# Learning in the Presence of Partial Explanations *

Sanjay Jain †
Dept. of Computer Science
University of Rochester
Rochester, New York 14627
jain@cs.rochester.edu

Arun Sharma ‡
Dept. of Computer & Information Sciences
University of Delaware
Newark, Delaware 19716
arun@udel.edu

March 12, 2007

## Abstract

*A preliminary version of this paper was presented at the Third Conference on Theoretical Aspects of Reasoning about Knowledge, 1990, Asilomar, California.

†Supported by NSF grant CCR 832-0136.

‡Supported by NSF grant CCR 871-3846.

The effect of a partial explanation as additional information in the learning process is investigated. A scientist performs experiments to gather experimental data about some phenomenon, and then, tries to construct an explanation (or theory) for the phenomenon. A plausible model for the practice of science is an inductive inference machine (scientist) learning a program (explanation) from graph (set of experiments) of a recursive function (phenomenon). It is argued that this model of science is not an adequate one, as scientists, in addition to performing experiments, make use of some approximate partial explanation based on the "state of the art" knowledge about that phenomenon. An attempt has been made to model this partial explanation as an additional information in the scientific process. It is shown that inference capability of machines is improved in the presence of such a partial explanation. The quality of this additional information is modeled using certain "density" notions. It is shown that additional information about a "better" quality partial explanation enhances the inference capability of learning machines as scientists more than a "not so good" partial explanation. Similar enhancements to inference of approximations, a more sophisticated model of science, are demonstrated.

Inadequacies in Gold's paradigm of language learning are investigated. It is argued that Gold's model fails to incorporate certain additional information that children get from their environment. Children are sometimes told about some grammatical rule that enumerates elements of the language. It is argued that these rules are a kind of additional information. They enable children to see in advance elements that are yet to appear in their environment. Also, children are being given some information about what is not in the language. Sometimes, they are rebuked for making incorrect utterances, or are told of a rule that enumerates certain non-elements of the language. An attempt has been made to extend Gold's model to incorporate both the above types of additional information. It is shown that either type of additional information

enhances the learning capability of formal language learning devices.

# 1 Introduction

Consider the scenario in which a subject is attempting to learn 'its' environment. At any given time, the subject receives a finite piece of data about its environment, and based upon this finite information, conjectures an explanation about the environment. The subject is said to *learn* its environment, just in case, the explanations conjectured by the subject become fixed over time, and this fixed explanation is a correct representation of the subject's environment. Computational learning theory provides a framework for the study of above scenario when the subject is an algorithmic machine. This paper argues that a subject, in a number of learning situations, has some partial explanation about its environment as additional information. We introduce various formulations of this partial explanation and investigate the impact of providing such an additional information on the learning capability of algorithmic devices. The two learning situations investigated are the practice of science and language acquisition.

Picture a scientist performing all possible experiments (in arbitrary order) associated with a phenomenon, noting the result of each experiment, while simultaneously, but algorithmically, conjecturing a succession of candidate explanations for the phenomenon. A criterion of success is that the scientist eventually conjectures an explanation which he/she never gives up, and this final explanation correctly predicts the results of every experiment about the phenomenon. The set of all pairs of the form ⟨experiment, corresponding result⟩ associated with the phenomenon can be taken to be coded by a function from $\mathbb{N}$ to $\mathbb{N}$, where $\mathbb{N}$ is the set of natural numbers. If, the ever experimenting scientist in the above scenario is replaced by a machine, then algorithmic identification in the limit of a program for a recursive function

from its graph serves as a plausible model for the practice of science. This is essentially the theme of inductive inference studied by Gold [Gol67]. A machine **M** **Ex**-*identifies* a function iff (by definition) the scientist is replaced by machine **M** in the above scenario for success. **Ex** is defined to be the class of sets $\mathcal{S}$ of recursive functions such that some machine **Ex**-identifies each recursive function in $\mathcal{S}$.

We feel that the above model of science is somewhat inadequate. For one thing, a scientist has more information available than just the result of experiments. For another, the result of a scientist's investigation need not be a final theory. C. S. Peirce [Pei58, Rei70] argues that science is a non-terminating process of successive approximations. Finally, a scientist might have some partial explanation of the phenomenon based on the "state of the art" knowledge about that phenomenon and probably uses this additional information in coming up with an explanation. The model described above does not take in to account the presence of this additional information. In the present paper, we attempt to model this additional information.

Our approach to modeling a scientist's knowledge of partial explanations is described thus. We require a learning machine to be presented with any program which computes a partial recursive function that (1) agrees sufficiently (infinitely-often) with the function being learned; and (2) does not contradict the function being learned. In other words, a machine learning a function $f$, is fed, in addition to a graph of $f$, a program that computes an infinite subset of $f$ as additional information. For a number of function inference criteria, we show that such an additional information enhances the learning capability of machines.

We model the quality of partial explanations using certain "density" notions due to Royer [Roy86]. Intuitively, a good partial explanation has, in some sense, a greater agreement with the function being learned than a not so good partial explanation. We show that a better quality partial explana-

tion enhances the function inference capability of machines more than a not so good partial explanation.

The restriction that the partial function computed by the additional information program not contradict the function being learned, we feel, makes our approach a simplistic one, as there is no reason to believe that the state of the art partial explanation available to a scientist has only errors of omission and no errors of commission.

A related idea to "scientific" inference of functions is Gold's seminal notion of *identification* [Gol67]. We will refer to it as **TxtEx**-*identification* following [CL82]. In the following, a language is a *recursively enumerable* (r.e.) set, and a *grammar* (type 0) for a language is a program that enumerates the language [HU79] in some fixed acceptable programming system [Rog58, Rog67, MY78].

According to Gold's paradigm, a child (modeled as a machine) receives (in arbitrary order) all the well-defined strings of a language (a *text* for the language), and simultaneously, conjectures a succession of candidate grammars for the language being received. A criterion of success is for the child to eventually conjecture a correct grammar and to never change its conjecture thereafter. A machine **M** **TxtEx**-*identifies* a language iff (by definition) the child is replaced by machine **M** in the above scenario for success. Machine **M** is often called a language learning machine. **TxtEx** is defined to be the class of sets $\mathcal{L}$ of *r.e.* languages such that some machine **TxtEx**-identifies each language in $\mathcal{L}$.

Additional information, in the context of language learning, is modeled as a grammar for any infinite subset of the language being learned. Such an additional information to a language learning machine is justified, as it is not uncommon for an elder person (a parent or a teacher) to tell a child some small grammatical rule that enables the child to enumerate a list of elements of the language. Basically, this additional information, in the form

of a grammatical rule, enables the child to know certain elements of the language before these elements actually appear in the child's text.

It turns out that this kind of additional information, henceforth referred to as *positive* additional information, indeed increases the learning power of language learning machines. We further model the quality of positive additional information by measuring the "density of agreement" between the language being learned and the subset language whose grammar is provided as additional information. Not surprisingly, a "better quality" positive additional information enhances the learning capability of language learning devices more than a "not so good" positive additional information.

Gold's paradigm is based on the assumption that children are rarely informed of their grammatical errors. However, there are studies that refute this assumption [BB64, Dal76]. It is plausible that children are receiving some information about the complement of the language. A rebuke from an elder person for any ungrammatical utterance may act as a clue to a child about the absence of certain strings from the language. Better still, the elder person may provide the child with a rule that enumerates some ungrammatical strings in the language. We model such an additional information about what is not in the language by providing a language learning machine with any grammar that generates a subset of the *complement* of the language being learned. We refer to such additional information as *negative* additional information, and show that even negative additional information enhances the learning capability of language learning devices. We model the quality of negative additional information by measuring the density of agreement between the complement of the language being learned and the subset of the complement language whose grammar is provided as additional information. Even in this case, we show that a better quality negative additional information enhances the learning power of language learning devices more than a not so good negative additional information.

Finally, we consider language learning scenarios in which a machine is provided with both positive additional information and negative additional information.

In the present work, we are concerned with extending **TxtEx**-identification and **Ex**-identification by providing additional information to the learning machine. We briefly note other attempts to extending these fundamental learning paradigms. L. Blum and M. Blum [BB75] and Case and Smith [CS83], in the context of function inference, consider the case where the program inferred by the learning machine is allowed to make a finite number of mistakes. For language learning, Case and Lynes [CL82] and Osherson and Weinstein [OW82b, OW82a] consider learning criteria in which the grammar inferred is allowed to be a grammar for a finite variant of the language being learned. Smith [Smi82] considers the function inference criteria in which the learning machine is replaced by a "team" of learning machines and successful learning takes place if any one member of the team succeeds in learning the language. Osherson, Stob, and Weinstein [OSW86a] consider a generalized notion of team learning. Pitt [Pit84] has shown that the power of probabilistic machines can be neatly characterized in terms of teams [Smi82] of deterministic machines. Jain and Sharma [JS90b] consider team inference in the context of language learning. Royer [Roy86] and Smith and Velauthapillai [SV86] consider the case where the inferred program may have infinitely many anomalies, but the "density" of these anomalies is bounded. Recently, Case [Cas88] has considered language learning criteria in which the learning agent is allowed to converge in the limit to a finite set of grammars instead of one. Case, Jain and Sharma [CJS89] consider grammar size restrictions in Case's vacillating language learning criteria [Cas88]. Fulk [Ful85, Ful90a] and Jain and Sharma [JS89] consider other forms of additional information to learning machines.

# 2 Notation

Any unexplained recursion theoretic notation is from [Rog67]. $\mathbb{N}$ denotes the set of natural numbers, $\{0, 1, 2, 3, \ldots\}$. $\mathbb{N}^+$ denotes the set of positive integers, $\{1, 2, 3, \ldots\}$. Unless otherwise specified, $i, j, k, l, m, n$, with or without decorations, range over $\mathbb{N}$. $*$ denotes any finite number which is not prespecified. $a, b$ and $c$, with or without decorations, range over $(\mathbb{N} \cup \{*\})$. $\emptyset$ denotes the empty set. $\subseteq$ denotes subset. $\subset$ denotes proper subset. $S$, with or without decorations, ranges over subsets of $\mathbb{N}$. $\text{card}(S)$ denotes the cardinality of the set $S$. $\max, \min$ denote the maximum and minimum of a set, respectively. For $n \in \mathbb{N}$ and any two sets $S_1$ and $S_2$, $S_1 =^n S_2$ means $\text{card}((S_1 - S_2) \cup (S_2 - S_1)) \leq n$; $S_1 =^* S_2$ means $\text{card}((S_1 - S_2) \cup (S_2 - S_1))$ is finite.

$\eta$ and $\theta$ range over *partial* functions with arguments and values from $\mathbb{N}$. $f$ ranges over *total* functions with arguments and values from $\mathbb{N}$. For $n \in \mathbb{N}$ and partial functions $\eta$ and $\theta$, $\eta =^n \theta$ means that $\text{card}(\{x \mid \eta(x) \neq \theta(x)\}) \leq n$; $\eta =^* \theta$ means that $\text{card}(\{x \mid \eta(x) \neq \theta(x)\})$ is finite. $\text{domain}(\eta)$ and $\text{range}(\eta)$ denote the domain and range of the function $\eta$, respectively. For a set $S$, $\eta(S) = i$ means for all $x \in S, \eta(x) = i$.

$L$, with or without decorations, ranges over subsets of $\mathbb{N}$, usually construed as a *language*. $\mathcal{E}$ denotes the class of all *recursively enumerable* (r.e.) languages. $\mathcal{L}$ with or without decorations, ranges over subsets of $\mathcal{E}$, i.e., $\mathcal{L}$ is used to denote a class of r.e. languages. $\overline{L}$ denotes the complement of $L$, i.e., $\overline{L} = \mathbb{N} - L$.

$\varphi$ denotes a standard *acceptable* programming system [Rog58, Rog67, MY78]. $\Phi$ denotes an arbitrary Blum complexity measure [Blu67, HU79] for the $\varphi$-system. $\varphi_i$ denotes the partial computable function computed by program $i$ in the $\varphi$-system. $W_i = \text{domain}(\varphi_i)$. $W_i^s = \{x \leq s \mid \Phi_i(x) \leq s\}$. The set of all total recursive functions of one variable is denoted by $\mathcal{R}$. $\mathcal{S}, \mathcal{C}$, with or without decoration, range over subsets of $\mathcal{R}$. $\langle i, j \rangle$ stands for an arbi-

trary computable one to one encoding of all pairs of natural numbers onto $\mathbb{N}$ [Rog67] (we assume that $\langle i, j\rangle \geq \max(\{i, j\})$). $S \subseteq \mathbb{N}$ is called single-valued, just in case, $\{(x, y) \mid \langle x, y\rangle \in S\}$ represents a function. A single-valued set is said to be single-valued total, just in case, the function it represents is total. For $m$ and $n \in \mathbb{N}$, $[m, n]$ (respectively, $[m, n)$, $(m, n]$, and $[m, \infty)$) generally denotes $\{x \in \mathbb{N} \mid m \leq x \leq n\}$ (respectively, $\{x \in \mathbb{N} \mid m \leq x < n\}$, $\{x \in \mathbb{N} \mid m < x \leq n\}$, and $\{x \in \mathbb{N} \mid m \leq x\}$), although sometimes $[m, n], (m, n]$ and $[m, n)$ denote the corresponding interval in the real numbers. It will be clear from context which of these meanings is intended. Variable $d$, with or without decorations, ranges over real numbers in the real interval $[0, 1]$.

The quantifiers '$\forall^\infty$' and '$\exists^\infty$', essentially from [Blu67], mean 'for all but finitely many' and 'there exist infinitely many', respectively. The quantifier '$\exists!$' denotes 'there exists a unique'. $\square$ denotes the end of proof of a claim or a proposition; $\blacksquare$ denotes the end of proof of a theorem.

# 3 Function Inference

## 3.1 Fundamental Function Inference Paradigms

An *Inductive Inference Machine* (IIM) [Gol67] is an algorithmic device which takes as its input a set of data given one element at a time, and which from time to time, as it is receiving its input, outputs programs. IIMs have been used in the study of machine identification of programs for recursive functions as well as algorithmic learning of grammars for languages [BB75, CS83, Che81, Ful85, Gol67, OSW86b, Wie78]. For a survey of this work see [AS83, OSW86b, KW80, Cas86].

$\mathbf{M}$, with or without decorations, ranges over the class of inductive inference machines. For inference of a recursive function $f$ by an IIM $\mathbf{M}$, graph of $f$ is fed to $\mathbf{M}$ in any order. Without loss of generality [BB75,

CS83], we will assume that $\mathbf{M}$ is fed the graph of $f$ in the sequence $(0, f(0)), (1, f(1)), (2, f(2)), \ldots$. For all recursive functions $f$, $f|^n$ denotes the finite initial segment $((0, f(0)), (1, f(1)), \ldots, (n, f(n)))$. Variables $\sigma$ and $\tau$, with or without decorations, range over finite initial segments. $\mathbf{M}(\sigma)$ is the last output of $\mathbf{M}$ after receiving input $\sigma$ (note that $\sigma$ can be encoded as a natural number). We will assume, without loss of generality, that $\mathbf{M}(\sigma)$ is always defined. We say that $\mathbf{M}(f)$ converges to $i$ (written: $\mathbf{M}(f)\downarrow = i$) iff $(\forall^\infty n)[\mathbf{M}(f|^n) = i]$; $\mathbf{M}(f)$ is undefined otherwise.

**Definition 1** [Gol67, BB75, CS83] Suppose $a \in \mathbb{N} \cup \{*\}$.
(a) $\mathbf{M}$ $\mathbf{Ex}^a$-*identifies* a recursive function $f$ (written: $f \in \mathbf{Ex}^a(\mathbf{M})$) iff both $\mathbf{M}(f)\downarrow$ and $\varphi_{\mathbf{M}(f)} =^a f$.
(b) $\mathbf{Ex}^a = \{\mathcal{S} \subseteq \mathcal{R} \mid (\exists \mathbf{M})[\mathcal{S} \subseteq \mathbf{Ex}^a(\mathbf{M})]\}$.

Case and Smith [CS83] introduced another infinite hierarchy of identification criteria which we describe below. "**Bc**" stands for *behaviorally correct*. Barzdin [Bar74] independently introduced a similar notion.

**Definition 2** [CS83] Suppose $a \in \mathbb{N} \cup \{*\}$.
(a) $\mathbf{M}$ $\mathbf{Bc}^a$-*identifies* a recursive function $f$ (written: $f \in \mathbf{Bc}^a(\mathbf{M})$) iff $\mathbf{M}$, fed $f$, outputs over time an infinite sequence of programs $p_0, p_1, p_2, \ldots$ such that $(\forall^\infty n)[\varphi_{p_n} =^a f]$.
(b) $\mathbf{Bc}^a = \{\mathcal{S} \subseteq \mathcal{R} \mid (\exists \mathbf{M})[\mathcal{S} \subseteq \mathbf{Bc}^a(\mathbf{M})]\}$.

We usually write $\mathbf{Ex}$ for $\mathbf{Ex}^0$ and $\mathbf{Bc}$ for $\mathbf{Bc}^0$. Theorem 1 just below states some of the basic hierarchy results about the $\mathbf{Ex}^a$ and $\mathbf{Bc}^a$ classes.

**Theorem 1** *For all $n \in \mathbb{N}$,*
*(a)* $\mathbf{Ex}^n \subset \mathbf{Ex}^{n+1}$*;*
*(b)* $\bigcup_{n \in \mathbb{N}} \mathbf{Ex}^n \subset \mathbf{Ex}^*$*;*
*(c)* $\mathbf{Ex}^* \subset \mathbf{Bc}$*;*

*(d)* $\mathbf{Bc}^n \subset \mathbf{Bc}^{n+1}$*;*
*(e)* $\bigcup_{n \in \mathbb{N}} \mathbf{Bc}^n \subset \mathbf{Bc}^*$*; and*
*(f)* $\mathcal{R} \in \mathbf{Bc}^*$.

Parts (a), (b), (d), and (e) are due to Case and Smith [CS83]. John Steel first observed that $\mathbf{Ex}^* \subseteq \mathbf{Bc}$ and part (c) is due to Case and Smith [CS83]. Part (f) is due to Harrington [CS83]. Blum and Blum [BB75] first showed that $\mathbf{Ex} \subset \mathbf{Ex}^*$. Barzdin [Bar74] independently showed $\mathbf{Ex} \subset \mathbf{Bc}$.

## 3.2 Additional Information for Function Inference

We define the following notions of "density" from [Roy86]. Similar notions were also used by Smith and Velauthapillai [SV86] in the context of inductive inference.

**Definition 3** (S. Tennenbaum: see page 156 in [Rog67], [Roy86])
(a) Suppose that $A \subseteq \mathbb{N}$ and that $B$ is a finite, nonempty subset of $\mathbb{N}$. We define the *density of* $A$ *in* $B$ (denoted: $\mathbf{d}(A;B)$) as $\text{card}(A \bigcap B)/\text{card}(B)$.
(b) The *density* of a set $A$ (denoted: $\mathbf{d}(A)$) is $\lim_{n\to\infty} \inf(\{\mathbf{d}(A;\{z \mid z \leq x\}) \mid x \geq n\})$.

Intuitively, $\mathbf{d}(A;B)$ can be thought of as the probability of selecting an element of $A$ when choosing an arbitrary element from $B$.

We now describe our notion of additional information to an inductive inference machine learning a program from the graph of a recursive function. An IIM, trying to infer a program for a recursive function $f$, is given as additional information, a program for a partial recursive function $\eta$ which agrees with $f$ to some extent. In Definition 4 just below, we precisely define what we mean by "a partial function $\eta$ agrees with $f$ to some extent".

**Definition 4** Suppose $d$ is a real number in the interval $[0,1]$. A partial function $\eta$ is said to be *d-conforming* with a total function $f$ iff $\eta$ satisfies the following two conditions:

(1) $\eta \subseteq f$, i.e., $\eta$ does not contradict $f$; and

(2) $\mathbf{d}(\text{domain}(\eta)) \geq d$.

Using Definition 4, we define below our new learning criterion for identification of a program from graph of a recursive function in the presence of a partial explanation. In the following definition, **Ap** stands for **A**pproximate **p**artial additional information.

**Definition 5** Suppose $d$ is a real number in the interval $[0,1]$. Suppose $a \in \mathbb{N} \cup \{*\}$.

(a) A machine $\mathbf{M}$ $\mathbf{Ap}^d\mathbf{Ex}^a$-*identifies* a recursive function $f$ (written: $f \in \mathbf{Ap}^d\mathbf{Ex}^a(\mathbf{M})$) iff $\mathbf{M}$, fed $f$ and any program $p$ such that $\varphi_p$ is $d$-conforming with $f$, converges in the limit to a program $i$ such that $\varphi_i =^a f$.

(b) $\mathbf{Ap}^d\mathbf{Ex}^a = \{\mathcal{S} \subseteq \mathcal{R} \mid (\exists \mathbf{M})[\mathcal{S} \subseteq \mathbf{Ap}^d\mathbf{Ex}^a(\mathbf{M})]\}$.

We similarly define the corresponding identification criterion for **Bc** inference.

**Definition 6** Suppose $d$ is a real number in the interval $[0,1]$. Suppose $a \in \mathbb{N} \cup \{*\}$.

(a) A machine $\mathbf{M}$ $\mathbf{Ap}^d\mathbf{Bc}^a$-*identifies* a recursive function $f$ (written: $f \in \mathbf{Ap}^d\mathbf{Bc}^a(\mathbf{M})$) iff $\mathbf{M}$, fed $f$ and any program $p$ such that $\varphi_p$ is $d$-conforming with $f$, outputs an infinite sequence of programs $p_0, p_1, p_2, \ldots$ such that $(\forall^\infty n)[\varphi_{p_n} =^a f]$.

(b) $\mathbf{Ap}^d\mathbf{Bc}^a = \{\mathcal{S} \subseteq \mathcal{R} \mid (\exists \mathbf{M})[\mathcal{S} \subseteq \mathbf{Ap}^d\mathbf{Bc}^a(\mathbf{M})]\}$.

In the above identification criteria, $\varphi_p$ — an approximation to $f$, is a good plausible additional information to a machine trying to learn a program for

$f$ from a graph of $f$. However, $\varphi_p$ may be a very bad approximator locally for large intervals which may be of importance. To overcome this situation, we use the notion of "uniform density" from [Roy86] to define a new identification criterion.

**Definition 7** [Roy86] The *uniform density* of a set $A$ in intervals of length $\geq n$ (denoted: $\mathbf{ud}_n(A)$) is $\inf(\{d(A; \{z \mid x \leq z \leq y\}) \mid x, y \in \mathbb{N}$ *and* $y - x \geq n\})$. *Uniform density* of $A$ (denoted: $\mathbf{ud}(A)$) is $\lim_{n\to\infty} \mathbf{ud}_n(A)$.

Using the notion of *uniform density* we define an improved learning criterion. Definition 8 just below is an analogous notion to Definition 4 for this new density notion.

**Definition 8** Suppose $d$ is a real number in the interval $[0, 1]$. A partial function $\eta$ is said to be *d-uniform conforming* with a total function $f$ iff $\eta$ satisfies the following two conditions:
(1) $\eta \subseteq f$, i.e., $\eta$ does not contradict $f$; and
(2) $\mathbf{ud}(\text{domain}(\eta)) \geq d$.

In the following definition, **UAp** stands for **U**niform **A**pproximate **p**artial additional information.

**Definition 9** Suppose $d$ is a real number in the interval $[0, 1]$. Suppose $a \in \mathbb{N} \cup \{*\}$.
(a) A machine $\mathbf{M}$ $\mathbf{UAp}^d\mathbf{Ex}^a$-*identifies* a recursive function $f$ (written: $f \in \mathbf{UAp}^d\mathbf{Ex}^a(\mathbf{M})$) iff $\mathbf{M}$, fed $f$ and any program $p$ such that $\varphi_p$ is $d$-uniform conforming with $f$, converges in the limit to a program $i$ such that $\varphi_i =^a f$.
(b) $\mathbf{UAp}^d\mathbf{Ex}^a = \{\mathcal{S} \subseteq \mathcal{R} \mid (\exists \mathbf{M})[\mathcal{S} \subseteq \mathbf{UAp}^d\mathbf{Ex}^a(\mathbf{M})]\}$.

We similarly define the corresponding identification criterion for **Bc** inference.

**Definition 10** Suppose $d$ is a real number in the interval $[0,1]$. Suppose $a \in \mathbb{N} \cup \{*\}$.
(a) A machine $\mathbf{M}$ $\mathbf{UAp}^d\mathbf{Bc}^a$-*identifies* a recursive function $f$ (written: $f \in \mathbf{UAp}^d\mathbf{Bc}^a(\mathbf{M})$) iff $\mathbf{M}$, fed $f$ and any program $p$ such that $\varphi_p$ is $d$-uniform conforming with $f$, outputs an infinite sequence of programs $p_0, p_1, p_2, \ldots$ such that $(\forall^\infty n)[\varphi_{p_n} =^a f]$.
(b) $\mathbf{UAp}^d\mathbf{Bc}^a = \{\mathcal{S} \subseteq \mathcal{R} \mid (\exists \mathbf{M})[\mathcal{S} \subseteq \mathbf{UAp}^d\mathbf{Bc}^a(\mathbf{M})]\}$.

In what follows, we will refer to the two types of additional information as **Ap** and **UAp** type. Intuitively, **UAp** type additional information is a *better* kind of additional information; hence, we could expect the corresponding criteria of identification to be more general. Since, any $\mathbf{UAp}^d$ type additional information is also an $\mathbf{Ap}^d$ additional information we have the following two propositions.

**Proposition 1** $(\forall a \in \mathbb{N} \cup \{*\})(\forall d \in [0,1])$ $[\mathbf{Ap}^d\mathbf{Ex}^a \subseteq \mathbf{UAp}^d\mathbf{Ex}^a]$.

**Proposition 2** $(\forall a \in \mathbb{N} \cup \{*\})(\forall d \in [0,1])$ $[\mathbf{Ap}^d\mathbf{Bc}^a \subseteq \mathbf{UAp}^d\mathbf{Bc}^a]$.

Following theorems deal with the trade-offs between anomalies in the conjectured program, additional information, and types of identification criteria.

**Theorem 2** $(\forall d \in (0,1])(\forall m \in \mathbb{N})$ $[\mathbf{UAp}^d\mathbf{Ex} - \mathbf{Ap}^1\mathbf{Bc}^m \neq \emptyset]$.

Theorem 2 says that there are classes of recursive functions that can be **Ex**-identified with some **UAp** type additional information of non-zero density, but *cannot* be **Bc**-identified with any predetermined number of anomalies allowed per program and even the best possible **Ap** type additional information. In other words, the best possible **Ap** type additional information and a more general criterion of inference cannot, in general, compensate for any **UAp** type additional information of non-zero density.

**Proof of Theorem 2**: Let $N_0 = 0$. For $i \geq 0$, let $N_{2i+1} = N_{2i} + i + 1$ and $N_{2i+2} = N_{2i+1} * 2^i$. Let $S_j$ denote the set $\bigcup_{k \in \mathbb{N}}[N_{2*\langle j,k \rangle}, N_{2*\langle j,k \rangle+1})$. Consider the following class of functions:

$\mathcal{C} = \{f \in \mathcal{R} \mid$ the following conditions hold:
1. $f(\bigcup_{i \in \mathbb{N}}[N_{2i+1}, N_{2i+2})) = 0$
2. $(\forall j, x, y)[[x \in S_j \wedge y \in S_j] \Rightarrow [f(x) = f(y)]]$
$\}$

**Claim 1** $(\forall m \in \mathbb{N})[\mathcal{C} \notin \mathbf{Ap}^1\mathbf{Bc}^m]$.

**Proof of Claim 1**: Consider the following function $\eta$:

$$\eta(x) = \begin{cases} 0 & x \in \bigcup_{i \in \mathbb{N}}[N_{2i+1}, N_{2i+2}); \\ \uparrow & \text{otherwise.} \end{cases}$$

It is easy to see that $(\forall f \in \mathcal{C})[\mathbf{d}(\{x \mid \eta(x) = f(x)\}) = 1]$. Suppose by way of contradiction that machine $\mathbf{M}$, with a program for $\eta$ as the additional information, $\mathbf{Bc}^m$-identifies all $f$ in $\mathcal{C}$. It is, then, easy to convert $\mathbf{M}$ to $\mathbf{M}'$ such that $\mathbf{M}'$ $\mathbf{Bc}^m$-identifies all recursive functions. To see this, for a function $f$, define $f'$ as follows:

$$f'(x) = \begin{cases} 0 & \text{if } x \in \bigcup_{i \in \mathbb{N}}[N_{2i+1}, N_{2i+2}); \\ f(j) & \text{if } x \in S_j. \end{cases}.$$

Let $\mathcal{S} = \{f' \mid f \in \mathcal{R}\}$. Clearly, $[\mathcal{C} \in \mathbf{Ap}^1\mathbf{Bc}^m] \Rightarrow [\mathcal{S} \in \mathbf{Ap}^1\mathbf{Bc}^m] \Rightarrow [\mathcal{S} \in \mathbf{Bc}^m] \Rightarrow [\mathcal{R} \in \mathbf{Bc}^m]$. But $\mathcal{R} \notin \mathbf{Bc}^m$ [CS83]. Thus, no such machine $\mathbf{M}$ exists that $\mathbf{Ap}^1\mathbf{Bc}^m$-identifies $\mathcal{C}$. $\square$

**Claim 2** $(\forall d \in (0, 1])[\mathcal{C} \in \mathbf{UAp}^d\mathbf{Ex}]$.

**Proof of Claim 2**: Consider machine $\mathbf{M}$ which, on additional information program $s$, outputs a program $P(s)$ described as follows:

**begin** $\{\varphi_{P(s)}(x)\}$
  **if** $x \in \bigcup_{k\in\mathbb{N}}[N_{2k+1}, N_{2k+2})$
    **then**
      output 0
    **else**
      let $j$ be such that $x \in S_j$;
      search for $y$ such that $y \in S_j \bigwedge \varphi_s(y)\downarrow$;
      when such a $y$ is found output $\varphi_s(y)$
  **endif**
**end** $\{\varphi_{P(s)}(x)\}$

It is easy to see that if a program $s$ for $\varphi_s$ is additional information of type $\mathbf{UAp}^d, d > 0$, for $f \in \mathcal{C}$, then, for all $j$, there exists a $y$ such that $y \in S_j$ and $\varphi_s(y)\downarrow$. Thus, $\varphi_{P(s)} = f$. □

■ Theorem 2.

As a contrast to Theorem 2 above, Theorem 3 below says that there are classes of recursive functions that can be **Ex**-identified with **Ap** type additional information but cannot be **Bc**-identified with any predetermined number of anomalies and **UAp** type additional information if the density associated with **Ap** type additional information is better than the one associated with **UAp** type additional information.

**Theorem 3** $(\forall d_2 > d_1 \mid d_1, d_2 \in [0,1])(\forall l \in \mathbb{N})\ [\mathbf{Ap}^{d_2}\mathbf{Ex} - \mathbf{UAp}^{d_1}\mathbf{Bc}^l \neq \emptyset]$.

**Proof of Theorem 3**: Without loss of generality, let $d_2 = (m+3)/n$ and $d_1 = m/n$, where $m + 3 \leq n$ and $m, n \in \mathbb{N}$. Let $N_0 = -1$ and $N_i = n^i$. Let $S_j$ denote the set $\bigcup_{k\in\mathbb{N}}(N_{\langle j,k\rangle}, N_{\langle j,k\rangle+1}]$. Let $S'_j = S_j \bigcap \{x \mid x \geq m \ \mathbf{mod}\ n\}$. Consider the class $\mathcal{C}$ of recursive functions defined below.

$\mathcal{C} = \{f \in \mathcal{R} \mid$ the following two conditions hold:

1. $(\forall x)[[x < m \textbf{ mod } n] \Rightarrow [f(x) = 0]]$
2. $(\forall j, x, y)[[x \in S'_j \wedge y \in S'_j] \Rightarrow [f(x) = f(y)]]$

$\}$

**Claim 3** $(\forall k \in \mathbb{N})[\mathcal{C} \notin \mathbf{UAp}^{d_1}\mathbf{Bc}^k]$

**Proof of Claim 3**: Let $\eta$ be such that $\eta(x) = 0$ if $x < m \textbf{ mod } n$ and $\eta(x)$ is undefined otherwise. Clearly, any program for $\eta$ is an $\mathbf{UAp}^{d_1}$ additional information for all $f \in \mathcal{C}$. Now proceeding in the same way as in Claim 1 in Theorem 2, we have that $\mathcal{C} \notin \mathbf{UAp}^{d_1}\mathbf{Bc}^k$. □

**Claim 4** $\mathcal{C} \in \mathbf{Ap}^{d_2}\mathbf{Ex}$.

**Proof of Claim 4**: Consider machine **M**, which on additional information program $s$ outputs a program $P(s)$, defined as follows:

**begin** $\{\varphi_{P(s)}(x)\}$
  **if** $(x < m \textbf{ mod } n)$
    **then**
      output 0
    **else**
      let $j$ be such that $x \in S'_j$;
      search for $y \in S'_j$ such that $\varphi_s(y)\downarrow$;
      when such a $y$ is found output $\varphi_s(y)$
  **endif**
**end** $\{\varphi_{P(s)}(x)\}$

Let program $s$ for $\varphi_s$ be additional information of type $\mathbf{Ap}^{d_2}$ for $f \in \mathcal{C}$. Now for large enough $i$, $\text{card}(\{x \mid x \leq N_i \wedge \varphi_s(x) = f(x)\})/(N_i + 1) \geq (m + 2)/n$. Since $N_{i-1}/N_i = 1/n$, there exists a $y, N_{i-1} < y \leq N_i$ and $y \geq m \textbf{ mod } n$, such that $\varphi_s(y)\downarrow$. Thus, $\varphi_{P(s)} = f$. □

■ Theorem 3.

**Theorem 4** $(\forall i \in \mathbb{N})$

*1)* $\mathbf{Ex}^{i+1} - \mathbf{UAp}^1\mathbf{Ex}^i \neq \emptyset$.

*2)* $\mathbf{Bc}^{i+1} - \mathbf{UAp}^1\mathbf{Bc}^i \neq \emptyset$.

*3)* $\mathbf{Ex}^* - \bigcup_i \mathbf{UAp}^1\mathbf{Ex}^i \neq \emptyset$.

*4)* $\mathbf{Bc} - \mathbf{UAp}^1\mathbf{Ex}^* \neq \emptyset$.

**Proof of Theorem 4**: For all $f \in \mathcal{R}$, let $f'$ be defined as follows:

$$f'(x) = \begin{cases} f(y) & \text{if } (\exists y)[2^y = x]; \\ 0 & \text{otherwise.} \end{cases}$$

For any class of functions $\mathcal{C}$, let $\mathcal{C}' = \{f' \mid f \in \mathcal{C}\}$. It is easy to see that for all $a \in \mathbb{N} \bigcup \{*\}$, $\mathcal{C} \in \mathbf{Ex}^a \Leftrightarrow \mathcal{C}' \in \mathbf{Ex}^a \Leftrightarrow \mathcal{C}' \in \mathbf{UAp}^1\mathbf{Ex}^a$ and $\mathcal{C} \in \mathbf{Bc}^a \Leftrightarrow \mathcal{C}' \in \mathbf{Bc}^a \Leftrightarrow \mathcal{C}' \in \mathbf{UAp}^1\mathbf{Bc}^a$.

Theorem follows from the results in [CS83] (see theorem 1).

∎ Theorem 4.

The above theorems give the complete relationship between different **Ex** and **Bc** identification criteria formed with both **Ap** and **UAp** type additional information. We observe some of these relationships in Corollary 1 below which follows from results presented in this section and Theorem 1.

**Corollary 1** *Let* $d_1, d_2 \in [0, 1]$. *Let* $a, b \in \mathbb{N} \cup \{*\}$.

*a)* $\mathbf{Ap}^{d_1}\mathbf{Ex}^a \subseteq \mathbf{Ap}^{d_2}\mathbf{Ex}^b \Leftrightarrow [d_1 \leq d_2 \text{ and } a \leq b]$.

*b)* $\mathbf{Ap}^{d_1}\mathbf{Bc}^a \subseteq \mathbf{Ap}^{d_2}\mathbf{Bc}^b \Leftrightarrow [[b = *] \text{ or } [d_1 \leq d_2 \text{ and } a \leq b]]$.

*c)* $\mathbf{UAp}^{d_1}\mathbf{Ex}^a \subseteq \mathbf{UAp}^{d_2}\mathbf{Ex}^b \Leftrightarrow [d_1 \leq d_2 \text{ and } a \leq b]$.

*d)* $\mathbf{UAp}^{d_1}\mathbf{Bc}^a \subseteq \mathbf{UAp}^{d_2}\mathbf{Bc}^b \Leftrightarrow [[b = *] \text{ or } [d_1 \leq d_2 \text{ and } a \leq b]]$.

*e)* $(\forall d \in (0, 1])[\mathbf{Ap}^d\mathbf{Ex}^a \subset \mathbf{UAp}^d\mathbf{Ex}^a]$.

*f)* $(\forall d \in (0, 1])(\forall i \in \mathbb{N})[\mathbf{Ap}^d\mathbf{Bc}^i \subset \mathbf{UAp}^d\mathbf{Bc}^i]$.

## 3.3 Additional Information for Approximate Function Inference

Royer [Roy86] provides criticism of $\mathbf{Ex}^n$ and $\mathbf{Ex}^*$ criteria as models of science. They are too strict to reflect how anomalies occur in actual scientific theories. Case [Cas86] criticizes $\mathbf{Ex}^*$ criterion as being too impractical because under this criterion one can converge to an explanation for a phenomenon which is almost everywhere correct, but which is still incorrect on predicting all the experiments which one would care about. To address these issues, Royer [Roy86] considered the inductive inference criteria which permit infinitely many errors in explanations, but which require that the "density" of these errors be no more than a certain prespecified amount. Smith and Velauthapillai [SV86] also investigated similar criteria of inference. We investigate the effect of a partial explanation on such criteria. The following definitions are from [Roy86]. Also, see [SV86] for similar notions.

**Definition 11**
(a) [Roy86] The *asymptotic agreement* between two partial functions $\eta$ and $\theta$ (denoted: $\mathbf{aa}(\eta, \theta)$) is $\mathbf{d}(\{x \mid \eta(x) = \theta(x)\})$.
(b) [RU63, Roy86] The *Asymptotic disagreement* between two partial functions $\eta$ and $\theta$ (denoted: $\mathbf{ad}(\eta, \theta)$) is $1 - \mathbf{aa}(\eta, \theta)$.

**Definition 12** [Roy86] Let $d \in [0, 1]$.
(a) A machine $\mathbf{M}$ $\mathbf{Aex}^d$-*identifies* a recursive function $f$ (written: $f \in \mathbf{Aex}^d(\mathbf{M})$) iff $\mathbf{M}(f)\downarrow = i$ and $\mathbf{ad}(f, \varphi_i) \leq d$.
(b) $\mathbf{Aex}^d = \{\mathcal{C} \subseteq \mathcal{R} \mid (\exists \mathbf{M})[\mathcal{C} \subseteq \mathbf{Aex}^d(\mathbf{M})]\}$.

**Definition 13** [Roy86]
(a) The *asymptotic uniform agreement* between two partial functions $\eta$ and $\theta$ (denoted: $\mathbf{aua}(\eta, \theta)$) is $\mathbf{ud}(\{x \mid \eta(x) = \theta(x)\})$.
(b) The *Asymptotic uniform disagreement* between two partial functions $\eta$ and $\theta$ (denoted: $\mathbf{aud}(\eta, \theta)$) is $1 - \mathbf{aua}(\eta, \theta)$.

**Definition 14** [Roy86] Let $d \in [0,1]$.
(a) A machine $\mathbf{M}$ $\mathbf{UAex}^d$-*identifies* a recursive function $f$ (written: $f \in \mathbf{UAex}^d(\mathbf{M})$) iff $\mathbf{M}(f)\downarrow = i$ and $\mathbf{aud}(f, \varphi_i) \leq d$.
(b) $\mathbf{UAex}^d = \{\mathcal{C} \subseteq \mathcal{R} \mid (\exists \mathbf{M})[\mathcal{C} \subseteq \mathbf{UAex}^d(\mathbf{M})]\}$.

Above criteria can be extended to identification with additional information to give $\mathbf{Ap}^{d_1}\mathbf{Aex}^{d_2}$, $\mathbf{Ap}^{d_1}\mathbf{UAex}^{d_2}$, $\mathbf{UAp}^{d_1}\mathbf{Aex}^{d_2}$ and $\mathbf{UAp}^{d_1}\mathbf{UAex}^{d_2}$ criteria of identification.

Royer showed the following result about **Aex**-identification.

**Theorem 5** [Roy86] $(\forall d \in [0,1)\ )[\mathcal{R} \notin \mathbf{Aex}^d]$.

**Proposition 3** $(\forall d \in [0,1])[\mathcal{R} \in \mathbf{Ap}^d\mathbf{Aex}^{1-d}]$.

**Proof of Proposition 3:** A machine which just outputs the additional information program given to it $\mathbf{Ap}^d\mathbf{Aex}^{1-d}$-identifies $\mathcal{R}$. □

**Proposition 4** $(\forall d \in [0,1])[\mathcal{R} \in \mathbf{UAp}^d\mathbf{UAex}^{1-d}]$.

**Proof of Proposition 4:** A machine which just outputs the additional information program given to it $\mathbf{UAp}^d\mathbf{UAex}^{1-d}$-identifies $\mathcal{R}$. □

The following theorems give the relationship between different criteria of approximate identification with additional information.

**Theorem 6** $(\forall d_1 > 0)(\forall d_2, d_3 \mid d_2 + d_3 < 1)[\mathbf{UAp}^{d_1}\mathbf{Ex} - \mathbf{Ap}^{d_2}\mathbf{Aex}^{d_3} \neq \emptyset]$.

**Proof of Theorem 6**: Without loss of generality, assume that $d_1 = 2/n$, $d_2 = l/n$, $d_3 = (n-l-1)/n$, $n > 1$, where $l, n \in \mathbb{N}$. Let $N_0 = 0$, $N_{2i+1} = n^{i+1} + N_{2i}$, and $N_{2i+2} = N_{2i+1} + (i+1)*n$. Let $S_j = \bigcup_{k \in \mathbb{N}}[N_{2*\langle j,k\rangle+1}, N_{2*\langle j,k\rangle+2})$. Consider the following class of functions:

$\mathcal{C} = \{f \in \mathcal{R} \mid$ following two conditions hold:

1. $f(\{\bigcup_{i \in \mathbb{N}}[N_{2i}, N_{2i+1})\} \bigcap \{x \mid x < l \ \mathbf{mod}\ n\}) = 0$
2. $(\forall j)[[x \in S_j] \Rightarrow [f(x) = f(j)]]$

$\}$

To $\mathbf{UAp}^{d_1}\mathbf{Ex}$-identify $f$, $\mathbf{M}$, on additional information program $s$ for $\varphi_s$, outputs a program $P(s)$ described as follows:

**begin** $\{\varphi_{P(s)}(x)\}$
search for $y \in S_x$ such that $\varphi_s(y)\downarrow$;
when such a $y$ is found output $\varphi_s(y)$
**end** $\{\varphi_{P(s)}(x)\}$

It is easy to see that if, $f \in \mathcal{C}$, $\varphi_s$ is additional information of type $\mathbf{UAp}^{d_1}$, where $d_1 = 2/n$, then for all $x$, there exists a $y \in S_x$ such that $\varphi_s(y)\downarrow$. Thus, $\varphi_{P(s)} = f$.

Let $\eta$ be defined as follows:

$$\eta(x) = \begin{cases} 0 & [x < l \ \mathbf{mod}\ n] \bigwedge [x \in \bigcup_{i \in \mathbb{N}} [N_{2i}, N_{2i+1})]; \\ \uparrow & \text{otherwise.} \end{cases}$$

It is easy to see that $(\forall f \in \mathcal{C})[\mathbf{d}(\{x \mid \eta(x) = f(x)\}) = l/n]$. Suppose by way of contradiction that a machine $\mathbf{M}$ $\mathbf{Ap}^{d_2}\mathbf{Aex}^{d_3}$-identifies $\mathcal{C}$. It is, then, easy to convert $\mathbf{M}$ to a machine $\mathbf{M}'$ such that $\mathbf{M}'$ $\mathbf{Aex}^{(n-l-1)/(n-l)}$-identifies any $f \in \mathcal{R}$. Since this is not possible, no such machine $\mathbf{M}$ can exist.

∎ Theorem 6.

Similar proofs can be worked out for the following Theorems 7 and 8.

**Theorem 7** $(\forall d_1, d_2, d_3 \mid d_1 > d_3 \geq 0 \bigwedge d_2 + d_3 < 1)[\mathbf{UAex}^{d_1} - \mathbf{UAp}^{d_2}\mathbf{Aex}^{d_3} \neq \emptyset]$.

**Corollary 2** $(\forall d_1, d_2, d_3 \mid d_1 > d_3 \geq 0 \bigwedge d_2 + d_3 < 1)[\mathbf{UAex}^{d_1} - \mathbf{UAp}^{d_2}\mathbf{UAex}^{d_3} \neq \emptyset]$.

**Theorem 8** $(\forall d_1, d_2 \mid d_1 + d_2 < 1)[\mathbf{Aex}^0 - \mathbf{UAp}^{d_1}\mathbf{UAex}^{d_2} \neq \emptyset]$.

**Corollary 3** $(\forall d < 1)[\mathbf{Aex}^0 - \mathbf{Ap}^1\mathbf{UAex}^d \neq \emptyset]$.

**Theorem 9** $(\forall d_1 > 0)(\forall d_2 < 1)[\mathbf{UAp}^{d_1}\mathbf{Ex} - \mathbf{Ap}^1\mathbf{UAex}^{d_2} \neq \emptyset]$.

**Proof of Theorem 9**: Without loss of generality let $d_1 = 2/n$ and $d_2 = (n-2)/n$, where $n \in \mathbb{N}$. Let $N_0 = 0$, $N_{2i+1} = n^{i+1} + N_{2i}$, and $N_{2i+2} = N_{2i+1} + (i+1) * n$. Let $S_j = \bigcup_{k \in \mathbb{N}^+}[N_{2*\langle j,k\rangle+1}, N_{2*\langle j,k\rangle+2})$. Note that here $k$ ranges over $\mathbb{N}^+$ and not over $\mathbb{N}$. Consider the following class of functions:

Let $\mathcal{C} = \{f \in \mathcal{R} \mid$ following two conditions hold:

1. $f(\bigcup_{i \in \mathbb{N}}[N_{2i}, N_{2i+1})) = 0$
2. $(\forall j, x)[[x \in S_j] \Rightarrow [f(x) = f(j)]]$

$\}$

It is easy to see that $\mathcal{C} \in \mathbf{UAp}^{d_1}\mathbf{Ex}$. Define $\eta$ as follows:

$$\eta(x) = \begin{cases} 0 & x \in \bigcup_{i \in \mathbb{N}}[N_{2i}, N_{2i+1}); \\ \uparrow & \text{otherwise.} \end{cases}$$

Clearly, any program for $\eta$ is a valid additional information for any $f \in \mathcal{C}$. Suppose by way of contradiction that a machine $\mathbf{M}$ $\mathbf{Ap}^1\mathbf{UAex}^{d_2}$-identifies $\mathcal{C}$. It is, then, easy to convert $\mathbf{M}$ to $\mathbf{M}'$ such that $\mathbf{M}'$ $\mathbf{UAex}^{(n-2)/n}$-identifies $\mathcal{R}$. But by theorem 5, no such machine $\mathbf{M}'$ can exist. Thus, no such machine $\mathbf{M}$ exists.

∎ Theorem 9.

**Theorem 10** $(\forall d_1, d_2, d_3 \mid d_2 < d_1 \bigwedge d_3 < 1)[\mathbf{Ap}^{d_1}\mathbf{Ex} - \mathbf{Ap}^{d_2}\mathbf{UAex}^{d_3} \neq \emptyset]$.

**Proof of Theorem 10**: Without loss of generality, assume that $d_2 = l/n$, $d_1 = (l+2)/n$, and $d_3 = (n-2)/n$, where $l, n \in \mathbb{N}, n > 3$. Let $N_0 = 0$, $N_{2i+1} = n^{i+1} + N_{2i}$, and $N_{2i+2} = N_{2i+1} + (i+1) * n$. Let $S_j = \bigcup_{k \in \mathbb{N}}[N_{2*\langle j,k\rangle}, N_{2*\langle j,k\rangle+1})$. Let $S'_j = S_j \cap \{x \mid x \geq l \ \mathbf{mod}\ n\}$. Consider the following class of functions:

Let $\mathcal{C} = \{f \in \mathcal{R} \mid$ following two conditions hold:

1. $f(\{\bigcup_{i\in\mathbb{N}}[N_{2i}, N_{2i+1})\} \bigcap \{x \mid x < l \ \mathbf{mod}\ n\}) = 0$
2. $(\forall j, x)[[x \in S'_j] \Rightarrow [f(x) = f(j)]]$

$\}$

It is easy to see that $\mathcal{C} \in \mathbf{Ap}^{d_1}\mathbf{Ex}$. Define $\eta$ as follows:

$$\eta(x) = \begin{cases} 0 & [x < l \ \mathbf{mod}\ n] \bigwedge [x \in \bigcup_{i\in\mathbb{N}}[N_{2i}, N_{2i+1})]; \\ \uparrow & \text{otherwise.} \end{cases}$$

Since any program for $\eta$ is a valid additional information for any $f \in \mathcal{C}$, a machine $\mathbf{M}$ which $\mathbf{Ap}^{d_2}\mathbf{UAex}^{d_3}$-identifies $\mathcal{C}$ can be converted to a machine $\mathbf{M}'$ which $\mathbf{UAex}^{(n-2)/n}$-identifies $\mathcal{R}$. But by theorem 5, no such machine $\mathbf{M}'$ can exist.

∎ Theorem 10.

Theorems 11 and 12 below can be proved similarly.

**Theorem 11** $(\forall d_1, d_2 \mid d_1 > d_2)[\mathbf{UAex}^{d_1} - \mathbf{Ap}^1\mathbf{UAex}^{d_2} \neq \emptyset]$.

**Theorem 12** $(\forall d_1, d_2, d_3 \mid d_2 > d_1 \bigwedge d_1 + d_3 < 1)[\mathbf{Ap}^{d_2}\mathbf{Ex} - \mathbf{UAp}^{d_1}\mathbf{Aex}^{d_3} \neq \emptyset]$.

Results presented in this section give the complete relationship between different **Ex**, **Aex**, and **UAex** identification criteria formed with both **Ap** and **UAp** type additional information.

# 4 Language Learning

## 4.1 Fundamental Language Learning Paradigms

**Definition 15** [Gol67] A *text* for a language $L$ is a mapping $t$ from $\mathbb{N}$ into $(\mathbb{N} \cup \{\#\})$ such that $L$ is the set of natural numbers in the range of $t$.

Intuitively, a text for a language is an enumeration of the objects in the language with #'s representing pauses in the listing of such objects. For a finite initial segment $\sigma$, content($\sigma$) = range($\sigma$) $-$ {#} and $|\sigma|$ denotes the length of the finite initial segment $\sigma$, i.e., the number of elements in $\sigma$. $t, t'$ range over texts for languages. $\overline{t_n}$ denotes the initial segment of $t$ with length $n$. $\sigma \subset t$ means $\sigma$ is an initial segment of $t$. Similarly $\sigma \subseteq \sigma'$ means $\sigma$ is an initial segment of $\sigma'$. content($t$) = range($t$) $-$ {#}; intuitively, content($t$) is the set of meaningful things presented in text $t$. $\sigma_1 \diamond \sigma_2$ denotes the concatenation of $\sigma_1$ and $\sigma_2$, i.e.,

$$\sigma_1 \diamond \sigma_2(x) = \begin{cases} \sigma_1(x) & \text{if } x < |\sigma_1|; \\ \sigma_2(x - |\sigma_1|) & \text{if } x \geq |\sigma_1|. \end{cases}$$

$\mathbf{M}(t)\downarrow = i$ iff $(\forall^\infty n)[\mathbf{M}(\overline{t_n}) = i]$. We write $\mathbf{M}(t)\downarrow$ iff $(\exists i)[\mathbf{M}(t)\downarrow = i]$. If $L$ is a recursively enumerable language, then $i$ is a *grammar* for $L$ iff $W_i = L$. $\sigma$ is *in* $L$ iff content($\sigma$) $\subseteq L$.

**Definition 16** [Gol67, CL82, OW82a, OW82b]
(a) $\mathbf{M}$ $\mathbf{TxtEx}^a$-*identifies* an r.e. language $L$ (written: $L \in \mathbf{TxtEx}^a(\mathbf{M})$) iff for any text $t$ for $L$, $\mathbf{M}(t)\downarrow$ and $W_{\mathbf{M}(t)} =^a L$.
(b) $\mathbf{TxtEx}^a = \{\mathcal{L} \subseteq \mathcal{E} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{TxtEx}^a(\mathbf{M})]\}$.

**Definition 17** [Ful85, Ful90b] $\sigma$ is a $\mathbf{TxtEx}$-*stabilizing segment for* $\mathbf{M}$ *on* $L$ iff content($\sigma$) $\subseteq L$ and $(\forall \sigma' \mid \text{content}(\sigma') \subseteq L \ \wedge \ \sigma \subseteq \sigma')[\mathbf{M}(\sigma') = \mathbf{M}(\sigma)]$.

**Definition 18** [BB75, OW82a] $\sigma$ is a $\mathbf{TxtEx}^a$-*locking sequence for* $\mathbf{M}$ *on* $L$ iff $\sigma$ is a $\mathbf{TxtEx}$-stabilizing segment for $\mathbf{M}$ on $L$ and $W_{\mathbf{M}(\sigma)} =^a L$.

We often refer to $\mathbf{TxtEx}^a$-locking sequence by just locking sequence ($a$ will be clear from context). We now present a very important lemma in learning theory due to L. Blum and M. Blum [BB75]. We will have opportunity to use this lemma on many occasions.

**Lemma 1** [BB75, OW82a] *If* $\mathbf{M}$ $\mathbf{TxtEx}^a$*-identifies* $L$*, then there is a* $\mathbf{TxtEx}^a$*-locking sequence for* $\mathbf{M}$ *on* $L$*.*

Analogously to **Bc**-identification criteria in the context of function inference, we define a more general language learning criteria than **TxtEx**-identification.

**Definition 19** [CL82]
(a) $\mathbf{M}$ $\mathbf{TxtBc}^a$*-identifies* an r.e. language $L$ (written: $L \in \mathbf{TxtBc}^a(\mathbf{M})$) iff $\mathbf{M}$, fed any text $t$ for $L$, outputs over time an infinite sequence of grammars $p_0, p_1, p_2, \ldots$ such that $(\forall^\infty n)[W_{p_n} =^a L]$.
(b) $\mathbf{TxtBc}^a = \{\mathcal{L} \subseteq \mathcal{E} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{TxtBc}^a(\mathbf{M})]\}$.

We usually write **TxtEx** for $\mathbf{TxtEx}^0$ and **TxtBc** for $\mathbf{TxtBc}^0$.

Case [Cas88] considered the question whether humans converge to more than one distinct, but equivalent, correct grammars. He captured this notion through a new criterion of language learning, viz., **TxtFex**-identification — a more general criterion than Gold's **TxtEx**-identification. We also study the effect of additional information on this criterion.

**Definition 20** [Cas88] Suppose $\mathbf{M}$ is a learning machine and $t$ is a text. Then $\mathbf{M}(t)$ finitely-converges (written $\mathbf{M}(t)\Downarrow$) $\Leftrightarrow \{\mathbf{M}(\sigma) \mid \sigma \subset t\}$ is finite. If $\mathbf{M}(t)\Downarrow$ then $\mathbf{M}(t)$ is defined $= \{p \mid (\exists^\infty \sigma \subset t)[\mathbf{M}(\sigma) = p]\}$; otherwise, $\mathbf{M}(t)$ is undefined.

**Definition 21** [Cas88]
(a) For $b \in \mathbb{N}^+ \cup \{*\}$, a language learning machine, $\mathbf{M}$, $\mathbf{TxtFex}^a_b$*-identifies* an r.e. language $L$ (written: $L \in \mathbf{TxtFex}^a_b(\mathbf{M})$) $\Leftrightarrow$ ($\forall$ texts $t$ for $L$)[$\mathbf{M}(t)\Downarrow =$ a set of cardinality $\leq b$ and $(\forall p \in \mathbf{M}(t))[W_p =^a L]$].
(b) $\mathbf{TxtFex}^a_b = \{\mathcal{L} \subseteq \mathcal{E} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{TxtFex}^a_b(\mathbf{M})]\}$.

In $\mathbf{TxtFex}_b^a$-*identification*, the $b$ is a "bound" on the number of final grammars and the $a$ is a bound on the number of anomalies allowed in these final grammars. A bound of $*$ on the number of anomalies (or the number of final grammars) means that the number of anomalies (or the number of final grammars) is finite, however the bound is not prespecified.

The following definitions are analogue of Definitions 17 and 18 for **TxtFex** and **TxtBc** identification criteria.

**Definition 22** (Based on [BB75, Cas88]) Let $a, b \in \mathbb{N} \cup \{*\}$.
(a) $\sigma$ is a $\mathbf{TxtFex}_b$-*stabilizing segment for* $\mathbf{M}$ *on* $L$ iff [content$(\sigma) \subseteq L$] and there exists a set $S$ of cardinality at most $b$ such that

$(\exists \sigma' \subseteq \sigma)[S = \{\mathbf{M}(\sigma'') \mid \sigma' \subseteq \sigma'' \subseteq \sigma\}]$ and
$S = \{\mathbf{M}(\sigma''') \mid \sigma \subseteq \sigma''' \wedge \text{content}(\sigma''') \subseteq L\}$.

(b) $\sigma$ is a $\mathbf{TxtFex}_b^a$-*locking sequence for* $\mathbf{M}$ *on* $L$ iff $\sigma$ is a $\mathbf{TxtFex}_b$-stabilizing segment for $\mathbf{M}$ on $L$ and $(\forall \sigma' \mid \sigma \subseteq \sigma' \wedge \text{content}(\sigma') \subseteq L)[W_{\mathbf{M}(\sigma')} =^a L]$.

**Definition 23** (Based on [BB75, CL82]) Let $a \in \mathbb{N} \cup \{*\}$. $\sigma$ is a $\mathbf{TxtBc}^a$ *locking sequence for* $\mathbf{M}$ *on* $L$ iff content$(\sigma) \subseteq L$ and $(\forall \sigma' \mid [\sigma \subseteq \sigma'] \wedge [\text{content}(\sigma') \subseteq L])[W_{\mathbf{M}(\sigma')} =^a L]$.

There is an analogue of Lemma 1 for **TxtBc** [CL82] and **TxtFex** [Cas88] learning also.

**Lemma 2** (Based on [BB75, CL82, Cas88]) *If* $\mathbf{M}$ $\mathbf{TxtFex}_b^a$-*identifies* $L$, *then there is a* $\mathbf{TxtFex}_b^a$-*locking sequence for* $\mathbf{M}$ *on* $L$. *If* $\mathbf{M}$ $\mathbf{TxtBc}^a$-*identifies* $L$, *then there is a* $\mathbf{TxtBc}^a$-*locking sequence for* $\mathbf{M}$ *on* $L$.

Theorem 13 just below states some of the basic results in language learning.

**Theorem 13** *For all* $i, n \in \mathbb{N}$,
*(a)* $\mathbf{TxtEx}^{n+1} - \mathbf{TxtFex}_*^n \neq \emptyset$;

*(b)* $\mathbf{TxtEx}^{2n+1} - \mathbf{TxtBc}^n \neq \emptyset$*;*
*(c)* $\mathbf{TxtEx}^{2n} \subset \mathbf{TxtBc}^n$*;*
*(d)* $\mathbf{TxtFex}^0_{i+1} - \mathbf{TxtFex}^*_i \neq \emptyset$*;*
*(e)* $\bigcup_n \mathbf{TxtFex}^n_i \subset \mathbf{TxtFex}^*_i$*; and*
*(f)* $\bigcup_n \mathbf{TxtBc}^n \subset \mathbf{TxtBc}^*$.

Parts (a), (d) and (e) are due to Case [Cas88]. Parts (b) and (c) are due to Case and Lynes [CL82]. Part (f) follows from part (e) in Theorem 1. Osherson and Weinstein independently established that $\mathbf{TxtEx} \subset \mathbf{TxtFex}_*$ [OW82b].

## 4.2 Additional Information for Language learning

Formal language learning theory was originally motivated by the study of language learning in children. It relied on early claims of psycholinguists that children are rarely, if ever, informed of grammatical errors; instead, children are only exposed to strings in the language. Based on this, Gold [Gol67] developed the notion of **TxtEx**-identification. However, it turns out that the class **TxtEx**, which contains sets of *r.e.* languages that can be **TxtEx**-identified by some language learning machine, contains "small" classes of languages. For instance, none of the classes of languages in the Chomsky hierarchy (regular, context free, context sensitive, and r.e.) are contained in **TxtEx**. This led Gold to two possible conclusions. One was that the class of natural languages is much "smaller" than previously thought, and the other was that children are being given additional information in some subtle way. Angluin [Ang80a, Ang80b] and Wiehagen [Wie77, KW80] address the first conclusion of Gold. We will concern ourselves, in this section, with the second conclusion of Gold.

It is not uncommon for an elder person (a parent or teacher) to tell a child some small grammatical rule that enables the child to enumerate a

list of elements of the language. Basically, this additional information (the grammatical rule) enables the child to know certain elements of the language before these elements appear in the child's text. This kind of additional information can be modeled in Gold's paradigm by requiring that, in addition to a text for the language, the language learning device be provided with a grammar for an infinite subset of the language. It turns out that such an additional information indeed increases the language learning power of learning machines. We further model the quality of this additional information by measuring the "density of agreement" between the language being learned and the subset language whose grammar is provided as additional information. Not surprisingly, a "better quality" additional information enhances the learning power of language learning machines more than a "not so good" additional information. We now define this "density" notion and the new language learning criteria.

**Definition 24** Let $L_1$ and $L_2$ be any two languages. Let $x_1 < x_2 < x_3, \cdots$ be the elements of $L_2$.

The *relative density* of $L_1$ in $L_2$ (denoted by $\mathbf{rd}(L_1; L_2)$) is defined as follows:

$$\mathbf{rd}(L_1; L_2) = \begin{cases} \mathbf{d}(\{i \mid x_i \in L_1\}) & \text{If } L_2 \text{ is infinite;} \\ \mathbf{d}(L_1; L_2) & \text{otherwise.} \end{cases}$$

Similarly, *uniform relative density* of $L_1$ in $L_2$ (denoted: $\mathbf{urd}(L_1; L_2)$) is defined as follows:

$$\mathbf{urd}(L_1; L_2) = \begin{cases} \mathbf{ud}(\{i \mid x_i \in L_1\}) & \text{If } L_2 \text{ is infinite;} \\ \mathbf{ud}(L_1; L_2) & \text{otherwise.} \end{cases}$$

**Definition 25** Suppose $d$ is a real number in the interval $[0, 1]$.
(a) A language $L'$ is said to be *d-language conforming* with another language $L$ iff $L'$ satisfies the following two conditions:

(1) $L' \subseteq L$; and

(2) $\mathbf{rd}(L'; L) \geq d$.

(b) A language $L'$ is said to be *d-language uniform conforming* with another language $L$ iff $L'$ satisfies the following two conditions:

(1) $L' \subseteq L$; and

(2) $\mathbf{urd}(L'; L) \geq d$.

**Definition 26** Let $d \in [0, 1]$ and $a \in (\mathbb{N} \cup \{*\})$.

(a) A machine $\mathbf{M}$ $\mathbf{Ap}^d\mathbf{TxtEx}^a$-*identifies* an r.e. language $L$ (written: $L \in \mathbf{Ap}^d\mathbf{TxtEx}^a(\mathbf{M})$) iff $\mathbf{M}$, fed any text for $L$ and any grammar $p$ such that $W_p$ is $d$-language conforming with $L$, converges in the limit to a grammar $i$ such that $W_i =^a L$.

(b) $\mathbf{Ap}^d\mathbf{TxtEx}^a = \{\mathcal{L} \subseteq \mathcal{E} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{Ap}^d\mathbf{TxtEx}^a(\mathbf{M})]\}$.

We can similarly define $\mathbf{UAp}^d\mathbf{TxtEx}^a$, $\mathbf{Ap}^d\mathbf{TxtFex}^a_b$, $\mathbf{UAp}^d\mathbf{TxtFex}^a_b$, $\mathbf{Ap}^d\mathbf{TxtBc}^a$, and $\mathbf{UAp}^d\mathbf{TxtBc}^a$ criteria of language learning. Clearly, these criteria are analogs of the similar criteria for function inference. It should be noted that all the diagonalization theorems in function inference carry over to language learning case.

Above, we were concerned with additional information that *supplements* the information a child is already receiving in the form of a *text* for the language. In other words, the additional information that we just modeled, is about what is in the language and not about what is not in the language. However, literature of speech language pathology and linguistics contains extensive refutations of the claim that children receive no negative data [BB64, Dal76]. Intuitively, it is clear that children are receiving information about the complement of the language they are trying to learn. If a child's utterances do not have the desired effect, it somehow works as a clue that the utterance is not in the language. An elder person (a parent or a teacher) either rebukes the child or tells it specifically that something is not in the

language. Better still, an elder person can provide the child with a rule that enumerates a list of strings which are not members of the language. This kind of additional information can be modeled in Gold's paradigm by requiring that the language learning device be provided with a grammar for a subset of the complement of the language being learned. It turns out that even this kind of additional information enhances the learning power of language learning devices.

Fulk [Ful85] investigated a different approach to additional information about the complement of a language. He showed that being given *text* for a language $L$, and a grammar for the complement of $L$ is equivalent to being given a *text* for $L$ and an enumeration of a non-empty, finite sequence of grammars, the last of which is a grammar for the complement of $L$. However, we feel, a grammar for the complement of the language is too much additional information, and children certainly are not being given a rule that lists everything that is ungrammatical. We further employ the above density notions to differentiate a "good quality" additional information about the complement from a "not so good quality" additional information. As in the previous case, better the additional information, more is the enhancement achieved in learning power of language learning devices. We now define this notion. In the following definitions **ACp** stands for **A**pproximate **C**omplement **p**artial additional information.

**Definition 27** Let $d \in [0,1]$. Let $a \in (\mathbb{N} \cup \{*\})$.
(a) A machine $\mathbf{M}$ $\mathbf{ACp}^d\mathbf{TxtEx}^a$-*identifies* an r.e. language $L$ (written: $L \in \mathbf{ACp}^d\mathbf{TxtEx}^a(\mathbf{M})$) iff $\mathbf{M}$, fed any text for $L$ and any grammar $p$ such that $W_p$ is $d$-language conforming with the complement of $L$ (i.e. $\mathbb{N}-L$), converges in the limit to a grammar $i$ such that $W_i =^a L$.
(b) $\mathbf{ACp}^d\mathbf{TxtEx}^a = \{\mathcal{L} \subseteq \mathcal{E} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{ACp}^d\mathbf{TxtEx}^a(\mathbf{M})]\}$.

We can similarly define $\mathbf{UACp}^d\mathbf{TxtEx}^a$, $\mathbf{ACp}^d\mathbf{TxtFex}^a_b$, $\mathbf{UACp}^d\mathbf{TxtFex}^a_b$, $\mathbf{ACp}^d\mathbf{TxtBc}^a$, and $\mathbf{UACp}^d\mathbf{TxtBc}^a$ criteria of language learning.

Finally, we define a language learning criteria that incorporates additional information both about elements of the language (positive information) and about elements of the complement of the language (negative information). It turns out that this kind of additional information is better than just providing positive additional information or just providing negative additional information.

**Definition 28** Let $d_1, d_2 \in [0,1], a \in (\mathbb{N} \cup \{*\})$.
(a) A machine $\mathbf{M}$ $\mathbf{Ap}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtEx}^a$-*identifies* an r.e. language $L$ (written: $L \in \mathbf{Ap}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtEx}^a(\mathbf{M})$) iff $\mathbf{M}$, fed any text for $L$ and grammars $p_1$ and $p_2$ such that $W_{p_1}$ is $d_1$-language conforming with $L$ and $W_{p_2}$ is $d_2$-language conforming with the complement of $L$ ( i.e. $\mathbb{N} - L$), converges in the limit to a grammar $i$ such that $W_i =^a L$.
(b) $\mathbf{Ap}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtEx}^a = \{\mathcal{L} \subseteq \mathcal{E} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{Ap}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtEx}^a(\mathbf{M})]\}$.

We can similarly define the following criteria of language learning.

(1) $\mathbf{Ap}^{d_1}\mathbf{UACp}^{d_2}\mathbf{TxtEx}^a$;
(2) $\mathbf{UAp}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtEx}^a$;
(3) $\mathbf{UAp}^{d_1}\mathbf{UACp}^{d_2}\mathbf{TxtEx}^a$;
(4) $\mathbf{Ap}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtFex}^a_b$;
(5) $\mathbf{Ap}^{d_1}\mathbf{UACp}^{d_2}\mathbf{TxtFex}^a_b$;
(6) $\mathbf{UAp}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtFex}^a_b$;
(7) $\mathbf{UAp}^{d_1}\mathbf{UACp}^{d_2}\mathbf{TxtFex}^a_b$;
(8) $\mathbf{Ap}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtBc}^a$;
(9) $\mathbf{Ap}^{d_1}\mathbf{UACp}^{d_2}\mathbf{TxtBc}^a$;
(10) $\mathbf{UAp}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtBc}^a$; and
(11) $\mathbf{UAp}^{d_1}\mathbf{UACp}^{d_2}\mathbf{TxtBc}^a$.

All the results in function learning have a counterpart in language learning. The following theorems give results which are new to language learning.

**Proposition 5** $(\forall i \in \mathbb{N})[\{L \mid L =^{i+1} \mathbb{N}\} \notin \mathbf{TxtFex}^i_*]$.

**Proof of Proposition 5:** Suppose by way of contradiction that $\mathbf{M}$ $\mathbf{TxtFex}^i_*$-identifies the above class. Let $\sigma$ be a $\mathbf{TxtFex}^i_*$-locking sequence for $\mathbf{M}$ on $\mathbb{N}$. Let $S$ be the set of grammars output by $\mathbf{M}$ on $\sigma$ which are at most $i$ different from $\mathbb{N}$. Thus, for any extension $\tau$ of $\sigma$, $\mathbf{M}(\tau) \in S$. Let $T = \{x \mid (\exists j \in S)[x \notin W_j]\}$. Clearly, $T$ is finite. Let $L$ be a language $i+1$ different from $\mathbb{N}$ such that content$(\sigma) \bigcup T \subseteq L$. Now, for all $j \in S$, $W_j \neq^i L$. Thus, $\mathbf{M}$ does not $\mathbf{TxtFex}^i_*$-identify $L$. A contradiction. □

**Theorem 14** [CL82] $(\forall i \in \mathbb{N})[\{L \mid L =^{2i+1} \mathbb{N}\} \notin \mathbf{TxtBc}^i]$.

**Theorem 15** [CL82] $\{L \mid L$ *is finite or* $L = \mathbb{N}\} \notin \mathbf{TxtBc}^*$.

**Theorem 16** *For all* $k \in \mathbb{N}$,

*1)* $\mathbf{TxtEx}^{k+1} - \mathbf{UAp}^1\mathbf{UACp}^1\mathbf{TxtFex}^k_* \neq \emptyset$;
*2)* $\mathbf{TxtBc}^{k+1} - \mathbf{UAp}^1\mathbf{UACp}^1\mathbf{TxtBc}^k \neq \emptyset$;
*3)* $\mathbf{TxtEx}^* - \bigcup_k \mathbf{UAp}^1\mathbf{UACp}^1\mathbf{TxtFex}^k_* \neq \emptyset$;
*4)* $\mathbf{TxtBc} - \mathbf{UAp}^1\mathbf{UACp}^1\mathbf{TxtFex}^*_* \neq \emptyset$;
*5)* $\mathbf{TxtEx}^{2k+1} - \mathbf{UAp}^1\mathbf{UACp}^1\mathbf{TxtBc}^k \neq \emptyset$;
*6)* $\mathbf{Ap}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtEx}^{2k} \subseteq \mathbf{Ap}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtBc}^k$;
*7)* $\mathbf{UAp}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtEx}^{2k} \subseteq \mathbf{UAp}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtBc}^k$;
*8)* $\mathbf{Ap}^{d_1}\mathbf{UACp}^{d_2}\mathbf{TxtEx}^{2k} \subseteq \mathbf{Ap}^{d_1}\mathbf{UACp}^{d_2}\mathbf{TxtBc}^k$;
*9)* $\mathbf{UAp}^{d_1}\mathbf{UACp}^{d_2}\mathbf{TxtEx}^{2k} \subseteq \mathbf{UAp}^{d_1}\mathbf{UACp}^{d_2}\mathbf{TxtBc}^k$; *and*
*10)* $\mathcal{E} \notin \mathbf{UAp}^1\mathbf{UACp}^1\mathbf{TxtBc}^*$.

**Proof of Theorem 16**: (1) Let $N_0 = 0$, $N_{3i+1} = N_{3i} + n^i$, $N_{3i+2} = N_{3i+1} + n^i$, and $N_{3i+3} = N_{3i+2} + 1$, $n > 1$. Consider the following class of languages:

$\mathcal{L} = \{L \in \mathcal{E} \mid$ following conditions hold:

1. $\bigcup_{i \in \mathbb{N}} [N_{3i}, N_{3i+1}) \subseteq L$
2. $\bigcup_{i \in \mathbb{N}} [N_{3i+1}, N_{3i+2}) \subseteq \overline{L}$
3. $\text{card}(\overline{L} \cap [\bigcup_{i \in \mathbb{N}} \{N_{3i+2}\}]) \leq k+1$

}

It is easy to see that $\mathcal{L} \in \mathbf{TxtEx}^{k+1}$. Also, since grammars for $L_1 = \bigcup_{i \in \mathbb{N}}[N_{3i}, N_{3i+1})$ and $L_2 = \bigcup_{i \in \mathbb{N}}[N_{3i+1}, N_{3i+2})$ are valid additional information of type $\mathbf{UAp}^1$ and $\mathbf{UACp}^1$, $\mathcal{L} \in \mathbf{UAp}^1\mathbf{UACp}^1\mathbf{TxtFex}^k_* \Leftrightarrow \mathcal{L} \in \mathbf{TxtFex}^k_*$. Suppose by way of contradiction that $\mathbf{M}$ $\mathbf{TxtFex}^k_*$-identifies $\mathcal{L}$. It is, then, easy to convert $\mathbf{M}$ to $\mathbf{M}'$ such that $\mathbf{M}'$ $\mathbf{TxtFex}^k_*$-identifies $\{L \mid L =^{k+1} \mathbb{N}\}$. But this is not true (proposition 5). Thus, no such $\mathbf{M}$ can exist.

(2), (3), and (5) can be be proved similarly.

(4) Let the $N_i$'s be as defined in the proof of part 1. Consider the following class of languages:

$\mathcal{L} = \{L \in \mathcal{E} \mid$ following conditions hold:

1. $\bigcup_{i \in \mathbb{N}}[N_{3i}, N_{3i+1}) \subseteq L$
2. $\bigcup_{i \in \mathbb{N}}[N_{3i+1}, N_{3i+2}) \subseteq \overline{L}$
3. $(\forall i)(\exists! j)[N_{3\langle i,j\rangle+2} \in L]$
4. $(\forall^\infty i)[[N_{3\langle i,j\rangle+2} \in L] \Rightarrow [W_j = L]]$

}

It is easy to see that $\mathcal{L} \in \mathbf{TxtBc}$. Also, $\mathcal{L} \in \mathbf{UAp}^1\mathbf{UACp}^1\mathbf{TxtFex}^*_* \Leftrightarrow \mathcal{L} \in \mathbf{TxtFex}^*_* \Leftrightarrow \{L' \mid L \in \mathcal{L}\} \in \mathbf{TxtFex}^*_*$, where $L' = \{\langle i, j\rangle \mid N_{3\langle i,j\rangle+2} \in L\}$.

However, the proof of $\mathbf{Bc} - \mathbf{Ex}^* \neq \emptyset$ in [CS83] can easily be modified to show that $\{L' \mid L \in \mathcal{L}\} \notin \mathbf{TxtFex}^*_*$. Hence, $\mathcal{L} \notin \mathbf{UAp}^1\mathbf{UACp}^1\mathbf{TxtFex}^*_*$. This proves 4.

(6) This proof is the same as used in [CL82] to prove that $\mathbf{TxtEx}^{2k} \subseteq \mathbf{TxtBc}^k$. Let $\mathbf{M}$ $\mathbf{Ap}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtEx}^{2k}$-identify $\mathcal{C}$. $\mathbf{M}'$ can $\mathbf{Ap}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtBc}^k$-

identify $\mathcal{C}$ as follows. $\mathbf{M}'$ given $p_1$ (as positive additional information), $p_2$ (as negative additional information), and $\sigma$ behaves as described below. Let $|\sigma| = s$. Recall that as defined in Section 2, $W_j^s = \{x \leq s \mid \Phi_j(x) \leq s\}$. Let $\mathbf{M}$, given $p_1, p_2$, and $\sigma$, output $j$. Let $T = \{x \mid x \in W_j^s - \text{content}(\sigma)\}$. Let $S$ be the set of $k$ least elements of $T$ (if $\text{card}(T) < k$ then let $S = T$). Output $p(j)$ where $W_{p(j)} = W_j \cup \text{content}(\sigma) - S$. It is easy to see that $\mathbf{M}'$ $\mathbf{Ap}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtBc}^k$-identifies $\mathcal{C}$. This proves 6.

(7), (8), and (9) can be proved similarly.

(10) Let the $N_i$'s be as defined in the proof of part 1. For any language $L$, define $L'$ as follows:

1. $\bigcup_{i \in \mathbb{N}}[N_{3i}, N_{3i+1}) \subseteq L'$
2. $\bigcup_{i \in \mathbb{N}}[N_{3i+1}, N_{3i+2}) \subseteq \overline{L'}$
3. $(\forall i)[i \in L \Leftrightarrow N_{3i+2} \in L']]$

Clearly, $\{L' \mid L \in \mathcal{E}\} \in \mathbf{UAp}^1\mathbf{UACp}^1\mathbf{TxtBc}^* \Leftrightarrow \mathcal{E} \in \mathbf{TxtBc}^*$.

Since $\mathcal{E} \notin \mathbf{TxtBc}^*$ we have $\{L' \mid L \in \mathcal{E}\} \notin \mathbf{UAp}^1\mathbf{UACp}^1\mathbf{TxtBc}^*$. Thus, $\mathcal{E} \notin \mathbf{UAp}^1\mathbf{UACp}^1\mathbf{TxtBc}^*$. This proves the theorem.

∎ Theorem 16.

**Theorem 17** $(\forall d > 0)[\mathbf{UAp}^d\mathbf{TxtEx} - \mathbf{Ap}^1\mathbf{UACp}^1\mathbf{TxtBc}^* \neq \emptyset]$.

Proof of Theorem 17 is similar to the proof of Theorem 18 below.

**Theorem 18** $(\forall d > 0)[\mathbf{UACp}^d\mathbf{TxtEx} - \mathbf{UAp}^1\mathbf{ACp}^1\mathbf{TxtBc}^* \neq \emptyset]$.

**Proof of Theorem 18**: Let $N_0 = 0$, $N_{i(i+1)+1} = N_{i(i+1)} + n^i$, $N_{i(i+1)+2} = N_{i(i+1)+1} + n^i$, $N_{i(i+1)+2+2j+1} = N_{i(i+1)+2+(2j)} + 1$, and $N_{i(i+1)+2+2j+2} = N_{i(i+1)+2+(2j)+1} + n^i$, where $j < i$ and $n > 1$. Let $S_j = \bigcup_{k \in \mathbb{N}} \bigcup_{l < \langle j,k \rangle} \{N_{\langle j,k \rangle * (\langle j,k \rangle + 1) + 2 + 2l}\}$. Consider the following class of languages:

$\mathcal{L} = \{L \in \mathcal{E} \mid$ following conditions hold:

1. $\bigcup_{i \in \mathbb{N}} [N_{i(i+1)}, N_{i(i+1)+1}) \subseteq L$
2. $\bigcup_{i \in \mathbb{N}} [N_{i(i+1)+1}, N_{i(i+1)+2}) \subseteq \overline{L}$
3. $\bigcup_{i \in \mathbb{N}} \bigcup_{j<i} [N_{i(i+1)+2+2j+1}, N_{i(i+1)+2+2j+2}) \subseteq L$
4. $(\forall x, y, j)[[x \in S_j \wedge y \in S_j] \Rightarrow [x \in L \Leftrightarrow y \in L]]$
5. $\{j \mid S_j \subseteq L\}$ is finite or co-finite.

$\}$

It is easy to see that $\mathcal{L} \in \mathbf{UACp}^d\mathbf{TxtEx}$ (since the additional information gives the text for the complement, and finite-cofinite languages can be identified on characteristic function input).

Also, $\mathcal{L} \in \mathbf{UAp}^1\mathbf{ACp}^1\mathbf{TxtBc}^* \Leftrightarrow \mathcal{L} \in \mathbf{TxtBc}^*$, and $\mathcal{L} \in \mathbf{TxtBc}^* \Rightarrow \{L \mid L$ is finite or co-finite$\} \in \mathbf{TxtBc}^*$. But $\{L \mid L$ is finite or co-finite$\} \notin \mathbf{TxtBc}^*$. Hence, $\mathcal{L} \notin \mathbf{UAp}^1\mathbf{ACp}^1\mathbf{TxtBc}^*$. This proves the theorem.

∎ Theorem 18.

**Theorem 19** $(\forall d_1, d_2 \mid d_2 > d_1)[\mathbf{Ap}^{d_2}\mathbf{TxtEx} - \mathbf{UAp}^{d_1}\mathbf{UACp}^1\mathbf{TxtBc}^* \neq \emptyset]$.

Proof of Theorem 19 is similar to the proof of Theorem 20 below.

**Theorem 20** $(\forall d_1, d_2 \mid d_2 > d_1)[\mathbf{ACp}^{d_2}\mathbf{TxtEx} - \mathbf{UAp}^1\mathbf{UACp}^{d_1}\mathbf{TxtBc}^* \neq \emptyset]$.

**Proof of Theorem 20**: Without loss of generality, let $d_1 = l/n$, $d_2 = (l+3)/n$, $n > 3$, where $l, n \in \mathbb{N}$.

Let $N_0 = 0$.

For $j < n^i$, $i \geq 0$, let

$N_{2*(n^i-1)/(n-1)+2j+1} = N_{2*(n^i-1)/(n-1)+2j} + n$ and

$N_{2*(n^i-1)/(n-1)+2j+2} = N_{2*(n^i-1)/(n-1)+2j+1} + n^i$

Let $S_j = \bigcup_{k \in \mathbb{N}} \bigcup_{m<n^{\langle j,k \rangle}} \bigcup_{s \in \{x \mid l \leq x < n\}} \{N_{2*(n^{\langle j,k \rangle}-1)/(n-1)+2m} + s\}$.

Consider the following class of languages:

$\mathcal{L} = \{L \in \mathcal{E} \mid$ following conditions hold:

1. $\bigcup_{i \in \mathbb{N}} \bigcup_{j<n^i} [N_{2*(n^i-1)/(n-1)+2j+1}, N_{2*(n^i-1)/(n-1)+2j+2}) \subseteq L$
2. $\bigcup_{i \in \mathbb{N}} \bigcup_{j<n^i} \bigcup_{r<l} \{N_{2*(n^i-1)/(n-1)+2j} + r\} \subseteq \overline{L}$
3. $(\forall x, y, j)[[x \in S_j \wedge y \in S_j] \Rightarrow [x \in L \Leftrightarrow y \in L]]$
4. $\{j \mid S_j \subseteq L\}$ is finite or co-finite.

$\}$

It is easy to see that $\mathcal{L} \in \mathbf{ACp}^{d_2}\mathbf{TxtEx}$.

Also, $\mathcal{L} \in \mathbf{UAp}^1\mathbf{UACp}^{d_1}\mathbf{TxtBc}^* \Leftrightarrow \mathcal{L} \in \mathbf{TxtBc}^*$, and $\mathcal{L} \in \mathbf{TxtBc}^* \Rightarrow \{L \mid L$ is finite or co-finite$\} \in \mathbf{TxtBc}^*$. But $\{L \mid L$ is finite or co-finite$\} \notin \mathbf{TxtBc}^*$. Hence, $\mathcal{L} \notin \mathbf{UAp}^1\mathbf{UACp}^{d_1}\mathbf{TxtBc}^*$. This proves the theorem. ∎ Theorem 20.

The above theorems give the complete relationship between different language identification criteria introduced in this section. We observe some of these relationships in Corollary 4 below which follows from results presented in this section, language learning counterpart of results presented in section 3.2, and Theorem 13.

**Corollary 4** *Let* $d_1, d_2, d_3, d_4 \in [0, 1]$. *Let* $a, b \in \mathbb{N} \cup \{*\}$.
*a)* $\mathbf{Ap}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtEx}^a \subseteq \mathbf{Ap}^{d_3}\mathbf{ACp}^{d_4}\mathbf{TxtEx}^b \Leftrightarrow [d_1 \leq d_3$ *and* $d_2 \leq d_4$ *and* $a \leq b]$.
*b)* $\mathbf{Ap}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtBc}^a \subseteq \mathbf{Ap}^{d_3}\mathbf{ACp}^{d_4}\mathbf{TxtBc}^b \Leftrightarrow [d_1 \leq d_3$ *and* $d_2 \leq d_4$ *and* $a \leq b]$.
*c)* $\mathbf{UAp}^{d_1}\mathbf{UACp}^{d_2}\mathbf{TxtEx}^a \subseteq \mathbf{UAp}^{d_3}\mathbf{UACp}^{d_4}\mathbf{TxtEx}^b \Leftrightarrow [d_1 \leq d_3$ *and* $d_2 \leq d_4$ *and* $a \leq b]$.
*d)* $\mathbf{UAp}^{d_1}\mathbf{UACp}^{d_2}\mathbf{TxtBc}^a \subseteq \mathbf{UAp}^{d_3}\mathbf{UACp}^{d_4}\mathbf{TxtBc}^b \Leftrightarrow [d_1 \leq d_3$ *and* $d_2 \leq d_4$ *and* $a \leq b]$.
*e)* $(\forall d_1 \in (0, 1])[\mathbf{Ap}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtEx}^a \subset \mathbf{UAp}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtEx}^a]$.
*f)* $(\forall d_1 \in (0, 1])[\mathbf{Ap}^{d_1}\mathbf{UACp}^{d_2}\mathbf{TxtEx}^a \subset \mathbf{UAp}^{d_1}\mathbf{UACp}^{d_2}\mathbf{TxtEx}^a]$.

*g)* $(\forall d_2 \in (0,1])[\mathbf{Ap}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtEx}^a \subset \mathbf{Ap}^{d_1}\mathbf{UACp}^{d_2}\mathbf{TxtEx}^a]$.
*h)* $(\forall d_2 \in (0,1])[\mathbf{UAp}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtEx}^a \subset \mathbf{UAp}^{d_1}\mathbf{UACp}^{d_2}\mathbf{TxtEx}^a]$.
*i)* $(\forall d_1 \in (0,1])[\mathbf{Ap}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtBc}^a \subset \mathbf{UAp}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtBc}^a]$.
*j)* $(\forall d_1 \in (0,1])[\mathbf{Ap}^{d_1}\mathbf{UACp}^{d_2}\mathbf{TxtBc}^a \subset \mathbf{UAp}^{d_1}\mathbf{UACp}^{d_2}\mathbf{TxtBc}^a]$.
*k)* $(\forall d_2 \in (0,1][\mathbf{Ap}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtBc}^a \subset \mathbf{Ap}^{d_1}\mathbf{UACp}^{d_2}\mathbf{TxtBc}^a]$.
*l)* $(\forall d_2 \in (0,1][\mathbf{UAp}^{d_1}\mathbf{ACp}^{d_2}\mathbf{TxtBc}^a \subset \mathbf{UAp}^{d_1}\mathbf{UACp}^{d_2}\mathbf{TxtBc}^a]$.

# 5 Conclusions

The aim of this paper was to take a first step in modeling the presence of partial explanations in learning situations, and to investigate the effect of such additional information on the learning capability of algorithmic learning devices. Two learning situations were considered: practice of science modeled as inference of programs for recursive functions and language learning modeled as inference of type 0 grammars for recursively enumerable sets. It was shown, in both the learning situations, that the presence of partial explanation as additional information enhances the learning capability of machines. Furthermore, certain density notions were used to model the quality of partial explanation, and it was shown, in the context of both the learning situations, that a better quality partial explanation enhances the learning capability of algorithmic learning machines more than a not so good partial explanation.

Finally, we would like to state two shortcomings in this work which suggest obvious directions for further investigation.

In the context of "scientific" inference of functions, our partial explanations do not contradict the function being learned. This is clearly a very simplistic model of partial explanation, as there is no reason to believe that the state of the art explanation available to a scientist makes no errors of commission. Hence, a natural line of further investigation would be the study

of partial explanations that are correct on a set of certain density and either undefined or incorrect off that set.

Also, we would like to point out the ad hoc nature of approximate learning notions, as they are dependent on the choice of Gödel numbering used in encoding the experiments and their outcomes. A particular encoding of experiments and experimental outcomes is presupposed when a recursive function is used to model a phenomenon. The density of the codes of a class of experiments for a phenomenon could change with the change in the encoding scheme used. For instance, consider the predictions of Aristotelian Physics on experiments in classical mechanics[1]. There exist Gödel numbering of experiments for which Aristotelian Physics is correct on a set of density one, and at the same time there exist Gödel numbering of experiments for which Aristotelian Physics is correct only on a set of density of zero. Addressing this issue of the dependence of density notions on the choice of Gödel numbering used to encode experiments and experimental outcomes is an obvious future research direction.

# 6 Acknowledgements

We are grateful to an anonymous referee for several useful comments which have resulted in many improvements in the paper. We would like to thank John Case and Mark Fulk for timely advice and continuous encouragement. Zuzana Dobes, Lata Narayanan, and Rajeev Raman provided helpful discussions. Department of CS at SUNY Buffalo, Department of CIS at University of Delaware, and the Xerox University Grants Program to University of Rochester provided equipment support for the preparation of this manuscript. Sanjay Jain was supported by NSF grant CCR 832-0136 and Arun Sharma was supported by NSF grant CCR 871-3846.

[1]This example was pointed out to us by an anonymous referee.

# References


[Ang80a] D. Angluin. Finding patterns common to a set of strings. *Journal of Computer and System Sciences*, 21:46–62, 1980.

[Ang80b] D. Angluin. Inductive inference of formal languages from positive data. *Information and Control*, 45:117–135, 1980.

[AS83] D. Angluin and C. Smith. A survey of inductive inference: Theory and methods. *Computing Surveys*, 15:237–289, 1983.

[Bar74] J. M. Barzdin. Two theorems on the limiting synthesis of functions. *In Theory of Algorithms and Programs, Latvian State University, Riga*, 210:82–88, 1974. In Russian.

[BB64] R. Brown and U. Bellugi. Three processes in the child's acquisition of syntax. *Harvard Educational Review*, 34:133–151, 1964.

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

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

[Cas86] J. Case. Learning machines. In W. Demopoulos and A. Marras, editors, *Language Learning and Concept Acquisition*. Ablex Publishing Company, 1986.

[Cas88] J. Case. The power of vacillation. In D. Haussler and L. Pitt, editors, *Proceedings of the Workshop on Computational Learning Theory*, pages 133–142. Morgan Kaufmann Publishers, Inc., 1988.

[Che81] K. Chen. *Tradeoffs in Machine Inductive Inference*. PhD thesis, SUNY at Buffalo, 1981.

[CJS89] J. Case, S. Jain, and A. Sharma. Convergence to nearly minimal size grammars by vacillating learning machines. In R. Rivest, D. Haussler, and M. K. Warmuth, editors, *Proceedings of the Second Annual Workshop on Computational Learning Theory, Santa Cruz, California*, pages 189–199. Morgan Kaufmann Publishers, Inc., August 1989.

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

[CS83] J. Case and C. Smith. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science*, 25:193–220, 1983.

[Dal76] P. Dale. *Language Development, Structure and Function*. Holt, Reinhart, and Winston, New York, 1976.

[Ful85] M. Fulk. *A Study of Inductive Inference Machines*. PhD thesis, SUNY at Buffalo, 1985.

[Ful90a] M. Fulk. Inductive inference with additional information. *Journal of Computer and System Sciences*, 1990.

[Ful90b] M. Fulk. Prudence and other conditions on formal language learning. *Information and Computation*, 85:1–11, 1990.

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

[HU79] J. Hopcroft and J. Ullman. *Introduction to Automata Theory Languages and Computation*. Addison-Wesley Publishing Company, 1979.

[JS89] S. Jain and A. Sharma. Knowledge of an upper bound on grammar size helps language learning. Technical Report 283, University of Rochester, 1989.

[JS90a] S. Jain and A. Sharma. Hypothesis formation and language acquisition with an infinitely often correct teacher. In R. Parikh, editor, *Proceedings of the Third Conference on Theoretical Aspects of Reasoning About Knowledge*, pages 225–239. Morgan Kaufmann Publishers, Inc., March 1990.

[JS90b] S. Jain and A. Sharma. Language learning by a team. In M. S. Paterson, editor, *Proceedings of the 17th International Colloquium on Automata, Languages and Programming*, pages 153–166. Springer-Verlag, July 1990. Lecture Notes in Computer Science, 443.

[KW80] R. Klette and R. Wiehagen. Research in the theory of inductive inference by GDR mathematicians – A survey. *Information Sciences*, 22:149–169, 1980.

[MY78] M. Machtey and P. Young. *An Introduction to the General Theory of Algorithms*. North Holland, New York, 1978.

[OSW86a] D. Osherson, M. Stob, and S. Weinstein. Aggregating inductive expertise. *Information and Control*, 70:69–95, 1986.

[OSW86b] D. Osherson, M. Stob, and S. Weinstein. *Systems that Learn, An Introduction to Learning Theory for Cognitive and Computer Scientists*. MIT Press, Cambridge, Mass., 1986.

[OW82a] D. Osherson and S. Weinstein. Criteria of language learning. *Information and Control*, 52:123–138, 1982.

[OW82b] D. Osherson and S. Weinstein. A note on formal learning theory. *Cognition*, 11:77–88, 1982.

[Pei58] C. S. Peirce. *Collected Papers (Edited by A. W. Burks)*. Harvard University Press, Cambridge, Mass., 1958.

[Pit84] L. Pitt. *A characterization of probabilistic inference*. PhD thesis, Yale University, 1984.

[Rei70] F. E. Reilly. *Charles Peirce's Theory of Scientific Method*. Fordham University Press, New York, 1970.

[Rog58] H. Rogers. Gödel numberings of partial recursive functions. *Journal of Symbolic Logic*, 23:331–341, 1958.

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

[Roy86] J. Royer. Inductive inference of approximations. *Information and Control*, 70:156–178, 1986.

[RU63] G. F. Rose and J. S. Ullian. Approximations of functions on the integers. *Pacific J. Math.*, 13:693–701, 1963.

[Smi82] C. Smith. The power of pluralism for automatic program synthesis. *Journal of the ACM*, 29:1144–1165, 1982.

[SV86] C. Smith and M. Velauthapillai. On the inference of programs approximately computing the desired function. *Lecture Notes in Computer Science*, 265:164–176, 1986.

[Wie77] R. Wiehagen. Identification of formal languages. In *Mathematical Foundations of Computer Science, Proceedings, 6th Symposium, Tatranska Lomnica*, pages 571–579. Springer-Verlag, 1977. Lecture Notes in Computer Science 53.

[Wie78] R. Wiehagen. Characterization problems in the theory of inductive inference. *Lecture Notes in Computer Science*, 62:494–508, 1978.