \mathbb{N}$ and any natural number t, let

$$D^{\leq t} = \{\sigma \in (\mathbb{N} \cup \{\#\})^* \mid \text{content}(\sigma) \subseteq D \wedge |\sigma| \leq t\}$$

be the set of all sequences with content in D and length at most t. Again for any D and t, we define a set

$$M(D;t) = \{\sigma \in D^{\leq t} \mid \forall \tau \in D^{\leq t} \colon h(\sigma \diamond \tau) = h(\sigma)\}.$$

Intuitively, $M(D;t)$ collects candidates for possible locking sequences. By coding, $M(D;t)$ can be considered as a set of natural numbers and is therefore well-ordered. Note that, as a subset of $D^{\leq t}$, $M(D;t)$ is always finite and a description of it can be computed from D and t. Now consider the following partially set-driven learner:

$$h'(D;t) = \begin{cases} h(\min M(D;t)), & \text{if } M(D;t) \neq \emptyset; \\ 0, & \text{otherwise.} \end{cases}$$

As h is total, so is h'. Suppose $L \in \mathcal{L}$ to be a learnable language and $T \in \mathbf{Txt}(L)$ a text for L. Let M denote the set of all locking sequences for h on L and $\sigma = \min M$ (the Locking Lemma states that M is non-empty). By the

minimality of σ, for all sequences τ with $\langle\tau\rangle < \langle\sigma\rangle$, there is an index n such that $\forall n' \geq n\colon \tau \notin M(\mathrm{content}(T[n']); n')$. Let n_0 be the maximum of these (finitely many) numbers. W.l.o.g. $\mathrm{content}(\sigma) \subseteq \mathrm{content}(T[n_0])$ and $|\sigma| \leq n_0$. Finally, we get

$$\forall n' \geq n_0\colon h'(\mathrm{content}(T[n']); n') = h(\sigma).$$

It is now easy to see that h' identifies $\mathcal{L}$ and (I) to (V) of Definition 2.24 hold for the starred learner $(h')^*$.

qed

Corollary 2.26:
We have $[\mathcal{R}\mathbf{TxtPsdEx}] = [\mathbf{TxtPsdEx}] = [\mathbf{TxtGEx}]$.

■

In this section a first paradigm of computational language learning has been defined. We generalized the approach to a full-grown grading scheme of algorithmic learning. The application of standard techniques has enabled us to conclude a first set of relations between the criteria we are interested in. In the next section we will use this preparation to present our own work in the field of rearrangement-independent inference paired with several restrictions.

3 Restricted Partially Set-Driven Learning

The main information missing in partially set-driven inference is the particular order in which the data was presented. Fulk [14] indeed showed that taking away this a priori knowledge does no harm, as long as one's only objective is to identify a certain class of languages in the limit. If a target collection is unidentifiable by a rearrangement-independent learner, the reasons preventing successful learning are immanent to the framework of inductive inference itself, and are not due to the special trait of rearrangement-independence. As a remark, it is interesting that these reasons can be separated into two major groups: First, computational limits, as some concept classes are unidentifiable by a recursive learner, but are easily inferred using arbitrary functions; second, topological reasons, where the structure of the class itself impedes its learning. For a discussion, see e.g. Jain et al. [17] or Case & Kötzing [10].
As opposed to this, we will try to investigate the possibilities and limits partially set-driven learning adds to the theory of computational language learning. Unrestricted Gold-style learning was treated exhaustively in the last section, but from Fulk's theorem yet a new question immediately arises.

Is rearrangement-independent inference equally powerful as Gold-style learning even if paired with various learning restrictions?

This question will guide the further investigation presented in this work. Recall that every **Psd**-learner can be expressed using **G**-learning. Theorem 2.17 now states that, for arbitrary learning restrictions $\alpha; \delta$, we have

$$[\tau(\alpha)\mathbf{TxtPsd}\delta\mathbf{Ex}] \subseteq [\tau(\alpha)\mathbf{TxtG}\delta\mathbf{Ex}].$$

So partially set-driven learning can be *at most* as powerful as its full-information complement. It remains unknown, whether *strictly* less classes of languages can be inferred using certain learning restrictions. We are primarily interested in two categories of such restrictions, namely *content-based* and *delayable* ones.

3.1 Content-Based Learning Restrictions

The first collection of restrictions imposes constraints on the choice of potential conjectures. In content-based inference there might be, in every step, some unavailable hypotheses as they would infringe a desired strategy of learning. Not unexpectedly, content-based constraints derive from the content of the data the learner has seen so far.

Definition 3.1:
A learning restriction $\delta\colon \mathfrak{P} \times \mathbf{Txt} \to \{0;1\}$ is said to be *content-based* if there is a predicate $P\colon \mathbb{N} \times \mathcal{P}_{\text{fin}}(\mathbb{N}) \to \{0;1\}$, on pairs of natural numbers and finite sets thereof, such that, for all infinite sequences $p \in \mathfrak{P}$ and texts $T \in \mathbf{Txt}$,

$$\delta(p;T) \Leftrightarrow p \in \mathfrak{R} \wedge \forall n\colon P(p(n); \text{content}(T[n])).$$

□

Again we require successful content-based learning sequences to be total. Please notice that predicate P needs not to be recursive. Quite the opposite, in most of the cases found in literature P will be recursively isomorphic to the Halting Problem (see also below). The conjunction of two content-based restrictions is again content-based. We already know a prominent example of this category, *consistency*. Recall Definition 2.11, the restriction is content-based as witnessed by predicate

$$P(i; D) \Leftrightarrow D \subseteq W_i.$$

The main result, presented in the following proposition, is that partially set-drivenness does *not* weaken content-based learning either.

Proposition 3.2:
For any content-based learning restriction δ, we have $[\mathbf{TxtPsd}\delta\mathbf{Ex}] = [\mathbf{TxtG}\delta\mathbf{Ex}]$. *Moreover, any* $\mathbf{TxtG}\delta\mathbf{Ex}$*-identifiable class of languages can be so learned order-independently and strongly locking.*

The main idea is to verify that Fulk's [14] original construction preserves content-based learning. We adjust the proof of Theorem 2.25 to comprise partial learners. This is necessary as content-based learning cannot assumed to be total (Theorem 2.13).

Proof: (*Of Proposition 3.2.*) As the other inclusion is trivial, it is sufficient to show that every $\mathbf{TxtG}\delta\mathbf{Ex}$-learnable class of languages can be so learned partially set-driven. Suppose $h \in \mathcal{P}$ to be a learner and $\mathcal{L} = \mathbf{TxtG}\delta\mathbf{Ex}(h)$ a concept class.

Again, for any finite set D and any natural number t, let $D^{\leq t}$ denote the set of all sequences with content in D and length at most t and

$$M(D; t) = \{\sigma \in D^{\leq t} \mid h(\sigma)\!\downarrow \wedge\, \forall \tau \in D^{\leq t} \colon h(\sigma \diamond \tau)\!\downarrow = h(\sigma)\}$$

shall be the collection of candidate locking sequences on input $(D; t)$. If h is defined on every sequence in $D^{\leq 2t}$, once more a description of $M(D; t)$ can be computed from D and t. Furthermore, let $\sigma^t_D \in D^{\leq t}$ denote the sequence listing D in increasing order up to length t, possibly padded with repeating occurrences of $\max D$.

We define the following partially set-driven learner h':

$$h'(D; t) = \begin{cases} h(\min M(D; t)), & \text{if } \forall \sigma \in D^{\leq 2t} \colon h(\sigma)\!\downarrow \wedge\, M(D; t) \neq \emptyset; \\ h(\sigma^t_D), & \text{if } \forall \sigma \in D^{\leq 2t} \colon h(\sigma)\!\downarrow \wedge\, M(D; t) = \emptyset; \\ \uparrow, & \text{otherwise.} \end{cases}$$

Learner h' is partial recursive as the condition $\forall \sigma \in D^{\leq 2t} \colon h(\sigma)\!\downarrow$ is semi-decidable. If h is total, so is h'.

Claim 1*:* Learner h' identifies $\mathcal{L}$ order-independently and strongly locking.

Suppose $L \in \mathcal{L}$ to be a language and $T \in \mathbf{Txt}(L)$ a text for L. Learner h is defined on every initial sequence of T by assumption. Then the same holds for h'. The rest of the claim follows as in Theorem 2.25.

Claim 2*:* Learner h' respects restriction δ on texts for languages in $\mathcal{L}$.

Let P be a predicate for δ as given in Definition 3.1; again let $L \in \mathcal{L}$ and $T \in \mathbf{Txt}(L)$. In stage n, suppose $D = \mathrm{content}(T[n])$. $P(h(\sigma^n_D); D)$ holds since n is sufficiently large such that $\mathrm{content}(\sigma^n_D) = D$ and h $\mathbf{TxtG}\delta\mathbf{Ex}$-learns L from text $\sigma^n_D \diamond T \in \mathbf{Txt}(L)$.
It is left to prove that, for any number n and any sequence $\sigma \in M(\mathrm{content}(T[n]); n)$, predicate $P(h(\sigma); \mathrm{content}(T[n]))$ holds as well. This, however, follows directly from the definition of $M(\mathrm{content}(T[n]); n)$: We have $h(\sigma) = h(\sigma \diamond \tau)$ for all extensions $\tau \in (\mathrm{content}(T[n]))^{\leq n}$, especially for those with $\mathrm{content}(\sigma \diamond \tau) = \mathrm{content}(T[n])$, and $P(h(\sigma \diamond \tau); \mathrm{content}(\sigma \diamond \tau))$ holds since $\sigma \diamond \tau \diamond T$ is a text for L.

qed

The above proof, together with the observation that consistency is content-based, yields some interesting corollaries.

Corollary 3.3:
For every content-based learning restriction δ, we have

(i) $[\mathcal{R}\mathbf{TxtPsd}\delta\mathbf{Ex}] = [\mathcal{R}\mathbf{TxtG}\delta\mathbf{Ex}]$,

(ii) $[\tau(\delta)\mathbf{TxtPsdEx}] = [\tau(\delta)\mathbf{TxtGEx}]$.

Proof: This can be straightforwardly verified using the above construction.

qed

Corollary 3.4:
The following statements hold:

(i) $[\mathbf{TxtPsdConsEx}] = [\mathbf{TxtGConsEx}]$,

(ii) $[\mathcal{R}\mathbf{TxtPsdConsEx}] = [\mathcal{R}\mathbf{TxtGConsEx}]$,

(iii) $[\tau(\mathbf{Cons})\mathbf{TxtPsdEx}] = [\tau(\mathbf{Cons})\mathbf{TxtGEx}]$,

(iv) $[\tau(\mathbf{Cons})\mathbf{TxtPsdEx}] \subsetneq [\mathcal{R}\mathbf{TxtPsdConsEx}] \subsetneq [\mathbf{TxtPsdConsEx}]$.

■

Proposition 3.2 also reproves Fulk's original result without the assumption of a total learner. This can easily be seen by noticing that $\mathbf{T}$, the learning restriction which is always *true*, is of course content-based.

Jain et al. pointed out another restriction from this category—they are using the wider term *constraints on potential conjectures* [17]—namely *accountability.*

A learner is said to be accountable if it generalizes from the input with every hypothesis. Meaning that each conjectured set have to contain an unseen data point.

Definition 3.5 (cf. JAIN et al. [17])**:**
We define the learning restriction *accountability* as follows:
Suppose $p \in \mathfrak{P}$ to be an infinite sequence and $T \in \mathbf{Txt}$ a text,

$$\mathbf{Acc}(p;T) \Leftrightarrow p \in \mathfrak{R} \wedge \forall n\colon W_{p(n)} \setminus \operatorname{content}(T[n]) \neq \emptyset.$$

□

To require a learner to predict new data in every step is making its hypotheses falsifiable by future observations and, hence, is implementing a common strategy for scientific progress. For a further discussion of accountability in the psychology of learning and the theory of science itself see JAIN et al. [17] and POPPER [26, 27]. Accountable language learning turns out to be very restrictive. It is clear that no finite language can be inferred by an accountable scientist. This restriction, at least in the setting of Gold-style explanatory learning, is even equivalent to forcing a learner to *only* output hypotheses for infinite languages. On the other hand, there are classes of infinite languages which are not accountably learnable either. For both, see [17] again.

Content-based learning can be done rearrangement-independently as we have shown above. However, if we go back one more step in the hierarchy of interaction operators, this is no longer true: There is a content-based constraint such that *set-driven* scientists can infer strictly less classes of languages with respect to this restriction.

Proposition 3.6:
There is a class of languages consistently identifiable by a rearrangement-independent learner, which cannot be learned by any *set-driven scientist. Particularly, we have* $[\mathbf{TxtSdConsEx}] \subsetneq [\mathbf{TxtPsdConsEx}]$.

Proof: Again the inclusion is obvious. The concept class below was used by SCHÄFER-RICHTER [29] to separate full-information learning from set-driven inference. The proposition now follows from the observation that this class can even be learned consistently by a rearrangement-independent scientist.

Suppose collection $\mathcal{L}$ to contain language $L_e = \{\langle e; y\rangle \mid y \in \mathbb{N}\}$, whenever $\varphi_e(0)\!\uparrow$, and language $L'_e = \{\langle e; y\rangle \mid y \leq \varphi_e(0)\}$ instead, if $\varphi_e(0)\!\downarrow$.

Claim 1*:* $\mathcal{L} \in [\mathbf{TxtPsdConsEx}]$.

Recall that $\pi_1 \in \mathcal{R}$ denotes the total recursive function defined by $\forall x; y\colon \pi_1(\langle x; y\rangle) = x$ and Φ is the complexity measure associated to φ. Let p_0 be a W-index for the empty set, i.e. $W_{p_0} = \emptyset$. Using the Parameter Theorem, there are total recursive functions $p; \mathrm{ind} \in \mathcal{R}$ such for all e and each finite set $D \subset_{\mathrm{fin}} \mathbb{N}$,

$$W_{p(e)} = L_e \qquad \text{and} \qquad W_{\mathrm{ind}(D)} = D.$$

We consider the **Psd**-learner $h \in \mathcal{P}$ defined as follows for finite sets D and numbers t:

$$h(D;t) = \begin{cases} p_0, & \text{if } D = \emptyset; \\ p(\pi_1(\min D)), & \text{else, if } \Phi_{\pi_1(\min D)}(0) > t; \\ \mathrm{ind}(D), & \text{otherwise.} \end{cases}$$

On any text T for a language L_e or L'_e in $\mathcal{L}$, $\pi_1(\min D)$ defaults to index e as soon as T shows the first element. So consistency immediately follows: After this point h only conjectures the input set D itself or the proper superset L_e.
It remains to show that h infers $\mathcal{L}$.

Case 1: $\varphi_e(0)\uparrow$.

Let $T \in \mathbf{Txt}(L_e)$ be a text for language L_e. For all indices n, we have $\Phi_{\pi_1(\min \mathrm{content}(T[n]))}(0) = \Phi_e(0) > n$. Therefore, h constantly outputs the correct hypothesis $p(e)$.

Case 2: $\varphi_e(0)\downarrow$.

Now let $T \in \mathbf{Txt}(L'_e)$ be a text for the finite set L'_e. In this case there is an n' sufficiently large such that $L'_e = \mathrm{content}(T[n'])$ and $\Phi_e(0) \leq n'$. We get $h(\mathrm{content}(T[n]); n) = \mathrm{ind}(L'_e)$ for each $n \geq n'$, which implies the claim.

Claim 2: $\mathcal{L} \notin [\mathbf{TxtSdEx}]$.

By way of contradiction assume otherwise.
Suppose $\mathcal{L} \subseteq \mathbf{TxtSdEx}(h')$ to be identifiable by some set-driven learner $h \in \mathcal{P}$. With Kleene's Recursion Theorem (KRT) we get an index e such that, for every input x, $\varphi_e(x)$ is the first number m found by a particular search satisfying

$$\langle e; m+1 \rangle \in W_{h'(\{\langle e;y \rangle \mid y \leq m\})},$$

if there is any, and undefined otherwise.

Case 1: $\varphi_e(0)\uparrow$.

We have that $\forall m\colon \langle e; m+1 \rangle \notin W_{h'(\{\langle e;y \rangle \mid y \leq m\})}$. Hence, h' cannot learn L_e from text $T(n) = \langle e; n \rangle$ as it never makes a correct guess.

Case 2: $\varphi_e(0)\downarrow = m$.

Now scientist h' cannot learn the finite language L'_e from *any* text since we know from the convergence of $\varphi_e(0)$ that $\langle e; m+1 \rangle \in W_{h'(\{\langle e;y \rangle \mid y \leq m\})} = W_{h'(L'_e)} \neq L'_e$

qed

Corollary 3.7:
We have $[\mathbf{TxtSdConsEx}] \subsetneq [\mathbf{TxtGConsEx}]$.

■

3.2 Delayable Learning Restrictions

There are more learning restrictions found in literature [1, 2, 17, 18, 31]. They impose *constraints on the relation between conjectures* [17] also known as *update constraints* [22]. Case and Kötzing (among others) realized that many of them share a common trait, they are *delayable* [8, 9, 20, 22]. Intuitively, a delayable learning restriction allows to postpone the output of a certain conjecture arbitrarily, given that the hypothesis will occur in the limit eventually.

Definition 3.8 (Kötzing & Palenta [22])**:**
Suppose $\vec{R}$ to be the set of all non-decreasing numerical functions $r\colon \mathbb{N} \to \mathbb{N}$ such that, for each m, we have $\forall^\infty n\colon r(n) \geq m$.

A learning restriction $\delta\colon \mathfrak{P} \times \mathbf{Txt} \to \{0;1\}$ is said to be *delayable* if, for all texts $T; T' \in \mathbf{Txt}$ with $\mathrm{content}(T) = \mathrm{content}(T')$, all infinite sequences $p \in \mathfrak{P}$ and all functions $r \in \vec{R}$, we have

$$\delta(p;T) \wedge \forall n\colon \mathrm{content}(T[r(n)]) \subseteq \mathrm{content}(T'[n]) \;\Rightarrow\; \delta(p \circ r; T').$$

□

There is a special case of Definition 3.8 noteworthy: Above condition holds if the texts $T = T'$ are equal and $\forall n\colon r(n) \leq n$ [22]. This reflects the initial idea of deferring hypotheses a learner h outouts working on text T. Particularly, the conjecture $h(T[r(n)])$ is "delayed" until step n.

The essence of delayable learning becomes clear with more illustrating examples of according restrictions: A *strongly monotone* (**SMon**) scientist may only grow its conjectured sets [18]; a *weakly monotone* (**WMon**) one has to behave that way at least while consistent with the incoming data [24]. A *conservative* (**Conv**) learner even has to stay with its current hypothesis as long it is consistent [1]. In *monotone* (**Mon**) learning only incorrect data may be removed [31]. A *cautious* (**Caut**) learner is forbidden to ever conjecture a proper subset of a previously supposed set, cf. [17]. *Non-U-Shaped* or *strongly non-U-shaped* (**NU**/**SNU**) scientists will never semantically (syntactically) abandon a correct hypothesis [2, 11]. Complementary, in *decisive* and *strongly decisive* (**Dec**/**SDec**) learning a hypothesis once abandoned will never again be repeated semantically (syntactically) [17, 21]. More formally, we have the following definitions.

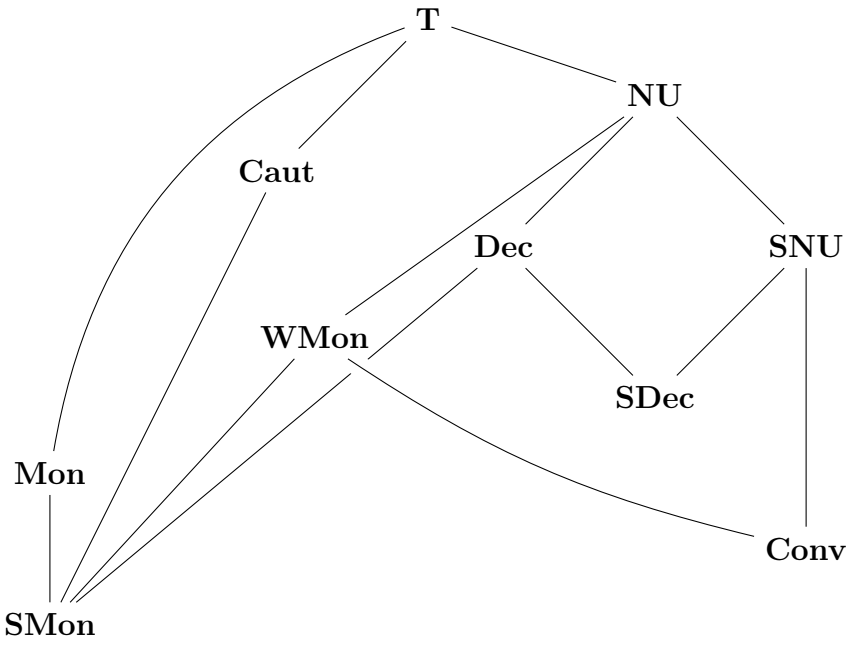

Figure 1: Relations between delayable learning restrictions.
(Taken from [22] with permission by the authors.)

Definition 3.9 (Kötzing & Palenta [22])**:**
Suppose $p \in \mathfrak{P}$ to be an infinite sequence and $T \in \mathbf{Txt}$ a text.

$$\begin{aligned}
\mathbf{Conv}(p;T) &\Leftrightarrow \forall i\colon \operatorname{content}(T[i+1]) \subseteq W_{p(i)} \Rightarrow p(i) = p(i+1);\\
\mathbf{Caut}(p;T) &\Leftrightarrow \forall i;j\colon W_{p(i)} \subsetneq W_{p(j)} \Rightarrow i < j;\\
\mathbf{NU}(p;T) &\Leftrightarrow \forall i;j;k\colon (i \leq j \leq k \wedge W_{p(i)} = W_{p(k)} = \operatorname{content}(T)) \Rightarrow W_{p(i)} = W_{p(j)};\\
\mathbf{Dec}(p;T) &\Leftrightarrow \forall i;j;k\colon (i \leq j \leq k \wedge W_{p(i)} = W_{p(k)}) \Rightarrow W_{p(i)} = W_{p(j)};\\
\mathbf{SNU}(p;T) &\Leftrightarrow \forall i;j;k\colon (i \leq j \leq k \wedge W_{p(i)} = W_{p(k)} = \operatorname{content}(T)) \Rightarrow p(i) = p(j);\\
\mathbf{SDec}(p;T) &\Leftrightarrow \forall i;j;k\colon (i \leq j \leq k \wedge W_{p(i)} = W_{p(k)}) \Rightarrow p(i) = p(j);\\
\mathbf{SMon}(p;T) &\Leftrightarrow \forall i;j\colon i \leq j \Rightarrow W_{p(i)} \subseteq W_{p(j)};\\
\mathbf{Mon}(p;T) &\Leftrightarrow \forall i;j\colon i \leq j \Rightarrow W_{p(i)} \cap \operatorname{content}(T) \subseteq W_{p(j)} \cap \operatorname{content}(T);\\
\mathbf{WMon}(p;T) &\Leftrightarrow \forall i;j\colon (i \leq j \wedge \operatorname{content}(T[j]) \subseteq W_{p(i)}) \Rightarrow W_{p(i)} \subseteq W_{p(j)}.
\end{aligned}$$

□

One verifies that all of the above restrictions are indeed delayable. Moreover, the restriction **T** and the success criterion **Ex** are also members of this category. The conjunction of two delayable restrcitions is again delayable. In general content-based restrictions are not delayable since they enforce an immediate behavior of the current hypothesis. It is easy to see that there are intimate relations between the restrictions named in Definition 3.9. These are presented in Figure 1: A solid black line indicates that the lower restriction implies the higher one. The conditionals directly carry over to the level of learnable classes via Theorem 2.17. There is a very helpful technical lemma concerning these restrictions. It outlines another basic property of delayable learning, at least in the case of full information:

Delayable Gold-style learning is total.

Theorem 3.10 (Kötzing & Palenta [22])**:**
For any delayable learning restriction δ, we have $[\mathbf{TxtG}\delta] = [\mathcal{R}\mathbf{TxtG}\delta]$.

■

This corresponds with Theorem 2.7 as **Ex** is delayable. There are more properties of Fulk's normal form that, in parts, carry over to delayable learning: Whenever a delayable criterion allows for rearrangement-independent learning we can additionally assume order-independence.

Proposition 3.11:
Rearrangement-independent delayable learning is order-independent. Particularly, for every delayable restriction δ, the following two statements hold:

(i) Every $\mathbf{TxtSd}\delta\mathbf{Ex}$*-learner is order-independent.*

(ii) Every $\mathbf{TxtPsd}\delta\mathbf{Ex}$*-learnable class of languages can be so learned order-independently.*

Proof:
For part (i): Suppose $h \in \mathcal{P}$ to be a learner and $L \in \mathbf{TxtSd}\delta\mathbf{Ex}(h)$ an identifiable language. Using the Locking Lemma, let σ be a locking sequence for the starred learner h^* on L. For every text $T \in \mathbf{Txt}(L)$, there is an index n_0 such that

$$\forall n \geq n_0 \colon \operatorname{content}(T[n]) \supseteq \operatorname{content}(\sigma).$$

Then, for every sufficiently large n, there is a sequence $\tau_n \in L^*$ such that $\operatorname{content}(\sigma \diamond \tau_n) = \operatorname{content}(T[n])$ and thus

$$\forall n \geq n_0 \colon h(\operatorname{content}(T[n])) = h^*(\sigma \diamond \tau_n) = h^*(\sigma)$$

as learner h is set-driven. So h is order-independent by the arbitrary choice of σ.

For part (ii): Let $\mathcal{L} = \mathbf{TxtPsd}\delta\mathbf{Ex}(h)$ for some learner $h \in \mathcal{P}$. Consider the **Psd**-learner defined by $h'(D;t) = h(D;2t)$ for every finite set D and number t.

Claim 1*:* Learner h' infers $\mathcal{L}$ with respect to δ.

Suppose $L \in \mathcal{L}$ to be a learnable language and $T \in \mathbf{Txt}(L)$ a text for L. Let T' derive from T by inserting a pause symbol at every other position: $\forall i \colon T'(2i) = T(i) \wedge T'(2i+1) = \#$. So T' is a text for L as well. Let, for every n, $r(n) = 2n$, thus, $r \in \vec{R}$. By construction, we have

$$\forall n \colon \operatorname{content}(T'[r(n)]) = \operatorname{content}(T[n]) \wedge \mathbf{Psd}(h';T)(n) = \mathbf{Psd}(h;T')(r(n)).$$

Scientist h learns L from T', hence, $\delta\mathbf{Ex}(\mathbf{Psd}(h;T');T')$. It follows that $\delta\mathbf{Ex}(\mathbf{Psd}(h';T);T)$ as $\delta\mathbf{Ex}$ is delayable. To put it another way, h' identifies L from T with respect to δ.

Claim 2: Learner h' is order-independent on languages in $\mathcal{L}$.

Using the Locking Lemma we get again a locking sequence σ for h^* on $L \in \mathcal{L}$. For every $T \in \mathbf{Txt}(L)$, there is an index n_0 such that

$$\forall n \geq n_0 \colon \operatorname{content}(T[n]) \supseteq \operatorname{content}(\sigma) \wedge n \geq |\sigma| - |\operatorname{content}(\sigma)|.$$

From this we get, for sufficiently large n,

$$|\operatorname{content}(T[n]) \setminus \operatorname{content}(\sigma)| + |\sigma| = |\operatorname{content}(T[n])| - |\operatorname{content}(\sigma)| + |\sigma| \leq 2n.$$

The last inequality is due to $|\operatorname{content}(T[n])| \leq n$. So there is a sequence τ_n listing $\operatorname{content}(T[n]) \setminus \operatorname{content}(\sigma)$, possibly padded with symbol #, such that $\operatorname{content}(\sigma \diamond \tau_n) = \operatorname{content}(T[n])$ and $|\sigma \diamond \tau_n| = 2n$. In conclusion, we get

$$\forall n \geq n_0 \colon h'(\operatorname{content}(T[n]); n) = h(\operatorname{content}(T[n]); 2n) = h^*(\sigma \diamond \tau_n) = h^*(\sigma),$$

which implies the claim.

qed

However, the answer to the question whether partially set-drivenness reduces the power of delayable language learning is not as clear as for content-based learning. For some of the above restriction is has already been proven that according scientists can w.l.o.g. assumed to be rearrangement-independent [9]. On the other hand, in the next section, we will present a delayable criterion for which **Psd**-learners can learn *strictly* less classes of languages than their full-information counterparts. The next proposition is a mere summary of theorems that have already been stated by Case & Kötzing in [9], combined with the initial result by Fulk [14] (Theorem 2.25).

Proposition 3.12:
The following learning criteria are all extensionally equivalent:

(i) [**TxtGEx**]

(ii) [**TxtGSNUEx**]

(iii) [**TxtGNUEx**]

(iv) [**TxtPsdEx**]

(v) [**TxtPsdSNUEx**]

(vi) [**TxtPsdNUEx**]

Moreover, in all cases the learners can be assumed to be total.

Proof: The following chains of inclusions obviously hold for the total and partial case, respectively:

$$[\mathcal{R}\mathbf{TxtPsdSNUEx}] \subseteq [\mathbf{TxtPsdSNUEx}] \subseteq [\mathbf{TxtPsdNUEx}] \subseteq [\mathbf{TxtPsdEx}];$$
$$[\mathbf{TxtPsdSNUEx}] \subseteq [\mathbf{TxtGSNUEx}] \subseteq [\mathbf{TxtGNUEx}],$$

Additionally, they are all contained in [**TxtGEx**] of course.

We have already seen that Gold-style learning without restrictions can be done rearrangement-independently, thus, [**TxtPsdEx**] = [**TxtGEx**]. The finding of Case & Kötzing [9] that partially set-driven learning is w.l.o.g. total and strongly non-U-shaped ([$\mathcal{R}$**TxtPsdSNUEx**] = [**TxtPsdEx**]) now completes the proof.

qed

For (strongly) non-U-shaped learning rearrangement-independence again is no weakness. The reason for this is simply that neither **NU** nor **SNU** mean *any* constraint for Gold-style learning whatsoever. As opposed to this, in the next section we will identify strong monotony (**SMon**) to be quite a severe restriction. This will yield a first criterion for which **Psd**-learning will be strictly weaker.

Like we did in the case of content-based learning, one can investigate whether rearrangement-independent delayable learning can even be done solely by set-driven scientists. An affirmative answer would give immediate new results regarding the question we asked at the beginning of this section: According to a theorem by Kinber & Stephan [19] (see Theorem 3.14 below) *all* set-driven learning is properly contained in conservative inductive inference. So if delayable **Psd**-learning could be done by set-driven scientists, then these learners could not even infer all classes contained in [**TxtGConvEx**], let alone the criteria higher above in the hierarchy shown in Figure 1. However, for almost all delayable restrictions it can be proven that set-driven learner can infer strictly less classes of languages than their partially set-driven analogons. The proof is presented below. Note **SMon** again taking the role of a significant exception. We will need the following two theorems.

Theorem 3.13 (Kötzing & Palenta [22])**:**
The following relations hold for set-driven delayable learning:

(i) [**TxtSdSMonEx**] $\subsetneq$ [**TxtSdMonEx**] $\subsetneq$ [**TxtSdEx**].

(ii) *For any restriction* $\delta \in \{$**Conv**; **Caut**; **NU**; **Dec**; **SNU**; **SDec**; **WMon**$\}$, *learning with respect to* δ *does not weaken set-driven learning, thus,* [**TxtSdEx**] = [**TxtSd**δ**Ex**].

■

Theorem 3.14 (Kinber & Stephan [19])**:**
Set-driven learning is properly contained in conservative learning with full information, thus, we have [**TxtSdEx**] $\subsetneq$ [**TxtGConvEx**].

■

Proposition 3.15:
For $\delta \in \{$**Conv**; **Caut**; **NU**; **Dec**; **SNU**; **SDec**; **Mon**; **WMon**$\}$, *we have* [**TxtSd**δ**Ex**] $\subsetneq$ [**TxtPsd**δ**Ex**].

Proof: All inclusions are trivial. The separations are shown in several cases.

Case 1: $\delta = \mathbf{NU}$ or $\delta = \mathbf{SNU}$.

We already know that neither form of non-U-shapedness is a restriction to rearrangement-independent learning (Proposition 3.12). By the above Theorem 3.13, the same holds for set-driven learning. So this case is obvious: Unrestricted partially set-driven inference equals Gold-style learning (Theorem 2.25) while set-driven learning is strictly less powerful (Theorem 3.14).

Case 2: $\delta \in \{\mathbf{Conv}; \mathbf{Caut}; \mathbf{WMon}\}$.

Kinber & Stephan [19] used the construction presented below to separate set-driven learning from Gold-style conservative learning (see also Theorem 3.14 above). In fact, the learner they considered is rearrangement-independent and infers the target class not only conservatively, but also cautiously and weakly monotone.

Suppose $\psi \in \mathcal{P}$ to be a $\{0;1\}$-valued partial recursive function which has *no* total recursive extension [28]. Let

$$G = \{\langle x; \psi(x)\rangle \mid x \in \mathrm{dom}(\psi)\} \subseteq \mathbb{N}$$

denote its graph. W.l.o.g. $\psi(0)\!\downarrow = 1$, hence, $\langle 0;1\rangle \in G$. Since G is r.e., we fix a procedure to enumerate it. For any number t, let G^t be the finite subset of G which is enumerated this way within t steps. A finite set D is said to be *incompatible with* G^t just in case there is a point x such that $\langle x;0\rangle; \langle x;1\rangle \in D \cup G^t$. Accordingly, D is said to be *incompatible with* G if there is a stage t such that D is incompatible with G^t. For some fixed t, it is decidable whether some set D is compatible with G^t.
Now suppose $\mathcal{L}$ to consist of G and all finite sets which are incompatible with G.

Claim 1: $\mathcal{L} \in [\mathbf{TxtPsdConvCautEx}]$.

Let e be an index for G and again $\mathrm{ind} \in \mathcal{R}$ be such that $W_{\mathrm{ind}(D)} = D$. We define the **Psd**-learner h as follows:

$$\forall D; t\colon h(D;t) = \begin{cases} e, & \text{if } D \text{ is compatible with } G^t; \\ \mathrm{ind}(D), & \text{otherwise.} \end{cases}$$

Clearly h identifies $\mathcal{L}$: For each $D \subset_{\mathrm{fin}} G$ and arbitrary t, we have $h(D;t) = e$. For a finite language L incompatible with G and a text $T \in \mathbf{Txt}(L)$ for L, there is an index n_0 sufficiently large such that $\mathrm{content}(T[n_0]) = L$ and L is incompatible with G^{n_0}, implying $\forall n \geq n_0\colon h(\mathrm{content}(T[n]); n) = \mathrm{ind}(\mathrm{content}(T[n]) = \mathrm{ind}(L)$.

It is left to show that h is both conservative and cautious on texts for languages in $\mathcal{L}$. The first mindchange of h from G to some set D occurs only if D is incompatible with G, thus, $D \not\subseteq G$. From this point on, h solely conjectures canonical indices for the current input set itself. Hence, h is conservative (and hence weakly monotone). The learner is cautious for the same reason as it never returns to a proper subset of a prior hypothesis.

Claim 2: $\mathcal{L} \notin [\mathbf{TxtSdEx}]$.

By way of contradiction assume otherwise.
Suppose $\mathcal{L} \subseteq \mathbf{TxtSdEx}(h')$ for some learner $h' \in \mathcal{P}$. Class $\mathcal{L}$ is *dense*: Recall that w.l.o.g. $\langle 0;1\rangle \in G$. So $D \cup \{\langle 0;0\rangle\} \in \mathcal{L}$ for every finite set D. Therefore, h' has to be total. Let e be an index for G and $D_0 \subset_{\text{fin}} G$ such that $h'(D) = e$ for all $D_0 \subseteq D \subset_{\text{fin}} G$. D_0 exists as h' learns G by assumption. Now we define a function f as follows:

$$\forall x\colon f(x) = \begin{cases} 0, & \text{if } h'(D_0 \cup \{\langle x;0\rangle\}) = e; \\ 1, & \text{otherwise.} \end{cases}$$

As h' is total recursive, so is f. If $\psi(x)\!\downarrow = 0$, then $\langle x;0\rangle \in G$ and thus $h'(D_0 \cup \{\langle x;0\rangle\}) = e$. If $\psi(x)\!\downarrow = 1$, then $D_0 \cup \{\langle x;0\rangle\} \in \mathcal{L} \setminus \{G\}$. So $h'(D_0 \cup \{\langle x;0\rangle\}) \neq e$ as h' has to infer this language. Putting these two facts together, we see that f is a total recursive extension of ψ, a contradiction to the choice of ψ.

Case 3: $\delta \in \{\mathbf{Dec}; \mathbf{SDec}; \mathbf{Mon}\}$.

To show this separation we can draw on a construction first introduced by Osherson, Stob & Weinstein [25]. They originally used it to separate unrestricted Gold-style learning from cautious inference. Kötzing & Palenta [22] extended it to comprise (strongly) decisive learning. Notation follows the latter.

Consider the **Psd**-learner $h \in \mathcal{P}$ defined by, $h(D;t) = \varphi_{\max D}(t)$ for all finite sets D and numbers t. Let $\mathcal{L} = \mathbf{TxtPsdSDecMonEx}(h)$ be the class of languages h infers. Thus, $\mathcal{L}$ is a *self-learning* class [9] as it is only implicitly defined by its learner h. By Theorem 3.14 above, it is now sufficient to show that $\mathcal{L}$ cannot be learned conservatively.

By way of contradiction assume otherwise.
Suppose $\mathcal{L} \subseteq \mathbf{TxtGConvEx}(h')$ to be identifiable w.l.o.g by some *total* learner $h' \in \mathcal{R}$ (Theorem 3.10). We now fix an *uniform* procedure to enumerate r.e. sets and, for any index i, let W_i^t denote the finite subset of W_i enumerated after t steps. We define a recursive predicate Q on sequences σ and numbers t:

$$Q(\sigma;t) \Leftrightarrow \text{content}(\sigma) \subsetneq W^t_{h'(\sigma)}.$$

Using the Operator Recursion Theorem (ORT), we get an index p and a total recursive function $e \in \mathcal{R}$ strongly monotone increasing such that

$$W_p = \text{range}(e);$$
$$\forall n;t\colon \varphi_{e(n)}(t) = \begin{cases} \text{ind}(\text{content}(e[n+1])), & \text{if } Q(e[n+1];t); \\ p, & \text{otherwise.} \end{cases}$$

If $Q(e[n+1];t)$ is false for any n and t, then $L = \text{range}(e) \in \mathcal{L}$ since h constantly outputs the correct conjecture p on texts for L.

On the other hand, h' does not learn L from text e as content$(e[n+1])$ is never a proper subset of $W_{h'(e[n+1])}$ although the target language is infinite.

Now consider the case that $Q(e[n+1];t)$ is true for some n and t. Let n_0 be minimal such that some t satisfy $Q(e[n_0+1];t)$ and, accordingly, t_0 minimal such that $Q(e[n_0+1];t_0)$ holds. This implies $Q(e[n_0+1];t)$ for all $t \geq t_0$ as well.
It follows that $L' = \text{content}(e[n_0+1])$ is in $\mathcal{L}$: Suppose $T \in \mathbf{Txt}(L')$ to be a text for L'. From the strong monotony of e we get $\max L' = e(n_0)$ and there is an index n' minimal such that $e(n_0) \in \text{content}(T[n'])$. W.l.o.g. $n' \geq t_0$, hence, for all $n \geq n'$, $h(\text{content}(T[n]);n) = \varphi_{e(n_0)}(n) = \text{ind}(\text{content}(e[n_0+1]))$. Before that, h working on T solely outputted index p by the minimality of n_0. So **SDec** and **Mon** both hold. But scientist h' does not learn L' conservatively from any text starting with prefix $e[n_0{+}1]$: As we know from predicate Q, $L' = \text{content}(e[n_0+1]) \subsetneq W_{h'(e[n_0+1])}$. Hence, h' can never change its conjecture back to a correct guess, a contradiction.

qed

Corollary 3.16:
For $\delta \in \{$**Conv**; **Caut**; **NU**; **Dec**; **SNU**; **SDec**; **Mon**; **WMon**$\}$, *we have* $[\mathbf{TxtSd}\delta\mathbf{Ex}] \subsetneq [\mathbf{TxtG}\delta\mathbf{Ex}]$.

■

Just like for content-based learning, for *almost* all delayable restrictions set-drivenness *does* weaken learning power. It is an open question whether Proposition 3.15 can be extended to comprise strongly monotone learning. There are some hints that it does and a sub-case of it is proven in the next section. We will provide further evidence there that strongly monotone inference takes an exceptional position among the delayable criteria.

4 Strongly Monotone Learning

On several occasions throughout this thesis we obtained the impression that strongly monotone language learning is somehow special. We will use the present section to explore this anomaly. As a strategy of learning, **SMon** seems to be quite a severe constraint, forcing a learner to only increase the conjectured set in size, never abandon a single data point once included. Furthermore, this implies that, on texts for identifiable languages, a strongly monotone learner *always* supposes a subset of the language to learn. Maybe these are mere fragments, but all elements contained in a hypothesis are certain to appear in the target language. Ultimately, we found the paradigm of *explanatory language learning from text by a strongly monotone scientist* to be sensitive to whether the learner perceives the input as a full sequence or only its content and length. This, Proposition 4.1, is the main result. To our knowledge **TxtGSMonEx** is the *very first* learning criterion in literature to which Fulk's theorem is not applicable. Besides that, in this section we will investigate strongly monotone learning when paired with different interaction operators and using an alternative success criterion.

4.1 Explanatory Learning

The next proposition is the main result of this thesis.

Proposition 4.1:
There is a language identifiable by an iterative, strongly monotone learner, which cannot be so learned partially set-driven.
Thus, we have [**TxtItSMonEx**] $\not\subseteq$ [**TxtPsdSMonEx**].

Proof: Recall that iterative scientists rely solely on their last hypothesis and the current data point to infer a language.

The Operator Recursion Theorem yields a total recursive function $a \in \mathcal{R}$ strongly monotone increasing such that, for all finite sets D and numbers $y; z$, we have

$$\varphi_{a(D)}(y; z) = \mathrm{ind}(D \cup \{a(D)\}).$$

From the strong monotony of a we get that $\mathrm{range}(a)$ is recursive and a is recursively invertible, i.e. we can regain the full information about D from value $a(D)$. W.l.o.g. $0 \notin \mathrm{range}(a)$. We define some auxiliary functions for finite sequences σ:

$$x_\sigma = \begin{cases} 0, & \text{if } \mathrm{content}(\sigma) \subseteq \{0\}; \\ \sigma(i^*), & \text{otherwise, with } i^* \text{ minimal such that } \sigma(i^*) \in \mathbb{N}\backslash\{0\}. \end{cases}$$

$$y_\sigma = \begin{cases} 0, & \text{if } |\mathrm{content}(\sigma)| \leq 1; \\ 1, & \text{otherwise;} \end{cases}$$

$$z_\sigma = \begin{cases} 0, & \text{if } \mathrm{content}(\sigma) \cap \mathrm{range}(a) = \emptyset; \\ \min(\mathrm{content}(\sigma) \cap \mathrm{range}(a)), & \text{otherwise.} \end{cases}$$

Intuitively, x_σ is the first non-zero element appearing in σ, y_σ tests whether the sequence's content comprises at least two elements and z_σ searches for the minimal value of function a. Observe that all these functions are total and can be computed iteratively. The computation of function y may need an intern variable which is either empty or stores the first natural number occurring in σ for comparison with the current data point. Let $\mathrm{pad} \in \mathcal{R}$ denote a total recursive padding function and p_0 be a W-index for the empty set.
So there is an **It**-learner $h \in \mathcal{P}$ whose *starred* learner h^* satisfies the following condition:

$$h^*(\sigma) = \begin{cases} \mathrm{pad}(p_0;0;0;0), & \text{if } \mathrm{content}(\sigma) \subseteq \{0\}; \\ \mathrm{pad}(\varphi_{x_\sigma}(y_\sigma;z_\sigma);\ x_\sigma;y_\sigma;z_\sigma) & \text{else, if } \varphi_{x_\sigma}(y_\sigma;z_\sigma)\downarrow; \\ \uparrow, & \text{otherwise.} \end{cases}$$

Let $\mathcal{L} = \mathbf{TxtItSMonEx}(h)$ denote the class of languages h infers.

By way of contradiction assume $\mathcal{L} \subseteq \mathbf{TxtPsdSMonEx}(g)$ for some learner $g \in \mathcal{P}$. Again let $g^*(\sigma) = g(\mathrm{content}(\sigma);|\sigma|)$ denote the *starred* learner of g.

The Parameter Theorem gives us another function $\mathrm{union} \in \mathcal{R}$ such that, for each index i and all finite sets D, we have $W_{\mathrm{union}(i;D)} = W_i \cup D$. Now using ORT once more, there is an index p, a total recursive function $e \in \mathcal{R}$ with $\mathrm{range}(a) \cap \mathrm{range}(e) = \emptyset$ as well as $0 \notin \mathrm{range}(e)$, and a computable sequence $(\sigma_i)_{i\in\mathbb{N}}$ of sequences such that the following construction holds:

The sequences σ_i are defined recursively,

$$\sigma_0 = \emptyset;$$

$$\forall i\colon \sigma_{i+1} = \sigma_i \diamond \begin{cases} e(2i)^t, & \text{for } t \text{ minimal such that } g^*(\sigma_i)\downarrow \neq g^*(\sigma_i \diamond e(2i)^t)\downarrow; \\ e(2i+1)^t, & \text{for } t \text{ minimal such that } g^*(\sigma_i)\downarrow \neq g^*(\sigma_i \diamond e(2i+1)^t)\downarrow, \end{cases}$$

for whatever case is detected first by a particular search, if any. This is possible since both conditions are semi-decidable and can be tested iteratively in parallel for increasing parameter t. Index p shall be such that

$$W_p = \bigcup_{i\in\mathbb{N};\ \sigma_i\downarrow} \mathrm{content}(\sigma_i).$$

Meaning that, at stage i. sequence σ_i is computed first and, if this computation halts, then $\mathrm{content}(\sigma_i)$ is enumerated. Furthermore, we have, for all sequences σ, and every index i,

$$\varphi_{e(i)}(y;z) = \begin{cases} \mathrm{ind}(\{e(i)\}), & \text{if } y = 0; \\ \mathrm{union}(p;\{e(i)\}), & \text{else, if } z = 0; \\ \mathrm{union}(p;D^* \cup \{a(D^*)\}), & \text{otherwise, with } z = a(D^*). \end{cases}$$

Case 1: Language $L = \bigcup_{i\in\mathbb{N}} \mathrm{content}(\sigma_i)$ is infinite.

Meaning sequence σ_i is defined for every i. Then we have $L \in \mathcal{L}$: Let $T \in \mathbf{Txt}(L)$ be a text for L and i' such that $x_T = e(i')$—the first element of L to appear in text T. The learner h working on T first conjectures the empty set $\emptyset$ and changes its mind to the singleton $\{e(i')\}$ when it sees $e(i')$ the first time. From this point on, h *semantically* behaves the same as $\varphi_{e(i')}$. It reaches its final and correct guess $W_p \cup \{e(i')\}$ as soon as it sees the first member of the (infinite) set L different from $e(i')$. All these mind changes are permitted using **SMon**. From $L \cap \mathrm{range}(a) = \emptyset$ and $|L| > 1$, we get that h also *syntactically* converges to hypothesis

$$\mathrm{pad}(\mathrm{union}(p; \{e(i')\}); e(i'); 1; 0).$$

Scientist g cannot learn L from text $\bigcup_{i \in \mathbb{N}} \sigma_i$ as it makes infinitely many mind changes by definition.

Case 2: Language $L = \bigcup_{i \in \mathbb{N}} \mathrm{content}(\sigma_i)$ is finite.

Hence, from one point on, sequences σ_i are undefined. Let σ_k be the last one defined. Then both of the following languages are in $\mathcal{L}$:

$$L_1 = \mathrm{content}(\sigma_k) \cup \{e(2k);\ a(\mathrm{content}(\sigma_k) \cup \{e(2k)\})\}$$
$$L_2 = \mathrm{content}(\sigma_k) \cup \{e(2k+1);\ a(\mathrm{content}(\sigma_k) \cup \{e(2k+1)\})\}$$

Suppose $T \in \mathbf{Txt}(L_1)$ to be a text for L_1. For ease of notation, let $D^* = \mathrm{content}(\sigma_k) \cup \{e(2k)\}$. We have either $x_T = a(D^*)$ or $x_T = e(i')$ for some i'. So learner h behaves like $\varphi_{a(D^*)}$ or $\varphi_{e(i')}$, respectively. In both cases it *semantically* converges to the correct conjecture $W_{\mathrm{union}(p; D^* \cup \{a(D^*)\})} = W_p \cup D^* \cup \{a(D^*)\} = L_1$. This is because $a(D^*)$ is the sole (and therefore minimal) member of $L_1 \cap \mathrm{range}(a)$ and L_1 has at least two elements. In both cases the resulting sequence of hypotheses suffices **SMon**. Again h also *syntactically* converges on T to

$$\mathrm{pad}(\mathrm{union}(p; D^* \cup \{a(D^*)\});\ x_T; 1; a(D^*)).$$

The reasoning for L_2 is the same.

Additionally, for every i, the singleton set $\{e(i)\}$ is in $\mathcal{L}$ since it can be inferred by function $\varphi_{e(i)}$. By assumption, we get $\{e(i)\} \in \mathbf{TxtPsdSMonEx}(g)$. Therefore, there are a numbers $t(i)$ minimal such that $e(i) \in W_{g^*(e(i)^{t(i)})}$ for every index i. In particular, from the strong monotony of g, we get

$$e(2k) \in W_{g^*(e(2k)^{t(2k)} \diamond \sigma_k)} \quad \text{and} \quad e(2k+1) \in W_{g^*(e(k+1)^{t(2k+1)} \diamond \sigma_k)}.$$

By definition of the sequences σ_i and the rearrangement-independence of g, we have

$$g^*(e(2k)^{t(2k)} \diamond \sigma_k) = g^*(\sigma_k) = g^*(e(2k+1)^{t(2k+1)} \diamond \sigma_k)$$

In conclusion, g can learn neither L_1 nor L_2 strongly monotone from texts starting with prefix σ_k, a contradiction.

qed

Corollary 4.2:
We have $[\mathbf{TxtPsdSMonEx}] \subsetneq [\mathbf{TxtGSMonEx}]$.

Proof: The inclusion again is trivial. On the other hand, the above proposition implies $[\mathbf{TxtGSMonEx}] \not\subseteq [\mathbf{TxtPsdSMonEx}]$ as every iterative learner can be simulated by a **G**-learner.

qed

So strongly monotone learning is different. Not only does it give us the first (and up until now *sole*) criterion for which **Psd**-learning is strictly weaker, among all delayable restrictions we considered, **SMon** is also the only one for which a result similar to Proposition 3.15 could not be obtained yet. A significant intermediate result though is proven next.

Proposition 4.3:
There is a class of languages which is rearrangement-independently learnable by a strongly monotone scientist, but cannot be inferred by a total *set-driven scientist. Particularly, we have* $[\mathcal{R}\mathbf{TxtSdSMonEx}] \subsetneq [\mathbf{TxtPsdSMonEx}]$.

Proof: For the separation we define a **Psd**-learner h: Let p_0 be an index of the empty set. For all finite sets D and numbers t, we have

$$h(D;t) = \begin{cases} p_0, & \text{if } D = \emptyset; \\ \varphi_{\max D}(D;t), & \text{otherwise.} \end{cases}$$

Let $\mathcal{L} = \mathbf{TxtPsdSMonEx}(h)$ be the collection of languages identified by h. It is now sufficient to prove $\mathcal{L}$ to be unidentifiable by any *total* set-driven learner.

By way of contradiction assume $\mathcal{L} \subseteq \mathbf{TxtSdEx}(h')$ for some learner $h' \in \mathcal{R}$. With ORT we get an index p and a total recursive function $e \in \mathcal{R}$ with $0 \notin \operatorname{range}(e)$ such that, for all i,

$$W_p = \{e(0)\} \cup \{e(i) \mid \forall 0 < j \leq i \colon h'(\{e(0); \ldots; e(j-1)\}) \neq h'(\{e(0); \ldots; e(j)\})\};$$

$$\varphi_{e(0)}(D;t) = p;$$
$$\varphi_{e(i+1)}(D;t) = \begin{cases} p, & \text{if } h'(\{e(0); \ldots; e(i)\}) \neq h'(\{e(0); \ldots; e(i+1)\}); \\ \operatorname{union}(p; D), & \text{otherwise.} \end{cases}$$

Case 1*:* W_p is infinite.

Then language $W_p = \operatorname{range}(e)$ is in $\mathcal{L}$: As $0 \notin \operatorname{range}(e)$, for every $\emptyset \neq D \subset_{\text{fin}} \operatorname{range}(e)$ and arbitrary t, we have $\varphi_{\max D}(D;t) = p$ since the condition $h'(\{e(0); \ldots; e(i)\}) \neq h'(\{e(0); \ldots; e(i+1)\})$ always holds.

Scientist h' cannot learn W_p from text e as it makes infinitely many mindchanges.

Case 2: W_p is finite.

In this case there is some k such that $W_p = \text{content}(e[k+1])$ and both finite sets $L_1 = W_p$ and $L_2 = W_p \cup \{e(k+1)\}$ are in $\mathcal{L}$. The reasoning for L_1 is the same as in the first case. Now let $T \in \mathbf{Txt}(L_2)$ be a text. There is some index n_0 minimal such that $L_2 = \text{content}(T[n])$ for all $n \geq n_0$. By the strong monotony of e, $e(k+1) = \max \text{content}(T[n])$ holds for these n. This implies

$$\forall n \geq n_0 \colon h(\text{content}(T[n]); n) = \varphi_{e(k+1)}(L_2; n) = \text{union}(p; L_2).$$

This is because L_2 is finite and $h'(\{e(0); \ldots; e(k)\}) = h'(\{e(0); \ldots; e(k+1)\})$ by assumption. Prior to n_0 the learner h working on T conjectured the subsets W_p, before the first occurrence of $e(k+1)$, and $W_p \cup \text{content}(T[n]) \subseteq L_2$ after it. These mindchanges suffice **SMon**.

Again h' cannot learn both L_1 and L_2 as we know from the definition of W_p that $h'(L_1) = h'(L_2)$, a contradiction.

qed

4.2 Behaviorally Correct Learning

In all the the previous work we only discussed explanatory language learning, requiring a learner to *syntactically* converge to a single explanation of the learnee. This is the oldest and most common setting in computational learning. But it is not the only one. Instead, one may ask for mere *semantical* convergence. Meaning that, from one point on, the scientist only gives correct conjectures. This idea is formalized in the notion of a *behaviorally correct* learner.

Definition 4.4 (Bārzdiņš [4]; cf. Kötzing [20])**:**
Suppose $p \in \mathcal{P}$ to be an infinite sequence and $T \in \mathbf{Txt}$ a text.
We define the following learning restriction, *behavioral correctness*:

$$\mathbf{Bc}(p; T) \Leftrightarrow \forall^\infty n \colon W_{p(n)} = \text{content}(T)$$

□

Evidently **Bc**-learning is an extension of explanatory learning and is delayable as well. There is a result in the field of function identification stating that behaviorally correct scientists can identify *strictly* more concept classes, it carries over to the case of language learning.

Theorem 4.5 (Case & Smith [12])**:**
We have $[\mathbf{TxtGEx}] \subsetneq [\mathbf{TxtGBc}]$.

■

The new freedom of convergence, however, is counteracted if paired with some of the delayable restrictions of Definition 3.8.

Definition 4.6 (Kötzing & Palenta [22])**:**
Suppose $p \in \mathfrak{P}$ to be a partial function and $T \in \mathbf{Txt}$ a text.
For every p, we define the following two sets:

$$\mathrm{Sem}(p) = \{p' \in \mathfrak{P} \mid \forall i\colon p(i)\!\downarrow \Rightarrow (p'(i)\!\downarrow \wedge\, W_{p(i)} = W_{p'(i)})\}$$
$$\mathrm{Mc}(p) = \{p' \in \mathfrak{P} \mid \forall i\colon p(i)\!\downarrow = p(i+1)\!\downarrow \Rightarrow p'(i)\!\downarrow = p'(i+1)\!\downarrow\}$$

A learning restriction $\delta\colon \mathfrak{P} \times \mathbf{Txt} \to \{0;1\}$ is said to be *semantic* if, for all p and T, the following condition holds

$$\delta(p;T) \Rightarrow \forall p' \in \mathrm{Sem}(p)\colon \delta(p';T)$$

and *pseudo-semantic* if

$$\delta(p;T) \Rightarrow \forall p' \in \mathrm{Sem}(p) \cap \mathrm{Mc}(p)\colon \delta(p';T).$$

□

Intuitively, a restriction is semantic if it allows all hypotheses to be replaced by semantically equivalent ones. A restriction is pseudo-semantic if such an alternation is required not to bring new mindchanges. The conjunction of two (pseudo-)semantic restrictions is again (pseudo-)semantic [22]. *All* restrictions considered in this thesis—content-based and delayable ones, as well as all success criteria—are pseudo-semantic and all but **Conv**, **SNU**, **SDec** and **Ex** are semantic. The next theorem can be obtained by realizing that pseudo-semantic restrictions which are not semantic enforce additionally syntactical update constraints. So the equalities stated below are already established on the level of learning restrictions.

Theorem 4.7:
For each sequence generating operator β, the following three equalities hold:

(i) $[\mathbf{Txt}\beta\mathbf{ConvBc}] = [\mathbf{Txt}\beta\mathbf{ConvEx}]$

(ii) $[\mathbf{Txt}\beta\mathbf{SNUBc}] = [\mathbf{Txt}\beta\mathbf{SNUEx}]$

(iii) $[\mathbf{Txt}\beta\mathbf{SDecBc}] = [\mathbf{Txt}\beta\mathbf{SDecEx}]$

■

Corollary 4.8:
We have $[\mathbf{TxtPsdSNUBc}] = [\mathbf{TxtGSNUBc}] = [\mathbf{TxtGEx}]$.

Proof: Immediate from Proposition 3.12.

qed

This excursion to behaviorally correct learning provides us with the concepts needed to illustrate a special trait of the result shown in Proposition 4.1. Recall that $\mathbf{Txt}\beta\mathbf{SMonEx}$ was the first scheme of learning criteria for which it makes a difference whether the order of the input is available or not. Surprisingly so, this is no longer true when one only requires semantical convergence. In fact, now the mere set of shown examples is sufficient to identify a language.

Proposition 4.9:
Every class of languages identifiable by a strongly monotone behaviorally correct learner, can be so learned by a set-driven scientist. Thus, we have $[\mathbf{TxtSdSMonBc}] = [\mathbf{TxtPsdSMonBc}] = [\mathbf{TxtGSMonBc}]$.

Moreover, we can assume the learners to be total.

Proof: It remains to show that every concept class in $[\mathbf{TxtGSMonBc}]$ can be so learned by a total and set-driven scientist.

So let $h \in \mathcal{R}$ be a *total* learner (Theorem 3.10) and $\mathcal{L} = \mathbf{TxtGSMonBc}(h)$. For every finite set D, let again $D^{\leq t}$ denote the set of all sequences with content in D and length at most t. Using s-m-n, there is a partial recursive function h' such that, for all D,

$$W_{h'(D)} = \bigcup_{t \in \mathbb{N}} \bigcup_{\sigma \in D^{\leq t}} W_{h(\sigma)}.$$

As h is total, so is h'.

Claim 1*:* Learner h' **Bc**-identifies $\mathcal{L}$.

Suppose $L \in \mathcal{L}$ to be a language and $T \in \mathbf{Txt}(L)$ a text for L. As h learns L from T behaviourally correctly there is an index n_0 such that

$$\forall n \geq n_0 \colon W_{h(T[n])} = W_{h(T[n_0])} = L.$$

Observe that $T[n] \in (\mathrm{content}(T[n]))^{\leq n}$, for any n. From the strong monotony of h, i.e. $W_{h(\sigma)} \subseteq L$ for all $\sigma \in (L \cup \{\#\})^*$, we now get

$$\forall n \geq n_0 \colon W_{h'(\mathrm{content}(T[n]))} = \bigcup_{t \in \mathbb{N}} \bigcup_{\sigma \in (\mathrm{content}(T[n]))^{\leq t}} W_{h(\sigma)} = L.$$

Claim 2*:* The learner h' is strongly monotone on texts for languages in $\mathcal{L}$.

By way of contradiction assume otherwise.
Let $L \in \mathcal{L}$ and $D \subseteq D' \subset_{\mathrm{fin}} L$ be such that $W_{h'(D)} \not\subseteq W_{h'(D')}$. That means, there is some number t' and sequence $\sigma \in D^{\leq t'}$ satisfying

$$W_{h(\sigma)} \not\subseteq \bigcup_{t \in \mathbb{N}} \bigcup_{\tau \in (D')^{\leq t}} W_{h(\tau)}.$$

The latter is equivalent to $W_{h(\sigma)} \not\subseteq W_{h(\tau)}$, for *any* $\tau \in (D')^{\leq t}$ and any t, especially those $\tau \sqsupseteq \sigma$ extending σ. This is a contradiction to h being a strongly monotone learner for language L.

qed

It becomes clear that the separation shown in the case of explanatory learning (Proposition 4.1), as well as the difficulties to settle the relation of set-driven and rearrangement-independent **SMon**-learning really depend on the syntactical

convergence of successful explanatory learning sequences. Moreover, behaviorally correct, strongly monotone learning can even be done by iterative learners. A class of inference machines normally considered to be even less powerful than set-driven scientists (compare Theorem 2.18).

Proposition 4.10:
Suppose a learning restriction δ to be semantic. Every class of languages identifiable with respect to δ can be so learned iteratively. Thus, we have $[\mathbf{TxtIt}\delta] = [\mathbf{TxtG}\delta]$.

Proof: It is sufficient to show that every $\mathbf{TxtG}\delta$-learnable class of languages can be inferred by an iterative scientist with respect to δ. Let $h \in \mathcal{P}$ be a learner and $\mathcal{L} = \mathbf{TxtG}\delta(h)$ the class of languages h identifies.

Let $\mathrm{pad} \in \mathcal{R}$ denote a padding function strictly monotone increasing and $\mathcal{R} \ni \mathrm{unpad}_2 \colon \mathrm{pad}(e;\sigma) \mapsto \sigma$ its total recursive inversion. Furthermore, let p_0 be an index for the empty set. Consider the following $\mathbf{It}$-learner $h' \in \mathcal{P}$:

$$\begin{aligned} h'(\emptyset) &= p_0; \\ h'(e;x) &= \mathrm{pad}(h(\mathrm{unpad}_2(e) \diamond x); \mathrm{unpad}_2(e) \diamond x). \end{aligned}$$

Intuitively speaking, h' unpads the previous input sequence $\sigma = \mathrm{unpad}_2(e)$ stored in its last conjecture and uses this information to simulate the computation of $h(\sigma \diamond x)$ in the next step with x being the new data point. The new guess again is padded with the updated input $\sigma \diamond x$. This is possible as semantic learning does not require syntactical convergence and $\sigma \diamond x$ is always finite, thus, an eligible padding input. If h is total, so is h'. It is easy to see that h' identifies $\mathcal{L}$ with respect to δ since by construction h and h' produce semantically equivalent sequences of hypotheses. We get, for any language $L \in \mathcal{L}$ and all texts $T \in \mathbf{Txt}(L)$,

$$\forall i \colon W_{\mathbf{It}(h';T)(i)} = W_{\mathbf{G}(h;T)(i)}.$$

qed

Corollary 4.11:
We have $[\mathbf{TxtItSMonBc}] = [\mathbf{TxtGSMonBc}]$.

Proof: Immediate from the observation that **SMonBc** is a semantic restriction.
qed

For behaviorally correct strongly monotone learning all considered interaction operators are equally powerful. This has two main reasons as outlined in the two propositions: Mere semantical convergence allows, in every stage, to unify previously conjectured sets with the current guess to ensure strong monotony. Furthermore, **Bc**-learning, via padding, enables a learner to use all its hypotheses as extern memory in fact providing it with full information.

4.3 Conclusion

In this thesis we examined the question to what extent FULK's result—that rearrangement-independence does not reduce the power of Gold-style language learning from text—can be transferred to learning with various learning restrictions. We certified content-based learning to have FULK's property. For delayable learning, however, we found a more complex situation. While some of the restrictions allowed for partially set-driven learning, strongly monotone explanatory learning is crucially depending on the knowledge of the order in which the members of the target language is presented. Moreover, we were able to prove that this dependency is due to the characteristics of explanatory learning, more precisely, the characteristics of syntactical convergence. The question whether restricted set-driven learning is equally powerful was solved negatively for almost all considered restrictions. Notwithstanding, still many questions remain unanswered. For many delayable restrictions including decisiveness, conservatism and the weaker variations of monotony it is still unknown whether strictly less concept classes can be inferred by rearrangement-independent learners. For strongly monotone explanatory inference the power of iterative and set-driven learners has to be examined more deeply. We are confident that a solution for these questions will soon be found.

The presented work was able to further clarify, at least for computational learning, what information is necessary to successfully recognize certain classes of languages. This knowledge will hopefully be useful for various applications of algorithmic language learning in the future.

Bibliography

[1] Dana Angluin. Inductive Inference of Formal Languages from Positive Data. *Inf. Contr.*, 45(2):117 – 135, 1980.

[2] Ganesh Baliga, John Case, Wolfgang Merkle, Frank Stephan, and Rolf Wiehagen. When Unlearning Helps. *Inf. Comput.*, 206(5):694–709, May 2008.

[3] Jānis M. Bārzdiņš. Inductive Inference of Automata, Functions and Programs. *Int. Math. Congress*, pages 771–776, 1974. Vancouver.

[4] Jānis M. Bārzdiņš. Two Theorems on the Limiting Synthesis of Functions. *Theory of Algorithms and Programs*, 1(210):82–88, 1974. Latvian State University (in Russian).

[5] Lenore Blum and Manuel Blum. Toward a Mathematical Theory of Inductive Inference. *Inf. Contr.*, 28(2):125 – 155, 1975.

[6] Manuel Blum. A Machine-Independent Theory of the Complexity of Recursive Functions. *J. ACM*, 14(2):322–336, 1967.

[7] John Case. Periodicity in Generations of Automata. *Mathematical Systems Theory*, 8(1):15–32, 1974.

[8] John Case and Timo Kötzing. Dynamically Delayed Postdictive Completeness and Consistency in Learning. In *Proc. of ALT*, pages 389–403, 2008.

[9] John Case and Timo Kötzing. Strongly Non-U-Shaped Learning Results by General Techniques. In *Proc. of COLT*, pages 181–193, 2010.

[10] John Case and Timo Kötzing. Topological Separations in Inductive Inference. In *Proc. of ALT*, pages 128–142, 2013.

[11] John Case and Samuel E. Moelius. Optimal Language Learning from Positive Data. *Inf. Comput.*, 209(10):1293–1311, 2011.

[12] John Case and Carl Smith. Comparison of Identification Criteria for Machine Inductive Inference. *Theoretical Computer Science*, 25(2):193 – 220, 1983.

[13] Rusins Freivalds and Rolf Wiehagen. Inductive Inference with Additional Information. *Elektronische Informationsverarbeitung und Kybernetik*, 15(4):179–185, 1979.

[14] Mark A. Fulk. Prudence and other Conditions on Formal Language Learning. *Inf. Comput.*, 85(1):1 – 11, 1990.

[15] Mark A. Fulk. Robust Separations in Inductive Inference. In *Proc. of the Third Annual Workshop on Computational Learning Theory*, COLT '90, pages 405–410, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.

[16] E. Mark Gold. Language Identification in the Limit. *Inf. Contr.*, 10(5):447–474, 1967.

[17] Sanjay Jain, Daniel N. Osherson, James S. Royer, and Arun Sharma. *Systems That learn : An Introduction to Learning Theory*. Learning, Development, and Conceptual Change. MIT Press, Cambridge, Massachusetts US, 2nd edition, 1999. A Bradford Book.

[18] Klaus P. Jantke. Monotonic and Non-monotonic Inductive Inference. *New Generation Comput.*, 8(4):349–360, 1991.

[19] Efim B. Kinber and Frank Stephan. Language Learning from Texts: Mindchanges, Limited Memory, and Monotonicity. *Inf. Comput.*, 123(2):224–241, 1995.

[20] Timo Kötzing. *Abstraction and Complexity in Computational Learning in the Limit*. PhD Thesis, University of Delaware, 2009.

[21] Timo Kötzing. A Solution to Wiehagen's Thesis. In *Proc. of STACS*, pages 494–505, 2014.

[22] Timo Kötzing and Raphaela J. Palenta. A Map of Update Constraints in Inductive Inference. In *Proc. of ALT*, 2014.

[23] Steffen Lange, Jochen Nessel, and Rolf Wiehagen. Learning Recursive Languages from Good Examples. *Ann. of Math. and Art. Intell.*, 23(1-2):27–52, January 1998.

[24] Steffen Lange and Thomas Zeugmann. Monotonic versus Non-Monotonic Language Learning. In G. Brewka, K. P. Jantke, and P. H. Schmitt, editors, *Nonmonotonic and Inductive Logic: Proc. of the Second International Workshop*, pages 254–269. Springer, Berlin, Heidelberg, 1993.

[25] Daniel N. Osherson, Michael Stob, and Scott A. Weinstein. *Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists*. MIT Press, Cambridge, Massachusetts US, 1986.

[26] Karl R. Popper. *Objective Knowledge: An Evolutionary Approach*. Oxford University Press, 1979.

[27] Karl R. Popper. *Karl Popper: Logik Der Forschung*. Klassiker Auslegen. Akademie Verlag GmbH, 1998.

[28] Hartley Rogers. *Theory of Recursive Functions and Effective Computability*. MIT Press, Cambridge, Massachusetts US, 1987.

[29] Gisela Schäfer-Richter. *Über Eingabeabhängigkeit und Komplexität von Inferenzstrategien*. PhD Thesis, Rheinisch-Westfälische Technische Hochschule Aachen, 1984.

[30] Kenneth Wexler and Peter W. Culicover. *Formal Principles of Language Acquisition*. MIT Press, Cambridge, MA, 1980.

[31] Rolf Wiehagen. A Thesis in Inductive Inference. In J. Dix, K. P. Jantke, and P. H. Schmitt, editors, *Nonmonotonic and Inductive Logic: Proc. of the 1st International Workshop*, pages 184–207. Springer, Berlin, Heidelberg, 1991.

Statement of Authorship

I hereby confirm that I composed the presented master thesis solely by myself and that it describes my own work, unless otherwise acknowledged in the text. I have used the tools and sources cited in the text only.

I agree that one copy of my presented master thesis may remain at the disposal of the archives of the Friedrich Schiller University Jena.

Selbstständigkeitserklärung

Ich erkläre, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Seitens des Verfassers bestehen keine Einwände die vorliegende Masterarbeit für die öffentliche Benutzung im Universitätsarchiv zur Verfügung zu stellen.

Martin Schirneck, Jena, March 2015