This report reproduces a dissertation presented to the Faculty of the Graduate School of Yale University in candidacy for the degree of Doctor of Philosophy, May, 1985.

This research was partially funded by the National Science Foundation under grant numbers MCS 8002447, MCS 8116678, MCS 8204246, and MCS 8404226.

PROBABILISTIC INDUCTIVE INFERENCE

Leonard Brian Pitt

YALEU/DCS/TR-400
June, 1985

ABSTRACT

PROBABILISTIC INDUCTIVE INFERENCE

Leonard Brian Pitt
Yale University
1985

Inductive inference machines are algorithms which accept as input only the values of an arbitrary recursive function f, and attempt to synthesize programs which compute f. The classes of functions which can be successfully inferred depends both on the type of inference strategy employed, as well as the assumed "identification criterion," or definition of successful inference.

In this dissertation we define ***probabilistic*** inductive inference machines, which follow randomized strategies to infer functions. Let $0 \leq p < 1$. We ask, for various identification criteria, whether a probabilistic inductive inference machine can infer larger classes of functions if the criterion of successful inference is relaxed to allow inference with probability at least p, as opposed to requiring certainty.

For the most basic identification criteria (EX and BC) we show that any class of functions which can be inferred from examples with probability exceeding $\frac{1}{2}$ can be inferred deterministically, and that for probabilities $p \leq \frac{1}{2}$ there is a discrete hierarchy of inferability parameterized by p.

We characterize the power of the probabilistic computational model by showing that the classes of functions inferable are identical to those classes inferable by ***teams*** of inductive inference machines, a previously investigated computational model of inference allowing a finite number of different strategies to be run simultaneously. For the criteria EX and BC we show in addition that *frequency* inference, a third model for inference, gives rise to the same classes of inferable functions. This unifying characterization suggests that there is a notion of "uncertainty of success" which is invariant across computational models of inference.

Acknowledgements

I am extremely grateful to Dana Angluin, my advisor and friend, for more reasons than I could hope to fit on these pages. She has been a constant source of encouragement and ideas since I met her five years ago. I have always found her door to be open, her manner to be friendly, and her advice (both technical and worldly) to be excellent. I only hope that I can repay my debt by striving to provide my students with an advisor half as wonderful.

I would also like to thank Neil Immerman and David Lichtenstein, the other members of my reading committee, for their many helpful, and more important, timely comments on a first draft of this dissertation. I am grateful to Robert Daley, for pointing out Podnieks' frequency results; and Carl Smith, for reading an early abstract, and for pointing out the excellent paper [10]. I am indebted to the taxpayers of the United States of America, whom, through National Science Foundation grants MCS 8002447, MCS 8204246, MCS 8116678, and MCS 8404226, have supported my education and research while at Yale.

That this is a "Theory" thesis and not an "AI" thesis is due in part to Dana Angluin, Dan Gusfield, and David Lichtenstein. I feel lucky that they noticed in me the seeds of discontent, and lured me from the third floor with their comfortable manner and their stimulating problems. David was especially encouraging during my first couple of years.

Greg Sullivan, my office-mate of many years, has taught me an enormous amount about everything from computer science to world affairs. He has been invigorating and challenging, and if I'm somewhat closer to knowing what a real proof is, then it is certainly to his credit. I will miss our late night discussions of politics and Gödel over pizza.

Many other people have helped me along the way, either through insightful comments, administrative support, or friendship. I'd like to thank Jim Arvo, William Bain, Sandeep Bhatt, Josh Cohen, young David Eisenstat/Angluin (... who helped me keep my perspective; it was difficult to take research too seriously when, in the midst of a meeting with Dana, he would gleefully mangle stacks of technical reports), Michael Fischer, Bill Gropp, Dan Jordan, Richard Kelsey, Phil Laird, David Littleboy, Steven Lytinen, Richard Morrow, Alex Nicolau, Ria Palumbo, Mukesh Prasad, Miriam Putterman, Christine Tattersall, David Wittenberg, and anyone who might be offended because I've forgotten to include them.

My parents and grandmother also played a major role toward the eventual production of this work. They motivated and challenged me from an early age, and through their confidence in me I learned to be confident. I don't quite know how to thank them, but I have the feeling that this dissertation will be thanks enough – they are certain to enjoy it much more than anyone who understands it.

Finally, I'd like to thank my wife, Claire, for her emotional support, her tolerance, her encouragement, her disgust (when appropriate), her understanding, and most important, her love, during the last few years. I'm sure this would have been much more difficult had she not been there to lean on after a bad day.

Contents

Index to Symbols

Chapter 1

Introduction

1.1 Overview

Consider the following situations:

- A scientist investigates a phenomenon by performing experiments and discovers a predictive theory of the phenomenon.
- A child implicitly acquires a grammar for English by listening to his elders.
- A test–taker determines the next value in a given sequence of numbers.

Each of these situations involves *inductive inference* — the process of determining a general rule from examples of the rule, a task at which human beings are quite proficient.

If our main concern is not the agent doing the inference, but rather the process of inference itself, then we find ourselves well within the domain of theoretical computer science. Research from this vantage point focuses on both general theoretical properties of inference techniques, and finding specific methods for inference within particular domains. Inductive inference has applications in linguistics, artificial intelligence, pattern recognition, cryptography, and the philosophy of science, among others.

To study inference formally, Gold [23] defines *inductive inference machines* (IIMs), an abstract model of computation for doing inference. An IIM is essentially any algorithmic device which attempts to infer rules from examples. In this model, a "rule" is any partial recursive function φ, and an "example" is a pair $\langle x, \varphi(x)\rangle$ for some x in the domain of φ. That this is a very general definition is clear, since any rule which might be useful (in the sense of specifying how predictions are to be made) is essentially a partial recursive function.[1] A predictive explanation for the rule φ is simply a program p which computes φ. Thus an IIM takes as input the values of some partial recursive function φ, and attempts to output a program p which computes φ, based on the examples it has seen. Note that if after seeing some finite number of examples, the IIM guesses the program p, the very next example might be inconsistent with p. For this reason, the inference of functions is seen as an infinite process, which occurs "in the limit".

There are two standard criteria of successful inference (or ***identification***) in the limit of an IIM on a given function φ: EX and BC. EX–identification requires that the infinite sequence of guesses of the IIM converge to a single program computing φ, while BC–identification requires only that after some finite initial segment, all the guesses be correct (but possibly different) programs for φ. Let ID denote either of these identification criteria. The class of functions $ID(M)$ consists of those functions φ such that M ID–identifies φ. There are two associated identifiability classes:

$$ID = \{U \mid U \subseteq ID(M) \text{ for some IIM } M\}.$$

Unfortunately, it has been shown [3,10] that there are classes of functions which are not in either class ID, *i.e.* classes of functions for which no single inference strategy (IIM) is correct. Thus no single strategy is general enough to infer every rule from examples, and since the definition of an IIM allows as much power as a Turing machine (the "most general" computing device), it is not likely that any other ***deterministic*** computational model will allow inference of all possible rules.

We are thus motivated to allow randomization as part of the inference machine, and then ask: "are more classes of functions identifiable if we only require the inference machine to be correct with some probability $p \leq 1$?" This is the main issue addressed in this dissertation.

We define ***probabilistic*** IIMs, which are IIMs having the ability to flip a fair coin. A probabilistic IIM P is allowed a (potentially infinite) sequence of 0–1 coin tosses. Now if we fix the (infinite) input sequence of examples of some function φ, then each 0–1 infinite sequence of coin tosses determines a sequence of guesses of P, which may or may not converge (in the EX or BC sense) to a program which computes φ. If we consider the usual Borel measure on the infinite coin toss sequences, then the set of guess sequences that converge to programs for φ is measurable, and is denoted $\Pr[P\ ID\text{–identifies } \varphi]$. For any value $p \geq 0$, the class $ID_p(P)$ consists of those functions φ such that $\Pr[P$

[1] We assume Church's Thesis.

ID–identifies $\varphi] \geq p$. Then the probabilistic identification class $ID_{prob}(p)$ is defined by

$$ID_{prob}(p) = \{U \mid U \subseteq ID_p(P) \text{ for some IIM } P\}.$$

Our results give a description of the structure of the classes $ID_{prob}(p)$ as a function of p. For both the EX and BC criteria there is a discrete hierarchy of classes, with "breakpoints" at the values $\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots$ That is, for all $n = 1, 2, 3, \ldots$, $ID_{prob}(\frac{1}{n})$ is a proper subclass of $ID_{prob}(\frac{1}{n+1})$; and if p_1 and p_2 are in the same half-open interval $(\frac{1}{n+1}, \frac{1}{n}]$, then $ID_{prob}(p_1) = ID_{prob}(p_2)$. Also, for both criteria, the sets of functions that can be identified by some machine with probability $p > \frac{1}{2}$ can be identified by some ***deterministic*** IIM.

This work is also one of unification, in that the precise statement of our main results gives an equivalence between three different models of computation for inductive inference: probabilistic identification as described above, frequency identification introduced by Podnieks [33], and identification by teams of machines introduced by Smith [35]. (It also settles an open problem of Podnieks for frequency identification.) This is unusual, for in many cases the introduction of new computational models for inductive inference gives rise to new and "orthogonal" hierarchies of identifiability.

In addition to EX and BC, there are a number of different identification criteria which have been previously investigated [10], with results reflecting the power and limitations of different types of inference strategies. We apply the probabilistic model of computation to many of these criteria, and give partial characterizations of the relationships between probabilistic computation and other computational models including teams, frequency, determinism, and nondeterminism.

This dissertation is organized as follows. In Chapter 2 we formally define the probabilistic model of computation for inductive inference, as well as several other computational models. The general notion of an identification criterion is also discussed. Chapter 3 contains our most central results showing the existence of a discrete probabilistic hierarchy, and the equivalence of the probabilistic model with teams and frequency for EX and BC. In Chapter 4 we apply the techniques developed in Chapter 3 to other identification criteria. Finally, in Chapter 5, we conclude by summarizing our results, and suggesting interesting areas for future research. We conclude this introduction with a review of the relevant inductive inference literature.

1.2 Previous Work

The reader interested in a general introduction to inductive inference may find an excellent survey of both the theoretical and more concrete results in [1]. Here we review previous work relating probability and inductive inference.

There have been a number of papers [12,22,25,27,40] involving the inference of ***stochastic grammars*** from randomly generated words. The inference model used is generally a deterministic one, whereas the function (grammar) is probabilistic. There is little relationship between the results describing effective techniques within this particular domain, and the abstract probabilistic model and characterization presented here.

Valiant [39] describes "concept learning" algorithms for boolean formulae. In his framework, examples of a ***concept*** (essentially a boolean vector of features) are provided by Nature randomly and according to some unknown but predetermined probability distribution. For certain classes of concepts (categorized by the types and sizes of the boolean formulae which describe them), deterministic algorithms are given which (with high probability) produce programs in polynomial time, which accurately classify examples and non–examples of a concept with high probability. The domain here is interesting, in that the criterion of success only requires the hypothesized program to be approximately correct (*i.e.* incorrect only on examples which have low probability (according to Nature) of being encountered); moreover, a hypothesis need not be produced if the set of examples seen are unrepresentative of the concept (*i.e.* have small total probability according to Nature's distribution.)

Work by Barzdin and Freivald [4], and Podnieks [34], has been concerned with the use of randomization to reduce the number of "mind changes" required by an IIM to identify functions. In their model, the IIM is required (with probability 1) to converge to a correct hypothesis, but is allowed to flip coins in order to (on the average) reduce the number of changes of hypotheses en route to a correct hypothesis. It is shown that if the final hypothesis is required to come from a particular Gödel numbering of the partial recursive functions, then the upper and lower bound for the number of mind changes required by an IIM to converge to a correct hypothesis for every function in some recursively enumerable class of total recursive functions is n, where n is the index of the function in the given numbering. Using a randomized strategy, however, this can be improved to an expected value of $\log n$.

A more recent paper by Wiehagen, Freivald, and Kinber [41] investigates the advantages of probabilistic inductive inference strategies over deterministic ones when the strategies are required to converge to a correct answer within some fixed number of changes in hypotheses. It is shown that for all $\epsilon > 0$, and all $n \geq 2$, there are classes of functions identifiable with probability at least $1 - \epsilon$ with at most n mind changes, but not by any deterministic strategy with at most n mind changes. Other results give that even when relaxing the criterion of successful inference to allow success with probability $p > \frac{1}{2}$, the speedup over deterministic strategies in the number of mind changes is at most linear. They also independently prove that $EX_{prob}(p) = EX$ when $p > \frac{1}{2}$, which is a special case of our Theorem 3.21. We discuss their results in greater detail in Section 4.3.1.

Freivald [21] investigates probabilistic inference in the setting of finite computations. In Freivald's model, an IIM P *finitely* identifies a set of functions U with probability p, if the probability is greater than p that P eventually halts and outputs a correct program for $f \in U$, given input/output examples of f. Freivald shows that if there is a probabilistic IIM which finitely identifies some set of functions U with probability $> \frac{2}{3}$, then there is a deterministic strategy finitely identifying U. He also gives a discrete hierarchy parameterized by p of those classes of functions finitely identifiable by a single machine with probability at least p. We review his results in more detail in Section 4.3.2, where we also extend them. Freivald's results for probabilistic finite identification motivated our work, and although many of the techniques used in the finite case do not extend to the limiting case, there is some overlap of ideas. In particular, the "threshold" programs described in Section 2.2 are a particularly useful tool.

Chapter 2

Preliminaries

2.1 Notation and Definitions

Unfortunately, in any technical document, precision, clarity, and brevity necessitate the introduction and use of many symbols. The reader is advised to keep a bookmark in this section. We also point out the Index to Symbols on page viii which gives the page where any symbol was first introduced.

The null or empty set is denoted by $\emptyset$. We use the symbols $\subseteq, \subset, \in, \bigcup$, and $\bigcap$ to denote the set operations containment, proper containment, membership, union, and intersection, respectively. The symbol $\biguplus$ is the set operation union, together with the assertion that the operands of the union are mutually disjoint; thus $S = \biguplus_{i\in N} S_i$ states that not only is S the union of the sets $\{S_i\}$, but also that for all $i \neq j$, $S_i \bigcap S_j = \emptyset$.

R denotes the set of real numbers, and $N = \{0,1,2,3,\ldots\}$ is the set of natural numbers. $N_t = \{0,1,\ldots,t-1\}$ = the first t elements of N. If $I \subseteq N$ is finite, then *max* I is the largest element of I, and *min* I is the least element. If $\{x_k\}_{k\in N}$ is a sequence of nonnegative real numbers, then $\liminf_{k\to\infty} x_k$ is defined as $\lim_{k\to\infty} \inf\{x_i | i \geq k\}$. If $x \in R$, then $\lfloor x \rfloor$ denotes the floor of x, or the greatest integer less than or equal to x, and $\lceil x \rceil$ is the ceiling of x, or the least integer greater than or equal to x. Intervals of real numbers are represented in the usual way, with round or square brackets to indicate exclusion or inclusion of the endpoint. For example, $(a,b] = \{p \in R \mid a < p \leq b\}$.

If S and T are sets, and I is a multiset, then $S - T$ is the set containing all elements of S which are not in T, and $T^c = N - T$ = the complement of T. $|S|$ denotes the cardinality of S; $|I|$ is the number of (not necessarily distinct) elements of the multiset I, and $|I \bigcap S|$ is the number of (not necessarily distinct) elements of the multiset I which are also elements of S. S^k (respectively S^∞) denotes the cartesian product of S with itself k (respectively an infinite number of) times, or equivalently, all sequences of k (respectively infinitely many) elements drawn from S. S^* denotes $\bigcup_{n \in \mathrm{N}} S^n$. (The set of all finite sequences of elements of S). If s is a finite or infinite sequence of elements of S (*i.e.* $s \in S^* \bigcup S^\infty$) then s_k denotes the k^{th} element of s (if it exists), $|s|$ denotes the length of s, and we say *s converges to x* (and write $s \downarrow= x$) iff $|s| = k < \infty$ and $s_k = x$ or $|s| = \infty$ and $s_k = x$ for all but finitely many values of k. The symbol $s|_k$ denotes the finite sequence $s_1, s_2, \ldots, s_k$ if $|s| \geq k$, otherwise $s|_k = s$.

If $E(k)$ is an equation containing the variable k, with k ranging over N, then we write "$(\forall_k^\infty)\ E(k)$" to indicate that the equation $E(k)$ is true *almost everywhere*, or for all but finitely many values of k. We write "$(\exists_k^\infty)\ E(k)$" to indicate that $E(k)$ is true *infinitely often*, or for infinitely many values of k.

Lower case letters $(i, j, k, l, m, n, \ldots)$ will generally represent natural numbers. Upper case letters and names will denote sets. The letter p will usually represent real numbers in the closed interval $[0, 1]$, and occasionally will denote a path of a tree.

The function f will range over all total recursive functions, and the function φ over all partial recursive functions. The graph of a function φ is the set of all pairs $\langle x, \varphi(x)\rangle$ for all x in the domain of φ. $f|_k$ is the finite initial segment of the graph of f consisting of $\langle 0, f(0)\rangle, \langle 1, f(1)\rangle, \ldots \langle k, f(k)\rangle$. $\mathcal{T}$ denotes the set of all total recursive functions. The set U will range over all subsets of $\mathcal{T}$. The set of all (not necessarily recursive) total functions from N into N is denoted by $\mathcal{H}$. We write $\varphi =^k f$ to indicate that $|\{x : \varphi(x) \neq f(x)\}| \leq k$. Similarly, $\varphi =^* f$ indicates that $\{x : \varphi(x) \neq f(x)\}$ is finite.

A Turing machine transducer is a machine which computes functions of one variable. We assume that a particular encoding of TM transducers as nonnegative integers has been chosen [24]. Hence the numbers $0, 1, 2, \ldots,$ are TM transducers, or ***programs***. (A number which is not the legitimate encoding of any program is viewed as a program computing the everywhere undefined function.) We denote the function computed by program i by φ_i. Thus $\langle \varphi_i \rangle_{i \in \mathrm{N}}$ is an acceptable numbering of all and only the partial recursive functions [26]. If $\varphi_i = f$, then we say that i is a *program index*, or simply an *index* of the function f.

If x is not in the domain of φ, then we say that φ, or any machine computing φ, *diverges* on input x and we write $\varphi(x) \uparrow$, otherwise $\varphi(x)$ is defined and we say $\varphi(x)$ converges and write $\varphi(x) \downarrow$. If $\varphi(x)$ is defined, and equals y we write $\varphi(x) \downarrow= y$. If $\varphi(x)$ is defined and not equal to y then we write $\varphi(x) \downarrow\neq y$.

If f is a total recursive function, then we define two sets, $GOOD_f$ and BAD_f which partition N as follows:

$$GOOD_f = \{i \mid \varphi_i = f\}.$$

$$BAD_f = \{i \mid \varphi_i \neq f\}.$$

We further partition BAD_f into two sets $WRONG_f$ and $SLOW_f$.

$$SLOW_f = \{i \mid \varphi_i \neq f, \text{ and } (\forall x)\ \varphi_i(x) \neq f(x) \Rightarrow \varphi_i(x)\uparrow\}.$$

$$WRONG_f = \{i \mid \varphi_i \neq f, (\exists x)\ \varphi_i(x) \downarrow\neq f(x)\}.$$

$GOOD_f$ is the set of "good" programs for f, BAD_f the set of "bad" programs, consisting of $SLOW_f$, the set of programs which are restrictions of f to some domain properly contained in N (thus wherever they differ from f they diverge, *i.e.* are slow), and $WRONG_f$, the set of programs which converge to a value "wrong" for f on at least one argument. Clearly $GOOD_f \uplus SLOW_f \uplus WRONG_f$ = N. Note that $WRONG_f$ is recursively enumerable.

2.2 Amalgamating Programs

Given a collection of programs, some of which compute a function f, it is sometimes useful to construct a new program from the given programs such that the constructed program computes the function f. We will use the following two program amalgamations in many of our constructions in later sections.

Definition 2.1 *Let I be a finite or recursively enumerable collection of program indices. Then define the program $RACE_I$ by:*

$RACE_I$

On input x, dovetail the computations $\{\varphi_i(x) \mid i \in I\}$ until for some j, the computation of $\varphi_j(x)$ halts with $\varphi_j(x) = y$. Output y.

Lemma 2.2 *If $I \bigcap GOOD_f \neq \emptyset = I \bigcap WRONG_f$ then $RACE_I$ computes f.*

Proof: Since $I \bigcap WRONG_f = \emptyset$ there does not exist an $i \in I$ such that $\varphi_i(x) \downarrow\neq f(x)$ for any x. Since $I \bigcap GOOD_f \neq \emptyset$ there does not exist an x such that $(\forall i \in I)\ \varphi_i(x)\uparrow$. Therefore, $(\forall x)(\exists i \in I)\ \varphi_i(x) \downarrow= f(x)$ and there is no $i \in I$ such that $\varphi_i(x) \downarrow\neq f(x)$. So $(\forall x)\ RACE_I(x) = f(x)$. □

Definition 2.3 *Let I be any finite ordered multiset of (not necessarily distinct) program indices* $I = \{i_1, i_2, \ldots, i_k\}$. *Let* $\vec{p} = p_1, p_2, \ldots, p_k$ *be any finite sequence of probabilities* ($p_i \in \mathrm{R}, \sum_{i=1}^{k} p_i \leq 1$), *and let t be any positive rational number. Then the program* $THRESHOLD_{t,I,\vec{p}}$ *is the program defined by:*

$THRESHOLD_{t,I,\vec{p}}$

> *On input x, dovetail the computations* $\{\varphi_i(x) \mid i \in I\}$ *until a number y and a multiset* $S \subseteq I$ *have been found such that* $(\forall i \in S)\ \varphi_i(x) = y$ *and* $\sum_{i_j \in S} p_j > t$. *Output y.*

Lemma 2.4 *If*

$$\sum_{i_j \in I \bigcap WRONG_f} p_j < t < \sum_{i_j \in I \bigcap GOOD_f} p_j$$

then $THRESHOLD_{t,I,\vec{p}}$ *computes f.*

Proof: If $THRESHOLD_{t,I,\vec{p}}(x) = y$ then $(\exists S)(\forall i \in S)\ \varphi_i(x) = y$ and $\sum_{i_j \in S} p_j > t$. If $y \neq f(x)$ then $S \subseteq WRONG_f$ and

$$\sum_{i_j \in I \bigcap WRONG_f} p_j \geq \sum_{i_j \in S} p_j > t,$$

a contradiction. Hence if $THRESHOLD_{t,I,\vec{p}}(x)$ is defined, then it is correct. To see that $THRESHOLD_{t,I,\vec{p}}(x)$ is in fact total, observe that $(\forall x)$ the set $S = I \bigcap GOOD_f$ satisfies the dovetail halting condition of $THRESHOLD_{t,I,\vec{p}}(x)$. Since S is finite, after some finite amount of computation $THRESHOLD_{t,I,\vec{p}}$ will witness this and halt with a value. □

2.3 Inductive Inference Machines

An *inductive inference machine* (IIM) is a machine which attempts to synthesize programs computing a function φ, when presented only with the graph of φ [23]. We adopt the definition of L. Blum and M. Blum [7]:

> An *inductive inference machine* is an algorithmic device, or Turing machine that works as follows. First the machine is put in some initial state with its tape memory completely blank. From there it proceeds algorithmically except that, from time to time, the device requests an input or produces an output. Each time it requests an input, an external agency feeds the machine a pair of natural numbers $\langle x, y \rangle$, or a $*$, and then returns control to the

machine. ...The outputs produced by the machine are all natural numbers [and represent M's guess for a program index of the function whose values it receives].

The $*$'s in the input to the IIM are to model "gaps" in the domain of the function to be inferred, and allow the external agency to feed partial functions to the IIM. Throughout this work we deal only with the inference of total recursive functions (with the exception of Section 4.6). We claim without guarantee that most of our arguments need only minor modification to cover the case of inference of partial recursive functions. Note that the assumption that the functions being inferred are total does not require that an IIM may hypothesize only programs computing total functions. (We discuss just such a restriction in Section 4.5.) Indeed, it is the possibility that the IIM outputs incomplete hypotheses which make many of our arguments difficult. (We should note that we drop the "total hypotheses only" restriction not because it increases the complexity of our proofs, but because the restriction significantly limits the power of the inference machine[9,10].)

For any (not necessarily recursive) total function $h \in \mathcal{H}$, let $\mathcal{G}(h)$ be the set of all infinite sequences s containing all and only the elements of the graph of h (each element of the graph of h may appear in s even infinitely many times), and let $\mathcal{G}(\mathcal{H}) = \bigcup_{h \in \mathcal{H}} \mathcal{G}(h)$. Then for any recursive function $f \in \mathcal{T}$, $\mathcal{G}(f)$ consists of all ***admissible presentations*** of f to some IIM [1]. Note that the sequences s of $\mathcal{G}(f)$ are not allowed to contain any $*$'s, since the functions to be inferred are all total functions.

If $s \in \mathcal{G}(\mathcal{H})$ then we write $M(s)$ to denote the (possibly infinite) sequence of outputs of M when fed s. Then (as defined in Section 2.1 for arbitrary sequences) $M(s)|_k$ is the first k elements of $M(s)$ if $|M(s)| \geq k$, and $M(s)$ otherwise.

2.4 Identification Criteria and Computational Models

Given a function $f \in \mathcal{T}$ and a presentation $s \in \mathcal{G}(f)$ of the function as input to an IIM M, there are a number of definitions for what constitutes a correct inference of f by M on s. Typically, most definitions, or "identification criteria" are predicates describing which output sequences $M(s)$ correspond to a correct inference of f. For example, a natural definition of successful inference might be that "M correctly infers f iff for any $s \in \mathcal{G}(f)$, $|M(s)| = 1$ (*i.e.* M on seeing the graph of f in any order outputs exactly one guess), and the single element of $M(s)$ is a program index for f. (This definition is considered in Section 4.3.2.) Alternatively, we might allow M on input $s \in \mathcal{G}(f)$ to output several hypotheses, changing its mind at most a finite number of times as it sees new values of f, as long as "eventually" it outputs a final correct hypothesis. (Thus $M(s)$ is finite, with last element a program index for f.) This is essentially the *EX* identification criterion which we will define later in this section. In attempting to define a probabilistic model of computation for inductive inference, we found that for some identification criteria it was not at all clear what the appropriate definition ought to be. Part of the confusion was due to the fact that "identification criterion" is not a well defined term in the literature. Consider the following example.

C. Smith [35] defines the class of sets of functions inferable by teams of machines for any identification criterion I by: $I_{team}(n) = \{U \mid (\exists M_1, M_2, \ldots, M_n)(\forall f \in U)(\exists i)\ M_i$ identifies f according to criterion $I\}$. This natural definition causes no problems for those criteria I that were investigated, but consider the general structure of the definition of the criterion *PEX* which we examine in Section 4.5:

An IIM is ***Popperian*** iff for any $s \in \mathcal{G}(\mathcal{H})$, the sequence of guesses $M(s)$ consists only of indices of total functions. Then M identifies f according to the *PEX* criterion iff M is Popperian, and for every $s \in \mathcal{G}(f)$, the sequence $M(s)$ satisfies property *EX*.

Now naïvely applying the definition of team inference above, we obtain $PEX_{team}(n) = \{U \mid (\exists M_1, M_2, \ldots, M_n)(\forall f \in U)(\exists i)\ M_i$ is Popperian and for every $s \in \mathcal{G}(f)$, $M_i(s)$ satisfies property $EX.\}$

Although this class might be of interest, we doubt that it is what the average inductive inference researcher would come up with were he placed in a room together with the definition of *PEX* and the notion of a team of IIMs. We believe the natural (and intended) definition to be the same as above, except that every machine M_i of the team must be Popperian. (One motivation for studying teams is to examine finite unions – modeled by our definition, and not the one above.)

Part of the confusion seems to be that the notion of identification criterion was mixed together with that of restricting the class of machines under consideration. In the example above, the identification criterion *PEX* is essentially a restriction of the class of IIMs under consideration (*i.e.* those which output only indices of total programs), together with a definition for correct identification (namely that the output sequence $M(s)$ satisfy property *EX*.) Since we are interested in examining many different identification criteria, and several different models of computation (deterministic, probabilistic, nondeterministic, teams) we find it useful to define abstractly the notion of an identification criterion and the effect the use of different computational models has when applied to the criterion.

Our general definition for an identification criterion will separate the two concepts of restricting the class of machines under consideration, and demanding some relationship between $M(s)$ $(s \in \mathcal{G}(f))$ and the input function f.

To allow an identification criterion to restrict the class of machines which can correctly infer a function f, we need to define the input–output sequence of a machine M on any given sequence of inputs.

Definition 2.5 *Let M be an IIM (possibly with access to an arbitrary set as oracle), and let $s \in \mathcal{G}(\mathcal{H})$. Then the sequence $IO_{M,s}$ consists of elements of $\mathrm{N} \times \{input, output\}$ and is a record of the input–output behavior of M when fed the sequence s as input. $IO_{M,s}$ is defined by running M forever with input s, and if whenever M queries for the next value (say it's s_k) from the sequence s, then the next element of $IO_{M,s}$ is $\langle s_k, input\rangle$; if instead, M next outputs a guess g, then the next element of $IO_{M,s}$ is $\langle g, output\rangle$.*

Thus the sequence $IO_{M,s}$ consists of the (possibly infinite) sequence $M(s)$, and (possibly only a finite initial segment of) the sequence s, interleaved in the exact order determined by M's sequence of queries and guesses.

Definition 2.6 *$IO = \bigcup IO_{M,s}$ where the union is taken over all sequences $s \in \mathcal{G}(\mathcal{H})$ and all IIMs (even those with an arbitrary set as oracle).*

Then a "behavioral restriction" on the class of IIMs is a (not necessarily recursive) predicate $\mathcal{B} : IO \rightarrow \{0,1\}$. For example, a behavioral restriction which captures the definition of a Popperian IIM is $(\forall s \in \mathcal{G}(\mathcal{H}))$ $\mathcal{B}(IO_{M,s}) = 1$ iff all "output" elements of the input–output sequence are indices of total programs. Note that Popperian IIMs must output total programs even when given as input the values of nonrecursive functions. There are other criteria (for example NV described in Section 4.5) where the behavior of the IIM on nonrecursive inputs is relevant.

Alongside the behavioral restriction, the other component of an identification criterion is the description of successful inference. For $s \in \mathcal{G}(f)$ input to M, since the sequence of outputs $M(s)$ may be infinite, typically the definition of successful inference is a limiting one — that is, "in the limit" the sequence $M(s)$ satisfies some predicate relative to f. Thus in the definition below, successful inference is defined partly by the limit of a predicate Q which depends on initial segments of $M(s)$, and the function f. For example, in Definition 2.12 the criterion EX specifies that a correct inference of

M on $s \in \mathcal{G}(f)$ occurs whenever the infinite sequence $M(s)$ consists of a single correct index for f, except for perhaps at a finite number of places. This is expressed as the limit (as $k \to \infty$) of a predicate Q_{EX} which equals 1 whenever the last two elements of $M(s)|_k$ are both equal to some single program index for f. Thus identification occurs iff $\lim_{k\to\infty} Q_{EX}(M(s)|_k)$ exists and $= 1$, *i.e.* the sequence converges to a single correct program index.

We are now ready to give a general description of an arbitrary identification criterion.

Definition 2.7

- *An identification criterion ID is a pair* (B_{ID}, Q_{ID}) *where* $B_{ID} : IO \to \{0,1\}$ *is a "behavioral restriction"– a (not necessarily recursive) predicate characterizing the types of IIMs (by behavior) under consideration. (Usually* B_{ID} *is identically equal to* 1 — *i.e. there is no restriction on the behavior of the machine.)* Q_{ID} *is a (not necessarily recursive) predicate* $Q_{ID} : \mathrm{N}^* \times \mathcal{T} \to \{0,1\}$.
- *M ID–identifies f iff* $(\forall s \in \mathcal{G}(\mathcal{X}))\ B_{ID}(IO_{M,s}) = 1$ *and* $(\forall s \in \mathcal{G}(f))\ \lim_{k\to\infty} Q_{ID}(M(s)|_k, f)$ *exists and* $= 1$.
- $ID(M) = \{f \mid M$ *ID–identifies* $f\}$.
- *M ID–identifies U iff* $U \subseteq ID(M)$.
- $ID = \{U \mid (\exists M)\ M$ *ID–identifies* $U\}$.

From here on, we write $B_{ID}(M) = 1$ to indicate that $(\forall s \in \mathcal{G}(\mathcal{X}))\ B_{ID}(IO_{M,s}) = 1$. This notation is adopted to indicate that B_{ID} is essentially a predicate on IIMs (its definition depends only on M). This will not cause any problems as long as we avoid predicates which depend on more than simply the input/output behavior of machines.

Recent work of Daley and Smith [15], and Freivald [18], has concerned the definition of the complexity of an inference. One such complexity measure is the number of mind changes made by an IIM en route to outputting a final hypothesis. There is no difficulty in capturing this definition with our general definition of an identification criterion. (We do so in Section 4.3.1.) We note however that there does not seem to be a straightforward way to incorporate the more general definition of the complexity of an inference given in [15]. This of course simply means that neither of the two definitions subsume the other, not that adopting one precludes the use of the other.

We now show that for many identification criteria, we may assume without loss of generality that all IIMs output infinitely many guesses, and that the sequence of outputs is independent of the particular order of the inputs.

Definition 2.8 ([10]) *An IIM M is* order independent[1] *iff* $(\forall h \in \mathcal{H})$ $s_1, s_2 \in \mathcal{G}(h) \Rightarrow M(s_1) = M(s_2)$.

Definition 2.9 *Let g be any finite or infinite sequence. Then the sequence g' is a* repetition variant *of g iff either*

- $|g| = k < \infty$ *and there is a sequence r of length $k-1$ of positive integers such that the sequence g' is the sequence obtained by repeating, in place, every element g_i of g exactly r_i times, $(1 \leq i \leq k-1)$ and repeating the element g_k infinitely many times, OR*
- *g is infinite and there is an infinite sequence r of positive integers such that g' is the sequence obtained by repeating, in place, every element g_k of g exactly r_k times.*

Definition 2.10 *A predicate $Q_{ID} : \mathrm{N}^* \times \mathcal{T} \rightarrow \{0,1\}$ is* limiting–invariant under repetition *iff the limit of Q_{ID} on initial segments of sequences is invariant to repetitions within the sequence. That is, if g is any sequence, and g' is a repetition variant of g, then the limits $\lim_{k\rightarrow\infty} Q_{ID}(g|_k, f)$ and $\lim_{k\rightarrow\infty} Q_{ID}(g'|_k, f)$ either both exist, or both do not exist, and if they exist, they are equal.*

Theorem 2.11 *Let ID be any identification criterion such that Q_{ID} is limiting–invariant under repetition. Then for all IIMs M there is an IIM M', uniform in M such that M' is order independent, and, regardless of the input, outputs infinitely many values (or none at all) and $ID(M) \subseteq ID(M')$.*

Proof: M' requests inputs, saves them, and simulates M on the inputs presented in the canonical order $\langle 0, f(0)\rangle, \langle 1, f(1)\rangle, \langle 2, f(2)\rangle, \ldots$. After each step of simulation, if M does not output a new guess, then M' outputs the last guess that M has output. Clearly M' on any input outputs infinitely many values (or none if M never outputs a value), and M' is order independent since its behavior does not depend on the order of input. To see that $ID(M) \subseteq ID(M')$, observe that if $ID(M)$ is nonempty then for every s, $B_{ID}(IO_{M,s}) = 1$ and since $IO_{M',s} = IO_{M,s'}$ for some s' (which is s in canonical order and without repeats), for every s, $B_{ID}(IO_{M',s}) = 1$. Further, if $f \in ID(M)$, then $(\forall s \in \mathcal{G}(f)) \lim_{k\rightarrow\infty} Q_{ID}(M(s)|_k, f)$ exists and $= 1$, and since for any $s \in \mathcal{G}(f)$, $M'(s)$ feeds the reordered sequence $s' \in \mathcal{G}(f)$ to M, $M'(s)$ is a repetition variant of $M(s')$,

$$\lim_{k\rightarrow\infty} Q_{ID}(M'(s)|_k, f) = \lim_{k\rightarrow\infty} Q_{ID}(M(s')|_k, f) = 1.$$

[1]This is different than a type of order independence of [7] which covers behavior on partial functions.

Thus we assume (without loss of generality when Q_{ID} is limiting–invariant under repetition) that any IIM outputs the same infinite sequence of guesses when given any presentation $s \in \mathcal{G}(h)$ for any $h \in \mathcal{H}$. We denote this unique sequence of outputs by $M(h)$. Usually, we will be dealing with recursive functions f, so we will write $M(f)$. Then (as defined in Section 2.1 for arbitrary sequences) $M(f)|_k$ is the first k elements of $M(f)$ if $|M(f)| \geq k$, and $M(f)$ otherwise. Furthermore, we shall assume for many of our arguments that any function given as input is presented in the canonical order $\langle 0, f(0)\rangle, \langle 1, f(1)\rangle, \langle 2, f(2)\rangle, \ldots$. We write "$M(f|_k)$," or "$M$ on $f|_k$" to denote the (possibly infinite) sequence of outputs obtained by running M with input f until (if ever) M asks for the $k+1^{st}$ pair $\langle k+1, f(k+1)\rangle$.

Two common and natural identification criteria are EX and BC.

Definition 2.12 *EX is the identification criterion defined by the pair (B_{EX}, Q_{EX}) where B_{EX} is always 1, and if g is a sequence with $|g| = k > 1$ then $Q_{EX}(g, f) = 1 \Leftrightarrow g_k = g_{k-1}$ and $g_k \in GOOD_f$; if $|g| = 1$ then $Q_{EX}(g, f) = 1 \Leftrightarrow g_1 \in GOOD_f$.*

Definition 2.13 *BC is the identification criterion defined by the pair (B_{BC}, Q_{BC}) where B_{BC} is always 1, and if g is a sequence with $|g| = k$ then $Q_{BC}(g, f) = 1 \Leftrightarrow g_k \in GOOD_f$.*

Thus the BC criterion requires that all guesses of M be correct past some finite initial number of incorrect guesses, whereas the EX criterion requires in addition that eventually these correct guesses be identical. Clearly $EX \subseteq BC$, and the containment is proper [3,10]. Our definition of EX and BC above are not the standard definitions ([7,10]), but are equivalent. Clearly both Q_{EX} and Q_{BC} are limiting–invariant under repetition. The identification criteria with which we will mostly be concerned throughout this work are EX, BC, and their variants.

Now we carefully define Smith's notion of *team inference* with respect to our definition of an identification criterion. In this model, a team of IIM's $M_1, M_2, \ldots, M_n$ identifies the function f iff there is at least one i such that M_i identifies f. Each member of the team carries out a separate computation, and there need not be communication between team members. Identification by a team of n machines may be viewed as a kind of finite nondeterminism; after an initial n–way nondeterministic choice among the machines, the computation is deterministic. In chapter 3 we relate the (yet undefined) notion of probabilistic inference to team inference. We will argue in Section 4.2 that team inference seems to be the only natural definition of nondeterminism for inductive inference.

Definition 2.14 *Let $ID = (B_{ID}, Q_{ID})$ be any identification criterion, and let the team $\{M_1, M_2, \ldots, M_n\}$ be a collection of IIMs. Then*

- *The team $\{M_1, M_2, \ldots, M_n\}$ ID–identifies f iff*
  $(\forall i)\ B_{ID}(M_i) = 1$ *and* $(\exists i)\ \lim_{k\to\infty} Q_{ID}(M_i(f)|_k, f)$ *exists and* $= 1$.
- $ID(M_1, M_2, \ldots, M_n) = \{f \mid (M_1, M_2, \ldots, M_n)$ *ID–identifies* $f\}$.
- *The team $\{M_1, M_2, \ldots, M_n\}$ ID–identifies U iff $U \subseteq ID(M_1, M_2, \ldots, M_n)$.*
- $ID_{team}(n) = \{U \mid (\exists M_1, M_2, \ldots, M_n)\ U \subseteq ID(M_1, M_2, \ldots, M_n)\}$.

Note that in the definition above, we might have replaced $(\forall i)\ B_{ID}(M_i) = 1$ by $(\exists i)$ $B_{ID}(M_i) = 1$, and for *PEX* briefly mentioned above, we would have the "unnatural" definition. Since most of the identification criteria we examine will have B_{ID} identically $= 1$, this alternate definition of team inference is of no consequence. An interesting question which we do not explore is what is the structure of team inference with respect to the alternate definition.

Smith shows for both *EX* and *BC* that for all n, there are classes of functions identifiable by a team of $n+1$ machines, but not by any collection of n machines. This gives an infinite hierarchy of "inferability" [35]:

Theorem 2.15 $(\forall n) EX_{team}(n) \subset EX_{team}(n+1)$ *and* $BC_{team}(n) \subset BC_{team}(n+1)$.

In the following two sections we define probabilistic and nondeterministic models of computation for arbitrary identification criteria.

2.5 Probabilistic Inductive Inference Machines

We now define ***probabilistic*** IIMs and for all identification criteria *ID*, we define the probability that a given probabilistic IIM *ID*–identifies a given function.

A probabilistic IIM P is simply a deterministic IIM which is equipped with a special "t–sided coin oracle". The oracle O is an infinite sequence of integers $i_1, i_2, \ldots$ such that $(\forall j)\ i_j \in \{0, 1, \ldots, t-1\} = \mathrm{N}_t$. We denote the set of these infinite t–ary sequences by N_t^∞. (N_t^∞ is simply the infinite cartesian product of N_t with itself). The oracle sequence, or "coin flips" are printed on a semi-infinite read-only one way tape. When P is fed f, it may from time to time advance the read head on the coin tape to the next square and read the value of the next coin flip.

If we run P with oracle $\mathcal{O}$ on input f, then P behaves like a deterministic IIM, which we denote by $P^{\mathcal{O}}$. If *regardless of the oracle* $\mathcal{O}$, $P^{\mathcal{O}}$ satisfies the behavioral restriction (*i.e.* $(\forall \mathcal{O})\ B_{ID}(P^{\mathcal{O}}) = 1$), then we say that P satisfies B_{ID}. If P satisfies B_{ID}, and in addition, for some particular oracle $\mathcal{O}$ the sequence of guesses output corresponds to a deterministic ID–identification of f, we write $P^{\mathcal{O}}$ ID–identifies f. We denote the sequence of guesses output by P with oracle $\mathcal{O}$ by $P^{\mathcal{O}}(f)$. We would like to define the probability that P ID–identifies f as the probability (taken over all oracles $\mathcal{O}$) that $P^{\mathcal{O}}$ ID–identifies f. We first review the necessary probability theory.

2.5.1 Some Probability Theory

Much of the material in this section may be found in [29]. Intuitively, a probability measure is a function Pr which assigns "probabilities" (real numbers between 0 and 1) to outcomes of some experiment which is to be performed. The outcomes are elements of some universal set Ω. In practice, it is useful to have the probability defined not only on elements of Ω, but on subsets of Ω as well. A probability measure should satisfy axioms which we believe intuitive, for example, $\Pr[\Omega]$ should equal 1; For all $A \subseteq \Omega$, $\Pr[A]$ should be between 0 and 1, and Pr should be additive in the following sense: If A is the disjoint union of the finite or countable collection $\{A_i\}$, then $\Pr[A]$ should equal $\sum_i \Pr[A_i]$.

As it turns out, it may not always be possible, given a set Ω, to define a probability function on *all* subsets of Ω, in a way which is consistent with the situation we want to model. For example, it can be shown that there is no function defined on all subsets of the interval [0,1], which satisfies the above three properties, *and* is such that for any interval $(a,b) \subseteq [0,1]$, $\Pr[(a,b)] = b - a$ (length is the natural definition of probability for an interval in [0,1]). The reason for this is that there are many different, and often bizarre ways to express sets as partitions of other sets, and then deduce by the properties of a probability measure above, that Pr must be defined to be two different values for a set so constructed.

The approach generally taken then, is to carefully delineate the class of subsets for which the function Pr is to be defined, and then show that Pr is in fact well–defined on this family of sets. We need the notion of a Borel field.

Definition 2.16 *A family of subsets $\mathcal{B}$ of a given set Ω is a Borel field iff the following three conditions hold:*

1. $\Omega \in \mathcal{B}$

2. $A \in \mathcal{B} \Rightarrow \Omega - A \in \mathcal{B}$.

3. *If* $\{A_i\}_{i \in I}$ *is a finite or countable collection of elements of* $\mathcal{B}$, *then* $\bigcup_{i \in I} A_i \in \mathcal{B}$ *and* $\bigcap_{i \in I} A_i \in \mathcal{B}$.

It follows that any Borel field is closed under complementation, and countable unions and intersections.

Now if $\mathcal{C}$ is a collection of subsets of Ω, then there is a unique "smallest" Borel field, denoted $\mathcal{B}(\mathcal{C})$, which contains every element of $\mathcal{C}$, and is closed under finite and countable unions and intersections of elements of $\mathcal{C}$ and their complements. ("Smallest" is with respect to containment.)

For example, consider the real line R, and let $\mathcal{I}$ be the family of all of the open intervals of the form $(-\infty, w)$, for $w \in$ R. Then $\mathcal{B}(\mathcal{I})$ contains just about any set of real numbers imaginable; in fact, one has to be somewhat clever to come up with a subset of R which isn't in $\mathcal{B}(\mathcal{I})$.

Now given Ω and $\mathcal{B}$, we can define a probability measure on elements of $\mathcal{B}$, rather than on *all* subsets of Ω.

Definition 2.17 *A probability measure* Pr *on a Borel field* $\mathcal{B}$ *of subsets of* Ω *is a function* Pr:$\mathcal{B} \to$ R *such that*

1. $\Pr[\Omega] = 1$

2. $(\forall A)\ A \in \mathcal{B} \Rightarrow \Pr[A] \geq 0$

3. *If* $\{A_i\}$ *is a finite or countable collection of mutually disjoint elements of* $\mathcal{B}$, *then* $\Pr[\biguplus_i A_i] = \sum_i \Pr[A_i]$.

This last property is called *countable additivity*, and we will use it liberally. Many other properties of probability measures follow from the definition above. For example, *monotonicity*: If $A \subseteq B$ and both are in $\mathcal{B}$, then $\Pr[A] \leq \Pr[B]$. We call the elements of $\mathcal{B}$ the *measurable* sets.

If $\{A_i\}$ is a countable collection of sets, then

$$\limsup_{k \to \infty} A_k = \bigcap_{k=0}^{\infty} \bigcup_{i=k}^{\infty} A_i$$

$$\liminf_{k \to \infty} A_k = \bigcup_{k=0}^{\infty} \bigcap_{i=k}^{\infty} A_i$$

The two sets are, respectively, the limit supremum, and limit infimum of the sequence $\{A_i\}$, and correspond to, respectively, the set of elements which are in infinitely many of the sets $\{A_i\}$, and the set of elements which are in all but finitely many of the sets $\{A_i\}$. If for the sequence of sets $\{A_i\}$ we have that the limit supremum and limit infimum are equal, then this is the limit of the sequence, *i.e.*

$$\lim_{k\to\infty} A_k = \liminf_{k\to\infty} A_k = \limsup_{k\to\infty} A_k$$

A sequence of sets is *monotone* iff either $(\forall k)\ A_k \subseteq A_{k+1}$ or $(\forall k)\ A_{k+1} \subseteq A_k$. Every monotone sequence of sets has a limit, and every Borel field is closed under lim inf and lim sup. If $\{A_i\}$ is a sequence of measurable sets for which the limit is defined, then

$$\Pr[\lim_{k\to\infty} A_k] = \lim_{k\to\infty} \Pr[A_k].$$

Finally, we define

Definition 2.18 *A probability space is a triple* $(\Omega, \mathcal{B}, \Pr)$ *of a sample space of events* Ω, *a Borel field* $\mathcal{B}$ *of subsets of* Ω, *and a probability measure* $\Pr$ *on* $\mathcal{B}$.

Now given a probability space $(\Omega, \mathcal{B}, \Pr)$ and a set $A \subseteq \Omega$, to find $\Pr[A]$ we need only show that A is measurable by showing that it may be expressed by countable intersections, unions, and complements of some known measurable sets $\{A_i\}$ (for which the values $\Pr[A_i]$ are known), and then applying the properties (some shown above) of probability measures to compute $\Pr[A]$ from the values $\Pr[A_i]$.

2.5.2 Probability of Identification

We now define a probability measure on our coin-flip oracles. For any natural number t, define the probability space $(\mathrm{N}_t, \mathcal{B}_t, \Pr_t)$ where $\mathcal{B}_t = \{S \mid S \subseteq \mathrm{N}_t\}$, and $\Pr_t[S] = \frac{|S|}{t}$.

Now let N_t^k be the cartesian product of N_t with itself k times (which is simply the set of all t–ary sequences of length k), and recall that N_t^∞ is the set of all infinite t–ary sequences, or infinite cartesian product of N_t with itself. Let $\mathcal{B}_t^\infty$ be the smallest Borel field of subsets of N_t^∞ containing all of the sets

$$\mathrm{N}_t^{j-1} \times A_j \times \mathrm{N}_t^\infty$$

where $(\forall j)\ A_j \in \mathcal{B}_t$.

Then $(\mathrm{N}_t^\infty, \mathcal{B}_t^\infty, \Pr_t^\infty)$ is a probability space [29], with $\Pr_t^\infty$ the unique measure on $\mathcal{B}_t^\infty$ which $(\forall n > 0)$, and for all choices of n integers $0 < i_1 < i_1 \cdots < i_n$ satisfies

$$\Pr_t^\infty[A_{i_1,i_2,\ldots i_n}] = \prod_{j=1}^{n} \Pr_t[A_{i_j}]$$

for all sets $A_{i_1,i_2,\ldots i_n}$, where $A_{i_1,i_2,\ldots i_n}$ denotes the set

$$\mathrm{N}_t^{i_1-1} \times A_{i_1} \times \mathrm{N}_t^{i_2-i_1-1} \times A_{i_2} \times \mathrm{N}_t^{i_3-i_2-1} \times A_{i_3} \times \cdots \times A_{i_n} \times \mathrm{N}_t^{\infty}$$

with $A_i \in \mathcal{B}_t$.

If the value of t is fixed and is clear from context, we will write Pr instead of Pr_t^{∞}. The measurable sets $A_{i_1,i_2,\ldots i_n}$ are called *rectangles*, corresponding to the intuitive view of the product space as an infinite dimensional cube, and the sets A_i constraining a side of the cube to have measure less than 1. Note in particular that the sets A_j generating $\mathcal{B}_t^{\infty}$ are rectangles.

We note a few facts about our probability space. Let $\mathcal{O}$ be a sequence of t–ary flips (*i.e.* $\mathcal{O} \in \mathrm{N}_t^{\infty}$), let $\mathcal{O}_j$ denote the j^{th} element in the sequence, and let $\mathcal{O}|_j$ denote the finite sequence $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_j$. Then for all j, $0 \leq j \leq t-1$, and for all $n > 1$, the set $\{\mathcal{O} \mid \mathcal{O}_n = j\}$ is a rectangle (and therefore measurable), and $\mathrm{Pr}[\{\mathcal{O} \mid \mathcal{O}_n = j\}] = \frac{1}{t}$. If $s = s_1, s_2, s_3, \ldots, s_k$ is a finite t–ary sequence of length k, then we say that $\mathcal{O}$ *extends* s iff $\mathcal{O}|_k = s$. Then for any sequence s of length k, $\{\mathcal{O} \mid \mathcal{O} \text{ extends } s\}$ is a rectangle, and $\mathrm{Pr}[\{\mathcal{O} \mid \mathcal{O} \text{ extends } s\}] = \frac{1}{t^k}$.

Given a probabilistic IIM P, we would like to define Pr[P ID–identifies f] as $\mathrm{Pr}[\{\mathcal{O} \mid P^{\mathcal{O}}\, ID\text{–identifies } f\}]$. The problem is that the set may not be measurable. We show that it always is.

Lemma 2.19 *If $ID = (B_{ID}, Q_{ID})$ is any identification criterion, P a probabilistic IIM which satisfies B_{ID}, and f a total recursive function, then $\{\mathcal{O} \mid P^{\mathcal{O}}\, ID\text{–identifies } f\}$ is a measurable set.*

Proof: Let $P^{\mathcal{O}}(f)|_j$ be defined as in Section 2.3. Then

$$\begin{aligned}
\{\mathcal{O} \mid P^{\mathcal{O}}\, ID\text{–identifies } f\} &= \{\mathcal{O} \mid \lim_{j\to\infty} Q_{ID}(P^{\mathcal{O}}(f)|_j, f) = 1\} \\
&= \{\mathcal{O} \mid (\exists k)(\forall j \geq k)\; Q_{ID}(P^{\mathcal{O}}(f)|_j, f) = 1\} \\
&= \bigcup_{k=1}^{\infty} \{\mathcal{O} \mid (\forall j \geq k)\; Q_{ID}(P^{\mathcal{O}}(f)|_j, f) = 1\} \\
&= \bigcup_{k=1}^{\infty} \bigcap_{j=k}^{\infty} \{\mathcal{O} \mid Q_{ID}(P^{\mathcal{O}}(f)|_j, f) = 1\} \\
&= \bigcup_{k=1}^{\infty} \bigcap_{j=k}^{\infty} \bigcup_{\substack{s \in \mathrm{N}_t^j \\ Q_{ID}(P^s(f), f) = 1}} \{\mathcal{O} \mid \mathcal{O} \text{ extends } s\}
\end{aligned}$$

Since the sets $\{\mathcal{O} \mid \mathcal{O} \text{ extends } s\}$ are rectangles, they are measurable, and we have thus shown that $\{\mathcal{O} \mid P^{\mathcal{O}}\ ID\text{–identifies } f\}$ is expressible as countable unions and intersections of measurable sets, and is therefore measurable. ☐

Definition 2.20 *Let ID be an identification criterion, and let P be a probabilistic IIM with a t–sided coin* ($t \geq 2$) *which satisfies* B_{ID}. *Then for any total recursive function f,*

$$\Pr_t^\infty[P\ ID\text{–identifies } f] = \Pr_t^\infty[\{\mathcal{O} \mid P^{\mathcal{O}}\ ID\text{–identifies } f\}]$$

We now show that without loss of generality, we may assume that all probabilistic IIMs have a two–sided coin.

Lemma 2.21 *Let P be a probabilistic IIM with at t–sided coin. Then there exists a probabilistic IIM* P_1 *with a 2–sided coin such that* $(\forall f)\ \Pr_2^\infty[P_1\ ID\text{–identifies } f] = \Pr_t^\infty[P\ ID\text{–identifies } f]$.

Proof: P_1 on input f simulates P. P_1 obtains the next value of f whenever P was about to, and P_1 outputs any guess whenever P does. The only problem occurs each time P wishes to flip its t–sided coin. P_1 simulates the t–sided coin flip by flipping the 2–sided coin t times, and if the coin comes up "heads" for exactly one of the flips, then P_1 feeds the appropriate value to P. If this doesn't occur, then P_1 continues flipping "blocks" of t flips until the first block is encountered such that there is exactly one head in the block. P_1 iterates this procedure whenever P attempts to flip its coin (read the next value on its oracle tape).

We make this procedure more precise. Let $\mathcal{O}$ be an infinite binary sequence $b = b_0, b_1, \ldots$. Define the k^{th} *block of* b to be the subsequence $b_{(k-1)t}, b_{(k-1)t+1}, \ldots, b_{(k-1)+t-1}$. The k^{th} block is *good* iff there is exactly one $i \in \{0, 1, \ldots, t-1\}$ such that $b_{(k-1)t+i} = 1$ (heads). Then $P_1^{\mathcal{O}}$ constructs the (possibly finite) t–ary sequence $c = c_0, c_1, \ldots$ such that $c_n = j$ $(0 \leq j \leq t-1)$ iff there are at least n good blocks, and if the n^{th} good block is block k, then $b_{(k-1)t}, \ldots, b_{(k-1)t+j}, \ldots, b_{(k-1)+t-1} = 0, \ldots, 1, \ldots, 0$.

Now $\Pr_2^\infty[\{\mathcal{O} \mid \text{the sequence constructed by } P_1 \text{ (in the limit) is finite}\}] = 0$, since it is the infinite product of rectangles with probability < 1, and by symmetry, the probability that $c_k = j$ is $\frac{1}{t}$ for each j, $0 \leq j \leq t-1$. Thus P_1 is able to construct (albeit slowly) an oracle for P, and successfully simulate P. ☐

Since the "number of sides" of the coin is irrelevant, we will use whatever value of t is most convenient for the argument which we are presenting. We let $t = 2$ unless otherwise specified.

Definition 2.22 *Let ID be an identification criterion, P a probabilistic IIM which satisfies B_{ID}, and f a total recursive function. Then*

- *P ID–identifies f with probability p iff* $\Pr[P$ *ID–identifies* $f] \geq p$.
- $ID_p(P) = \{f \mid P$ *ID–identifies f with probability* $p\}$.
- *P ID–identifies U with probability p iff* $U \subseteq ID_p(P)$.
- $ID_{prob}(p) = \{U \mid (\exists P)\ U \subseteq ID_p(P)\}$.

Note that our definition requires that given *any* oracle $\mathcal{O}$, $P^{\mathcal{O}}$ satisfies the behavioral restriction. As with team identification, interesting questions arise when this requirement is weakened to allow $P^{\mathcal{O}}$ to violate B_{ID} whenever the sequence of outputs $P^{\mathcal{O}}(f)$ didn't satisfy the predicate Q_{ID}. As mentioned when discussing the definition of team inference, for most of our work this difference is of no consequence since in most cases we will have that B_{ID} is identically 1.

Theorem 2.23 *If ID is any identification criterion then* $(\forall n \geq 1)\ ID_{team}(n) \subseteq ID_{prob}(\frac{1}{n})$.

Proof: If $U \in ID_{team}(n)$ then $(\exists M_1, M_2, \ldots, M_n)$ such that $(\forall i)\ B_{ID}(M_i) = 1$ and $(\forall f \in U)(\exists i)\ M_i$ *ID*–identifies f. Let P be the probabilistic IIM which has an n–sided coin, which it flips once and obtains with probability $\frac{1}{n}$ a particular value $i \in \{0, 1, \ldots, n-1\}$. P then simulates machine $i+1$. Clearly P satisfies B_{ID} and $(\forall f \in U)$ $\Pr[P$ *ID*–identifies $f] \geq \frac{1}{n}$. □

Theorem 2.23 shows that for any identification criterion, a probabilistic IIM can simulate (with success $\frac{1}{n}$) a team of n IIMs. Much of Chapter 3 involves showing that teams of IIMs can simulate probabilistic IIMs.

Corollary 2.24 $(\forall n \geq 1)\ EX_{team}(n) \subseteq EX_{prob}(\frac{1}{n})$ *and* $BC_{team}(n) \subseteq BC_{prob}(\frac{1}{n})$.

We say that a probabilistic IIM P *behaves nicely* iff P obeys the following computation sequence:

1. Receive a value of f
2. Guess a program index
3. Flip a 2–sided coin
4. Execute a finite number of deterministic steps
5. Go to Step 1.

It will be convenient for many of our arguments to assume that all probabilistic IIMs behave nicely. We now show that we may do this without any loss of generality for many types of identification criteria.

Lemma 2.25 *Let P be a probabilistic IIM and ID an identification criterion such that Q_{ID} is limiting–invariant under repetition. Then there is a probabilistic IIM P_1 which behaves nicely and such that* $(\forall f)$ $\Pr[P$ *ID–identifies* $f] = \Pr[P_1$ *ID–identifies* $f]$.

Proof: By Lemma 2.21 we may assume that P flips a two–sided coin. Thus we only need to construct a machine P_1 which behaves nicely and for all f is correct with the same probability as P. P_1 simulates P, and P_1 must be sure to obey the required computation sequence (receive value – guess program – flip coin – compute – repeat). This is easily accomplished. P_1 uses three tapes as buffers to store inputs to and outputs from its simulated computation of P. The three buffers are: a coin buffer to which P_1 writes and from which P reads coin flips; a value buffer, on which P_1 writes the sequence of values of f which it receives during its computation, and from which P reads, and a guess buffer on which P writes its guesses which P_1 saves and uses from time to time. Now P_1 begins to execute the above computation sequence. Whenever it receives a value for f, it writes it on the value buffer. Whenever it is time (according to the above sequence) to make a new guess, P_1 outputs the last value written to the guess buffer by P. Whenever it is time to flip the coin, P_1 writes the result of the coin flip to the coin buffer, and whenever it is time to compute deterministically, P_1 simulates P until either P outputs a new guess, or is awaiting the next (not yet written) value from either the coin or value buffer. P_1 then returns to step 1. of the computation sequence.

Clearly the sequence of outputs of P_1 with oracle $\mathcal{O}$ and input f is a repetition variant of the sequence which P outputs with oracle $\mathcal{O}$ and input f. Now since Q_{ID} is limiting–invariant under repetition, we conclude that $\Pr[P_1^{\mathcal{O}}\ ID\text{–identifies } f] = \Pr[P^{\mathcal{O}}\ ID\text{–identifies } f]$, completing the proof of Lemma 2.25. □

Since Q_{EX} and Q_{BC} are limiting–invariant under repetition, we assume without loss of generality that all IIMs *EX*– and *BC*–identifying sets of functions behave nicely. Whenever we introduce a new identification criterion *ID*, we immediately have that Theorem 2.23 applies. Furthermore, if Q_{ID} is limiting–invariant under repetition we will assume with impunity that all IIMs under consideration behave nicely.

2.5.3 Infinite Computation Trees

For the sake of intuition, as well as cleaner arguments, for any probabilistic IIM P and any function f, we define an infinite complete binary "computation tree" $T_{P,f}$ which represents all of the possible computations of P with input f. (Each determined by the particular sequence of coin flips defining the coin oracle).

The nodes of $T_{P,f}$ will correspond to *configurations* of P and the edges will correspond to the results of coin flips. (A *configuration* is a structure which specifies the state of P, the contents of all of its tapes, and the positions of all of its read and write heads [24].)

In particular, the root node will correspond to the configuration of P immediately after P makes its first guess. Since P behaves nicely, P's next step after this guess will be to flip the coin. The left edge leaving the root node will correspond to a coin flip which comes up "heads", the right edge "tails". After an initial guess and a coin flip, P, (according to the niceness conventions) executes a finite number of transitions, receives the next value of f and then guesses again. The left child of the root node will correspond to the configuration that P reaches just after it makes its second guess, given that the first flip was heads. In general, a node of depth d in $T_{P,f}$ will correspond to the configuration of P reached if P were to run through d iterations of the read–guess–flip–compute loop and the sequence of d coin flips that P received was exactly the sequence of heads and tails which lead to node n in the tree.

The nodes of $T_{P,f}$ are numbered in breadth first search order (across levels left to right, starting with the root node, which is numbered '1'). The depth of a node n in $T_{P,f}$ is denoted $d(n)$, where $d(n) = \lfloor \log_2 n \rfloor$. (Hence node 1 has depth 0, nodes 2 and 3 have depth 1, etc). *Parent*(n) denotes the immediate ancestor of node n in $T_{P,f}$. When we write "n", we sometimes are referring to the node numbered n, or to the value n itself, the meaning will be clear from context. Finally, we define the labeling function ind:N $\rightarrow$ N on the nodes of $T_{P,f}$ by: $ind(n) =$ the guess that P has just output when it is in the configuration corresponding to node n. If $ind(n) = j$, then we say that j is the *index* of node n, to indicate that j is P's guess for a program index for f. Note that for any probabilistic IIM P, any function f, and any number k, there is a Turing machine which when fed the first k values of f, and the description of P, constructs $T_{P,f}$ through the k^{th} level.

A *path* p of $T_{P,f}$ is an infinite sequence of adjacent nodes $\langle t_0, t_1, \ldots, \rangle$, starting at the root node ($t_0 = 1$), and going "down the tree, never changing directions", so that for all i, the i^{th} node t_i on p is a node occurring at depth i of $T_{P,f}$. We observe that there is an isomorphism between the set of coin oracles and the set of paths in $T_{P,f}$, (namely each path corresponds to a particular coin oracle), and the sequence of guesses along path p is exactly the sequence $P^{\mathcal{O}}(f)$ where $\mathcal{O}$ corresponds to the coin flips along p.

2.5.4 Probability on Infinite Computation Trees

As we have observed, the tree $T_{P,f}$ is a structure which nicely represents all of P's possible computations, and a particular path p in $T_{P,f}$ corresponds exactly to an oracle sequence, together with the corresponding infinite sequence of outputs $P^{\mathcal{O}}(f)$. Thus the function Pr defined above can be extended to sets of paths S in $T_{P,f}$ by $\Pr[S] = \Pr[\{\mathcal{O} \mid \mathcal{O}$ corresponds to a path $p \in S\}]$. We pay particular attention to some measurable sets of paths which we will find useful.

Definition 2.26 *For each node* $n \in T_{P,f}$, $P_n = \{\text{paths } p \in T_{P,f} \mid p \textit{ contains node } n\}$.

Clearly $\Pr[P_n] = \frac{1}{2^{d(n)}}$ since P_n corresponds to the set of oracles $\{\mathcal{O} \mid \mathcal{O}$ extends $s\}$ where s is the finite path segment of length $d(n)$ which leads to node n. This makes sense, since if we think of P as flipping coins and following down some path in $T_{P,f}$, P must pass through exactly one node at depth (or *level*) $d(n)$, and these should be equiprobable. The sets $\{P_n\}$ are (correspond to) rectangles, and we use them to construct other measurable sets of paths in Sections 3.1.1 and 3.2.1.

2.6 Nondeterministic Inductive Inference Machines

We briefly define a nondeterministic model of computation for any identification criterion ID.

A nondeterministic IIM is exactly a probabilistic IIM; *i.e.* it is a deterministic IIM which has access to an oracle $\mathcal{O}$. The only difference is the definition of successful nondeterministic identification:

Definition 2.27 *Let N be a nondeterministic IIM, ID be any identification criterion, and f any total recursive function. Then*

- *N nondeterministically ID–identifies f iff N satisfies B_{ID} (i.e. for every oracle $\mathcal{O}$ $B_{ID}(N^{\mathcal{O}}) = 1$) and $(\exists \mathcal{O})$ $N^{\mathcal{O}}$ ID–identifies f (i.e. $\lim_{k\to\infty} Q_{ID}(N^{\mathcal{O}}(f)|_k, f) = 1$).*
- *N nondeterministically ID–identifies U iff $(\forall f \in U)$ N nondeterministically ID–identifies f.*
- *$ID_{nondet} = \{U \mid (\exists N)\ N$ nondeterministically identifies $U\}$.*

For example, the class EX_{nondet} consists of those sets of functions for which there is a single nondeterministic IIM identifying every function in the set. It is easy to show that the class of all partial recursive functions is contained in EX_{nondet}. We will discuss this in greater detail in Section 4.2.

Chapter 3

Probability, Teams, EX, and BC

In this chapter we examine the relationship between team and probabilistic inference strategies for the most natural definitions of successful inference. We will demonstrate for *EX* and *BC* that there is an infinite hierarchy of probabilistic inference classes, and that this hierarchy is identical to the hierarchy of team inference shown in [35], and given by Theorem 2.15. This is achieved by showing that teams of IIMs can "simulate" probabilistic IIMs, and combining these results with Corollary 2.24. Section 3.1 proves the main results for the *BC* identification criterion, and Section 3.2 for *EX*. We put all of these results together in Section 3.3 and examine the probabilistic identification hierarchies in light of the relationship between teams and probability. In Section 3.4 we introduce Podnieks' "frequency" identification for *EX* and *BC* and relate frequency to probability and teams. In Section 3.5 we consider different definitions of probabilistic *EX* and *BC* identification and show that they define the same classes. Finally we consider other aspects of probabilistic inference for *EX* and *BC* in Section 3.6.

3.1 BC Probability and Teams

3.1.1 BC Convergence in $T_{P,f}$

We will carry out our arguments using the computation trees $T_{P,f}$. We define some useful sets of paths in these trees, along with some helpful lemmas.

Definition 3.1 *Let $p = \langle t_0, t_1, \ldots, \rangle$, be a path in $T_{P,f}$, and $A \subseteq \mathrm{N}$. The path p BC–converges to A iff $(\forall_k^\infty)$ $ind(t_k) \in A$.*

If path p *BC*–converges to A, then p corresponds to a possible computation of P with input f, for which P, after some initial sequence of guesses, outputs only indices from the set A.

Path $p = \langle t_0, t_1, \ldots, \rangle$ *BC–converges to A at node n* iff

- p passes through node n. ($t_{d(n)} = n$)
- p BC–converges to A.
- $(\forall k \geq d(n))\ ind(t_k) \in A$.
- there does not exist $k < d(n)$ such that $(\forall m \geq k)\ ind(t_m) \in A$.

This simply requires that on path p, all nodes from n and beyond have index in the set A, and node n is the least depth at which this convergence occurs.

If a path p BC–converges to the set $GOOD_f$, then p contains a sequence of coin flips which causes P to output a sequence of guesses corresponding to a single deterministic BC–identification of f.

Definition 3.2

- $B(A) = \{p \mid p$ *is a path in* $T_{P,f}$*, and* p *BC–converges to* $A\}$.
- $B_j(A) = \{p \mid p$ *is a path in* $T_{P,f}$*, and* p *BC–converges to* A *at node* $j\}$.

Note that for all $j \neq m$, and A, $B_j(A) \bigcap B_m(A) = \emptyset$. If neither j nor m is an ancestor of the other, then no paths pass through both. If one is the ancestor of the other, then any path which BC–converges to A must, by definition, converge at *exactly* one node.

We say that a path $p = \langle t_0, t_1, \ldots, \rangle$ is ***k–consistent*** with $B_j(A)$ iff the following three conditions hold:

1. $t_{d(j)} = j$ (the path passes through node j).
2. $(\forall i)\ d(j) \leq i \leq k \Rightarrow ind(t_i) \in A$.
3. j is the root OR $ind(t_{j-1}) \notin A$.

Definition 3.3 $B_{j,k}(A) = \{p \mid p$ *is k–consistent with* $B_j(A)\}$.

Intuitively, $B_{j,k}(A)$ is the set of paths p such that if we examine the nodes on p only through depth k, p seems to be a path in $B_j(A)$. Another way of stating this is that it is not possible to deduce that p is not in $B_j(A)$ from looking only at the first k levels of the tree.

Clearly, $B(A) = \biguplus_{j \in \mathrm{N}} B_j(A)$. It is also true that $B_j(A) = \bigcap_{k=d(j)}^{\infty} B_{j,k}(A)$. Thus to show $B(A)$ measurable for all A, we need only show that $B_{j,k}(A)$ is measurable for all $j, k \geq d(j)$, and A. We will express $B_{j,k}(A)$ using the rectangles $\{P_n\}$ (Definition 2.26) but we will first need the following definition.

Definition 3.4 $N_{j,k}(A) = \{n \mid d(n) = k,$ *and* $\exists$ *path* $p \in B_{j,k}(A)$ *passing through node* $n\}$.

This set of nodes is intuitively, the set of nodes which terminate partial paths which converge at node j, "through level k". In other words, if $ind(j) \in A$, and $ind(parent(j)) \notin A$ (or j is the root), and $ind(i) \in A$ for each node i on the path from j through level k, then the node at level k is in $N_{j,k}(A)$.

Lemma 3.5 $(\forall j)(\forall k \geq d(j))(\forall A \subseteq \mathrm{N})$ $B_{j,k}(A)$ *is measurable and* $\Pr[B_{j,k}(A)] = \frac{|N_{j,k}(A)|}{2^k}$.

Proof: We claim that

$$B_{j,k}(A) = \biguplus_{x \in N_{j,k}(A)} P_x.$$

To see that the sets in the union are disjoint, note that if $x \neq y$ and both are in $N_{j,k}(A)$, then $d(x) = d(y) = k$, and every path must pass through *exactly* one node at each level; thus $P_x \bigcap P_y = \emptyset$.

($\subseteq$) If $p \in B_{j,k}(A)$, then p passes through some node y at level k, and $y \in N_{j,k}(A)$. Therefore $p \in P_y \subseteq \biguplus_{x \in N_{j,k}(A)} P_x$.

($\supseteq$) If $p \in \biguplus_{x \in N_{j,k}(A)} P_x$, and y is the node at depth k on p, then since the definition of $N_{j,k}(A)$ doesn't depend on nodes deeper than depth k, all paths passing through y must be in $B_{j,k}(A)$.

We now have

$$\begin{aligned} \Pr[B_{j,k}(A)] &= \Pr[\biguplus_{x \in N_{j,k}(A)} P_x] \\ &= \sum_{x \in N_{j,k}(A)} \Pr[P_x] \\ &= \sum_{x \in N_{j,k}(A)} 2^{-d(x)} \\ &= \sum_{x \in N_{j,k}(A)} 2^{-k} \\ &= \frac{|N_{j,k}(A)|}{2^k} \end{aligned}$$

□

Thus for all A, $B_{j,k}(A), B_j(A)$, and $B(A)$ are measurable. Note now that $B(GOOD_f)$ are exactly those paths such that the sequence of outputs along the nodes of the path corresponds to a correct BC–identification. Therefore we have $\Pr[P\ BC\text{–identifies } f] = \Pr[B(GOOD_f)]$.

We end this section by proving several lemmas which will be useful in subsequent sections.

The following lemma asserts that the sets $\{B_{j,k}(A)\}$ are increasingly better estimates of the set $B_j(A)$ as k gets larger.

Lemma 3.6 *For all nodes j, for all $A \subseteq \mathrm{N}$, and for all $T_{P,f}$,*

1. $(\forall k \geq d(j))\ B_{j,k}(A) \supseteq B_{j,k+1}(A)$.
2. $(\forall k \geq d(j))\ \Pr[B_{j,k}(A)] \geq \Pr[B_j(A)]$.
3. $\Pr[B_j(A)] = \lim_{k\to\infty} \Pr[B_{j,k}(A)]$.

Proof: Property 1 is immediate from the definition of $B_{j,k}(A)$. Property 2 and property 3 follow from property 1, the monotonicity property of probability measures, and the fact that $B_j(A) = \bigcap_{k=d(j)}^{\infty} B_{j,k}(A)$. □

The following lemma gives us insight into how and when paths BC–converge in any tree $T_{P,f}$. In particular, suppose that the probability of paths converging to a set A is greater than p. (*i.e.* $\Pr[B(A)] > p$). The convergence of different paths to A may occur at many different nodes. We show however that there are nodes where "significant chunks" of paths converge to A. This must occur because there are an uncountable number of paths, but only countably many nodes.

Lemma 3.7 *For all $A \subseteq \mathrm{N}$, for all $p \in \mathrm{R}$ such that $0 \leq p \leq 1$, if $\Pr[B(A)] > p$, then there is a least numbered node v such that*

$$\Pr[\biguplus_{j=1}^{v} B_j(A)] = \sum_{j=1}^{v} \Pr[B_j(A)] > p.$$

Proof:

$$\biguplus_{j=1}^{\infty} B_j(A) = B(A),$$

so

$$\sum_{j=1}^{\infty} \Pr[B_j(A)] = \Pr[\biguplus_{j=1}^{\infty} B_j(A)] = \Pr[B(A)] > p$$

and by the definition of an infinite sum there is a least v such that

$$\sum_{j=1}^{v} \Pr[B_j(A)] > p.$$

□

3.1.2 Team Simulation of Probabilistic IIMs

We saw that by Corollary 2.24 a probabilistic IIM can simulate a team of IIMs. In this section we show under what circumstances a team of IIMs may be used to simulate a probabilistic IIM for BC–identification.

Theorem 3.8 $(\forall n \geq 1)(\forall p)\ \frac{1}{n+1} < p \leq 1 \Rightarrow BC_{prob}(p) \subseteq BC_{team}(n)$.

Corollary 3.9 $(\forall n \geq 1)\ BC_{prob}(\frac{1}{n}) = BC_{team}(n)$.

The corollary follows from Theorem 3.8 and Corollary 2.24. Thus the probabilistic BC–identification hierarchy contains the team BC–identification hierarchy. Theorem 3.8 also implies that the probabilistic hierarchy is "no finer" than the team hierarchy. We shall discuss this more in Section 3.3.

We first note that a special case of Theorem 3.8 has a very simple proof. If $n = 1$, then the theorem asserts that if there is a probabilistic IIM P which BC–identifies a set of functions U with probability $p > \frac{1}{2}$, then there is a deterministic IIM M_1 which BC–identifies U. To prove this, we merely need to argue that since $\Pr[B(GOOD_f)] > \frac{1}{2}$, the k^{th} level of $T_{P,f}$ consists of $> \frac{1}{2}$ correct programs for all but finitely many levels k. The machine M_1 which will identify U deterministically will, given $f \in U$, construct $T_{P,f}$, and for its k^{th} guess output the index of a *THRESHOLD* program (Section 2.2) which essentially does a majority vote of the computations of the programs whose indices occur at the k^{th} level of $T_{P,f}$. A simple argument shows that M_1 BC–identifies U.

In order to prove Theorem 3.8 in general, we need only show that if $U \subseteq BC_p(P)$ with $p > \frac{1}{n+1}$ then there is a team $\{M_1, M_2, \ldots, M_n\}$ such that for every $f \in U$, there is an i such that M_i BC–identifies f.

For a particular $f \in U$, we'll informally use the term "weight" of a set of paths P in $T_{P,f}$ to mean $\Pr[P]$, as this term more accurately suggests the appropriate intuition. (The entire tree $T_{P,f}$ has weight 1.) The intuition behind the proof of Theorem 3.8 is as follows. Since the weight of paths which BC–converge to correct programs for f is $> \frac{1}{n+1}$, we can show that the fraction of correct programs at each level of the computation tree $T_{P,f}$ is greater than $\frac{1}{n+1}$ for all but finitely many levels of the tree. (In the case that $n = 1$, we have the argument described earlier.) A deterministic strategy to BC–identify f might simply output a *THRESHOLD* program with threshold $> \frac{1}{n+1}$ constructed from the program indices found at each level, since as we will show below, this group of indices will contain greater than the fraction $\frac{1}{n+1}$ of $GOOD_f$ indices. The problem with this strategy is that each level might contain greater than the fraction $\frac{1}{n+1}$ of $WRONG_f$ indices, thus not satisfy the threshold condition. Elements of $WRONG_f$ have the pleasing property that they can be identified (in the limit, given values of f), by simulation and comparison with f. If the deterministic strategy knew roughly how many $WRONG_f$ programs there were at a given level of the tree, then it could eliminate most of them, and output the remaining programs at that level. If enough $WRONG_f$ programs were eliminated at each level, then the deterministic strategy could output a correct threshold program, and thus BC–identify f. The team of n IIMs is used to guess roughly what the fraction of $WRONG_f$ indices is (in the limit) at each level of the tree.

Definition 3.10 *Let $T_{P,f}$ be a computation tree, and $A \subseteq N$ be a set of program indices. Then*

- $L_k = \{n \mid n \text{ is a node at level } k \text{ of } T_{P,f}\}$.
- $L_k(A) = \{n \in L_k \mid ind(n) \in A\}$.

Note that $|L_k| = 2^k$. The sets which will most concern us are the sets $L_k(GOOD_f)$ and $L_k(WRONG_f)$, the sets of nodes at level k which have $GOOD$ and $WRONG$ indices of f respectively.

Lemma 3.11 $(\forall A \subseteq N)$ $(\forall f)$ $(\forall$ *IIMs* $P)$ $\Pr[B(A)] > p \Rightarrow (\overset{\infty}{\forall}_k)$ $|L_k(A)| > p2^k$.

That is, the fraction of nodes at level k with indices in A is greater than p for all but finitely many levels.

Proof:

If $\Pr[B(A)] > p$, then by Lemma 3.7 there exists a least numbered node v such that

$$\Pr[\biguplus_{j=1}^{v} B_j(A)] = \sum_{j=1}^{v} \Pr[B_j(A)] > p.$$

By Lemma 3.6, for all $k \geq \max\{d(i) \mid 1 \leq i \leq v\}$, we have that

$$\sum_{j=1}^{v} \Pr[B_{j,k}(A)] \geq \sum_{j=1}^{v} \Pr[B_j(A)] > p.$$

Then by Lemma 3.5,

$$\sum_{j=1}^{v} \frac{|N_{j,k}(A)|}{2^k} = \sum_{j=1}^{v} \Pr[B_{j,k}(A)] > p$$

or

$$\sum_{j=1}^{v} |N_{j,k}(A)| > p2^k.$$

Now by definition, every element $N_{j,k}(A)$ is at depth k, and has index in A. Also note that $N_{j,k}(A) \bigcap N_{m,k}(A) = \emptyset$ if $j \neq m$. (Since if j and m are on the same path, then one of the sets is empty, for convergence can happen at exactly one node on any path; otherwise, their descendants at level k are disjoint.) Therefore

$$|\biguplus_{j=1}^{v} N_{j,k}(A)| > p2^k$$

and there are $> p2^k$ nodes at level k with index in the set A, *i.e.*

$$(\forall_k^\infty)\ |L_k(A)| > p2^k,$$

proving the lemma. □

We are now ready to prove Theorem 3.8. Let $U \in BC_{prob}(p)$, with $p > \frac{1}{n+1}$. Then there is a probabilistic IIM P which BC–identifies every $f \in U$ with probability $\geq p > \frac{1}{n+1}$. We construct a team of n deterministic IIMs such that for all $f \in U$, there is a team member which BC–identifies f.

Consider any $f \in U$, and the tree $T_{P,f}$. Then by the definition of probabilistic BC–identification,

$$\Pr[B(GOOD_f)] \geq p > \frac{1}{n+1}.$$

Lemma 3.11 asserts that

$$(\forall_k^\infty)\ |L_k(GOOD_f)| > \frac{2^k}{n+1}.$$

Now since

$$(\forall k)\ |L_k(GOOD_f)| + |L_k(SLOW_f)| + |L_k(WRONG_f)| = 2^k$$

we have

$$(\forall_k^\infty)\ |L_k(WRONG_f)| < \frac{n2^k}{n+1}.$$

There are then n distinct and mutually exclusive possibilities about how $|L_k(WRONG_f)|$ behaves "in the limit."

$$\begin{array}{llcl}
\text{Possibility } n: & (\forall_k^\infty)|L_k(WRONG_f)| & < & \frac{n2^k}{n+1} \text{ and} \\
 & (\exists_k^\infty)|L_k(WRONG_f)| & \geq & \frac{(n-1)2^k}{n+1} \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\text{Possibility } i: & (\forall_k^\infty)|L_k(WRONG_f)| & < & \frac{i2^k}{n+1} \text{ and} \\
 & (\exists_k^\infty)|L_k(WRONG_f)| & \geq & \frac{(i-1)2^k}{n+1} \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\text{Possibility } 1: & (\forall_k^\infty)|L_k(WRONG_f)| & < & \frac{2^k}{n+1} \text{ and} \\
 & (\exists_k^\infty)|L_k(WRONG_f)| & \geq & 0
\end{array}$$

We use the team of n deterministic IIM's to guess which case will hold for a particular f. The machine whose guess is correct will BC–identify f.

The idea behind the construction is fairly simple. If a machine M_i knows roughly what the fraction of $WRONG_f$ guesses there are at each level of the tree, it can cancel most of them by witnessing that they differ from f. Machine M_i will search for deeper and deeper levels of the tree $T_{P,f}$ such that the fraction of $WRONG_f$ guesses among those output is at least $\frac{i-1}{n+1}$, and then cancel these wrong guesses.

If it is also true that past some point, the fraction of $WRONG_f$ guesses is bounded above by $\frac{i}{n+1}$, then M_i will be able to form (in the limit) sets of indices for which at least the fraction $\frac{1}{n+1}$ are correct, and strictly less than this are $WRONG_f$ indices. Thus M_i will be able to output a threshold program.

Machine M_i

1. $k_{old} \leftarrow 0$
2. LOOP:
3. Simulate M on input values received from f, and build $T_{P,f}$.
4. DOVETAIL the computations of $\varphi_{ind(s)}(j)$ for all nodes s and numbers j, comparing the outputs of completed computations with actual values of f, UNTIL for some level $k > k_{old}$, there are $\geq \frac{2^k(i-1)}{n+1}$ nodes in the set $CANCEL_k$, the set of nodes at level k whose indices have been observed to be in $WRONG_f$.
5. $I_k \leftarrow$ The ordered multiset of indices of nodes in $L_k - CANCEL_k$
6. $\vec{p} = p_1, p_2, \ldots, p_{|I_k|}$, with $(\forall i)\ p_i = \frac{1}{2^k}$.

7. OUTPUT the index of the program $THRESHOLD_{\frac{1}{n+1}, I_k, \vec{p}}$

8. $k_{old} \leftarrow k$

9. GO TO LOOP

Consider the instructions for machine M_i above. We clarify the dovetail of line 4: $CANCEL_k$ starts out empty. A node n at level k is placed in $CANCEL_k$ when for some x, $\varphi_{ind(n)}(x) \downarrow\neq f(x)$. Thus $CANCEL_k$ contains only elements of $L_k(WRONG_f)$.

Now let $U \subseteq BC_p(P)$, with $p > \frac{1}{n+1}$, and let $M_1, M_2, \ldots, M_n$ be defined as above. Then to prove Theorem 3.8 we only need to prove the following

Claim 3.12 *Let $f \in U$, and let M_i be the machine defined above which "guesses correctly", i.e. $|L_k(WRONG_f)|$ satisfies the i^{th} possibility stated previously. Then*

1. *M_i outputs infinitely many indices of programs $THRESHOLD_{\frac{1}{n+1}, I_k, \vec{p}}$*
2. $(\forall_k^\infty)$ $|I_k \bigcap WRONG_f| < \frac{2^k}{n+1}$.
3. $(\forall_k^\infty)$ $|I_k \bigcap GOOD_f| > \frac{2^k}{n+1}$.

Theorem 3.8 follows from Claim 3.12 because if M_i satisfies the three conditions, then $(\forall_k^\infty)$ the program $THRESHOLD_{\frac{1}{n+1}, I_k, \vec{p}}$ (whose index M_i outputs) satisfies the hypothesis of Lemma 2.4, and thus $(\forall_k^\infty)$ M_i outputs the index of a program which computes f, which is exactly the definition of BC–identification.

Proof of Claim 3.12

To show 1. we note that the only possible way for M_i to output the indices of only finitely many programs $THRESHOLD_{\frac{1}{n+1}, I_k, \vec{p}}$ is that for some value k_{old}, the dovetail of step 4 of M_i fails to satisfy its halting condition. By assumption on i, $(\exists_k^\infty)$ $|L_k(WRONG_f)| \geq \frac{2^k}{n+1}$, therefore there is some $k > k_{old}$ with $|L_k(WRONG_f)| \geq \frac{2^k}{n+1}$. Now $|L_k(WRONG_f)|$ is at most 2^k, hence finite, and after some finite number of steps of simulation, M_i would be able to witness that all of these nodes have indices in $WRONG_f$, hence they would be placed into $CANCEL_k$. Thus $CANCEL_k$ at some point must contain $\geq \frac{2^k}{n+1}$ nodes and therefore the halting condition is satisfied.

We prove 2. The number of elements of $WRONG_f$ which are in I_k can be at most the total number of nodes at level k with indices in the set $WRONG_f$ $(= |L_k(WRONG_f)|,)$ minus the number of nodes which have been cancelled. Thus

$$|I_k \bigcap WRONG_f| = |L_k(WRONG_f)| - |CANCEL_k|.$$

By assumption on i,

$$(\forall_k^\infty)\ |L_k(WRONG_f)| < \frac{i2^k}{n+1}$$

and by the dovetail halting condition, for all k found by M_i ,

$$|CANCEL_k| \geq \frac{(i-1)2^k}{n+1}.$$

Thus

$$(\forall_k^\infty)\ |I_k \bigcap WRONG_f| < \frac{i2^k}{n+1} - \frac{(i-1)2^k}{n+1} = \frac{2^k}{n+1}.$$

Finally (part 3.), to see that $(\forall_k^\infty)\ |I_k \bigcap GOOD_f| > \frac{2^k}{n+1}$, note that no node in $|L_k(GOOD_f)|$ is *ever* cancelled, so the multiset I_k contains the index of *every* node in $|L_k(GOOD_f)|$. Thus

$$|I_k \bigcap GOOD_f| = |L_k(GOOD_f)|.$$

That is, the number of *GOOD* indices in I_k equals the number of nodes with *GOOD* indices at level k. Now since $\Pr[B(GOOD_f)] \geq p > \frac{1}{n+1}$, as noted earlier, Lemma 3.11 implies

$$(\forall_k^\infty)\ |I_k \bigcap GOOD_f| = |L_k(GOOD_f)| > \frac{2^k}{n+1}.$$

This completes the proof of Claim 3.12 and Theorem 3.8. □

3.2 EX Probability and Teams

In this section we prove theorems analogous to those relating probabilistic *BC*–identification and team *BC*–identification. As we shall see, the restrictions of *EX*–identification disallow some of the proof techniques of the previous sections, so we will need more complicated machinery.

3.2.1 EX Convergence in $T_{P,f}$

We begin by defining a more natural notion of convergence of a path in a tree $T_{P,f}$ than that of *BC*–convergence.

Definition 3.13 *Let* $p = \langle t_0, t_1, \ldots \rangle$ *be a path in* $T_{P,f}$ *, and* j *be a program index. The path* p converges to j *iff* $(\forall_k^\infty)\ ind(t_k) = j$.

If path p converges to j then p corresponds to a possible computation of P with input f for which P (in the limit) converges to outputting "j" as its guess for a program index for f.

Definition 3.14 *Path* $p = \langle t_0, t_1, \ldots \rangle$ converges at node n *iff*

- *p passes through node n.* ($t_{d(n)} = n$).
- $(\forall k \geq d(n))$ $ind(t_k) = ind(n)$ *(hence p converges to ind(n)).*
- *there does not exist* $k < d(n)$ *such that* $(\forall m \geq k)$ $ind(t_m) = ind(n)$.

This definition simply requires that all nodes past n on path p have the same index as node n, and node n is the least depth at which this convergence occurs.

It is important to note that a path which converges at a node n, converges to $ind(n)$. Thus if we know where a path converges, we know what index it converges to. Note that if p converges to j, then $p \in B(\{j\})$. We develop a new notation to represent paths which converge, rather than abuse the old notation for paths which *BC*-converge.

Definition 3.15 $C(A) = \{$*paths* $p \in T_{P,f} \mid (\exists a \in A)$ *p converges to a*$\}$.

Let P be a probabilistic IIM. Then $C(GOOD_f)$ contains exactly those paths such that the sequence of outputs along the nodes of the path corresponds to a correct *EX*–identification. Therefore we have $\Pr[P\ EX\text{–identifies } f] = \Pr[C(GOOD_f)]$.

Thus the probability that P *EX*-identifies f is the fraction of paths of $T_{P,f}$ which converge to a correct program index for f; or the fraction of P's possible computations which correspond to a single deterministic *EX*–identification of the function f.

Definition 3.16 $C_j = \{p \mid$ *p is a path in* $T_{P,f}$ *and p converges at node* $j\}$.

A path $p = \langle t_0, t_1, \ldots \rangle$ is *k–consistent* with C_j iff the following three conditions hold:

1. $t_{d(j)} = j$ (the path passes through node j).
2. $(\forall i)\ d(j) \leq i \leq k \Rightarrow ind(t_i) = ind(j)$.
3. j is the root OR $ind(t_{d(j-1)}) \neq ind(j)$.

Definition 3.17 $C_{j,k} = \{p \mid$ *p is k-consistent with* $C_j\}$.

Thus $C_{j,k}$ consists of paths p satisfying:

- p passes through node j.
- P outputs a different index at *parent*(j) than at j (or j is the root).
- All nodes after j on p down to depth k have the same index as j.

Intuitively, $C_{j,k}$ is the set of paths which appear to be converging to $ind(j)$ and appear to converge *at* j when we examine $T_{P,f}$ for k levels only.

Clearly $C(A) = \biguplus_{ind(j)\in A} C_j$, and $C_j = \bigcap_{k=d(j)}^{\infty} C_{j,k}$.

Note that $C_{j,k}$ is the set of paths which converge at node j (to $ind(j)$) "through level k"; thus $C_{j,k} = B_{j,k}(\{ind(j)\})$. We have already shown that for all A, $B_{j,k}(A)$ is measurable, therefore, $C_{j,k}$, C_j, and $C(A)$ are all measurable.

We end this section with some important lemmas. The following two lemmas are analogues of Lemmas 3.6 and 3.7.

Lemma 3.18 *For all nodes j, and for all $T_{P,f}$*

1. $(\forall k \geq d(j))\ C_{j,k} \supseteq C_{j,k+1}$.
2. $(\forall k \geq d(j))\ \Pr[C_{j,k}] \geq \Pr[C_j]$.
3. $\Pr[C_j] = \lim_{k\to\infty} \Pr[C_{j,k}]$.

Proof: $C_{j,k} = B_{j,k}(\{ind(j)\})$ and then the lemma follows immediately from Lemma 3.6. Thus the sets $\{C_{j,k}\}$ are increasingly better estimates of the set C_j as k increases. □

Lemma 3.19 *For all $A \subseteq$ N and for all $p \in [0,1]$, if $\Pr[C(A)] > p$, then there exists nodes $\{n_1, n_2, \ldots, n_k\}$ such that $(\forall i)$ $ind(n_i) \in A$, and $\Pr[\biguplus_{j=1}^{k} C_{n_j}] > p$.*

Proof:

$$\biguplus_{ind(j)\in A} C_j = C(A)$$

so

$$\sum_{ind(j)\in A} \Pr[C_j] = \Pr[\biguplus_{ind(j)\in A} C_j] = \Pr[C(A)] > p$$

so (by a simple limit property) there must be a finite set of nodes $\{n_1, n_2, \ldots, n_k\}$ such that

$$\Pr[\biguplus_{j=1}^{k} C_{n_j}] = \sum_{j=1}^{k} \Pr[C_{n_j}] > p.$$

□

So most of the paths which converge to any index in the set A converge at one of a finite collection of nodes. The justification for the partition in the expression above is that a path can converge at *at most* one node.

We introduce one more lemma for which there is no analogue in the *BC* case.

Lemma 3.20 $\Pr[C_{j,k}]$ *is computable from the first k levels of $T_{P,f}$* .

Proof: $\Pr[C_{j,k}] = \Pr[B_{j,k}(\{ind(j)\})] = \frac{|N_{j,k}(\{ind(j)\})|}{2^k}$, by Lemma 3.5. Thus we need only show that for all k, $N_{j,k}(\{ind(j)\})$ is computable from the first k levels of $T_{P,f}$. But $N_{j,k}(\{ind(j)\})$ is simply the set of nodes at level k through which a path in $B_{j,k}(\{ind(j)\})$ passes. These are nodes m at level k such that m is a descendant of j, and all nodes x on the path between j and m (inclusive), have $ind(x) = ind(j)$, and either j is the root node, or $ind(parent(j)) \neq ind(j)$. Thus membership in $N_{j,k}(\{ind(j)\})$ depends only on the indices of nodes of $T_{P,f}$ in the first k levels. □

Note that in general, $\Pr[B_{j,k}(A)]$ is not necessarily computable, since membership in A might not be decidable.

3.2.2 Team Simulation of Probabilistic IIMs

We are surprised to find that analogues of Theorem 3.8 and Corollary 3.9 exist for EX–identification, since the majority voting techniques do not seem to work in this case.

Theorem 3.21 $(\forall n \geq 1)(\forall p)\ \frac{1}{n+1} < p \leq 1 \Rightarrow EX_{prob}(p) \subseteq EX_{team}(n)$.

Corollary 3.22 $(\forall n \geq 1)\ EX_{prob}(\frac{1}{n}) = EX_{team}(n)$.

The corollary follows from Theorem 3.21 and Corollary 2.24. To prove Theorem 3.21, we will need the following definitions and lemma.

Definition 3.23 *A set I of program indices is a* correct list *for f iff $I \bigcap GOOD_f \neq \emptyset$.*

The class OEX was introduced in [10][1]

Definition 3.24

- *M OEX-identifies f (written $f \in OEX(M)$), iff M, when fed the graph of f in any order, outputs an infinite sequence $\{I_k\}$ of finite lists, and there is a correct list I such that $(\forall_k^\infty)\ I_k = I$.*
- *$OEX = \{U \mid (\exists M)\ U \subseteq OEX(M)\}$.*

Case and Smith [10] prove a generalization of the following lemma.

Lemma 3.25 [2] *$OEX = EX$.*

[1] Our definition is somewhat different, but it is easy to show that the two definitions are equivalent.

[2] Note that the lemma does not contradict the team hierarchy theorem by the following fallacious reasoning. A single IIM can simulate each member of a team of n IIMs and output a list containing

Proof: Clearly $EX \subseteq OEX$. We show that $OEX \subseteq EX$. Let $U \in OEX(M)$. We construct an IIM M' which *EX*–identifies U. The idea behind the proof is that M' can simulate M, and once M converges to outputting a correct list I, M' can cancel (in the limit) every element of $I \bigcap WRONG_f$. Then M' can output the program $RACE_I$ which will satisfy the hypothesis of Lemma 2.2 and therefore will be a correct program for f.

M' on input $f|_k$ simulates M on input $f|_k$ and obtains the list I_k. M' tries to compute, allowing k steps for each computation, the values $\{\varphi_i(j) \mid i \in I_k, 1 \leq j \leq k\}$. M' sets L_k to $I_k - \{i \in I_k \mid (\exists j \leq k)$ and in $\leq k$ steps, $\varphi_i(j)\downarrow \neq f(j)\}$. M' then outputs the index of the program $RACE_{L_k}$.

To see that M' *EX*–identifies U, let $f \in U \subseteq OEX(M)$. Then let k_0 be large enough so that for all $k > k_0$, $I_k = I$, with $I \bigcap GOOD_f \neq \emptyset$, and $(\forall i \in I \bigcap WRONG_f)$, M' cancels i within k steps. No further cancellations occur once $k \geq k_0$, and the sequence of lists $\{L_k\}$ converges to the list L. Then M' converges to outputting the index of $RACE_L$. Now since I is a correct list for f, $I \bigcap GOOD_f \neq \emptyset$, furthermore, no element in $I \bigcap GOOD_f$ is ever cancelled, so the set L satisfies the hypothesis of Lemma 2.2. Hence M' converges to outputting the program $RACE_L$ which computes f. □

It is now clear that Theorem 3.21 follows from:

Lemma 3.26 *Let $U \in EX_{prob}(p)$, with $p > \frac{1}{n+1}$. Then $(\exists M_1, M_2, \ldots, M_n)$ (deterministic IIMs) such that $(\forall f \in U)(\exists i)$ M_i OEX–identifies f.*

At first glance, it would appear that a technique similar to that used in the *BC* proof could be employed here. But even in the more obvious case where $n = 1$ this approach doesn't seem to work. For example, if P is a probabilistic IIM which *EX*–identifies U with probability $p > \frac{1}{2}$, then certainly a program which did a "majority vote" of the programs at each level of $T_{P,f}$ would (in the limit) be correct. However, in order to *EX*–identify a function, the program output must be the same in the limit, and the majority vote program would change from level to level. We might think that since P *EX*–identifies f with probability $> \frac{1}{2}$ that there is some correct index of f, say i, such that in the limit $> \frac{1}{2}$ of the nodes at each level of $T_{P,f}$ had index i. This is unfortunately not true, since probabilistic *EX*–identification was defined to capture the intuitive notion that "when you run P with input f, the probability that you get a correct *EX*–identification

the last element output by each IIM of the team. If the outputs of *each* team member converged, then indeed the lists output by the single IIM would converge to a correct list, and the IIM would *OEX* and hence *EX*–identify the function. Unfortunately, this need not happen, as any team members incorrect on a given function may be incorrect by not converging to a single hypothesis.

is at least p." This doesn't imply that there is some single index for f which P will converge to with probability p. We note that this less natural definition would allow a more straightforward proof, similar to the proof in the BC case. We investigate this apparently different notion of probabilistic EX–identification in Section 3.5.

The idea behind the proof of Lemma 3.26 is that rather than look only at levels of the tree, each deterministic IIM of the team will have to scan the tree to identify converging paths and nodes at which this convergence occurs.

Proof of Lemma 3.26:

Let P be the probabilistic machine which identifies U with probability $p > \frac{1}{n+1}$. We will show that if a deterministic machine has a reasonable estimate of the "weight" of the set of ***all converging paths*** (paths which converge to *any* index), then it can converge to a correct list for f, hence OEX–identify f.

The finite nondeterminism of the team of n machines is used in the following way: each team member guesses a different range which the weight of the converging paths may fall into. In particular, for $1 \leq i \leq n$, M_i assumes that the total weight of all converging paths is in the half-open interval $(\frac{i}{n+1}, \frac{i+1}{n+1}]$. Depending on the function f chosen from U, the weight of converging paths will fall into one of these intervals, and the associated machine will converge to a correct list for f.

Let P be a probabilistic IIM with $U \subseteq EX_p(P)$ and $p > \frac{1}{n+1}$. Then let the team $\{M_1, M_2, \ldots, M_n\}$ be those defined below.

Machine M_i

1. On input $f|_k$, simulate P with input $f|_k$, and construct $T_k =$ the finite tree consisting of the first k levels of $T_{P,f}$.
2. FOR each node j in T_k compute $\Pr[C_{j,k}]$
3. Let c_k be the least numbered node in T_k such that $\sum_{j=1}^{c_k} \Pr[C_{j,k}] \geq \frac{i}{n+1}$ (If no such c_k exists, then output $\emptyset$.
4. Output $\{ind(i) \mid 1 \leq i \leq c_k\}$.

Note that each step of M_i is a simple, computable operation: Step 2 can be done by Lemma 3.20. We comment that c_k exists for all k, but this is not necessary for the proof.

We show that for every $f \in U$, there is an i such that M_i converges to outputting a correct list for f.

As mentioned above, the team member which is correct will be the one with the best estimate for the weight of the converging paths.

More precisely: $\Pr[C(GOOD_f)] \geq p > \frac{1}{n+1}$ by the definition of "P EX–identifies f with probability p." $C(\mathrm{N})$ is the set of paths which converge to ***any*** index (good or bad), so clearly

$$C(GOOD_f) \subseteq C(\mathrm{N})$$

and therefore

$$\Pr[C(\mathrm{N})] \geq \Pr[C(GOOD_f)] > \frac{1}{n+1}.$$

Let $m = \max\{i \mid \frac{i}{n+1} < \Pr[C(\mathrm{N})]\}$. The value m is well defined, since $\frac{1}{n+1} < \Pr[C(\mathrm{N})] \leq \frac{n+1}{n+1}$. In particular, $1 \leq m \leq n$. We will show that M_m converges to a correct list for f.

M_m "knows" that the weight of the paths which converge to any index is greater than $\frac{m}{n+1}$. By Lemma 3.19 there exists a finite set of nodes V with the weight of the paths converging at a node in V greater than $\frac{m}{n+1}$. M_m will look for these nodes, find them (in the limit), and output their indices. (Actually, M_m will output the indices of nodes $1, 2, \ldots, n_{max}$, where n_{max}, is the greatest numbered node in the set V.)

M_m attempts to compute for every node j the weight of paths which converge at node j ($\Pr[C_j]$). It cannot do this, since it is not a finite computation. M_m can, however, compute $\Pr[C_{j,k}]$, which we know is an upper bound for $\Pr[C_j]$ (see Lemma 3.18), and will converge to $\Pr[C_j]$ from above as k increases (Step 2).

M_m outputs the indices of the first c_k nodes, where c_k is the smallest numbered node such that the (estimated) weight of paths converging to any of the nodes $\{1, 2, \ldots, c_k\}$ is greater than $\frac{m}{n+1}$. (Steps 3, 4)

M_m will eventually converge to outputting some fixed list, because there is some smallest numbered node s such that the weight of the paths converging to a node in $\{1, 2, 3, \ldots, s\}$ is $\geq \frac{m}{n+1}$, and the estimate of these weights are becoming better in the limit. More formally:

By the definition of m,

$$\frac{m}{n+1} < \Pr[C(\mathrm{N})] \leq \frac{m+1}{n+1}.$$

Since

$$\Pr[C(\mathrm{N})] > \frac{m}{n+1}$$

Lemma 3.19 gives nodes $\{n_1, n_1, \ldots, n_v\}$ with

$$ind(n_i) \in \mathrm{N} \text{ and } \sum_{i=1}^{v} \Pr[C_{n_i}] > \frac{m}{n+1}.$$

Since *all* nodes j have $ind(j) \in \mathrm{N}$, this implies that there exists a smallest numbered node s, such that

$$\sum_{j=1}^{s} \Pr[C_j] > \frac{m}{n+1}.$$

(Choosing $s \geq \max\{n_i\}$ will certainly satisfy the inequality). Now for all $k \geq d(s)$, nodes $1, 2, \ldots s$ will be in T_k, and furthermore, by Lemma 3.18

$$\sum_{j=1}^{s} \Pr[C_{j,k}] \geq \sum_{j=1}^{s} \Pr[C_j] \geq \frac{m}{n+1}$$

hence $(\forall_k^\infty)\ c_k \leq s$ in Step 3 of M_m. Now Lemma 3.26 follows from:

Claim 3.27

1. *M_m converges to the list $I = \{ind(1), ind(2), \ldots, ind(s)\}$.*
2. *I contains a correct program index for f.*

Proof:

(Part 1): We have already shown that $(\forall_k^\infty)\ c_k \leq s$. Now, by Lemma 3.18, for all j, and for all $k \geq d(j)$, $\Pr[C_{j,k}] \geq \Pr[C_{j,k+1}]$. It follows that the sequence $\{c_k\}$ is nondecreasing (for all sufficiently large k) since c_k was chosen as the ***smallest*** value satisfying the inequality $\sum_{j=1}^{c_k} \Pr[C_{j,k}] \geq \frac{m}{n+1}$, and since the summands are non-increasing, $\{c_k\}$ must be non-decreasing. Since $\{c_k\}$ is a nondecreasing sequence of integers bounded above by s, it converges. Suppose that $\{c_k\}$ converged to a number $t < s$. Then for all sufficiently large k, $\sum_{j=1}^{t} \Pr[C_{j,k}] \geq \frac{m}{n+1}$. This implies that $\sum_{j=1}^{t} \Pr[C_j] \geq \frac{m}{n+1}$, since the latter is the limit of the former. This is a contradiction, since s is the *least* integer satisfying that inequality. Therefore, $\{c_k\}$ converges to s, and the list of program indices output by M converges to $I = \{ind(1), ind(2), \ldots, ind(s)\}$.

(Part 2): We now argue that the list of indices which M_m outputs contains a correct index for f. This is straightforward: The weight of paths converging to correct indices for f is greater than $\frac{1}{n+1}$, and the weight of paths converging to any index is less than or equal to $\frac{m+1}{n+1}$. It follows that the weight of paths converging to an index which is not an index for f is strictly less than $\frac{m}{n+1}$. But M has found a list of indices with weight greater than or equal to $\frac{m}{n+1}$, hence not all of the indices on the list can be incorrect. More formally, Since

$$\mathrm{N} = GOOD_f \biguplus BAD_f$$

we have

$$C(\mathrm{N}) = C(GOOD_f) \biguplus C(BAD_f)$$

so that

$$\Pr[C(\mathrm{N})] = \Pr[C(GOOD_f)] + \Pr[C(BAD_f)].$$

We also know that

$$\Pr[C(\mathrm{N})] \leq \frac{m+1}{n+1}$$

and

$$\Pr[C(GOOD_f)] > \frac{1}{n+1}.$$

We conclude that

$$\Pr[C(BAD_f)] < \frac{m}{n+1}.$$

Now observe that the set of indices $I = \{ind(1), ind(2), \ldots, ind(s)\}$ has the property that $\sum_{j=1}^{s} \Pr[C_j] \geq \frac{m}{n+1}$, that is, $\Pr[C(I)] \geq \frac{m}{n+1}$, therefore at least one element of I must be a correct program index for f, otherwise $I \subseteq BAD_f$, $C(I) \subseteq C(BAD_f)$, and $\Pr[C(BAD_f)] \geq \frac{m}{n+1}$ — a contradiction. This completes the proof of Claim 3.27, Lemma 3.26, and Theorem 3.21. □

3.3 Team and Probabilistic Inference Hierarchies

We now examine the consequences of the theorems in the preceding sections of this chapter. For this section only, let *ID* denote either of *EX* or *BC* (as opposed to an arbitrary identification criterion). Corollary 2.24, Theorems 3.8, 3.21 and Corollaries 3.9 and 3.22 can then be generalized as

Theorem 3.28

1. $(\forall n \geq 1)\ ID_{team}(n) \subseteq ID_{prob}(\frac{1}{n})$.
2. $(\forall n \geq 1)(\forall p)\ \frac{1}{n+1} < p \leq 1 \Rightarrow ID_{prob}(p) \subseteq ID_{team}(n)$.
3. $(\forall n \geq 1)\ ID_{prob}(\frac{1}{n}) = ID_{team}(n)$.

Part 3., together with the team hierarchy theorem (Theorem 2.15), implies that

Theorem 3.29 $(\forall n \geq 1)\ ID_{prob}(\frac{1}{n}) \subset ID_{prob}(\frac{1}{n+1})$.

Thus the team hierarchy is contained in the probabilistic hierarchy. We now note that this probabilistic hierarchy is no finer, and that it is identical to the team hierarchy:

Suppose that $\frac{1}{n+1} < p \leq \frac{1}{n}$. Clearly $ID_{prob}(\frac{1}{n}) \subseteq ID_{prob}(p)$. Parts 2 and 1 of Theorem 3.28 give the two containments

$$ID_{prob}(p) \subseteq ID_{team}(n) \subseteq ID_{prob}(\tfrac{1}{n})$$

and therefore

$$ID_{prob}(p) = ID_{prob}(\tfrac{1}{n}).$$

Thus for all of the "intermediate" probabilities $p \in (\frac{1}{n+1}, \frac{1}{n}]$, $ID_{prob}(p)$ "collapses" to $ID_{prob}(\frac{1}{n})$. The following corollary contains restatements of the same result.

Corollary 3.30 $(\forall n \geq 1)(\forall p)$

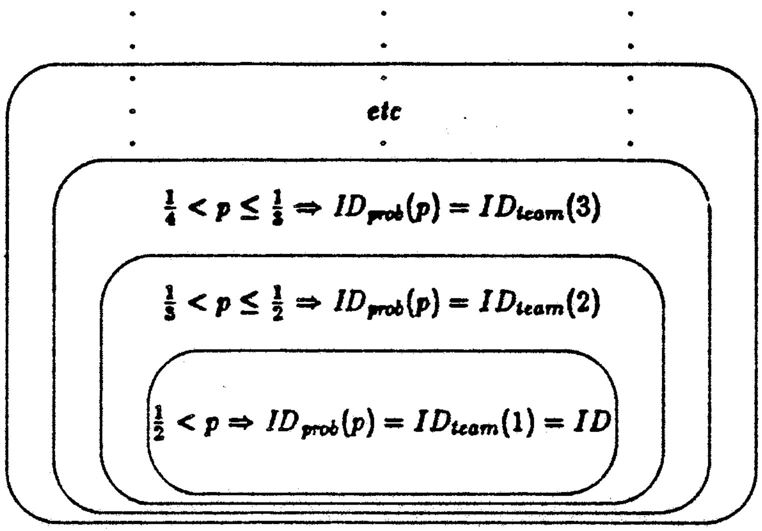
- $\frac{1}{n+1} < p \leq \frac{1}{n} \Rightarrow ID_{prob}(p) = ID_{prob}(\frac{1}{n})$.
- $(\forall p_1 < p_2)$ *If both* p_1 *and* p_2 *are in the same interval* $(\frac{1}{n+1}, \frac{1}{n}]$ *then* $ID_{prob}(p_1) = ID_{prob}(p_2)$. *If* p_1 *and* p_2 *are in different intervals, then* $ID_{prob}(p_1) \supset ID_{prob}(p_2)$.

We conclude that the probabilistic hierarchy is *exactly* the team hierarchy. Of particular interest is the following special case ($n = 1$) of our results:

$$p > \frac{1}{2} \Rightarrow ID_{prob}(p) = ID_{team}(1) = ID$$

That is, if we have a probabilistic IIM which *EX*– (*BC*–) identifies the set of functions U with probability $p > \frac{1}{2}$, then there is a deterministic IIM which *EX*– (*BC*–) identifies U. This result is shown independently by Wiehagen, Freivald, and Kinber [41] for the *EX* case.

The picture on the following page illustrates the relationship between the probabilistic and team hierarchies.



We give an example which demonstrates possible "practical" applications of our results. The following is a modification of a scenario suggested by J. Case, which appeared in [35].

We wish to send a collection of robots to investigate some alien planet. Since there may be possibly unforeseen natural disasters on this planet, we equip each robot with an inference algorithm, which it uses to predict possible occurrences such as floods, etc. based on the soil samples or other data that it collects. We would like to send the fewest number of robots possible, but would like to ensure that at least 3 will learn enough about the planet to survive, and thus carry out some particular distributed experiments and transmit the results back to Earth.

Suppose now that we are able to construct a team of 11 such robots (with possibly different inference algorithms) with the property that at least 3 of the 11 will survive. Clearly then there exists a single probabilistic IIM robot which survives with probability $\geq \frac{3}{11}$. Since $\frac{3}{11} > \frac{1}{4}$ we know that there exists a team of 3 IIM robots with at least 1 member having survival ability. By simply replicating this trio, we end up with a set of 9 robots containing at least 3 with survival ability. Thus we have a savings of 2 robots. Furthermore, since the proofs of our theorems are constructive, we can actually build the robots.

3.4 Frequency Identification

3.4.1 BC Frequency Identification

Suppose M is a deterministic IIM, and on input f, M keeps changing its guess, but "in the limit", the fraction p of M's guesses are correct. The following definitions are due to Podnieks [33], and capture this intuitive notion.

Let M be a deterministic IIM, and for each $i \in \mathrm{N}$ let g_i be the i^{th} guess of M with input f. For each k, define

$$F_k = \frac{|\{i : \varphi_{g_i} = f, 1 \leq i \leq k\}|}{k}.$$

That is, F_k is the fraction of correct guesses of M among the first k guesses (on input $f|_k$). We say that M is correct with "frequency" p if $F_k \geq p$ "in the limit". More formally,

Definition 3.31

- *M BC–identifies f with frequency p iff when fed the graph of f in any order*[3]

$$\liminf_{k \to \infty} F_k \geq p.$$

- *M BC–identifies U with frequency p iff* $(\forall f \in U)$ *M BC–identifies f with frequency p.*
- $BC_{freq}(p) = \{U \mid (\exists M)$ *M BC–identifies U with frequency p*$\}$.

We comment that it is not possible to capture "lim inf $F_k \geq p$" as an identification criterion; any predicate Q_{freq} would need to have access to the entire output sequence $M(s)$ to verify that the lim inf was $\geq p$. On the other hand, it is not difficult to express "lim inf $F_k > p$" formally as an identification criterion.

In [33], Podnieks shows that $(\forall n \geq 1)$ $BC_{freq}(\frac{1}{n}) \subset BC_{freq}(\frac{1}{n+1})$. (Actually, his results include the stronger statement that $(\forall \epsilon > 0)$ $BC_{freq}(\frac{1}{n} + \epsilon) \subset BC_{freq}(\frac{1}{n})$.)

He conjectures that for all p_1, p_2 such that $0 \leq p_1 < p_2 \leq 1$, $BC_{freq}(p_1) \supset BC_{freq}(p_2)$. We show that this conjecture is false, and that the "breakpoints" for this hierarchy are at exactly the numbers of the form $\frac{1}{n}$. More specifically, we show that this "frequency" hierarchy is identical to the BC team hierarchy.

[3]Podnieks' definition was that M need only have correct frequency behavior when fed the graph of f in the canonical order $\langle f(0), f(1), \ldots \rangle$. It seems desirable to have this aspect of our definitions uniform across all inference types, thus we adopt the definition here. Researchers partial to the less restrictive definition of "identify when input in canonical order only" will realize that if the definitions of all other classes in this paper were modified similarly, then all of the theorems would still hold.

Theorem 3.32 $(\forall n \geq 1)\ BC_{team}(n) \subseteq BC_{freq}(\frac{1}{n})$.

Proof: Let $U \in BC_{team}(n)$. Then there is a team $\{M_1, M_2, \ldots, M_n\}$ of deterministic IIMs which identify U. Let M be a deterministic IIM which on input f, does the following: M simulates $M_1, M_2, \ldots, M_n$ on input f, and outputs as its guesses the guesses output by $M_1, M_2, \ldots, M_n$ in a rotating order. M's first n guesses will be the first guesses of $M_1, M_2, \ldots, M_n$. M's next n guesses will be the second guesses of $M_1, M_2, \ldots, M_n$, etc. We must show that M BC-identifies U with frequency $\frac{1}{n}$. To do this, it is sufficient to show that for every $f \in U$, and for all $\epsilon > 0$, $(\forall_k^\infty)\ F_k > \frac{1}{n} - \epsilon$.

Now if $f \in U$, then there is some j such that M_j identifies f. Therefore there is some c such that M_j outputs only correct guesses for f after c initial guesses. Hence after the first cn guesses of M, there is at least one correct guess in each subsequent group of n guesses of M. Let $ROUND_k$ denote the guesses of M numbered $(k-1)n+1$ through kn, *i.e.* $ROUND_k$ contains the k^{th} guesses of each of $M_1, M_2, \ldots, M_n$. Then for all $i > c$, $ROUND_k$ contains at least one correct guess.

Let $\epsilon > 0$. Let x_0 be large enough so that for all $x \geq x_0$,

- $x > c$.
- $\frac{1}{cn + xn} < \frac{\epsilon}{2}$.
- $\frac{x}{cn + xn} > \frac{1}{n} - \frac{\epsilon}{2}$.

Now consider any guess g_k which falls in $ROUND_{x+c}$ with $x \geq x_0$. Then

$$F_k = \frac{\#\text{ correct guesses among the first } k}{k}.$$

The numerator is at least $x - 1$, since there is at least one correct guess in each of the rounds numbered $c+1, c+2, \ldots, c+x-1$. The denominator is at most $n(x+c)$ since guess k falls in the $(x+c)^{\text{th}}$ round. Hence:

$$F_k \geq \frac{x-1}{n(x+c)} = \frac{x}{cn+xn} - \frac{1}{cn+xn} > \frac{1}{n} - \frac{\epsilon}{2} - \frac{\epsilon}{2} = \frac{1}{n} - \epsilon.$$

We've shown that $(\forall \epsilon > 0)(\forall f \in U)$ M outputs a sequence such that $(\forall_k^\infty)\ F_k > \frac{1}{n} - \epsilon$. It follows that M BC-identifies U with frequency $\frac{1}{n}$, which completes the proof of Theorem 3.32. □

Now we show that

Theorem 3.33 $(\forall n \geq 1)(\forall p)\ \frac{1}{n+1} < p \leq 1 \Rightarrow BC_{freq}(p) \subseteq BC_{team}(n)$.

Theorem 3.33 states that the relationship between frequency BC–identification and team BC–identification is the same as that between probabilistic BC–identification and team BC–identification. The proof is very similar to that of Theorem 3.8.

Proof:

Let $U \in BC_{freq}(p)$, with $p > \frac{1}{n+1}$, and let M BC–identify U with frequency p. To prove Theorem 3.33, we construct a team $\{M_1, M_2, \ldots, M_n\}$ such that $(\forall f \in U)$ $(\exists i)$ M_i BC–identifies f.

If A is a set of program indices, let $I_k(A)$ denote the multiset of guesses of M on input f, among the first k, which are in the set A. In particular, we are interested in the sets $I_k(GOOD_f), I_k(SLOW_f)$, and $I_k(WRONG_f)$.

Clearly $|I_k(GOOD_f)| + |I_k(SLOW_f)| + |I_k(WRONG_f)| = k$. Note also that since M BC–identifies U with frequency $p > \frac{1}{n+1}$, $(\forall_k^\infty)$ $|I_k(GOOD_f)| > \frac{k}{n+1}$, and therefore $(\forall_k^\infty)$ $|I_k(WRONG_f)| < \frac{kn}{n+1}$.

As in the proof of Theorem 3.8, there are n distinct and mutually exclusive possibilities for how the sequence $I_k(WRONG_f)$ behaves in the limit – let $1 \leq i \leq n$. Then the i^{th} possibility is:

$$\text{Possibility } i: \quad \begin{array}{rcl} (\forall_k^\infty)|I_k(WRONG_f)| & < & \frac{ik}{n+1} \text{ and} \\ (\exists_k^\infty)|I_k(WRONG_f)| & \geq & \frac{(i-1)k}{n+1}. \end{array}$$

The construction and proof now follow that on page 37. The instructions for machine M_i are:

Machine M_i

1. $k_{old} \leftarrow 0$
2. LOOP:
3. Simulate M on input values received from f, and let $g_1, g_2, \ldots$ be the sequence of guesses output by M.
4. DOVETAIL the computations $\{\varphi_{g_i}(j)\}$ for all pairs of numbers i and j, comparing the outputs of completed computations with actual values of f, UNTIL for some number $k > k_{old}$, there are $\geq \frac{(i-1)k}{n+1}$ elements in the ordered multiset $CANCEL_k$, the multiset of guesses among the first k guesses of M which have been observed to be in $WRONG_f$.
5. $S_k \leftarrow$ the ordered multiset $\{g_i \mid 1 \leq i \leq k\} - CANCEL_k$
6. $\vec{p} = p_1, p_2, \ldots, p_{|S_k|}$, with $(\forall i)$ $p_i = \frac{1}{k}$.
7. OUTPUT the index of the program $THRESHOLD_{\frac{k}{n+1}, S_k, \vec{p}}$

8. $k_{old} \leftarrow k$

9. GO TO LOOP

Let M_i satisfy the i^{th} possibility. We've already observed that $(\forall_k^\infty)\ |I_k(GOOD_f)| > \frac{k}{n+1}$. Now since no $GOOD_f$ program is ever placed into $CANCEL_k$ for any k, then $I_k(GOOD_f) \subseteq S_k$, and therefore $(\forall_k^\infty)\ |S_k \bigcap GOOD_f| > \frac{k}{n+1}$. Also, by assumption on i, $(\exists_k^\infty)\ |I_k(WRONG_f)| \geq \frac{(i-1)k}{n+1}$, hence M_i can find successively larger values of k for which it outputs the index of $THRESHOLD_{\frac{k}{n+1}, S_k, \vec{p}}$ such that by assumption on i,

$$\begin{aligned} |S_k \bigcap WRONG_f| &= |I_k(WRONG_f)| - |CANCEL_k| \\ &< \frac{ik}{n+1} - \frac{(i-1)k}{n+1} \\ &= \frac{k}{n+1}. \end{aligned}$$

Thus $(\forall_k^\infty)$ the programs $THRESHOLD_{\frac{k}{n+1}, S_k, \vec{p}}$ satisfy the hypothesis of Lemma 2.4, therefore compute f, and M_i BC–identifies f, completing the proof of Theorem 3.33. □

3.4.2 EX Frequency Identification

We now introduce what is essentially the EX version of Podnieks' BC–frequency identification, and prove that the analogous theorems are true.

Let M be a deterministic IIM, which on input f, outputs the sequence of guesses $g_1, g_2, \ldots$. Let

$$F_k(g_i) = \frac{|\{j : 1 \leq j \leq k \text{ and } g_i = g_j\}|}{k}.$$

Definition 3.34

- *M EX–identifies f with frequency p iff there exists a guess g_i such that* $\liminf_{k\to\infty} F_k(g_i) \geq p$, *and* $\varphi_{g_i} = f$.
- *M EX–identifies U with frequency p iff* $(\forall f \in U)$ *M EX–identifies f with frequency p.*
- $EX_{freq}(p) = \{U \mid (\exists M)$ *M EX–identifies U with frequency p*$\}$.

If M EX–identifies f with frequency p, there is some *particular* correct guess of f, that occurs in M's output sequence with frequency p.

It is clear that if $p_1 \leq p_2$ then $EX_{freq}(p_1) \supseteq EX_{freq}(p_2)$. We show that

Theorem 3.35

1. $(\forall n \geq 1)\ EX_{team}(n) \subseteq EX_{freq}(\frac{1}{n})$.

2. $(\forall n \geq 1)(\forall p)\ \frac{1}{n+1} < p \leq 1 \Rightarrow EX_{freq}(p) \subseteq EX_{team}(n)$.

Theorem 3.35 asserts that the relationship between frequency *EX*–identification and team *EX*–identification is the same as that between probabilistic *EX*–identification and team *EX*–identification.

Proof:

The proof of the first part of the theorem is nearly identical to the proof of Theorem 3.32. If $U \in EX_{team}(n)$, and is *EX*–identified by the team $\{M_1, M_2, \ldots, M_n\}$, then we construct M which $EX_{freq}(\frac{1}{n})$–identifies U. On input $f \in U$, M simulates each M_i, and outputs their guesses in a round–robin fashion; its first n guesses being the first guesses of each team member, its second n guesses being the second guesses of each team member, etc. Since some M_i *EX*–identifies f, it follows by an argument nearly identical to that proving Theorem 3.32, that M *EX*–identifies f with frequency $\frac{1}{n}$.

We prove the second part of the theorem. Let $U \in EX_{freq}(p)$, with $p > \frac{1}{n+1}$. Let M be an IIM which *EX*–identifies U with frequency p. To show $U \in EX_{team}(n)$, we construct a team $M_1, M_2, \ldots, M_n$ of IIMs which *EX*–identify U. The idea behind the construction is the following: If $f \in U$, then we know that there is some correct guess g of M and (in the limit) the fraction of guesses of M which are "g" is greater than $\frac{1}{n+1}$. How many other distinct guesses of M can have this property? At most $n-1$, since the total number of distinct guesses of M which occur with limit frequency greater than $\frac{1}{n+1}$ can be at most n. Each member in the team of n IIMs will choose one of these, and output it. We must show that there is a single team member which "settles" on guessing the correct index, instead of having team members alternate guessing the correct index.

Let $FREQ_k = \{g_i \mid F_k(g_i) > \frac{1}{n+1}\}$. $FREQ_k$ is the set of guesses of M, which, if we look at the sequence of guesses through the k^{th} guess, occur frequently (*i.e.* $> \frac{k}{n+1}$ times). Clearly $|FREQ_k| \leq n$, and we note that since f is *EX*–identified by M with frequency $p > \frac{1}{n+1}$, there must exist a guess $g \in GOOD_f$ such that $(\forall_k^\infty)\ g \in FREQ_k$.

For each k, we define the function W_k which tells us for each $i \in FREQ_k$ at what point x in the sequence i first occurred as a guess with $F_x(i) > \frac{1}{n+1}$, and for each y such that $x \leq y \leq k$, $F_y(i) > \frac{1}{n+1}$.

More precisely,

$$W_k(i) = \begin{cases} k+1 & \textit{if } i \notin FREQ_k. \\ k & \textit{if } i \in FREQ_k - FREQ_{k-1}. \\ W_{k-1}(i) & \textit{otherwise.} \end{cases}$$

Clearly, for each i, $\{W_k(i)\}_{k\to\infty}$ is monotone nondecreasing. We now give the description of the machines M_i.

Machine M_i

1. On input $f|_k$, simulate M on input $f|_k$, and obtain the guesses $g_1, g_2, \ldots, g_k$.
2. Compute $FREQ_k$ and $W_k(s)$ for each $s \in FREQ_k$.
3. If $|FREQ_k| < i$ then output "0".
4. Otherwise, sort[4] the elements of $FREQ_k$ in order of increasing values of $W_k(s)$, and output the i^{th} element of the sorted set $FREQ_k$.

We must show that $(\forall f \in U)(\exists i)$ M_i EX–identifies f. If $f \in U$, then $(\exists g \in GOOD_f)$ $(\forall_k^\infty)$ $g \in FREQ_k$. We argue that g eventually occupies the same position in the ordered sets $FREQ_k$.

Since $(\forall_k^\infty)$ $g \in FREQ_k$, there must be a number k_0 such that $(\forall k \geq k_0)$ $W_k(g) \leq k_0$, by the definition of W_k. Let the function $pos(k)$ denote the position that g occupies in the ordering of elements in $FREQ_k$. So $(\forall k \geq k_0)$ $1 \leq pos(k) \leq n$. We claim that as k increases, $pos(k)$ is monotone nonincreasing. To see this, let us suppose that $pos(k)$ increases somewhere. This means that for some $k \geq k_0$, $pos(k) = j$, and $pos(k+1) = j+x$. The only way that this can happen is that there is some guess $h \in FREQ_{k+1}$ with $W_{k+1}(h) < W_{k+1}(g)$, and one of the following true:

1. $h \notin FREQ_k$.

2. $h \in FREQ_k$, and $W_k(h) > W_k(g)$.

If the first case holds, then by the definition of W_{k+1}, we must have that $W_{k+1}(h) = k+1 > k_0 \geq W_k(g) = W_{k+1}(g)$, which contradicts the fact that $W_{k+1}(h) < W_{k+1}(g)$. The second case cannot hold either, since by the definition of W_k, $W_{k+1}(h) = W_k(h)$ and $W_{k+1}(g) = W_k(g)$.

Hence pos is a monotone nonincreasing function of integers, bounded below by 1. It therefore has a limit $j, 1 \leq j \leq n$, and for all sufficiently large k, g will occupy the j^{th}

[4]It is easily seen that there can be no ties in this ordering, but this is unnecessary for the proof to follow, since ties could be broken by ordering on the actual value of the guess.

position in the ordered set $FREQ_k$. It follows that M_j will converge to outputting "g" as a guess. Hence M_j EX–identifies f, which completes the proof of Theorem 3.35. □

3.4.3 Frequency, Probability, and Team Hierarchies

Theorems 3.32, 3.33, and 3.35 relating frequency identification to team identification for both the EX and BC criteria are easily assembled to show that the frequency hierarchies are identical to the team hierarchies (which are identical to the probabilistic hierarchies). In particular, Theorems 3.28, 3.29, and Corollary 3.30 are all true if "frequency" is substituted for "probability".

We conclude that if ID is either of the criteria EX and BC, then

Theorem 3.36 $(\forall n \geq 1)(\forall p)\ \frac{1}{n+1} < p \leq \frac{1}{n} \Rightarrow ID_{freq}(p) = ID_{prob}(p) = ID_{team}(n).$

3.5 Invariance under Change of Definitions

In this section we consider a reasonable variation of the definition of probabilistic inference and show that the classes $EX_{prob}(p)$ and $BC_{prob}(p)$ remain unchanged. We also briefly consider a change of definition of team inference and show that $EX_{team}(n)$ and $BC_{team}(n)$ remain the same. This supports the position that the probabilistic and team inference classes defined are not arbitrary, but natural.

3.5.1 Probabilistic Inference Redefined

The definition of "Pr[P BC–identifies f] $\geq p$" was motivated by the notion that if you were to run P with input f many times (each run is infinite), the percentage of runs which resulted in correct BC–identifications would be at least p. In other words, the probability of obtaining a sequence of outputs which were **all correct** past some point would be at least p. A reasonable weakening of this definition would be to require only that past some point in the output sequence the probability of the next guess being correct is at least p. In other words, there is a value k_0 such that were you to run P on input f many times, then $(\forall k \geq k_0)$ the probability that P's k^{th} guess computes f is at least p. The corresponding notion for EX would be that there is some *particular* guess g such that the probability that the k^{th} guess of P on $f = g$ is at least p. We define this more precisely in terms of the computation tree $T_{P,f}$. Recall that for any set A, $L_k(A)$ is the set of nodes n at level k of $T_{P,f}$ such that $ind(n) \in A$.

Definition 3.37 *The probability measure* Pr′ *is defined by*

$$\Pr'[P\ BC\text{-identifies } f] = \liminf_{k\to\infty} \frac{|L_k(GOOD_f)|}{2^k}$$

$$\Pr'[P\ EX\text{-identifies } f] = \sup_{g\in GOOD_f} \left\{ \liminf_{k\to\infty} \frac{|L_k(\{g\})|}{2^k} \right\}$$

where sup *denotes the supremum, or least upper bound.*

Lemma 3.11 gives that

$$\Pr[P\ BC\text{-identifies } f] > p \Rightarrow (\forall_k^\infty)|L_k(GOOD_f)| > p2^k,$$

so

$$\Pr[P\ BC\text{-identifies } f] \leq \Pr'[P\ BC\text{-identifies } f].$$

The same could be shown for EX with a trivial modification of Lemma 3.11. Thus Pr′ is a "weaker" definition of probability.

Since the definition of Pr requires a relationship between the locations of correct programs at different levels of $T_{P,f}$ (namely that they lie beneath one another and form converging paths), it is not necessarily the case that

$$\Pr'[P\ ID\text{-identifies } f] \leq \Pr[P\ ID\text{-identifies } f]$$

for $ID \in \{EX, BC\}$.

We show that there are indeed computation trees for which the different definitions of probability separate.

Theorem 3.38 *There is a probabilistic IIM P and a function f such that*

$$\Pr'[P\ EX\text{-identifies } f] = \Pr'[P\ BC\text{-identifies } f] = 1$$

but

$$\Pr[P\ EX\text{-identifies } f] = \Pr[P\ BC\text{-identifies } f] = 0.$$

Proof: Clearly

$$\Pr[P\ BC\text{-identifies } f] \geq \Pr[P\ EX\text{-identifies } f]$$

and

$$\Pr'[P\ BC\text{-identifies } f] \geq \Pr'[P\ EX\text{-identifies } f]$$

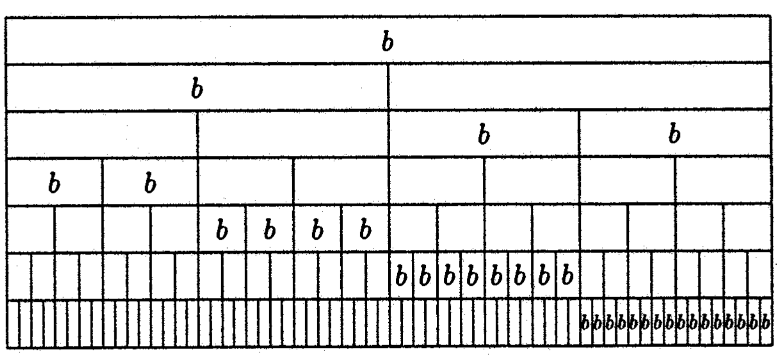
so we only need show that P EX-identifies every f with probability 1 (in the new sense of Pr′) and BC-identifies f with probability 0 (in the old sense).

Let f be any total recursive function, and let P have "hardcoded" into its program an index $g \in GOOD_f$ and $b \in BAD_f$. We describe the behavior of P on input f by giving an effective assignment of $ind(n)$ to every node n of the computation tree $T_{P,f}$. The value $ind(n)$ will either be g or b for every node n in the tree. The construction guarantees that the limit infimum of the fraction of guesses which are g at each level is 1, but at the same time ***every*** path in $T_{P,f}$ will contain an infinite number of b's — thus there will not be a single BC–identifying path.

We describe the values $\{ind(n)\}$ level by level. When we write "level $k = g^i, b^j, g^l, \ldots$ we mean that the first i nodes across level k (going left to right) have $ind(n) = g$, the next j nodes have $ind(n) = b$, the next l nodes have $ind(n) = g$, etc. Level $0 = b$. The next two levels will have $\frac{1}{2}$ of the nodes labeled b, and $\frac{1}{2}$ of the nodes labeled g, but the halves will be staggered so that level 1 is labeled b, g and level 2 is labeled g^2, b^2. The next four levels will have $\frac{1}{4}$ of the nodes labeled b and $\frac{3}{4}$ labeled g, again staggering the h's so that they "march across the tree."

- Level $3 = b^2, g^2, g^2, g^2$
- Level $4 = g^4, b^4, g^4, g^4$
- Level $5 = g^8, g^8, b^8, g^8$
- Level $6 = g^{16}, g^{16}, g^{16}, b^{16}$

The next eight levels have $\frac{1}{8}$ of the nodes labeled b, again staggered across the tree. The construction continues in this manner. The following picture shows the first several levels of the tree, with the b's written in, and the g's left out for clarity.



Clearly by construction every path in the tree has infinitely many nodes with index b. Furthermore, by construction,

$$\liminf_{k\to\infty} \frac{|L_k(\{g\})|}{2^k} = 1,$$

since $(\forall \epsilon > 0)(\forall_k^\infty)\ \frac{|L_k(\{g\})|}{2^k} > 1 - \epsilon$. Thus there is no BC–identifying path, and we have $Pr'[P\ EX\text{–identifies } f] = 1$. □

The above result can be extended to show that if $U \in EX$ then there is a P which EX–identifies every $f \in U$ with probability (in the new sense) $= 1$, and BC–identifies f with probability (in the old sense) $= 0$. The machine P on input $f \in U$ would first pick any $b \in WRONG_f$ (the null program will do), then simulate the machine M such that $U \subseteq EX(M)$. The levels of $T_{P,f}$ would be as above, except at level k the value g would be replaced by the last value of the sequence $M(f)$ which P was able to compute within k steps.

Thus the two probability measures are indeed different. We now consider the "new" classes $BC_{prob'}(p)$ and $EX_{prob'}(p)$ defined exactly as are $BC_{prob}(p)$ and $EX_{prob}(p)$ except with the measure Pr′ instead of Pr. In light of Theorem 3.38 we are surprised to find

Theorem 3.39 $(\forall p)$ $BC_{prob'}(p) = BC_{prob}(p)$ *and* $EX_{prob'}(p) = EX_{prob}(p)$.

Proof: Clearly $BC_{prob}(p) \subseteq BC_{prob'}(p)$ since $(\forall P)(\forall f)$ $\Pr[P\ BC\text{–identifies } f] \le \Pr'[P\ BC\text{–identifies } f]$. We show $BC_{prob}(p) \supseteq BC_{prob'}(p)$. Let $U \in BC_{prob'}(p)$. Consider the least n such that $p \in (\frac{1}{n+1}, \frac{1}{n}]$. Since $p > \frac{1}{n+1}$,

$$\liminf_{k\to\infty} \frac{|L_k(GOOD_f)|}{2^k} > \frac{1}{n+1}$$

and

$$(\forall_k^\infty)\ |L_k(GOOD_f)| > \frac{2^k}{n+1}.$$

The proof of Theorem 3.8 needs only this last inequality to prove that $U \in BC_{team}(n)$. Thus

$$BC_{prob'}(p) \subseteq BC_{team}(n) \subseteq BC_{prob}(\frac{1}{n}) \subseteq BC_{prob}(p),$$

the last two containments due to Corollary 2.24 and the fact that $p < \frac{1}{n}$.

That $EX_{prob}(p) \subseteq EX_{prob'}(p)$ is not as straightforward, since there can be *many* hypotheses which make up the fraction p of converging paths. In fact "many" is at most n if $p > \frac{1}{n+1}$.

Let $p \in (\frac{1}{n+1}, \frac{1}{n}]$. Then since $EX_{prob}(p) \subseteq EX_{team}(n)$ (Theorem 3.21), if $U \in EX_{prob}(p)$ then there is a team $\{M_1, M_2, \ldots, M_n\}$ such that $(\forall f \in U)(\exists i)$ M_i EX–identifies f. Now consider the probabilistic IIM P which chooses with probability $\frac{1}{n}$ to simulate M_i for each i. Then $(\forall f \in U)(\exists g \in GOOD_f)\ \Pr[P(f)\downarrow= g] \geq \frac{1}{n}$. It follows from Lemma 3.11(with the set "A" $= \{g\}$) that $(\forall q < \frac{1}{n})(\forall_k^\infty)$ the fraction of nodes at level k which have index $= g$ is $> q$. Thus the limit infimum of that fraction is $\frac{1}{n}$, and $U \in EX_{prob'}(\frac{1}{n})$. So $EX_{prob}(p) \subseteq EX_{prob'}(\frac{1}{n})$ and $EX_{prob'}(\frac{1}{n}) \subseteq EX_{prob'}(p)$ (since $p \leq \frac{1}{n}$).

To see that $EX_{prob}(p) \supseteq EX_{prob'}(p)$ consider the least n such that $p \in (\frac{1}{n+1}, \frac{1}{n}]$. We show

$$EX_{prob}(p) \supseteq EX_{prob}(\frac{1}{n}) \supseteq EX_{team}(n) \supseteq EX_{prob'}(p).$$

The first two containments follow from the fact that $p \leq \frac{1}{n}$ and Theorem 3.21 respectively. We show $EX_{team}(n) \supseteq EX_{prob'}(p)$. The proof is nearly identical to the proof of Theorem 3.35, so we give only a sketch.

Let $U \in EX_{prob'}(p)$ with $p > \frac{1}{n+1}$. We construct $\{M_1, M_2, \ldots, M_n\}$. Let $f \in U$. From the definition of $EX_{prob'}(p)$ there must be a $g \in GOOD_f$ such that

$$\liminf_{k\to\infty} \frac{|L_k(\{g\})|}{2^k} > \frac{1}{n+1}.$$

This gives that

$$(\forall_k^\infty)\ \frac{|L_k(\{g\})|}{2^k} > \frac{1}{n+1}.$$

Now each M_i simulates P while receiving values of f and builds $T_{P,f}$ level by level. M_i keeps a queue Q of the at most n guesses which satisfy $|L_k(\{g\})| > \frac{2^k}{n+1}$. Then Q is ordered by how many previous levels the guess satisfied the inequality. M_i outputs the i^{th} element (if it exists) of Q. Since eventually some guess g enters Q and never leaves, it eventually moves up in Q to some position $pos(g)$ and never moves again. Then $M_{pos(g)}$ EX–identifies f. □

3.5.2 Team Inference Redefined

The apparent modification in the definition of probabilistic identification motivates a similar change of definition for team inference which has yet to be considered. As one motivation of his definition of team inference (the one we have been using) Smith [35] writes

> ... in their study of the scientific method the philosophers seem to regard a scientific community as a whole. Indeed, when several researchers are investigating the same phenomenon, the results of every experiment are eventually known to all the researchers. Often there is no consensus on the proper

explanation of the phenomenon under investigation. The theory presented below provides a model of the scientific method admitting several, possibly contradictory, opinions as to an explanation of some phenomenon.

Thus, the bold assertion "science is correct" might be interpreted as "for every phenomenon there is at least one scientist who ultimately believes a particular (correct) theory." However, as we well know, scientists are a fickle lot, and many are not faithful to old or current theories as they receive new information, discuss research with their colleagues, and in general move on to sexier and trendier hypotheses.

Despite the possible infidelity of scientists, a particular theory might be so appealing that there always seems to be at least one scientist lobbying its cause, but not so appealing that there is a single crusader who refuses to move on to other theories. With this motivation, we relax our interpretation and say that "science is correct" for a particular phenomenon if past some point in time, there is some particular correct theory which is always in the current literature.

Definition 3.40 $\{M_1, M_2, \ldots, M_n\} EX_{team'}$*–identify* U *iff*
$(\forall f \in U)(\exists g \in GOOD_f)(\forall_k^\infty)\ (\exists i)\ M_i$*'s* k^{th} *guess is* g.

Thus the team members can switch among them which is guessing g as long as past some point in their output sequences, as least one team member outputs g. We have a corresponding definition for BC.

Definition 3.41 $\{M_1, M_2, \ldots, M_n\} BC_{team'}$*–identify* U *iff*
$(\forall f \in U)(\forall_k^\infty)\ (\exists i)\ M_i$*'s* k^{th} *guess is an element of* $GOOD_f$.

We now show

Theorem 3.42 $(\forall n)\ EX_{team'}(n) = EX_{team}(n)$ *and* $BC_{team'}(n) = BC_{team}(n)$.

Proof: (EX) Clearly $EX_{team'}(n) \supseteq EX_{team}(n)$. We show $EX_{team'}(n) \subseteq EX_{team}(n)$. Consider M which outputs the guesses of $M_1, M_2, \ldots, M_n$ in a round-robin order, as in the proof of Theorem 3.35. We immediately have $EX_{team'}(n) \subseteq EX_{freq}(\frac{1}{n}) \subseteq EX_{team}(n)$.

(BC) Clearly $BC_{team'}(n) \supseteq BC_{team}(n)$. To see that $BC_{team'}(n) \subseteq BC_{team}(n)$, use the same construction as above to get $BC_{team'}(n) \subseteq BC_{freq}(\frac{1}{n}) \subseteq BC_{team}(n)$. □

3.6 Other Properties of Probabilistic IIMs.

Throughout this section, "identify" refers to both EX and BC identification, and ID denotes both EX and BC.

3.6.1 Identifying Functions Drawn from a Hat

In the models of identification presented so far, we have assumed that functions were taken from some set U, and we have been interested in when there are IIMs (deterministic, probabilistic, nondeterministic,...) which can identify the function. Suppose that we know *a priori* that the function being presented to M is chosen randomly from $\mathcal{T}$ according to some known probability distribution. This might be the case for scientists having certain empirical evidence suggesting that the rules governing observed behavior occur randomly with certain probabilities.

We now ask the following question: Are probabilistic IIMs better ***on the average*** than deterministic IIMs at identifying functions?

Let $D : \mathcal{T} \rightarrow [0,1]$ be a probability distribution which assigns to every total recursive function f, a real number in [0,1] such that $\sum_{f \in \mathcal{T}} D(f) = 1$. Note that since $\mathcal{T}$ is a countable set, the distribution D is discrete. Let P be a probabilistic IIM, and M a deterministic IIM. Define $M(f)$ to be 1 if M identifies f, 0 otherwise, and $P(f)$ to be the probability that P identifies f.

Then the ***average performance***, $A(M,D)$ of M with respect to D is defined by

$$A(M,D) = \sum_{f \in \mathcal{T}} D(f) \cdot M(f)$$

and the average performance of P is

$$A(P,D) = \sum_{f \in \mathcal{T}} D(f) \cdot P(f).$$

Theorem 3.43 *For all distributions D on $\mathcal{T}$, and for all probabilistic IIMs P, there exists a deterministic IIM M such that $A(M,D) \geq A(P,D)$.*

Proof: There are two cases:

- (Case 1) $(\forall f)\ D(f) > 0 \Rightarrow P(f) = 1$.
- (Case 2) $(\exists f)\ D(f) > 0$ and $P(f) < 1$.

If Case 1 holds, then let $U = \{f \mid P(f) = 1\} \supseteq \{f \mid D(f) > 0\}$. Then since P identifies U with probability $1 > \frac{1}{2}$, by Theorem 3.28 there is a deterministic IIM M which identifies U, and thus

$$A(M,D) = \sum_{f \in U} D(f) = 1.$$

If Case 2 holds, then $A(P, D) = 1 - \epsilon$ for some $\epsilon > 0$. Now since $\sum_{f \in T} D(f) = 1$, there exists a finite number of *distinct* functions $\{f_1, f_2, \ldots, f_k\}$ such that $\sum_{i=1}^{k} D(f_i) > 1 - \frac{\epsilon}{2}$. Then there is a deterministic IIM M, which has "built in" a list of the indices of these functions. When given examples of a function f to be identified, M asks for enough values until it witnesses that all but one of the functions $\{f_1, f_2, \ldots, f_k\}$ differ from f, and then M outputs the index of the remaining function. (M outputs the index "0" while it eliminates the above functions.) Clearly M identifies each of the functions $\{f_1, f_2, \ldots, f_k\}$ and it follows that

$$A(M, D) > 1 - \frac{\epsilon}{2} > 1 - \epsilon = A(P, D)$$

and the theorem follows. □

We leave as an interesting question whether or not M can effectively be constructed from P. (In the event that P uses its coin only to make an initial choice between simulating one of a finite number of deterministic strategies $\{M_1, M_2, \ldots, M_n\}$, then there is some i such that $A(M_i, D) \geq A(P, D)$) [32]).

3.6.2 Restricted Choice Probabilistic IIMs

Consider probabilistic IIMs which use their coin only to initially choose to simulate one of a finite number of deterministic alternatives. Is there a difference between these probabilistic IIMs and probabilistic IIMs which are not of this special form? We call the former type of IIM ***restricted choice probabilistic IIMs***, and the latter ***unrestricted choice probabilistic IIMs***.

The criterion for successful probabilistic identification which we have used thus far has only been concerned with whether the probability of identification is above some threshold (p.) Within this framework, our results imply that restricted choice probabilistic IIMs are as powerful as unrestricted choice probabilistic IIMs; for if P is any probabilistic IIM which identifies the set U of functions with probability p, then consider the least positive integer n such that $\frac{1}{n+1} < p \leq \frac{1}{n}$. Then by part 2 of Theorem 3.28 there is a team of n deterministic IIMs identifying U, and by part 1 of Theorem 3.28 there is a restricted choice probabilistic IIM which identifies the set U with probability $\frac{1}{n} \geq p$.

Suppose now that we are concerned with how well a probabilistic IIM identifies ***every*** function. For every unrestricted choice probabilistic IIM P_u does there exist a restricted choice probabilistic IIM which identifies every total recursive function with probability at least as great as P_u? The answer is "no".

Theorem 3.44 *There exists an unrestricted choice probabilistic IIM P_u such that for any restricted choice probabilistic IIM P_r, there exists a total recursive function f such that* $\Pr[P_r$ *identifies* $f] < \Pr[P_u$ *identifies* $f]$.

Proof: P_u uses coin flips in such a way that P_u guesses the index "0" with probability $\frac{1}{2}$, "1" with probability $\frac{1}{4}, \ldots$, "n" with probability $\frac{1}{2^{n+1}}$. Thus the probability that P_u identifies any given total recursive function is greater than 0. Suppose there was a restricted choice probabilistic IIM P_r which chooses from deterministic strategies $\{M_1, M_2, \ldots, M_k\}$ with probabilities $\{p_1, p_2, \ldots, p_k\}$, respectively. Then if for all f, $Pr[P_r$ identifies $f] \geq \Pr[P_u$ identifies $f]$, it must be the case that (for all f) $\Pr[P_r$ identifies $f] > 0$. It follows that $\mathcal{T} \subseteq \bigcup_{i=1}^{k} ID(M_i)$ violating the team hierarchy theorem (Theorem 2.15). □

Chapter 4

Other Identification Criteria

4.1 Identification with Anomalous Hypotheses

Allowing randomization and some probability of error for identification is only one possible way to expand the classes of functions which are identifiable. Another manner in which the definition for correct identification may be relaxed is that of allowing the hypothesized programs to disagree with the function being identified on some number of arguments.

4.1.1 Anomalous EX Identification

For each $a \in \mathrm{N} \bigcup \{*\}$ the identification criterion EX^a has been introduced [10].

Definition 4.1 *Let M be a deterministic IIM, and $a \in \mathrm{N} \bigcup \{*\}$. Then M EX^a–identifies f iff when fed the graph of f in any order, $M(f) \downarrow = i$ and $\varphi_i =^a f$.*

Our definition for the EX^a criterion is the pair (B_{EX^a}, Q_{EX^a}) where B_{EX^a} is always satisfied, and $Q_{EX^a}(g|_k, f) = 1 \Leftrightarrow g_{k-1} = g_k$ and $\varphi_{g_k} =^a f$. Clearly $Q_{EX^a}(g|_k, f)$ is limiting–invariant under repetitions in the sequence g. Thus all of the definitions of Section 2.5.2 apply for probabilistic EX^a–identification, as well as the definitions of the trees $T_{P,f}$ in Section 2.5.3.

Note that if M EX^k–identifies f, the program φ to which M converges need not be total, *i.e.* φ could differ from f because $\varphi(x)$ is undefined, whereas $f(x)$ is defined.

The criterion EX^* (*i.e.* $a = *$) is the same as that of *a.e. identification* introduced in [7], and *sub–identification* in [28].

The following theorem is proved in [10].

Theorem 4.2 $(\forall k \in \mathrm{N})\ EX^{k+1} - EX^k \neq \emptyset$ *and* $EX^* - \bigcup_{k\in\mathrm{N}} EX^k \neq \emptyset$.

Smith [35] considers *team* inference for EX^a and shows that

$$(\forall a \in \mathrm{N}\bigcup\{*\})(\forall n \geq 1)\ EX^a_{team}(n) \subset EX^a_{team}(n+1).$$

Interesting tradeoffs are also given between the number of team members, the number of anomalies, and a complexity measure — the number of "mind changes" made by an IIM before converging to a correct program. Discussion of these tradeoffs are beyond the scope of this dissertation; the reader is encouraged to consult [35] for further details.

We define frequency EX^a–identification in the natural way.

Definition 4.3 *If M is a deterministic IIM, and on input f, M outputs the sequence of guesses $g_1, g_2, \ldots$, and $a \in \mathrm{N}\bigcup\{*\}$ then*

- *M EX^a–identifies f with frequency p iff* $(\exists g_i)\ \liminf_{k\to\infty} F_k(g_i) \geq p$ *and* $\varphi_{g_i} =^a f$.
- $EX^a_{freq}(p) = \{U \mid (\exists M)(\forall f \in U)$ *M EX^a–identifies U with frequency p*$\}$.

We now state

Theorem 4.4 $(\forall a \in \mathrm{N}\bigcup\{*\})(\forall n \geq 1)(\forall p)$
$\frac{1}{n+1} < p \leq \frac{1}{n} \Rightarrow EX^a_{prob}(p) = EX^a_{freq}(p) = EX^a_{team}(n)$.

Before we prove Theorem 4.4, we give the generalizations of the definition of the class OEX, and Lemma 3.25 which appear in [10].

Definition 4.5 $(\forall a \in \mathrm{N}\bigcup\{*\})$

- *M OEX^a–identifies f (written $f \in OEX^a(M)$) iff M, when fed the graph of f in any order, outputs an infinite sequence $\{I_k\}$ of finite lists, and there is a list I such that* $(\forall^\infty_k)\ I_k = I$, *and* $(\exists i \in I)\ \varphi_i =^a f$.
- $OEX^a = \{U \mid (\exists M)\ U \subseteq OEX^a(M)\}$.

Lemma 4.6 [10]

1. $(\forall k \in \mathrm{N})\ OEX^k = EX^k$.
2. $OEX^* - EX^* \neq \emptyset$.

The proof of part 1 of Lemma 4.6 is similar in spirit to the proof of Lemma 3.25.

To prove Theorem 4.4, we first note that the proof of Theorem 3.35 does not involve any simulation of the hypothesized programs, hence by simply substituting EX^a for

EX, EX^a_{freq} for EX_{freq}, and EX^a_{team} for EX_{team}, we have proved that if the hypothesis of Theorem 4.4 holds, then $EX^a_{freq}(p) = EX^a_{team}(n)$.

Now we show that $EX^a_{prob}(p) = EX^a_{team}(n)$ for $\frac{1}{n+1} < p \leq \frac{1}{n}$ and for $a \in \mathbf{N} \bigcup \{*\}$. We have that $EX^a_{team}(n) \subseteq EX^a_{prob}(\frac{1}{n})$ by Theorem 2.23. We now only need show that for $p > \frac{1}{n+1}$, $EX^a_{prob}(p) \subseteq EX^a_{team}(n)$.

Consider the case that $a \in \mathbf{N}$. Then by part 1 of Lemma 4.6, the analogues of Theorem 3.21 and Lemma 3.26 with a anomalies all hold, and we have that $EX^a_{prob}(p) \subseteq EX^a_{team}(n)$ when $p > \frac{1}{n+1}$.

These proofs do not work however, to show the corresponding result for any finite number of anomalies, since by part 2 of Lemma 4.6, simply converging to a list of programs containing at least one finite variant of f is not sufficient. We must employ other techniques.

We show that

Lemma 4.7 $(\forall n \geq 1)(\forall p)\ \frac{1}{n+1} < p \Rightarrow EX^*_{prob}(p) \subseteq EX^*_{team}(n)$.

Let $U \in EX^*_{prob}(p)$, and let P EX^*–identify U with probability $p > \frac{1}{n+1}$. We construct a team $M_1, M_2, \ldots, M_n$ which EX^*–identifies U.

Each member M_i of the team will proceed in phases. On input $f|_k$, M_i simulates P and constructs T_k, the finite subtree of $T_{P,f}$ through level k. M_i will keep a priority queue Q from phase to phase. At phase k Q will contain the nodes of T_k in some order. M_i will simulate the guesses made by P and order Q roughly by how many anomalies each has been observed to have. Since every program which converges $\neq f$ for infinitely many arguments will be pushed to the end of the queue infinitely often, M_i will be able to eliminate these guesses.

We denote the j^{th} element of the queue by $Q(j)$. $Q(1)$ is the beginning, or top of the queue. Q starts out empty.

Phase k of M_i

1. Receive the k^{th} input value $f(k)$, and build the k^{th} level of $T_{P,f}$ by simulating P.
2. Add j to the end of Q for each node j in the k^{th} level of $T_{P,f}$.
3. Allowing k steps for each computation, try to compute each of $\{\varphi_{ind(j)}(x) \mid j \in Q$, and $x \leq k\}$.
   For each j such that a value $x \leq k$ is found (within k steps of simulation) such that $\varphi_{ind(j)}(x) \neq f(x)$, and this inequality was not witnessed in any previous phase, move j to the end of Q.

4. Compute $\Pr[C_{j,k}]$ for each $j \in Q$, and let $I_k = \{ind(Q(j)) \mid 1 \leq j \leq c_k\}$, where c_k is the smallest number such that

$$\sum_{j=1}^{c_k} \Pr[C_{Q(j),k}] \geq \frac{i}{n+1}.$$

I_k is simply the indices of the smallest initial set of nodes, ordered by Q, which have total estimated probability $\geq \frac{i}{n+1}$.

5. Output the index of the program $RACE_{I_k}$.

We must show that $(\forall f \in U)(\exists i)$ M_i EX^*–identifies f. We define

- $GOOD^*_f = \{i \mid \varphi_i =^* f\}$.
- $WRONG^*_f = \{i \mid \{x : \varphi_i(x) \downarrow\neq f(x)\}$ is infinite$\}$.
- $SLOW^*_f = \mathrm{N} - (GOOD^*_f \bigcup WRONG^*_f)$.

Clearly $GOOD^*_f$, $WRONG^*_f$, and $SLOW^*_f$ partition N. Note that $SLOW^*_f$ consists of those indices such that the corresponding program is not a finite variant of f, but there are at most finitely many arguments for which it converges $\neq f$.

Let $GS = GOOD^*_f \bigcup SLOW^*_f$. Then $\Pr[C(GS)]$ is in some half open interval $(\frac{j}{n+1}, \frac{j+1}{n+1}]$ for some j, $1 \leq j \leq n$. Suppose that it falls in the interval $(\frac{i}{n+1}, \frac{i+1}{n+1}]$, then we show that M_i EX^*–identifies f, proving Lemma 4.7.

We will show that the sequence of lists $\{I_k\}$ in program M_i converges to a list I, such that $I \subseteq GS$, and $I \bigcap GOOD^*_f \neq \emptyset$. If this is the case, then M_i converges to outputting the index of a fixed program $RACE_I$. Then observe that $RACE_I =^* f$ since I contains at least one element of $GOOD^*_f$, a finite number of programs in $SLOW^*_f$, each of which converges $\neq f$ in only finitely many places, and no programs which converge $\neq f$ for infinitely many inputs.

Now, since $\Pr[C(GS)] > \frac{i}{n+1}$ by assumption on i, by Lemma 3.19 there must be a *finite* collection of nodes V such that

$$\sum_{j \in V} \Pr[C_j] > \frac{i}{n+1}, \text{ and } (\forall j \in V)\ ind(j) \in GS.$$

Note that if $j \in GS$, then there are only finitely many arguments for which $\varphi_j \downarrow\neq f$. Thus M_i will move each node in V to the end of Q at most finitely many times. Also, if $m \in Q$, and $ind(m) \in WRONG^*_f$, then m will be moved to the end of Q infinitely many times. It follows that (for all sufficiently large k) the order of elements from the beginning of Q to the highest numbered position v of Q which contains an element of V will remain constant.

Further, for all sufficiently large k, by Lemma 3.18,

$$\sum_{j\in V} \Pr[C_{j,k}] \geq \sum_{j\in V} \Pr[C_j] > \frac{i}{n+1}$$

and we have that $(\forall_k^\infty)\ c_k \leq v$.

Finally, by an argument similar to that in the proof of Lemma 3.26, the sequence $\{c_k\}$ converges to s, where s is the smallest value $1 \leq s \leq v$ such that

$$\sum_{j=1}^{s} \Pr[C_{Q(j)}] \geq \frac{i}{n+1}.$$

Then the sequence of lists $\{I_k\}$ converges to $I = \{ind(Q(j)) \mid 1 \leq j \leq s\}$. Clearly $I \subseteq GS$. Now suppose that $I \bigcap GOOD_f^* = \emptyset$. Then

$$\frac{i}{n+1} \leq \sum_{j=1}^{s} \Pr[C_{Q(j)}] \leq \Pr[C(I)] \leq \Pr[C(SLOW_f^*)].$$

Also, since $\Pr[C(GOOD_f^*)] > \frac{1}{n+1}$, it follows that $\Pr[C(GS)] > \frac{i+1}{n+1}$, which contradicts our assumption on i. Thus M_i converges to outputting the index of $RACE_I$ which computes a finite variant of f, and Lemma 4.7 and Theorem 4.4 follow. □

4.1.2 Anomalous BC Identification

Identification with anomalous hypotheses has been studied for BC–identification as well.

Definition 4.8 *Let M be a deterministic IIM, $a \in \mathrm{N} \bigcup \{*\}$. Then M BC^a–identifies f iff when fed the graph of f in any order, M outputs the infinite sequence $g_1, g_2, \ldots$ and $(\forall_k^\infty)\ g_k =^a f$.*

Analogous with our definition of EX^a as an identification criterion, our definition for the BC^a criterion is the pair (B_{BC^a}, Q_{BC^a}) where B_{BC^a} is always satisfied, and $Q_{BC^a}(g|_k, f) = 1 \Leftrightarrow \varphi_{g_k} =^a f$. Clearly $Q_{BC^a}(g|_k, f)$ is limiting–invariant under repetitions in the sequence g. Thus all of the definitions of Section 2.5.2 apply for probabilistic BC^a–identification, as well as the definitions of the trees $T_{P,f}$ in Section 2.5.3.

Case and Smith [10] show that $BC^{k+1} - BC^k \neq \emptyset$ and $BC^* - \bigcup_{k\in \mathrm{N}} BC^k \neq \emptyset$.

L. Harrington has shown that BC^* contains the class of partial recursive functions (this result appears in [10]), so there is no reason to consider more general (probabilistic, team, frequency) criteria for BC^*. In [35], Smith gives definitions for BC–identification with a finite number of anomalies by teams. It is shown that there is a proper hierarchy $(\forall k)(\forall n \geq 1)\ BC^k_{team}(n) \subset BC^k_{team}(n+1)$.

Daley [13] proves interesting tradeoffs, analogous to those shown for EX in [35], relating number of team members, number of anomalies, and number of mind changes required for BC–identification.

BC^k ($k \in \mathrm{N}$) is a limiting repetition–invariant identification criterion, so all of our definitions of probabilistic identification apply.

We define BC–frequency identification with anomalies. We say M BC^k–identifies f with frequency p iff the limit infimum of the fraction of guesses output by M which $=^k f$ is at least p.

Definition 4.9 $BC^k_{freq}(p) = \{U \mid (\exists M)(\forall f \in U)\ M\ BC^k\text{–}\textit{identifies } U \textit{ with frequency } p\}$.

It seems appropriate to form the following

Conjecture 4.10 $(\forall k)(\forall n \geq 1)(\forall p)$
$\frac{1}{n+1} < p \leq \frac{1}{n} \Rightarrow BC^k_{prob}(p) = BC^k_{team}(n) = BC^k_{freq}(p)$.

4.1.3 An Anomalous Corollary

The following corollary was pointed out to us by C. Smith [36].

Corollary 4.11 $(\forall k \in \mathrm{N})\ EX^k \subseteq EX_{prob}(\frac{1}{k+1})$.

Proof: From [35] we have $EX^k \subseteq EX_{team}(k+1)$. By Corollary 2.24, $EX_{team}(k+1) \subseteq EX_{prob}(\frac{1}{k+1})$. □

Alternate Proof:

Let

$$EX^{=k} = \{U \mid (\exists M)(\forall f \in U)\ M(f)\downarrow= g \text{ and } |\{x : \varphi_g(x) \neq f(x)\}| = k\}.$$

It has been shown [10] that $(\forall k)\ EX^{=k} = EX$. (The idea behind the proof is that an IIM M' which EX–identifies can be constructed from an IIM M which $EX^{=k}$–identifies by simulating M, obtaining its most recent guess, and altering it by correcting (if a wrong value has been witnessed) and/or "patching" the correct value in (if the guess has failed to converge) for the k spots where the current hypothesis of M appears to be incorrect.)

Lemma 4.12 $(\forall U \in EX^k)(\exists U_0, U_1, \ldots, U_k)$ *with* $(\forall i)(U_i \in EX^{=i})$ *and* $U = \biguplus_{i=0}^{k} U_i$.

Proof: If M EX^k-identifies U then $(\forall f \in U)$ $M(f) \downarrow= g$ and $0 \le |\{x : \varphi_g(x) \neq f(x)\}| \le k$. Let $U_i = \{f : |\{x : \varphi_g(x) \neq f(x)\}| = k\}$. □

To see that $EX^k \subseteq EX_{prob}(\frac{1}{k+1})$, let $U \in EX^k$. Then by Lemma 4.12 $U = \biguplus_{i=0}^{k} U_i$ with $U_i \in EX^{=i}$. Then let P be a probabilistic IIM which on input f flips a $k+1$–sided coin and guesses that $f \in U_i$. P then simulates the M which witnesses that $U \in EX^{=i}$.

□

We also have

Corollary 4.13 $(\forall k \in \mathrm{N})$ $BC^k \subseteq BC_{prob}(\frac{1}{k+1})$.

Proof: From [13] we have $BC^k \subseteq BC_{team}(k+1)$. By Corollary 2.24, $BC_{team}(k+1) \subseteq BC_{prob}(\frac{1}{k+1})$. □

4.2 Reliable Inference Strategies

We have mentioned that team inference may be viewed as nondeterminism restricted to choosing from among a finite number of deterministic strategies. We now consider unrestricted nondeterministic IIMs, and give a simple argument showing why this model is too powerful to be interesting. We then consider a type of behavioral restriction, that of "reliability" [7], and show that reliable nondeterministic IIMs are no more powerful than deterministic IIMs.

Consider the classes EX_{nondet} and BC_{nondet} given by Definition 2.27. Since Q_{EX} and Q_{BC} are limiting–invariant under repetition, we assume without loss of generality that all nondeterministic IIMs behave nicely, that for any nondeterministic IIM N and function f the computation tree $T_{N,f}$ is defined as for probabilistic IIMs, as are the definitions of any set of paths in a tree $T_{N,f}$ which may have been defined in the last chapter.

Thus a nondeterministic IIM N EX–(BC–)identifies the function f iff there exists at least one path in $T_{N,f}$ corresponding to a single deterministic EX–(BC–)identification of f. It is immediately clear that there is a single nondeterministic IIM N which EX–(and hence BC–)identifies $\mathcal{T}$, the class of *all* total recursive functions (in fact N identifies every partial recursive function): N nondeterministically receives a sequence of bits from its oracle. N prints every odd numbered bit it receives on a work tape, until an even numbered bit is received which is a "1". The binary number written on the work tape is used as the guess for an index of f, and N simply guesses that index at every step from then on. Clearly every possible number can be generated by N nondeterministically in this manner, so there is a computation of N which EX–(BC–)identifies any $f \in \mathcal{T}$

(without even seeing a value)! Thus unrestricted nondeterminism is too powerful a model to be of interest.

For EX–identification, a natural restriction for IIMs is that of ***reliability***.[1] An IIM M is ***reliable*** (on $\mathcal{T}$) iff $(\forall f \in \mathcal{T})\ M(f) \downarrow= g \Rightarrow \varphi_g = f$. Reliable IIMs are simply IIMs which cannot mislead us by converging to a wrong value. Reliable inference strategies have been studied in [7,10,28].

The identification criterion REX is defined by the pair (B_{REX}, Q_{EX}), where $B_{REX}(M) = 1$ iff M is reliable.[2]

Minicozzi [28] showed that the class REX is closed under finite and recursively enumerable union. What we show below is that REX is closed under a certain type of ***uncountable*** union.

We consider the following question about reliable nondeterministic IIMs: Are they too powerful, as are nondeterministic IIMs, or does the reliability restriction prohibit the type of unlimited guessing that allowed a single nondeterministic IIM to identify all $f \in \mathcal{T}$? We are surprised to find that

Theorem 4.14 $REX_{nondet} = REX \subset EX$.

Hence the class of reliably–nondeterministic–identifiable subsets of $\mathcal{T}$ is properly contained in EX, showing that reliability is too strong a restriction for nondeterministic IIMs to yield interesting identifiability classes.

Proof:

The proper containment follows from results in [10]. Clearly $REX \subseteq REX_{nondet}$. We show $REX_{nondet} \subseteq REX$.

Let N be a reliable nondeterministic IIM which REX–identifies the set U of functions. We construct a deterministic IIM M which REX–identifies U. M, given values from the graph of f, constructs $T_{N,f}$ level by level and makes a list of nodes. On input $f|_k$ M constructs T_k, the finite tree consisting of the first k levels of $T_{N,f}$, and then determines for each node n of T_k whether $C_{n,k}$ is empty. (Note that this computation depends only on the nodes through level k in $T_{N,f}$ – see Lemma 3.20). M then outputs the index of the least numbered node n such that $C_{n,k} \neq \emptyset$.

[1] Also called ***strong*** in [28]. Reliability is not a meaningful notion for BC–identification.

[2] We define B_{REX} more precisely, as in Section 2.4. $B_{REX}(IO_{M,s}) = 1$ iff either $s \in \mathcal{G}(f)$ for some $f \in \mathcal{T}$ and the "output" elements of s converge to an element of $GOOD_f$, or else s is not in $\mathcal{G}(f)$ for any total recursive function.

M attempts to find the *first* converging path. We argue that since $T_{N,f}$ must contain at least one converging path, and no converging path can converge to a wrong index (N is reliable), M must be correct:

Since N nondeterministically REX–identifies U, for every function $f \in U$ there is at least one path (oracle) in $T_{N,f}$ which converges to a correct index for f. Let s be the least numbered node such that there exists a path converging at s. Then $(\forall k > d(s))$ s will be in T_k, and $C_{s,k} \neq \emptyset$, hence M on input $f|_k$ outputs either $ind(s)$, or the index of some node t with $t < s$.

If for every node $n < s$ there was some level k_n such that $C_{n,k_n} = \emptyset$ then M will eventually witness this, and M will then converge to $ind(s)$, hence identify f.

Alternatively, suppose there was a node $n < s$, such that $(\forall k \geq d(n))$ $C_{n,k} \neq \emptyset$. We must show this is not possible. Since s is the least node at which convergence occurs, we know that every path passing through n cannot converge at n. In other words, $C_n = \emptyset$.

We have that $(\forall k \geq d(n))$ $C_{n,k} \neq \emptyset$. Consider all nodes at depth $\geq d(n)$. We will color some of these nodes red. In particular, color node m red if and only if $d(m) \geq d(n)$ and $C_{n,d(m)} \bigcap P_m \neq \emptyset$. Thus node m is colored red iff there is a path going through n, and then m, and the nodes on the partial path from n through m all have the same index as n.

We note two facts about our coloring:

1. There are infinitely many red nodes. This is the case because $(\forall k \geq d(n))$ $C_{n,k} \neq \emptyset$. So there is some path in $C_{n,k}$. In other words there is a path passing through n which "converges through level k". Then the node on that path at level k is red. Thus for each level, there is at least one red node, so there are infinitely many red nodes.

2. The subgraph of $T_{N,f}$ induced by the red nodes is a tree. Since $T_{N,f}$ is a tree, clearly the subgraph induced by the red nodes is a forest. To see that it is connected, observe that if m is red, then $parent(m)$ is red also.

We now have a rooted red tree (the root is n) with infinitely many nodes, and finite branching at each node. König's Infinity Lemma ([16]) asserts that there must be an infinite path in this red tree. But an infinite red path in $T_{N,f}$ corresponds to a path which is in C_n, hence C_n is not empty. Therefore, it cannot be the case that $(\forall k \geq d(n))$ $C_{n,k} \neq \emptyset$. This completes the proof of Theorem 4.14. □

Thus "unrestricted" nondeterministic IIMs are too powerful, and reliable nondeterministic IIMs are no more powerful than deterministic ones. This supports our view that *team inference* is the most natural notion of nondeterminism for inductive inference.

4.3 Probabilistic Finite Identification

In this section we extend the work of Freivald [21] for probabilistic finite inference, and give partial results concerning the relationships between team and probabilistic finite inference.

4.3.1 Bounded Mind Changes

Modifications of EX–identification have been investigated [5,10] which stipulate that the number of times the IIM changes its hypothesis en route to an EX–identification should be at most n for some number n.

Definition 4.15 *$(\forall n)$ the identification criterion EX_n is the pair (B_{EX_n}, Q_{EX_n}) where B_{EX_n} is always satisfied, and $Q_{EX_n}(\langle g_0, g_1, \ldots, g_k\rangle, f) = 1 \Leftrightarrow g_k = g_{k-1}, g_k \in GOOD_f$, and $|\{i : g_i \neq g_{i-1}\}| \leq n$.*

The probabilistic and team identification classes are given by Definitions 2.22 and 2.14. Our only difficulty with these definitions however, is that we have too many subscripts. We hope the reader will forgive our inconsistency by letting the team and probabilistic classes be denoted by $EX_n team(n)$ and $EX_n prob(p)$ respectively.

Freivald [21] considered probabilistic identification for the EX_0 case, which he called *finite identification*. We will discuss his results shortly.

Wiehagen, Freivald, and Kinber [41] consider probabilistic EX_n identification (their model is equivalent to ours). They are mostly concerned with probabilities $p > \frac{1}{2}$. They pose the following two questions:

1. Do there exist classes of recursive functions such that, with 'high' probability, these classes are limit identifiable with n changes of hypotheses by probabilistic strategies, but they cannot be limit identified with n changes of hypotheses by any deterministic strategy?
2. Does there exist a $k \in \mathrm{N}$ 'considerably' greater than n such that there are classes limit identifiable with n changes of hypotheses by probabilistic strategies, but even with k changes of hypotheses they are not limit identifiable by any deterministic strategy?

Their results include:

- $(\forall p > \frac{6}{7})\ EX_1prob(p) = EX_1$, and $EX_1prob(\frac{6}{7}) \supset EX_1$.
- $(\forall n \geq 2)(\exists U)(\forall \epsilon > 0)\ U \in EX_nprob(1-\epsilon) - EX_n$.
- $(\forall n)(\forall p > \frac{1}{2})\ EX_{2n} \subset EX_nprob(p) \subset EX_{8n+2}$.

Thus the answer to their first question is "yes" (when $n \geq 2$), and the answer to their second question is that probabilistic strategies (with $p > \frac{1}{2}$) can give at most a linear "speedup" in the number of mind changes. They also independently show that $EX_{prob}(p) = EX$ when $p > \frac{1}{2}$, a special case of our Theorem 3.21.

4.3.2 Freivald's Results

Freivald [21] introduced what are essentially the classes $EX_0prob(p)$, which he called *finite* identification by probabilistic machines. His model is equivalent to ours above, and it will be easier to discuss since it is free from the excess baggage of generality.

Simply put, every probabilistic IIM P has a 2-sided coin, and

Definition 4.16 [3] *P EX_0-identifies f with probability p iff* $\Pr[P$*, when fed f, halts and outputs an element of $GOOD_f$ after some finite amount of time*$] \geq p$.

We let $P(f)$ denote the output of P when fed f (which depends on the coins flipped). Freivald was mostly interested in values $p \geq \frac{1}{2}$. He shows

Theorem 4.17 [21] $(\forall p_1 < p_2)\ p_1, p_2 \in$ *the same interval* $(\frac{n+2}{2n+3}, \frac{n+1}{2n+1}] \Rightarrow EX_0prob(p_1) = EX_0prob(p_2)$. *Otherwise,* $EX_0prob(p_1) \supset EX_0prob(p_2)$.

In other words, there is a discrete hierarchy with breakpoints $\frac{2}{3}, \frac{3}{5}, \frac{4}{7}, \frac{5}{9}, \ldots$, and in particular, if $p > \frac{2}{3}$ then $EX_0prob(p) = EX_0$.

4.3.3 Teams for Probability Exceeding One Half

Theorem 4.18 $(\forall p)(\forall n \geq 1)\ p > \frac{n+1}{2n+1} \Rightarrow EX_0prob(p) \subseteq EX_0team(n)$.

Proof: Note that for $n = 1$ this is Freivald's result that if $p > \frac{2}{3}$ then $EX_0prob(p) = EX_0$. Let P (finitely) identify every $f \in U$ with probability $p > \frac{n+1}{2n+1}$. We construct $M_1, M_2, \ldots, M_n$ which $EX_0team(n)$– identify every $f \in U$. $(\forall i)$ M_i on input f simulates

[3] Freivald defines this with "$> p$" rather that "$\geq p$." We maintain consistency with the rest of this work and use "$\geq p$," and modify the statements of his theorems accordingly.

P with all finite sequences of coin flips, and builds a list $I = \{i_1, i_2, \ldots, i_k\}$ of possible outputs of P. M_i also keeps track, for each possible output i_k, of the associated probability p_k that P outputs i_k. M_i continues to search until (if ever) it finds a set $I = \{i_1, i_2, \ldots, i_k\}$ with associated list of probabilities $\vec{p} = \langle p_1, \ldots, p_k \rangle$ such that

$$\Pr[P(f) = i_j] \geq p_j \text{ and } \sum_{j=1}^{k} p_j > \frac{n+i}{2n+1}.$$

M_i then outputs the index of the program $THRESHOLD_{\frac{n}{2n+1}, I, \vec{p}}$.

Let $M_i(f)$ denote the single index output (if it exists) of M_i when fed f. We argue that $(\forall f \in U)(\exists i)\ M_i(f) \in GOOD_f$. By definition, $\Pr[P(f) \in GOOD_f] > \frac{n+1}{2n+1}$, therefore, for some i with $1 \leq i \leq n$ we have

$$\frac{i-1}{2n+1} \leq \Pr[P(f) \in BAD_f] < \frac{i}{2n+1}.$$

Then we show that M_i identifies f. By assumption on i,

$$\Pr[P(f)\downarrow] > \frac{n+1}{2n+1} + \frac{i-1}{2n+1} = \frac{n+i}{2n+1}$$

so M_i will find a set I of possible guesses and a vector $\vec{p}$ of associated probabilities, and output the index of the program $THRESHOLD_{\frac{n}{2n+1}, I, \vec{p}}$. Now

$$\Pr[P(f) \in WRONG_f] \leq \Pr[P(f) \in BAD_f] < \frac{i}{2n+1}$$

so

$$\sum_{j \in I \bigcap WRONG_f} p_j < \frac{i}{2n+1} \leq \frac{n}{2n+1}.$$

Further, since

$$\sum_{j \in I \bigcap WRONG_f} p_j < \frac{i}{2n+1} \text{ and } \sum_{j \in I} p_j > \frac{n+i}{2n+1}$$

we have

$$\sum_{j \in I \bigcap GOOD_f} p_j > \frac{n}{2n+1},$$

and by Lemma 2.4 $M_i(f) \in GOOD_f$. □

We've thus shown that probabilities $p > \frac{n+1}{2n+1}$ can be simulated by a team of n machines. We might conjecture that, for finite inference, teams of $n = 1, 2, 3, \ldots$ correspond to Freivald's hierarchy for $p > \frac{1}{2}$. This is not the case however, as all of the containments of Theorem 4.18 are proper. We will show that no probabilistic machine has the power of a team of 2 IIMs if the probability of success is $p > \frac{1}{2}$.

4.3.4 Probability Below One Half

We show that there is an infinite hierarchy of finite identification for $p \leq \frac{1}{2}$ which is proper *at least* at the values $\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots$

Theorem 4.19 $(\forall n \geq 1)$ $EX_0prob(\frac{1}{n}) \subset EX_0prob(\frac{1}{n+1})$.

Proof: The containment is immediate; that is proper follows from the second part of Lemma 4.20 given below. □

Let the sets $F_{n+1} = \{f \mid f =^n 0\}$, *i.e.* every $f \in F_{n+1}$ is non–zero on at most n arguments.

Lemma 4.20

1. $(\forall n \geq 1)$ $F_{n+1} \in EX_0team(n+1) - EX_0team(n)$.
2. $(\forall n \geq 1)(\forall p > \frac{1}{n+1})$ $F_{n+1} \in EX_0prob(\frac{1}{n+1}) - EX_0prob(p)$.

In other words, F_{n+1} can be identified by a team of $n+1$, and with probability $\frac{1}{n+1}$, but not by any team of n or with any probability exceeding $\frac{1}{n+1}$ (in particular, $\frac{1}{n}$).

Proof:

We need only show that $F_{n+1} \in EX_0team(n+1)$ and $(\forall p > \frac{1}{n+1})$ $F_{n+1} \notin EX_0prob(p)$, since $\frac{1}{n} > \frac{1}{n+1}$ and $EX_0team(n) \subseteq EX_0prob(\frac{1}{n})$.

$F_{n+1} \in EX_0team(n+1)$ by $M_0, M_1, M_2, \ldots, M_n$ where M_i assumes that $|\{x : f(x) \neq 0\}| = i$, waits until it sees these values of f, and then outputs the index of a program which does table look up for one of these values, or outputs 0 if the argument is not in the table.

Let $p > \frac{1}{n+1}$. We must show that $F_{n+1} \notin EX_0prob(p)$. Let P be any probabilistic IIM. We construct a function $f \in F_{n+1}$ such that Pr[P EX_0–identifies f] $< \frac{1}{n+1}$. We denote the function f by a list of its values on arguments 0,1,2,3,.... For example, the sequence of values $0^t j010^\infty$ denotes the function f such that $f(x) = j$ if $x = t$, 1 if $x = t+2$, and 0 elsewhere. We construct f in stages. We begin with $s =$ the null sequence, and $k = 1$.

Stage k

Feed P the sequence $s0^\infty$ until a finite set H_k of indices, and values t_k and x_k are found such that

- $(\forall j < k)\ H_k \bigcap H_j = \emptyset$.

- $\Pr[P(s0^\infty) \in H_k] > \frac{1}{n+1}$ and P makes the guesses of H_k seeing at most the values $s0^{t_k}$.

- $(\forall h \in H_k)\ \varphi_h$ is witnessed (within x_k steps of simulation) to agree on its first $|s| + t_k + 1$ values with the finite function $s0^{t_k}0$.

Then set $s \leftarrow s0^{t_k}1$. If $k < n$ then go to stage $k+1$ else set $f = s0^\infty$ and halt.

If all n stages of the construction of f are completed, then f is a total recursive function, which $\neq 0$ in at most n places, hence is in F_{n+1}. Further, by construction, P on input f outputs n groups of distinct indices $H_1, H_2, \ldots, H_n$ such that $(\forall i)\ H_i \subseteq BAD_f$ and $\Pr[P(f) \in H_i] > \frac{1}{n+1}$. Therefore

$$\Pr[P(f) \in BAD_f] > \frac{n}{n+1}$$

so

$$\Pr[P(f) \in GOOD_f] < \frac{1}{n+1},$$

and P doesn't identify every $f \in F_{n+1}$ with probability $p > \frac{1}{n+1}$.

If, on the other hand, not all stages of the construction of f were completed, then let k be the last stage completed ($k < n$). Then $s = 0^{t_1}10^{t_2}1 \ldots 0^{t_k}1$. Now consider the total recursive function $f = s0^\infty \in F_{n+1}$. By construction, all of the elements of $H_1, H_2, \ldots, H_k$ found do not compute f, since the elements of H_i disagree where f's i^{th} "1" is defined. Furthermore, $P(s0^\infty)$ never outputs a set of hypotheses H_{k+1} with total probability $> \frac{1}{n+1}$ such that $(\forall h \in H_k)\ \varphi_h$ agrees with the finite function $s0^{t_k+1}0$ for any value of t_k. But if P identifies F_{n+1} with probability $p > \frac{1}{n+1}$ then P identifies $s0^\infty$ with probability $> \frac{1}{n+1}$ and P would have output, after seeing some finite segment of values, a set of hypotheses H_{k+1} with the properties above. □

Corollary 4.21 $(\forall p > \frac{1}{2})\ EX_0team(2) - EX_0prob(p) \neq \emptyset$.

We might conjecture that finite identification is similar to EX–identification, and that probability $\frac{1}{n}$ is identical to a team of n, thus giving a discrete hierarchy for probability below $\frac{1}{2}$. We do not believe this is the case however. The reasons will become clear in the next section.

4.3.5 Many Team Members are Sufficient

We give an upper bound on the number of machines needed to EX_0–identify a set of functions identified by some probabilistic machine with probability $p > \frac{1}{n+1}$.

For limiting inference (EX) we showed that n team members were sufficient. Recall that this was achieved as follows. Each M_i assumed a value for the "total amount of convergence" of the probabilistic machine P on f, found a set of programs (in the limit) containing a correct program for f, and (in the limit) eliminated the $WRONG_f$ programs from the list, allowing the construction of a correct $RACE$ program.

EX_0 identification is much more limiting (since it is not limiting). In particular, obtaining a list of programs containing a correct program is not sufficient, since one can't eliminate the $WRONG_f$ programs from the list (unless you know exactly how many there are). The strategy used here is similar to that used for BC identification – the program ultimately output will be a $THRESHOLD$ program. To do this correctly however, the team members must have an estimate of both $\Pr[P(f) \in GOOD_f]$ and $\Pr[P(f)\downarrow]$. Unfortunately, $2n^2+n$ different machines seem to be necessary to guarantee that at least one has a close enough estimate to carry out the procedure properly.

Definition 4.22 $wt(X) = \sum_{x\in X} \Pr[P(f) = x]$.

Note that $wt(\mathrm{N})$ is simply the probability that P halts on input f.

Let P EX_0–identify U with probability $p > \frac{1}{n+1} = \frac{2}{2n+2}$. Let the machines $M_{i,j}$ for $1 \le i \le 2n$ and $i+1 \le j \le 2n+1$ be defined as follows. Machine $M_{i,j}$ assumes that

- $\frac{i-1}{2n+2} < wt(WRONG_f) \le \frac{i}{2n+2}$ ("$\le$" on the left if $i = 1$).
- $\frac{j}{2n+2} < wt(\mathrm{N}) \le \frac{j+1}{2n+2}$.

On input f, $M_{i,j}$ builds $T_{P,f}$, and dovetails the computations of all of the programs found until (if ever) the following two conditions are satisfied:

1. A set W is found such that $wt(W) > \frac{i-1}{2n+2}$ ($\ge$ if $i = 1$) and $(\exists x)(\forall w \in W)(\exists y \le x)$ $\varphi_w(y) \ne f(y)$.
2. A set H is found such that $W \bigcap H = \emptyset$, and $wt(H \bigcup W) > \frac{j}{2n+2}$.

If $M_{i,j}$ finds sets W and H then it outputs the index of program $THRESHOLD_{\frac{1}{2n+2},H,\vec{p}}$ where $\vec{p}$ is the vector of probabilities associated with the elements of H.

First we observe that since (by definition of P) $wt(GOOD_f) > \frac{2}{2n+2}$, there is some i between 1 and $2n$ such that $wt(WRONG_f)$ satisfies the first assumption of $M_{i,j}$ for every j. Further, since then we have that $wt(GOOD_f) + wt(WRONG_f) > \frac{2}{2n+2} + \frac{i-1}{2n+2}$, $wt(\mathrm{N}) > \frac{i+1}{2n+2}$ so for some j $wt(\mathrm{N})$ will satisfy $M_{i,j}$'s second assumption. It follows that $M_{i,j}$ will find a set H and W as specified.

We now need only prove that the threshold program chosen by $M_{i,j}$ (the team member which assumes correctly) satisfies $wt(H \bigcap WRONG_f) < \frac{1}{2n+2} < wt(H \bigcap GOOD_f)$ and by Lemma 2.4 $M_{i,j}$ will have EX_0–identified f.

Since $wt(WRONG_f) \leq \frac{i}{2n+2}$ and W is found such that $wt(W) > \frac{i-1}{2n+2}$ with $W \bigcap H = \emptyset$, we have that $wt(H \bigcap WRONG_f) < \frac{1}{2n+2}$. Now suppose by way of contradiction that $wt(H \bigcap GOOD_f) \leq \frac{1}{2n+2}$. Then since $wt(GOOD_f) > \frac{2}{2n+2}$, $wt(H^c \bigcap GOOD_f) > \frac{1}{2n+2}$, and $wt((H \bigcup W)^c \bigcap GOOD_f) > \frac{1}{2n+2}$ (since $W \bigcap GOOD_f = \emptyset$.) Thus

$$wt(\mathrm{N}) \geq wt(H \bigcup W) + wt((H \bigcup W)^c \bigcap GOOD_f) > \frac{j}{2n+2} + \frac{1}{2n+2} = \frac{j+1}{2n+2},$$

a contradiction since by assumption on j, $wt(\mathrm{N}) \leq \frac{j+1}{2n+2}$. We conclude that

$$wt(H \bigcap GOOD_f) > \frac{1}{2n+2}$$

and the program $THRESHOLD_{\frac{1}{2n+2}, H, \vec{p}}$ computes f.

Knowing the likelihood of P converging, together with the probability that it converges correctly is sufficient to allow a threshold strategy. The number of machines $M_{i,j}$ is

$$\sum_{i=1}^{2n} \sum_{j=i+1}^{2n+1} 1 = \sum_{i=1}^{2n} \sum_{j=i}^{2n} 1 = \sum_{i=1}^{2n} 2n + i - 1 = \sum_{i=1}^{2n} i = 2n^2 + n.$$

We've therefore shown

Theorem 4.23 $(\forall n \geq 1)$ $p > \frac{1}{n+1} \Rightarrow EX_0prob(p) \subseteq EX_0team(2n^2 + n)$.

We leave as open problems the exact correspondence between teams and probability for finite identification. We do not see how to extend our techniques to improve the $2n^2 + n$ upper bound. (Of course $EX_0team(n) \subseteq EX_0prob(\frac{1}{n})$.)

If the machine P is "Popperian" (see Sections 2.4 and 4.5) then $wt(SLOW_f) = 0$ and one degree of freedom is eliminated. It is not difficult to imagine that for Popperian EX_0–identification, the quadratic $2n^2 + n$ drops down to linear (somewhere near $2n$.) The factor of 2 seems necessary because the $OEX = EX$ tricks do not work here due to lack of "in the limit" time. A threshold vote must be made, and if the threshold were set at $\frac{1}{n+1}$ rather than half that value, then M_i would have to guarantee that it found nearly *all* of the $GOOD_f$ programs output. There seems to be no way of doing this if we can only estimate to within $\frac{1}{n+1}$. For these reasons we believe that probability and teams are very different for EX_0, and would be surprised to find that $(\forall n \geq 1)$ $p > \frac{1}{n+1} \Rightarrow EX_0prob(p) \subseteq EX_0team(n)$.

Finally, a problem still open and possibly independent of the relationship between teams and probability is whether the hierarchy $EX_0prob(\frac{1}{2}) \subset EX_0prob(\frac{1}{3}) \subset \cdots$ is the finest possible, has other discrete "breakpoints", or separates between all values $p_1 < p_2$.

4.4 Teams vs. Mind Changes for Finite Identification

Before beginning, we warn the reader that this is an orphan section which was mercifully adopted by this chapter. It has nothing to do with probability, but rather relates (for a special case) finite identification by teams with mind changes.

Bounding the number of mind changes en route to convergence can be seen as a sort of resource bound or complexity measure for inference. (An axiomatic treatment of complexity of inference can be found in [15] and [18].)

It has been shown [10] that the more mind changes available to an IIM, the more power it has to identify functions, *i.e.* $(\forall n)\ EX_n \subset EX_{n+1}$. We would like to compare the power of a team of n EX_0 machines with a single EX_m machine. A team of n EX_0 machines may be viewed as a *parallel* version of finite inference, whereas a single EX_m machine might be viewed as a *serial* version of finite inference. For example, it can be seen rather easily ([35]) that a team of n EX_0 machines are as powerful as a single EX_{n-1} machine, because a single EX_{n-1} machine makes at most n different hypotheses, each of which can be delegated to a different team member. Thus we have

Theorem 4.24 (Theorem 4.1 of [35]) $(\forall n)\ EX_{n-1} \subseteq EX_0team(n)$.

Many results have been given in [35] and recently [37] approaching a complete characterization of the relationships between mind changes, anomalies, and the number of team members for EX identification. However, the relationship between finite identification and mind changes has not been completely characterized. Here we take a step closer toward settling the remaining open problems.

Theorem 4.25 (*from* [37]) $(\forall n \geq 1)\ EX_0team(n) \subseteq EX_{2n-2}$.

Proof: Let $U \in EX_0(M_1, M_2, \ldots, M_n)$, then let M EX_{2n-2}-identify U as follows. For any $f \in U$, on input $f|_k$ M simulates each of $M_1, M_2, \ldots, M_n$ for k steps each on $f|_k$ and obtains the collection of guesses $I_k = \{g \mid (\exists i)\ M_i(f) \downarrow= g \text{ within } k \text{ steps}\}$. M dovetails for k steps each of the computations $\{\varphi_i(x) \mid i \in I_k,\ x \leq k\}$ and attempts to witness for each $i \in I_k$ that $i \in WRONG_f$. If $S_k \subseteq I_k$ is the set of indices for which M successfully witnesses this, then M sets $I_k \leftarrow I_k - S$. If I_k is nonempty, then M outputs $RACE_{I_k}$.

We show that M identifies every $f \in U$ with at most $2n-2$ mind changes. Since $(\exists i)$ $M_i(f) \downarrow= g$ and $g \in GOOD_f$ (by the definition of team identification), $(\forall_k^{\infty})$ I_k contains g. Since I_k can contain at most n elements (the values $\{M_i(f)\}$ if they exist), and at most $n-1$ elements of I_k can be eliminated by M observing that they are in $WRONG_f$, then in the worst case, each value $M_i(f)$ enters and leaves I_k exactly once, except for the value g which never gets eliminated. This is a total of $2n-1$ changes, but observe that M doesn't output anything until in fact I_k is nonempty, so the number of mind changes is at most $2n-2$. Also note that since the sequence of sets I_k have limit I, then the programs $RACE_{I_k}$ have limit $RACE_I$, $I \bigcap GOOD_f \neq \emptyset$ (since I contains g), and $I \bigcap WRONG_f = \emptyset$, otherwise I is not the limit of I_k. Now by Lemma 2.2, the program $RACE_I$ computes f. □

Putting this together with the last theorem, we have

$$(\forall n \geq 1)\ EX_{n-1} \subseteq EX_0team(n) \subseteq EX_{2n-2}.$$

What remains to be shown is the relationship between the each of the classes

$$\{EX_n, EX_{n+1}, \ldots, EX_{2n-3}\}$$

and $EX_0team(n)$. A dilution of Theorem 4.2 in [35] gives

$$EX_n - EX_0team(n) \neq \emptyset,$$

thus $EX_0team(n) \subset EX_{2n-2}$. This leaves three possibilities for the relationship of $EX_0team(n)$ with mind changes:

1. $EX_0team(n) = EX_{n-1}$.
2. for some k with $n \leq k < 2n-2$, $EX_0team(n) \subset EX_k$.
3. $(\forall k)$ $n \leq k \leq 2n-3 \Rightarrow EX_0team(n)$ and EX_k are incomparable.

We show that the first case does not happen when $n \geq 2$ (Clearly if $n = 1$ then $EX_0team(n) = EX_{n-1}$.) We prove that

Theorem 4.26 $(\forall n \geq 2)$ $EX_0team(n) \supset EX_{n-1}$.

The containment follows (as mentioned above) from Theorem 4.1 of [35]. To show that it is proper, we show that $(\forall n \geq 2)$ $\exists S_n \in EX_0team(n) - EX_{n-1}$. We assume without loss of generality, that the the numbers 0 and 1 are indices of the everywhere undefined function. Let $S_n = \{f \mid$ there are at most n values of x such that $f(x) > 1$, and $(\exists x)\ f(x) > 1$ and $f(x) \in GOOD_f.\}$

To see that $S_n \in EX_0team(n)$ let $M_1, M_2, \ldots, M_n$ be defined by: M_i, on input $f|_k$ waits until (if ever) it receives the i^{th} value $f(x) > 1$ which it then outputs.

To show that $S_n \notin EX_{n-1}$ we use the following version of the recursion theorem, due to Smullyan [38]:

Lemma 4.27 [38] *Let $k \in \mathrm{N}$ and $f_1, f_2, \ldots, f_k$ be any recursive functions of $k+1$ variables. Then there are numbers $i_1, i_2, \ldots, i_k \in \mathrm{N}$ such that* $(\forall x)$

$$\begin{aligned} \varphi_{i_1}(x) &= f_1(i_1, i_2, \ldots, i_k, x). \\ \varphi_{i_2}(x) &= f_2(i_1, i_2, \ldots, i_k, x). \\ &\ \ \vdots \\ \varphi_{i_k}(x) &= f_k(i_1, i_2, \ldots, i_k, x). \end{aligned}$$

As in Section 4.3.4 we let a sequence of values denote the function with those values. We prove that $S_n \notin EX_{n-1}$ by induction on n.

Base Case $(n = 2)$

Let M be any IIM. We show there is at least one function $f \in S_2$ which M doesn't identify within 1 change of mind. Define f_1, f_2, and f_3, functions of four variables, as follows.

$$f_1(i,j,k,x) = f_2(i,j,k,x) = f_3(i,j,k,x) = i0^t \ldots \text{ for } t = 1,2,3\ldots$$

such that the strategy M on input function $i0^\infty$ does not produce its first hypothesis within t steps of computation. Note that if $M(i0^\infty)$ never outputs any hypothesis, then the function $f_1 = f_2 = f_3 = i0^\infty$, and by Lemma 4.27 there are values i, j, k such that $\varphi_i(x) = f_1(i,j,k,x)$. In particular, $\varphi_i(0) = f_1(i,j,k,0) = i$, thus the function $i0^\infty$ is not identified by M but is in S_2.

Then let t_0 be the least number such that M outputs its 1^{st} hypothesis h_0 within t_0 steps of computation. Suspend defining f_1 and f_2 and continue defining

$$f_3 = i0^{t_0}k0^t \ldots \text{ for } t = 1,2,3\ldots$$

until (if ever) a number t_1 is found such that one of the following occurs:

1. φ_{h_0}, when simulated (in order) on arguments $0, 1, 2, \ldots$ outputs a sequence of values $i0^{t_0}k\ldots$ within t_1 steps of simulation.

2. M on input $i0^{t_0}k0^{\infty}$ changes its guess from h_0 to h_1 within t_1 steps of simulation.

Note that either 1. or 2. *must* occur. If neither occur, then $f_3 = i0^{t_0}k0^{\infty}$ and by Lemma 4.27 there are values i, j, k such that $\varphi_k(x) = f_3(i, j, k, x) = i0^{t_0}k0^{\infty} \in S_2$. Now since M outputs only h_0 and never changes its mind, and h_0 simulated on successive inputs for longer amounts of time never is witnessed to output the initial sequence of values $i0^{t_0}k$, M doesn't identify f.

If 1. occurs, then resume defining f_1 and f_2 by

$$f_1 = f_2 = i0^{t_0}0^t \; for \; t = 1, 2, 3, \ldots$$

until (if ever) M on input $i0^{t_0}0^t$ changes its hypothesis from h_0 (which it guessed after seeing at most $i0^{t_0}$) to h_1 within t_1 steps of simulation. This must occur, for otherwise, $f_1 = i0^{\infty}$ and Lemma 4.27 gives a numbers i, j, k such that $f(i, j, k, x) = \varphi_i(x) = i0^{\infty} \in S_2$, but not identified by M because M output only the guess h_0 which agrees with $i0^{t_0}k\ldots \neq f_1$. So on input $i0^{t_0}0^{t_1}$ M has output 2 hypotheses, h_0 and h_1, and has run out of its single allowed mind change. Now define

$$\begin{aligned} f_1 &= i0^{t_0}0^{t_1}10^{\infty}. \\ f_2 &= i0^{t_0}0^{t_1}0j0^{\infty}. \end{aligned}$$

Again, by Lemma 4.27 there are values i, j, k such that $\varphi_i(x) = f_1(i, j, k, x) = i0^{t_0}0^{t_1}10^{\infty}$ and $\varphi_j(x) = f_2(i, j, k, x) = i0^{t_0}0^{t_1}0j0^{\infty}$ Clearly both f_1 and f_2 are in S_2, are different, and M has output the guess h_1 which must be incorrect for at least one of $\{f_1, f_2\}$. Thus M doesn't EX_1–identify S_2

If, on the other hand, 2. occurs, then M on input $i0^{t_0}k0^{t_1}$ has output two hypotheses h_0 and h_1, and we then complete the definition of f_1 (which so far $= i0^{t_0}$), and f_3 (which so far $= i0^{t_0}k0^{t_1}$), by

$$\begin{aligned} f_1 &= i0^{t_0}k0^{t_1}0^{\infty}. \\ f_3 &= i0^{t_0}k0^{t_1}10^{\infty}. \end{aligned}$$

Then by Lemma 4.27 there are numbers i, j, k, such that $\varphi_i(x) = f_1(i, j, k, x) = i0^{t_0}k0^{t_1}0^{\infty}$ and $\varphi_k(x) = f_3(i, j, k, x) = i0^{t_0}k0^{t_1}10^{\infty}$, both in S_2, and M outputs its last guess h_1 when either are presented as input. M cannot EX_1–identify both. This completes the base case $n = 2$ of the induction.

Inductive Step

Now assume as our inductive hypothesis that $S_n \notin EX_{n-1}$ and we show that $S_{n+1} \notin EX_n$. Now $(\forall M)(\exists f \in S_n)$ such that M ***doesn't*** EX_{n-1}–identify f, and we'll construct f_1 and f_2 from f such that M doesn't EX_n–identify at least one of $\{f_1, f_2\}$: Let M and f be given such that M doesn't identify f within $n-1$ mind changes. (*i.e.* M needs at least n mind changes to identify f.)

Define $f_1(i,j,x)$ and $f_2(i,j,x)$ from f as follows:

$$f_1(i,j,x) = f_2(i,j,x) = f(x) \text{ for } 1 \leq x \leq t$$

for all t such that M on input $f(0), f(1), \ldots,$ has not changed its mind $\geq n$ times within t steps of computation and output an $n+1^{st}$ hypothesis h_n.

If M on input $f(0), f(1), \ldots$ changes its hypothesis $< n$ times, then $f_1 = f_2 = f$ and by choice of f, M doesn't identify the function(s), otherwise $f \in EX_{n-1}(M)$, a contradiction. Otherwise, let t be the smallest value such that within t steps of computation M (which could have received at most the values $f(0), \ldots, f(t)$) changes its mind n times, and outputs (its last) hypothesis h_n. Then define

$$\begin{aligned} f_1(i,j,x) &= f(0)f(1)\ldots f(t)i0^{\infty}. \\ f_2(i,j,x) &= f(0)f(1)\ldots f(t)j10^{\infty}. \end{aligned}$$

Since $f_1 \neq f_2$, h_n is not a correct index for both f_1 and f_2, so M doesn't EX_n–identify at least one of $\{f_1, f_2\}$. Furthermore, by Lemma 4.27 there are numbers i,j such that $\varphi_i(x) = f_1(i,j,x) \neq f_2(i,j,x) = \varphi_j(x)$. Also, since $f \in S_n$ there are at most n values x such that $f(x) > 1$ for $0 \leq x \leq t$, and thus there are at most $n+1$ values of x such that $f_1(x) > 1$, $f_2(x) > 1$, with one of those values a correct index – thus $f_1, f_2 \in S_{n+1}$. □

We believe (but have not yet been able to show) that these proof techniques could be used to conclude that $(\forall n \geq 2)(\forall k)\ n+1 \leq k \leq 2n-3 \Rightarrow EX_0team(n) - EX_k \neq \emptyset$. (*i.e.* we believe that the third possibility describing the relationship between $EX_0team(n)$ and mind changes holds).

4.5 Probabilistic Prediction

All of the definitions and results of the previous sections have been concerned with program synthesis from examples, as opposed to sequence extrapolation or prediction.

In this section we briefly investigate the probabilistic and team models of identification for the problem of sequence prediction.

A prediction method is simply an IIM M which (rather than attempting to output a program index) attempts to predict the next value when given the first k values of a total recursive function. [1,2,7]. M is given the values $f|_k$ and may either halt with a prediction for $f(k+1)$, or diverge.

Definition 4.28 *M NV–identifies (or predicts) f iff $(\forall x_1, x_2, \ldots, x_n)$, the computation $M(x_1, x_2, \ldots, x_n)$ is defined and $(\forall_k^{\infty})\ M(f(0), f(1), \ldots f(k)) = f(k+1)$.*

NV stands for Next Value. Although NV is not really an identification criterion, (it is a prediction criterion – whatever that is), it is not too difficult to squeeze NV into a formal definition of an identification criterion. We do this only so that we may use all of the tools and definitions which we have developed, and note that there are other "prediction criteria" which are not expressible as identification criteria as defined in Section 2.4. First note that the definition of NV–prediction does not depend on the sequential behavior of the machine on an infinite input (the graph of the function), but by the behavior of the machine on every finite initial segment of the graph of the function. Then it is easily seen that the definition above is equivalent to the following identification criterion:

Definition 4.29 *NV is the pair (B_{NV}, Q_{NV}), where*

- *$B_{NV}(M) = 1$ iff M is a total extrapolating machine, i.e. for every $s \in \mathcal{G}(\mathcal{H})$, the sequence $IO_{M,s}$ consists of alternating "output" elements and "input" elements.*
- *$Q_{NV}(\langle s_0, s_1, \ldots, s_k\rangle, f) = 1 \Leftrightarrow s_k = f(k)$ and $s_{k-1} = f(k-1)$.*

We let $\mathcal{E}$ denote the class of total extrapolating IIMs.

4.5.1 A Degenerate Hierarchy

In Section 2.4 we used the class PEX as an example to point out issues involved in defining identification criteria. Recall that an IIM is *Popperian* iff $(\forall s \in \mathcal{G}(\mathcal{H}))$, $M(s)$ consists only of indices of total programs. Then

Definition 4.30 *PEX is the pair (B_{PEX}, Q_{EX}) where $B_{PEX}(M) = 1$ iff M is Popperian. (More formally, $B_{PEX}(IO_{M,s}) = 1$ iff every "guess" element of $IO_{M,s}$ is the index of a total recursive function.)*

Thus *PEX* is simply *EX*–identification by machines restricted to outputting only total programs.

Theorem 4.31 *$PEX = NV$.*

This result originates from J. van Leeuwen, and independently from J. M. Barzdin. The proof appears in [10].

Recall the definition of reliable IIMs from Section 4.2, and let *RPEX* be the identification criterion *PEX* with the additional behavioral restriction that the IIM be reliable. (*i.e.* $B_{RPEX}(M) = 1$ iff M is Popperian and reliable). A trivial observation about the proof of Lemma 4.5 in [9] gives that $RPEX = PEX$.

Lemma 4.32 *For all total extrapolating machines M_1 there is a reliable Popperian IIM M_2 (uniform in M_1) such that for any f, the partial sequence of outputs $M_2(f|_k)$ depends only on the outputs $M_1(f(0)), M_1(f(0), f(1)), \ldots, M_1(f(0), f(1), \ldots, f(k))$, and M_1 NV–predicts $f \Leftrightarrow M_2$ RPEX–identifies f.*

Proof: This theorem is essentially a combination of theorems in [9] and [10], together with the observation that the construction is effective and that the k^{th} program output by M_2 can be generated by simulating M_1 only on $f|_k$.

M_2 on input $f|_k$ computes $M_1(f(0), f(1), \ldots, f(k))$. M_2 finds the set $D_k = \{x \mid M_1(f(0), f(1), \ldots, f(x-1)) \neq f(x) \text{ and } x \leq k\}$. M_2 outputs the index of the program p_k which on input x, outputs $f(x)$ if $x \in D_k$, otherwise outputs $M_1(p_k(0), p_k(1), \ldots, p_k(x-1))$.

Clearly the construction of M_2 from M_1 is effective. Now observe that

- Since M_1 is defined for all input sequences, the programs p_k are all total, so that M_2 is Popperian.
- If the sequence of programs $\{p_k\}$ output by M_2 converge to a program p, then the sequence of sets $\{D_k\}$ must converge to a set D. Furthermore, p has found and "patched" all of the anomalies in the set D, hence p computes f, and M_2 is reliable.
- If M_1 NV–predicts f, then by definition, there is a set D such that for all sufficiently large k, $D_k = D$. Now since p_k depends solely on D_k, there is a program p such that $(\forall_k^\infty)\ p_k = p$, and p computes f since M_2 is reliable. □

We now show that the classes $NV_{prob}(p)$ degenerate to NV. Let $U \in NV_{nondet}(M_1)$. Then for every oracle $\mathcal{O}$, $M_1^{\mathcal{O}}$ satisfies B_{NV}, and there is some oracle $\mathcal{O}$ such that $M_1^{\mathcal{O}}$

NV–predicts f. Now define the reliable, Popperian, nondeterministic IIM M_2. $M_2^{\mathcal{O}}(f)$ uses $\mathcal{O}$, M_1, and the input f to simulate $M_1^{\mathcal{O}}$ and effectively construct a sequence of total programs as in the proof of Lemma 4.32 such that $M_2^{\mathcal{O}}$ is Popperian, reliable, and $M_1^{\mathcal{O}}$ NV–predicts $f \Rightarrow M_2^{\mathcal{O}}$ $RPEX$–identifies f.

We've just shown that $NV_{nondet} \subseteq RPEX_{nondet}$. Finally, the proof of Theorem 4.14 applies to give $RPEX_{nondet} = RPEX$ by observing that all of the programs in the nondeterministic $RPEX$ tree are total. Further, since $RPEX = PEX = NV$ we have shown that

Theorem 4.33 $NV_{nondet} = NV$.

Corollary 4.34 $(\forall \epsilon > 0)$ $NV_{prob}(\epsilon) = NV$.

4.5.2 Redefining Probabilistic and Team Prediction

Since the probabilistic classes $NV_{prob}(p)$ did not prove too interesting, we consider an alternate definition of NV probabilistic prediction, similar to the alternate definition of probability and teams in Section 3.5 for EX and BC. Contrary to the results in Section 3.5 however, we find that the new definitions do alter the classes which are predictable.

Definition 4.35

- $NV_{prob'}(p) = \{U \mid (\exists P \in \mathcal{E})$ *such that*

$$(\forall f \in U)(\forall_k^\infty) \ \Pr[\{\mathcal{O} : P^{\mathcal{O}}(f(0), f(1), \ldots, f(k-1)) = f(k)\}] \geq p\}.$$

- $NV_{team'}(n) = \{U \mid (\exists M_1, M_2, \ldots, M_n \in \mathcal{E})$ *such that*

$$(\forall f \in U)(\forall_k^\infty)(\exists i) \ M_i(f(0), f(1), \ldots, f(k-1)) = f(k)\}.$$

We avoid using our probabilistic trees in the following arguments because the predicate defining NV isn't limiting–invariant under repetition. We first show that there *is* a hierarchy of team′ prediction for NV.

Theorem 4.36 $(\forall n \geq 1)$ $NV_{team'}(n) \subset NV_{team'}(n+1)$.

Proof: The containment is obvious. We show it is proper by showing $(\forall n \geq 1)$ $\exists U_{n+1} \in NV_{team'}(n+1) - NV_{team'}(n)$. Let

$$U_{n+1} = \{f \in \mathcal{T} \mid range(f) \subseteq \{1, 2, \ldots, n+1\}\}.$$

Clearly $U_{n+1} \in NV_{team'}(M_1, M_2, \ldots, M_n, M_{n+1})$, where $M_i(x_1, x_2, \ldots, x_k) = i$ regardless of the values $\{x_i\}$. (In fact, the team $M_1, M_2, \ldots, M_n, M_{n+1}$ successfully predicts even *uncomputable* functions with range contained in $\{1, 2, \ldots, n+1\}$.) To see that $U_{n+1} \notin NV_{team'}(n)$ let $M_1, M_2, \ldots, M_n$ be any collection of machines in $\mathcal{E}$. We construct $f_{M_1,M_2,\ldots,M_n} \in U_{n+1} - NV_{team'}(M_1, M_2, \ldots, M_n)$.

$f_{M_1,M_2,\ldots,M_n}$ is the total recursive function defined by:

$$f_{M_1,M_2,\ldots,M_n}(x) = \begin{cases} 0 & \text{if } x = 0. \\ \min\{i : 1 \le i \le n+1 \text{ and } (\forall j) M_j(f(0), \ldots, f(x-1)) \ne i\} & \text{if } x \ne 0. \end{cases}$$

Clearly $f_{M_1,M_2,\ldots,M_n}$ is total recursive, $\in U_{n+1}$, and contradicts the next guess of each of $M_1, M_2, \ldots, M_n$ for every value > 0. □

Now we show that for this model of probability and teams, we have the same relationships as for EX and BC.

Theorem 4.37 $(\forall n \ge 1)(\forall p)\ \frac{1}{n+1} < p \le \frac{1}{n} \Rightarrow NV_{prob'}(p) = NV_{team'}(n)$.

Proof: ($\supseteq$) A machine P flips an n–sided coin at *each* next guess and outputs the next guess of M_i where i is the result of the coin flip.

($\subseteq$) Let P NV–predict every $f \in U$ with probability (in the sense of *prob'*) $p > \frac{1}{n+1}$. Let $\epsilon > 0$ be such that $p > \frac{1}{n+1} + \epsilon$. We construct $M_1, M_2, \ldots, M_n$ which $NV_{team'}$–predict U.

By the definition of $NV_{prob'}(p)$, for every oracle $\mathcal{O}$, $P^{\mathcal{O}}$ satisfies B_{NV}. This means that for all k–tuples $\langle x_1, x_2, \ldots, x_k \rangle$, $P^{\mathcal{O}}(x_1, x_2, \ldots, x_k)$ is defined. It follows that $(\forall k)(\exists c)$ $\Pr[\{\mathcal{O} \mid P^{\mathcal{O}}(f(0), f(1), \ldots, f(k-1))$ halts with a prediction for $f(k)$ using at most c flips$\}] > 1 - \epsilon$.

Now $(\forall i)$ M_i on input $f(0), f(1), \ldots, f(k-1)$ must predict $f(k)$. Each M_i feeds the values $f(0), f(1), \ldots, f(k-1)$ to P along with every finite sequence of coin flips (in order of increasing length), and observes for each finite sequence s of coin flips whether $P^{\mathcal{O}}(f(0), \ldots, f(k-1))$ with sequence s predicts a value for $f(k)$. M_i does this for as many finite sequences as necessary, until it finds a set of finite sequences S such that

$$\sum_{s \in S} \Pr[P^s(f(0), \ldots, f(k-1)) \downarrow] > 1 - \epsilon.$$

Let $wt(x) = \Pr[\{\mathcal{O} \mid P^{\mathcal{O}}(f(0), \ldots, f(k-1)) = x\}]$, and let $wt_S(x)$ be an estimate of $wt(x)$ which each M_i observes from P's simulated behavior using the finite sequences of S. Note that $wt_S(x) \le wt(x) < wt_S(x) + \epsilon$, since the probability that P outputs its prediction for $f(k)$ using only sequences in S is $> 1 - \epsilon$.

Then define the two sets X_1 and X_2 as follows:

$$X_1 = \{x \mid \frac{1}{n+1} < wt_S(x) \leq \frac{1}{n+1} + \epsilon\}.$$
$$X_2 = \{x \mid \frac{1}{n+1} + \epsilon < wt_S(x)\}.$$

Clearly $|X_1| + |X_2| \leq n$ since $\sum_x wt_S(x) \leq \sum_x wt(x) = 1$. Then $(\forall i)$ M_i outputs the i^{th} greatest value $x \in X_1 \bigcup X_2$ (if it exists), and outputs 0 otherwise.

Now all M_i are in $\mathcal{E}$, since the above computation always halts. We show $(\forall_k^\infty)(\exists i)$ $M_i(f(0), \ldots, f(k-1)) = f(k)$. By definition of $P\ NV_{prob'}$–predicting f,

$$(\forall_k^\infty)\ wt(f(k)) = \Pr[\{O \mid P^O(f(0), \ldots, f(k-1)) = f(k)\}] > \frac{1}{n+1} + \epsilon.$$

And since

$$(\forall x)\ wt(x) < wt_S(x) + \epsilon$$

we have

$$(\forall_k^\infty)\ \frac{1}{n+1} + \epsilon < wt(f(k)) < wt_S(f(k)) + \epsilon$$

and

$$(\forall_k^\infty)\ wt_S(f(k)) > \frac{1}{n+1}.$$

Hence $(\forall_k^\infty)$ $f(k)$ will be in $X_1 \bigcup X_2$, and for some i, M_i will predict $f(k)$. □

This theorem is really about total probabilistic transducers rather than inductive inference machines. In general, you can increase the computational certainty (from $> \frac{1}{n+1}$ to $\frac{1}{n}$) with an exponential simulation. Freivald [19] obtains similar results for *limiting* probabilistic computations, although his definitions require that a correct answer appears (in the limit) with probability $> \frac{1}{n+1}$, and all incorrect answers appear with probability $< \frac{1}{n+1}$.

Definition 4.38 *Let M be an IIM, $f \in \mathcal{T}$, and $k \in \mathbb{N}$. Then*

$$F_{M,f,k} = \frac{|\{1 \leq x \leq k : M(f(0), \ldots, f(x-1)) = f(x)\}|}{k}.$$

Definition 4.39 $NV_{freq}(p) = \{U \mid (\exists M \in \mathcal{E})(\forall f \in U) \liminf_{k \to \infty} F_{M,f,k} \geq p\}$.

We've show for (EX and BC) that when the IIMs output programs as opposed to predictions, the computational models of frequency, teams, and probability define the same classes. This is basically because the behavior of the hypothesized programs can be observed, and the incorrect programs identified to some extent. With prediction the situation is quite different. It is not surprising that there are sequences (functions) which are easy to predict for most arguments (*i.e.* with high frequency), but not at all predictable with any reasonable probability for some infrequent, "hard" values:

Theorem 4.40 $(\forall \epsilon > 0)(\forall n \geq 1)$

- $NV_{freq}(1) - NV_{team'}(n) \neq \emptyset$.
- $NV_{freq}(1) - NV_{prob'}(\epsilon) \neq \emptyset$.

Proof: For every $\in \mathcal{T}$, let the function $\#0_f : \mathrm{N} \rightarrow \mathrm{R}$ be defined by

$$\#0_f(k) = \frac{|\{1 \leq x \leq k : f(x) = 0\}|}{k}.$$

Then let

$$U = \{f \in \mathcal{T} \mid \liminf_{k\rightarrow\infty} \#0_f(k) = 1\}.$$

Clearly $U \in NV_{freq}(1)$ by the machine which always predicts "0." Let the collection of machines $M_1, M_2, \ldots, M_n \in \mathcal{E}$ and let $f_{M_1,M_2,\ldots,M_n}$ be defined by

$$f_{M_1,M_2,\ldots,M_n}(x) = \begin{cases} 1 + \max\{M_i(f(0), \ldots, f(x-1)) \mid 1 \leq i \leq n\} & \textit{if } x = 2^k \textit{ for some } k. \\ 0 & \textit{otherwise.} \end{cases}$$

Clearly $f_{M_1,M_2,\ldots,M_n}$ is total, is in U, and contradicts the predictions of each of the machines $M_1, M_2, \ldots, M_n$ infinitely often. Since n was arbitrary, we have part 1. of the theorem. Part 2. follows since $(\forall \epsilon > 0)(\exists n)\ NV_{prob'}(\epsilon) \subseteq NV_{team'}(n)$. □

4.6 Probabilistic Language Identification

In this section we apply the probabilistic model to the problem of grammar inference from texts (arbitrary enumerations of formal languages.) We consider the situation where an IIM is fed elements from some formal language L, and M attempts to output (in the limit) a grammar which enumerates all and only the elements of L. The IIM receives explicit information about L, and only implicit information about the complement of L since no element of the complement of L ever appears in the presentation. This contrasts with the situation where an IIM is fed the characteristic function of some language L.

We fix some finite alphabet Σ and let Σ^* be the set of all finite length strings over Σ. A language L is any subset of Σ^*. The elements of L are called the words of L. A formal grammar for L in the most general case (type 0 [24]) is equivalent to a program which, when started with empty input, outputs all and only the elements of L (not necessarily in any order, and possibly with repeats). We let $GOOD_L$ denote the set of indices of programs which enumerate L, and BAD_L denote $\mathrm{N} - GOOD_L$.

A *text* for a language L is a function $t : \mathrm{N} \rightarrow L \bigcup \{*\}$ such that $range(t) - \{*\} = L$. The "*" comes from [7] (see Section 2.3) and helps model gaps in texts, and allows us to deal with finite languages.

Definition 4.41 [23]

- *M TXTEX–identifies L iff for all texts t for L, M on input $t(0), t(1), \ldots$ outputs an infinite sequence $g_1, g_2, \ldots$ which converges to some $g \in GOOD_L$.*
- *$TXTEX = \{\mathcal{L} \mid \mathcal{L}$ is a class of recursively enumerable languages and $(\forall L \in \mathcal{L})(\exists M)$ M TXTEX–identifies $L\}$.*

BC inference of languages has been defined as well:

Definition 4.42 [8,30,31]

- *M TXTBC–identifies L iff for all texts t for L, M on input $t(0), t(1), \ldots$ outputs an infinite sequence $g_1, g_2, \ldots$ such that $(\forall_k^\infty)$ $g_k \in GOOD_L$.*
- *$TXTBC = \{\mathcal{L} \mid \mathcal{L}$ is a class of recursively enumerable languages and $(\forall L \in \mathcal{L})(\exists M)$ M TXTBC–identifies $L\}$.*

It has been shown [8,30,31] that $TXTBC$ properly contains $TXTEX$.

Since the input to language–identifying machines are not total functions, (or recursive languages) we do not assume that the elements of L are fed to an IIM in any canonical order. If t is a text for L, we let $M(t)$ denote the (possibly infinite) sequence of M's outputs when fed t. We define $TXTEX$ and $TXTBC$ with respect to our formalization of identification criterion so that we may benefit from the general definition of the probabilistic model. $TXTEX$ is the pair (B_{TXTEX}, Q_{TXTEX}) where B_{TXTEX} always $= 1$ (*i.e.* no restriction on the class of machines considered), and $Q_{TXTEX}(\langle g_1, g_2, \ldots, g_k\rangle, L) = 1 \Leftrightarrow g_k = g_{k-1}$ and $g_k \in GOOD_L$. Then M $TXTEX$–identifies L iff *for every text t of L*, $\lim_{k\rightarrow\infty} Q_{TXTEX}(M(t)|_k, L) = 1$. Our formal definition of $TXTBC$ is identical, except that $Q_{TXTBC}(\langle g_1, g_2, \ldots, g_k\rangle, L) = 1 \Leftrightarrow g_k \in GOOD_L$. Let the definitions of $TXTEX_{prob}$ and $TXTBC_{prob}$ be given by Definition 2.22. Clearly Q_{TXTEX} and Q_{TXTBC} are limiting–invariant under repetition, so we freely use the probabilistic computation tree model, and let $T_{P,t}$ denote the computation tree of P on input text t.

In his seminal paper [23], Gold showed that no class of languages in $TXTEX$ contains both an infinite language and all of its finite sublanguages, and the same result for $TXTBC$ appears in [8]. It is easy to see however [23], that the class of all finite languages is contained in $TXTEX$.

Theorem 4.43 *$TXTEX \subset TXTEX_{prob}(\frac{1}{2})$ and $TXTBC \subset TXTBC_{prob}(\frac{1}{2})$.*

Proof: The containments are immediate; we only need to show that they are proper. It is sufficient to show that there is a class of recursively enumerable languages $\mathcal{L}$ such that $\mathcal{L} \in TXTEX_{prob}(\frac{1}{2}) - TXTBC$. Let

$$\mathcal{L} = \{L \mid L \text{ is finite or } L = \Sigma^*\}$$

By our remarks above, since $\mathcal{L}$ contains both an infinite language (Σ^*) and all of its finite sublanguages, $\mathcal{L} \notin TXTBC$. To see that $\mathcal{L} \in TXTEX_{prob}(\frac{1}{2})$ consider the probabilistic IIM P which on input any text t of $L \in \mathcal{L}$ flips a coin, and with probability $\frac{1}{2}$ outputs the index of a program which enumerates Σ^*, and with probability $\frac{1}{2}$ follows a strategy which $TXTEX$–identifies all of the finite languages. □

For $TXTBC$ we show that this separation is the best possible, but first we define a program similar to the *THRESHOLD* program of Section 2.2, but for enumerating languages rather than computing functions.

Definition 4.44 *Let I be any finite ordered multiset of (not necessarily distinct) indices of enumerating programs $I = \{i_1, i_2, \ldots, i_k\}$. Let $\vec{p} = p_1, p_2, \ldots, p_k$ be any finite sequence of probabilities ($p_i \in \mathrm{R}, \sum_{i=1}^{k} p_i \leq 1$), and let v be any positive rational number. Then the program $VOTE_{v,I,\vec{p}}$ is the program defined by:*

$VOTE_{v,I,\vec{p}}$

> *On null input, dovetail the enumerations $\{\varphi_i \mid i \in I\}$ (all on null input). If at any point there is a word $w \in \Sigma^*$ such that a subset (possibly a multiset) $S \subseteq I$ has been found such that $(\forall i \in S)$ φ_i has enumerated w and $\sum_{i_j \in S} p_j > v$. Then output w. (Meanwhile, output $*$ every 100 steps or so.)*

Lemma 4.45 *If $\sum_{i_j \in I \bigcap BAD_L} p_j < v < \sum_{i_j \in I \bigcap GOOD_L} p_j$ then the program $VOTE_{v,I,\vec{p}}$ enumerates L.*

Proof: Obvious by analogy with the proof of Lemma 2.4.

Theorem 4.46 $(\forall p > \frac{1}{2})$ $TXTBC_{prob}(p) = TXTBC$.

Proof: Let P $TXTBC$–identify every $L \in \mathcal{L}$ with probability $p > \frac{1}{2}$. Then for every text t for L, Pr[{paths corresponding to correct $TXTBC$–identifications}] $> \frac{1}{2}$. By Lemma 3.11 $(\forall_k^\infty)$ the fraction of program indices at level k of $T_{P,t}$ which are in $GOOD_L$ is $> \frac{1}{2}$. We construct a deterministic IIM M which $TXTBC$–identifies $\mathcal{L}$. On input text t of $L \in \mathcal{L}$, M feeds t to P, constructs $T_{P,t}$, and outputs the index of program $VOTE_{\frac{1}{2},I_k,p^{(k)}}$, where I_k is the multiset of indices of nodes at level k of $T_{P,t}$ and the corresponding vector $p^{(k)}$ consists of the value $\frac{1}{2^k}$ repeated 2^k times.

Since $(\forall_k^\infty)$ greater than half of the indices at level k of $T_{P,t}$ are in $GOOD_L$, and strictly less than half of the elements are in BAD_L, Lemma 4.45 gives that $(\forall_k^\infty)$ the program output by M enumerates L, which is the definition of $TXTBC$–identification.

□

For $TXTEX$ we are not able to prove the corresponding theorem. We prove something weaker which is in the same spirit as Freivald's proof that $EX_0prob(p) = EX_0$ when $p > \frac{2}{3}$. Moreover, we believe that the similarities are more than coincidental. In the finite inference case the IIM is limited to a single guess. There is not sufficient time to witness that certain hypotheses produced in the probabilistic tree are bad (*i.e.* in $WRONG_f$). These $WRONG_f$ hypotheses cannot be eliminated, so the amalgamation program $RACE$ described in Section 2.2 cannot be employed. With text presentations of languages, although the IIM is allowed "limiting time" to eliminate hypotheses which enumerate incorrect words, the set of $WRONG$ enumerators cannot be identified because the presentation is only an enumeration, and the IIM can never prove wrong any hypothesis which enumerates a suspicious word, since the text offers no information about the complement of the language presented. In short, for finite identification the information is there but there isn't enough time to use it, and for $TXTEX$–identification there's plenty of time, but no information. For this reason, we believe that the structure of the probabilistic $TXTEX$ classes might be identical to those for finite identification.

Theorem 4.47 $(\forall p > \frac{2}{3})$ $TXTEX_{prob}(p) = TXTEX$.

Proof: Clearly $TXTEX_{prob}(p) \supseteq TXTEX$. Now let $p > \frac{2}{3} + \epsilon$, and let P $TXTEX$–identify every $L \in \mathcal{L}$ with probability $\geq p$. Consider the machine M which begins its computation by setting $I_0 \leftarrow \emptyset$, $c_0 \leftarrow +\infty$, $k_{old} \leftarrow 0$, and the vector $p^{(k_{old})} = \langle 1 \rangle$.

Phase k

1. On input text t, build $T_{P,t}$ to the k^{th} level, and compute $\Pr[C_{j,k}]$ for every node j in the partial tree.
2. Let c_k be the smallest number such that $\sum_{j=1}^{c_k} \Pr[C_{j,k}] > \frac{2}{3} + \epsilon$.
3. IF $c_k \neq c_{k-1}$ THEN
    - 3.1. $I_k \leftarrow$ the ordered multiset of indices $\{ind(1), ind(2), \ldots, ind(c_k)\}$.
    - 3.2. $p^{(k)} \leftarrow$ the sequence of values $p_j^{(k)} = \Pr[C_{j,k}]$ for $1 \leq j \leq k$.
    - 3.3. $k_{old} \leftarrow k$.
    - 3.4. GO TO STEP 5.
4. ELSE $(c_k = c_{k-1})$
    - 4.1. $I_k \leftarrow I_{k-1}$
    - 4.2. If $\sum_{j=1}^{c_k} \Pr[C_{j,k}] \geq \sum_{j=1}^{c_{k-1}} p_j^{(k_{old})} - \frac{\epsilon}{2}$ then $(\forall j)\ p_j^{(k)} \leftarrow p_j^{(k_{old})}$ otherwise $(\forall j)\ p_j^{(k)} \leftarrow \Pr[C_{j,k}]$.
5. Output the index of the program $VOTE_{\frac{1}{3}, I_k, p^{(k)}}$.
6. Go to phase $k+1$.

M finds (in the limit) a set of nodes of "weight" $> \frac{2}{3}$. Of these, there must be a group of weight $> \frac{1}{3}$ which are in $GOOD_L$, and weight $< \frac{1}{3}$ can be in BAD_L. Thus the program output will be correct. The only problem is that the estimated probabilities $\Pr[C_{j,k}]$ can change forever, so it is not clear what probability to associate with each hypothesis in the voting set I. This problem is solved by *ignoring* the changes in probabilities once these changes become small $(< \frac{\epsilon}{2})$.

By the construction on page 43 and the argument preceding the statement of Claim 3.27, the sequence of values $\{c_k\}$ converges to some number c, and hence the multisets I_k converge to the multiset $I = \{ind(1), ind(2), \ldots, ind(c)\}$. Now note that the sequence of probability vectors $p^{(1)}, p^{(2)}, \ldots$ must converge to some fixed vector of probabilities $\vec{p}$, for if not, then $\sum_{j=1}^{c} \Pr[C_{j,k}]$ decreases by $\frac{\epsilon}{2}$ infinitely often, a contradiction. Therefore, M converges to the index of the program $VOTE_{\frac{1}{3}, I, \vec{p}}$. We now show that

$$\sum_{\substack{1 \leq j \leq c \\ ind(j) \in BAD_L}} p_j < \frac{1}{3} < \sum_{\substack{1 \leq j \leq c \\ ind(j) \in GOOD_L}} p_j$$

and by Lemma 4.45 M $TXTEX$-identifies L proving the theorem. Since the values $\Pr[C_{j,k}]$ converge to $\Pr[C_j]$ from above (Lemma 3.18) we have

$$\sum_{j=1}^{c} p_j \leq \sum_{j=1}^{c} \Pr[C_j] + \frac{\epsilon}{2},$$

therefore

$$\begin{aligned}
\sum_{\substack{1 \leq j \leq c \\ ind(j) \in BAD_L}} p_j &\leq \sum_{\substack{1 \leq j \leq c \\ ind(j) \in BAD_L}} \Pr[C_j] + \frac{\epsilon}{2} \\
&\leq \sum_{ind(j) \in BAD_L} \Pr[C_j] + \frac{\epsilon}{2} \\
&= \Pr[C(BAD_L)] + \frac{\epsilon}{2} \\
&\leq \frac{1}{3} - \epsilon + \frac{\epsilon}{2} < \frac{1}{3}.
\end{aligned}$$

Finally,

$$\sum_{\substack{1 \leq j \leq c \\ ind(j) \in GOOD_L}} p_j = \sum_{1 \leq j \leq c} p_j - \sum_{\substack{1 \leq j \leq c \\ ind(j) \in BAD_L}} p_j > \frac{2}{3} + \epsilon - \frac{1}{3} = \frac{1}{3} + \epsilon > \frac{1}{3}.$$

□

4.7 Probabilistically Finding Concise Explanations

Among competing hypotheses for explaining a rule, scientists often choose the simplest hypothesis as the best explanation. This criterion for selection is called "Occam's razor."[4] The concept of simplicity has been modeled within inductive inference. The inference of "simple" programs from examples has been investigated, where "simple" is interpreted as "concise", or "small". In this section we show that if there is a probabilistic IIM which can infer small programs for a class of functions from examples with probability exceeding $\frac{1}{2}$, then there is a deterministic IIM which can infer small programs for the class of functions.

Let $\{M_i\}$ be a recursively enumerable sequence of all Turing machine transducers (or any other general model of computation powerful enough to compute all of the partial recursive functions), and let $\langle\varphi_i\rangle_{i\in N}$ be the corresponding acceptable numbering. Then we have the following definition from [6].

Definition 4.48 *The function* $size_M : \mathrm{N} \rightarrow \mathrm{N}$ *is a program size measure iff the following two conditions hold:*

[4] I believe Occam originally said something like "... entities should not be multiplied unnecessarily."

1. $(\forall x)\ \{i \mid size_M(i) = x\}$ *is finite.*

2. There is some transducer T *such that for every* x, $T(x)$ *halts with the elements of the set* $\{i \mid size_M(i) = x\}$ *on its output tape.*

We then say $size_M$ is the size of machine M_i or of program i.

Let $min_M(f) = \min\{size_M(i) \mid \varphi_i = f\}$; the size of the smallest program which computes f. Freivald [20] considered EX–identification of minimal size programs from values, and showed that this notion of identification was dependent on the particular program system (and hence acceptable numbering) chosen (whereas the classes EX, BC, and their variants are all well defined and independent of any particular acceptable numbering.)

For this reason, EX–identification of minimal size functions *modulo a recursive "fudge" factor* was introduced. It was shown [20] that this notion and the associated class MEX defined below are in fact independent of the acceptable numbering chosen. We assume then that a class of TM transducers $\{M_i\}$ has been fixed and that the function $size_M$ for these machines is the number of tuples defining them [24]. We now drop the subscript M from $size_M(f)$ and $min_M(f)$.

The identification criterion MEX is not defined for identification of functions, but rather for classes of functions. Rather than trying to force–fit it into our general definition, we treat it as an exception and define it as follows.

Let $GOOD_f^{(h)} = \{i \in GOOD_f \mid size(ind(i)) \leq h(min(f))\}$. For any $h \in \mathcal{T}$ let the predicate $Q_{h-EX} : (\mathrm{N}^* \times \mathcal{T}) \rightarrow \{0,1\}$ be defined by

$$Q_{h-EX}(\langle g_1, g_2, \ldots, g_k \rangle, f) = 1 \Leftrightarrow g_k = g_{k-1} \text{ and } g_k \in GOOD_f^{(h)}.$$

Definition 4.49 ([11,20]) *Let M be an IIM. Then*

- M *MEX–identifies* U *iff* $(\exists h)(\forall f \in U)\ \lim_{k\to\infty} Q_{h-EX}(M(f)|_k, f) = 1$.
- $MEX = \{U \mid M$ *MEX–identifies* $U\}$.

MEX is simply EX with the added condition that the hypothesis converged to is not larger than $h(min(f))$. Chen [11] gives many results for MEX (mind changes, anomalies, etc.) Note that the corresponding class for BC is not interesting, since there are at most finitely many programs of size at most $h(min(f))$, and any M which (in the limit) alternated choosing its outputs from among this finite set, could OEX (and hence EX identify f.

Since MEX is defined for classes of functions rather than functions, we explicitly define probabilistic MEX identification.

Definition 4.50

- $\Pr[P \textit{ MEX-identifies } U] \geq p \Leftrightarrow \Pr[\{O \mid P^O \textit{ MEX-identifies } U\}] \geq p$.
- $MEX_{prob}(p) = \{U \mid (\exists P)\ P \textit{ MEX-identifies } U \textit{ with probability} \geq p\}$.

Since for all h, Q_{h-EX} is limiting-invariant under repetition, we assume that all probabilistic IIMs P behave nicely and we make liberal use of the tree definitions for probabilistic IIMs. In particular, recall that for any set of indices A, the set $C(A)$ consists of those paths in the tree $T_{P,f}$ which (EX–) converge to an element of the set A. Then we have P MEX–identifies U with probability p iff $(\exists h)(\forall f \in U)\ \Pr[C(GOOD_f^{(h)})] \geq p$.

Chen gives the following nonunion theorem for MEX.

Theorem 4.51 $(\exists U_1, U_2 \in MEX)\ U_1 \bigcup U_2 \notin MEX$.

This gives rise to the following corollary.

Corollary 4.52 $MEX \subset MEX_{prob}(\frac{1}{2})$.

Proof: Containment is immediate. We show that it is proper. Let U_1 and U_2 be defined as in the nonunion theorem above. Then $U_1 \bigcup U_2 \notin MEX$. We show $U_1 \bigcup U_2 \in MEX_{prob}(\frac{1}{2})$. If $U_1 \in MEX$ is witnessed by machine M_1 with recursive function h_1, and $U_2 \in MEX$ is witnessed by machine M_2 with recursive function h_2, then let P be a probabilistic IIM which flips a coin and on input f, simulates either M_1 or M_2 equiprobably. Let h be the recursive function defined by $h(x) = \max\{h_1(x), h_2(x)\}$. Then $(\forall f \in U)$ $\Pr[P$ MEX–identifies f (within recursive function h)$] \geq \frac{1}{2}$. □

We now show that this separation is the best possible, *i.e.*

Theorem 4.53 $(\forall p > \frac{1}{2})\ MEX_{prob}(p) = MEX$.

To prove the theorem, we essentially redo the proof of the special case $n = 1$ of Theorem 3.21. (We explain later why the proof doesn't easily extend to show $p > \frac{1}{n+1} \Rightarrow MEX_{prob}(p) \subseteq MEX_{team}(n)$.) The idea is to show that a deterministic IIM can find (in the limit, by simulating P) a finite collection I of *small* programs containing at least one program which computes f.

Definition 4.54

- M *MOEX–identifies* $U \Leftrightarrow (\exists h)(\forall f \in U)\ M(f) \downarrow I$, *a finite list of distinct indices such that*

1. $(\forall i \in I)\ size(i) \leq h(min(f))$.

2. $(\exists i \in I)\ i \in GOOD_f^{(h)}$.

- $MOEX = \{U \mid (\exists M)\ M\ MOEX\text{-identifies } U\}$.

Lemma 4.55 $MOEX = MEX$.

Proof: In the proof of Lemma 3.25 a machine M' was constructed from a machine M such that if $M \downarrow I$, $M' \downarrow RACE_I$. Since the construction of $RACE_I$ from I in Section 2.2 is effective, and I is finite, there is a recursive g such that

$$size(RACE_I) \leq g(\langle size(i)\rangle_{i \in I}).$$

For the definition of size as number of tuples, $size(RACE_I)$ is roughly $c + \sum_{i \in I} size(i)$ for some constant c.

Now by the axioms of program size (Definition 4.48) there is a recursive function r such that $(\forall x)$ the number of programs of size $\leq x$ is at most $r(x)$. Therefore, the number of different programs of size at most $h(min(f))$ is at most $r(h(min(f)))$.

Since all elements i of I are distinct and have $size(i) \leq h(min(f))$ we have $(\exists r)$ $|I| \leq r(h(min(f)))$. Finally, $size(RACE_I) \leq g($at most $r(h(min(f)))$ indices all of size $\leq h(min(f)))$. Now $M' \downarrow RACE_I$ and by the argument in the proof of Lemma 3.25, $RACE_I$ computes f. □

Proof of Theorem 4.53

By Lemma 4.55, we need only show that $(\forall p > \frac{1}{2})\ MEX_{prob}(p) \subsetneq MOEX$ and the theorem follows. Let P MEX–identify U with probability $p > \frac{1}{2}$ and within recursive function h. We assume without loss of generality that h is monotone nondecreasing. We construct M which $MOEX$–identifies U (within recursive function $h \circ h$). M on input $f|_k$ simulates P, constructs the partial tree T_k = the first k levels of $T_{P,f}$, computes $\Pr[C_{j,k}]$, for every node $j \in T_k$, and finds the lexicographically least ordered list of nodes J_k such that

1. $\sum_{j \in J} \Pr[C_{j,k}] > \frac{1}{2}$.

2. $\max\{size(ind(j)) \mid j \in J\} \leq h(\min\{size(ind(j)) \mid j \in J\})$.

Now M outputs $I_k = \{ind(j) \mid j \in J\}$.

We show that M $MOEX$–identifies U. First we show that the sequence $\{J_k\}$ converges to some list J of nodes as k increases. By definition of P MEX–identifying U with probability $p > \frac{1}{2}$, and by Lemma 3.11, there is a set of nodes $V = \{n_1, n_2, \ldots, n_v\}$ such that $\sum_{j \in V} \Pr[C_j] > \frac{1}{2}$ and $ind(j) \in GOOD_f^{(h)}$. Then the set V satisfies the conditions 1. and 2. of M since for sufficiently large k, $\Pr[C_{j,k}] \geq \Pr[C_j]$ (Lemma 3.18), and therefore for sufficiently large k M will be able to find such a list J_k. Since there *is* a list of nodes V satisfying the two conditions of M, there is a lexicographically least such list, and the sequence $\{J_k\}$ must converge to a list J since there are only a finite number of lists which are lexicographically less than V. Thus the sequence of outputs $\{I_k\}$ of M converges to the set $I = \{ind(j) \mid j \in J\}$.

Now note that since $\Pr[C(I)] \geq \frac{1}{2}$ and $\Pr[C(\mathrm{N} - GOOD_f^{(h)})] < \frac{1}{2}$ we must have that $I \bigcap GOOD_f^{(h)} \neq \emptyset$. Let $g \in I \bigcap GOOD_f^{(h)}$, and thus condition 2. of $MOEX$–identification is satisfied. We must show that all of the other elements of I are small.

By the choice of M,

$$\max\{size(ind(j)) \mid j \in J\} \leq h(\min\{size(ind(j)) \mid j \in J\}).$$

But $g \in I \bigcap GOOD_f^{(h)}$ so

$$h(\min\{size(ind(j)) \mid j \in J\}) \leq h(size(g)) \leq h(h(min(f)))$$

and $(\forall i \in I)\ size(i) \leq h(h(min(f)))$. Thus I satisfies the first condition of the definition of $MOEX$–identification (with recursive function $h \circ h$). Thus M $MOEX$–identifies U. □

We leave as an open question whether there is a nondegenerate team hierarchy (for $n > 2$) for MEX (we conjecture there is), and whether $\frac{1}{n+1} < p \leq \frac{1}{n} \Rightarrow MEX_{prob}(p) = MEX_{team}(n)$. The reason the proof here doesn't seem to extend nicely to the general case $p > \frac{1}{n+1}$ is that the i^{th} team member of the corresponding construction would assume that $\Pr[C(\mathrm{N})] \in (\frac{i}{n+1}, \frac{i+1}{n+1}]$ and would attempt to find a set of nodes satisfying

1. $\sum_{j \in J} \Pr[C_{j,k}] > \frac{i}{n+1}$.
2. $\max\{size(ind(j)) \mid j \in J\} \leq h(\min\{size(ind(j)) \mid j \in J\})$.

Now condition 2 would not necessarily ever be satisfied, since we can only guarantee that the fraction $\frac{1}{n+1}$ would be elements of $GOOD_f^{(h)}$. The problem is that there could be many *large* ($> h(min(f))$) hypotheses which were in $WRONG_f$.

Chapter 5

Conclusion

In this dissertation we have defined a general probabilistic model of computation for inductive inference, and, with respect to the most basic identification criteria, have characterized the power of the model by unification with two previously investigated models of inference.

Furthermore, we have shown that more classes of functions can be inferred if we are willing to allow the inference strategy to fail *some* of the time. It is surprising that the concept of uncertainty of inference, whether applied to the team, probabilistic, or frequency model, gives rise to a *discrete* hierarchy, and that the relationships between models of computation are invariant across different types of identification criteria.

Many of the results in Chapter 4 give only partial relationships between team and probabilistic computations; we have indicated many open problems in passing. There are a number of identification criteria yet to be investigated, including ***consistent*** strategies, ***complete*** strategies, ***prudent*** strategies, and NV' and NV'' prediction.

In addition to extending the results given here, there are more general and pressing concerns for inductive inference. Perhaps the most basic is to close the gap between theoretical results and practical inference methods. We believe the general models for inferring recursive functions are too broad, and must be constrained if they are to yield more insight into how inference should be done. Incorporating a definition of complexity of inference and limiting the inference task to particular domains is paramount to obtaining a theory which can speak to practical concerns. The relationship between computational models and complexity of inference will likely be a rich area of investigation. We conclude with a brief summary of our results:

5.1 Summary of Results

- A general framework was developed for defining identification criteria and their relationships with different models of computation for inductive inference.

- $(\forall a \in \mathbf{N}\bigcup\{*\})\ (\forall p)\ (\forall n \geq 1)\ \frac{1}{n+1} < p \leq \frac{1}{n} \Rightarrow$

$$EX^a_{team}(n) = EX^a_{prob}(p) = EX^a_{freq}(p).$$

$$BC_{team}(n) = BC_{prob}(p) = BC_{freq}(p).$$

- The probabilistic and team classes for EX and BC are invariant under reasonable change of definition.

- $\mathcal{T} \in EX_{nondet} \supset REX_{nondet} = REX$.
- For finite identification,

  $p > \frac{n+1}{2n+1} \Rightarrow EX_0prob(p) \subseteq EX_0team(n)$.
  $(\forall n \geq 1)\ EX_0prob(\frac{1}{n}) \subset EX_0prob(\frac{1}{n+1})$.
  $(\forall p)(\forall n \geq 1)\ p > \frac{1}{n+1} \Rightarrow EX_0prob(p) \subseteq EX_0team(2n^2+n)$.
  $(\forall n \geq 2)\ EX_{n-1} \subset EX_0team(n)$.

- $(\forall \epsilon > 0)\ NV_{nondet} = NV = NV_{prob}(\epsilon)$, and for alternate definitions of probability and teams for NV,

$$(\forall n \geq 1)(\forall p)\ \frac{1}{n+1} < p \leq \frac{1}{n} \Rightarrow NV_{prob'}(p) = NV_{team'}(n) \neq NV_{freq}(p).$$

- For identification of languages,

  $TXTBC_{prob}(\frac{1}{2}) \supset TXTBC$ and $(\forall p > \frac{1}{2})\ TXTBC_{prob}(p) = TXTBC$.
  $TXTEX_{prob}(\frac{1}{2}) \supset TXTEX$ and $(\forall p > \frac{2}{3}\ TXTEX_{prob}(p) = TXTEX$.

- $(\forall p > \frac{1}{2})\ MEX_{prob}(p) = MEX$.

Bibliography

[1] D. Angluin and C. Smith.
Inductive inference: theory and methods.
Computing Surveys 15(3): 237-269, September, 1983.

[2] J. M. Barzdin.
Prognostication of automata and functions.
In *Information and Processing '71,* pages 81-84, North–Holland, Amsterdam, 1972.

[3] J. M. Barzdin.
Two theorems on the limiting synthesis of functions.
Latvii gosudarst. Univ. Ucenye Zapiski 210:82-88, 1974.
(in Russian)

[4] J. M. Barzdin and R. V. Freivald.
On the prediction of general recursive functions.
Soviet Math. Dokl. 13:1224-1228, 1972.

[5] J. M. Barzdin and R. V. Freivald.
Prediction and limiting synthesis of recursively enumerable classes of functions.
In *Theory of Algorithms and Programs, Vol. 1.,* pages 112-128.
Latvian State University, Riga, U.S.S.R. 1974.
(in Russian)

[6] M. Blum.
On the size of machines.
Inform. Contr. 11:257-265, 1967.

[7] L. Blum and M. Blum.
Toward a mathematical theory of inductive inference.
Inform. Contr. 28:125-155, 1975.

[8] J. Case and C. Lynes.
Machine inductive inference and language identification.
In *Proceedings of the* 9^{th} *Colloquium on Automata, Languages, and Programming*,
Aarhus, Denmark, July, 1982.
Lecture Notes in Compter Science 140, pages 107-115, Springer, Berlin, 1982.

[9] J. Case and S. N. Manguelle.
Refinements of inductive inference by Popperian machines.
Tech. Rept. 152, Dept. of Comp. Sci., SUNY at Buffalo, NY, 1979.

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

[11] K. Chen.
Tradeoffs in machine inductive inference.
Ph.D. Dissertation, Comp. Sci. Dept., SUNY at Buffalo, NY, 1981.

[12] C. M. Cook, A. Rosenfeld, and A. R. Aronson.
Grammatical inference by hill–climbing.
Information Sciences 10:59-80, 1976.

[13] R. Daley.
On the error correcting power of pluralism in BC–type inductive inference.
Theoretical Computer Science 24:95-104, 1983.

[14] R. Daley.
Private communication, 1985.

[15] R. Daley and C. Smith.
On the complexity of inductive inference.
Tech. Rpt. 83-4, Dept. of Comp. Sci., Univ. of Pittsburgh, Pittsburgh, PA, 1983.
Inform. Contr., (to appear).

[16] S. Even.
Graph Algorithms.
Computer Science Press, Potomac, MD, 1979.

[17] J. A. Feldman.
Some decidability results in grammatical inference and complexity.
Inform. Contr. 20:244-262, 1972.

[18] R. V. Freivald.
On the complexity and optimality of computation in limit.
In *Theory of Algorithms and Programs, Vol. 1.,* pages 155-173.
Latvian State University, Riga, U.S.S.R. 1974.
(in Russian)

[19] R. V. Freivald.
Functions computable in the limit by probabilistic machines.
In *Mathematical Foundations of Computer Science*
(3^{rd} Symposium at Jadwisin near Warsaw, 1974), Springer–Verlag, 1975.

[20] R. V. Freivald.
Minimal Gödel numbers and their identification in the limit.
In *Lecture Notes in Computer Science,* 32, 219-225, 1975.

[21] R. V. Freivald.
Finite identification of general recursive functions by probabilistic strategies.
In *Proceedings of the Conference on Algebraic, Arithmetic, and Categorial Methods in Computation Theory,* pages 138-145, Akadamie–Verlag, 1979.

[22] B. R. Gaines.
Maryanski's grammatical inferencer.
IEEE Trans. on Computers C-28:62-64, 1978.

[23] E. M. Gold.
Language identification in the limit.
Inform. Contr. 10:447-474, 1967.

[24] J. E. Hopcroft and J. D. Ullman.
Introduction to Automata Theory, Languages, and Computation.
Addison–Wesley, Reading, MA, 1979.

[25] J. J. Horning.
A study of grammatical inference.
Ph.D. Dissertation, Comp. Sci. Dept., Stanford University, 1969.

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

[27] F. J. Maryanski and T. L. Booth.
Inference of finite state probabilistic grammars.
IEEE Trans. on Computers C-26:521-536, 1977.

[28] E. Minicozzi.
Some natural properties of strong identification in inductive inference.
Theoretical Computer Science 2:345-360, 1976.

[29] M. F. Neuts.
Probability.
Allyn and Bacon, Boston, MA, 1973.

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

[31] D. Osherson and S. Weinstein.
Criteria of language learning.
Inform. Contr. 52:123-138, 1982.

[32] L. Pitt.
A characterization of probabilistic inference.
Tech. Rept. YALEU/DCS/TR-319, Yale University, New Haven, CT, 1984.

[33] K. M. Podnieks.
Comparing various concepts of function prediction.
Latvii gosudarst. Univ. Ucenye Zapiski 210:68-81, 1974.
(in Russian)

[34] K. M. Podnieks.
Probabilistic synthesis of enumerated classes of functions.
Soviet Math. Dokl. 16:1042-1045, 1975.

[35] C. Smith.
The power of pluralism for automatic program synthesis.
J. ACM. 29(4):1144-1165, 1982.

[36] C. Smith.
Private communication, 1984.

[37] C. Smith and M. Velauthapillai.
Private communication, 1985.

[38] R. M. Smullyan.
Theory of formal systems.
Ann. Math. Studies 47, 1961.

[39] L. G. Valiant.
A theory of the learnable.
In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing.*
Association for Computing Machinery, 1984.

[40] A. Van der Mude and A. Walker.
On the inference of stochastic regular grammars.
Inform. and Contr. 38:310-329, 1978.

[41] R. Wiehagen, R. Freivald, and E. B. Kinber.
On the power of probabilistic strategies in inductive inference.
Theoretical Computer Science 28:111-133, 1984.