A CHARACTERIZATION OF PROBABILISTIC INFERENCE

Leonard Pitt

YALEU/DCS/TR-319

June, 1984

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

Abstract

Inductive Inference Machines (IIMs) attempt to identify functions given only input-output pairs of the functions. *Probabilistic* IIMs are defined, as is the corresponding probability that a probabilistic IIM identifies a function with respect to two common identification criteria: 'EX' and 'BC'. Let ID denote either of these criteria. Then $\mathrm{ID}_{\mathrm{prob}}(p)$ is the family of sets of functions U for which there is a probabilistic IIM identifying every $f \in U$ with probability $\geq p$. It is shown that for all positive integers n, $\mathrm{ID}_{\mathrm{prob}}(1/n)$ is properly contained in $\mathrm{ID}_{\mathrm{prob}}(1/(n+1))$, and that this discrete hierarchy is the "finest" possible. This hierarchy is shown equivalent to a hierarchy of teams of deterministic IIMs [20], and a hierarchy of frequency identification, introduced in [18], and settling an open problem. A special case of these results is that every class of functions which can be identified by some probabilistic IIM with probability $> 1/2$, can be identified deterministically. Other properties of nondeterministic and probabilistic inductive inference machines are briefly investigated.

Table of Contents

1. Introduction

1.1. Overview

Inductive inference is the study of algorithms which attempt to synthesize programs computing a function when given only examples of the function as input. Research focuses on both general theoretical properties of inference techniques, and finding specific methods for inference within particular domains. Inductive inference has applications in linguistics (the study of language acquisition), artificial intelligence, pattern recognition, cryptography, and the philosophy of science, among others. The reader interested in machine inference may find an excellent survey of both the theoretical and more concrete results in [1].

An *Inductive Inference Machine* (IIM) takes as input initial segments of the values of some total recursive function, $f(0)$, $f(1)$, $f(2)$, ..., and outputs guesses of programs based on the examples it has seen. Note that since new input values may not be consistent with a current guess, the IIM may not be able to determine at any point whether a particular guess is correct. For this reason, identification of functions is seen as an infinite process which happens "in the limit".

There are two standard criteria of successful identification in the limit of an IIM on a given function f: 'EX' and 'BC'. EX-identification requires that the infinite sequence of guesses converge to a single program computing f, while BC-identification only requires that after some finite initial segment, all the guesses be correct programs for f. There are two associated identifiability classes:

EX = $\{U \mid U$ is a set of total recursive functions such that there exists an IIM M which EX-identifies every $f \in U\}$. The class BC is defined analogously.

In this paper we investigate ***probabilistic*** Inductive Inference Machines. It is has been shown [2, 5] that there is no single deterministic inductive inference machine which can synthesize a correct program for *every* partial recursive function given only input/output pairs of the function. It seems natural to allow randomization as part of the inference machine, and then ask: "are larger 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 paper.

Ultimately, we would like to understand the interactions between randomization and various resource measures for inductive inference processes: time, space, number of examples, number of mind changes, number of anomalies of the hypothesized program, quality of intermediate hypotheses, etc. The present paper establishes the most basic framework for such investigations, by characterizing the power of probabilistic inductive inference machines with unrestricted resources.

We define probabilistic IIMs as follows: The machine M is allowed a (potentially infinite) sequence of 0-1 coin tosses; if we fix the input, $f(0)$, $f(1)$, $f(2)$, ..., each 0-1 infinite sequence then determines a sequence of guesses of M, which may or may not converge (in the EX or BC sense) to a program for f. If we consider the usual Borel measure on the infinite coin toss sequences, then the set of guess sequences that converge to programs for f (in either sense) is measurable, and is denoted Pr[M EX-identifies f] or Pr[M BC-identifies f], respectively. For $p \geq 0$, define

$\mathrm{EX}_{\mathrm{prob}}(p) = \{U \mid U$ is a set of functions such that there exists an IIM M such that Pr[M EX-identifies f] $\geq p$ for every $f \in U\}$. The class $\mathrm{BC}_{\mathrm{prob}}(p)$ is defined similarly.

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

This paper 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 Podneiks [18], and identification by teams of machines introduced by Smith [20]. (It also settles an open problem of Podneiks for frequency identification.) This is somewhat unusual, for in many cases the introduction of new computational models for inductive inference gives rise to new and "orthogonal" hierarchies of identifiability.

Previous work relating probability and inductive inference include results for identification of stochastic grammars [6, 10, 13, 15, 22], the use of randomization to reduce the number of "mind-changes" required by an IIM to identify functions [3, 19], probabilistic "concept learning" algorithms for boolean formulae [21], and more related to this investigation, the work of Freivald on the probabilistic *finite* identification hierarchy [9]:

In Freivald's model, an IIM M ***finitely*** identifies a set of functions U with probability p, if the probability is greater than p that M 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 $> 2/3$, then there is a deterministic strategy finitely identifying U. It is also shown that if $p_1 < p_2$, and for some $n \geq 0$, p_1 and p_2 both fall into the same interval $[(n+2)/(2n+3)\ ,\ (n+1)/(2n+1))$, then there is no difference between the classes which are finitely identifiable with probability p_1, and those classes finitely identifiable with probability p_2; whereas if p_1 and p_2 fall into different intervals, then larger classes of functions can be finitely identified by machines only required to be correct with probability $p_1 < p_2$.

Freivald's results motivated the work in this paper, and are superficially similar. Not many of the proof techniques for finite inference are amenable to adaptation to the more prevalent notions of "in the limit" inference.

Finally, a recent paper by Wiehagen, Freivald, and Kinber [23] 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 ϵ, 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 are shown, mostly for probabilities greater than 1/2. It is also independently proved that $\mathrm{EX}_{\mathrm{prob}}(p) = \mathrm{EX}$ when $p > 1/2$, which is a special case of our Theorem 40.

1.2. Notation

The null or empty set is denoted by $\emptyset$. We use the symbols $\subseteq$, $\subset$, $\in$, $\cup$, and $\cap$ to denote the set operations containment, proper containment, membership, union, and intersection, respectively. The symbol $\uplus$ is the set operation union, together with the assertion that the operands of the union are mutually disjoint; thus $S = \biguplus_{i \in \mathbf{N}} S_i$ states that not only is S the union of the sets $\{S_i\}$, but also that for all $i \neq j$, $S_i \cap S_j = \emptyset$. 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. $|S|$ denotes the cardinality of S; $|I|$ is the number of (not necessarily distinct) elements of the multiset I, and $|I \cap S|$ is the number of (not necessarily distinct) elements of the multiset I which are also elements of S.

The symbols $\forall$ and $\exists$ are universal and existential quantifiers, and represent "for all", and "there exists", respectively. $\mathbf{N}$ denotes the set of natural numbers $= \{0,1,2,3,...\}$. The set of real numbers is denoted by $\mathbf{R}$, and the set of rationals by $\mathbf{Q}$. If $I \subseteq \mathbf{N}$ is finite, then $max\ I$ is the largest element of I. If $\{x_k\}_{k \in \mathbf{N}}$ is a sequence of non-negative real numbers, then $\underline{\lim}_k\ x_k$ is defined as $\lim_{k \to \infty}$ infimum $\{x_i \mid i \geq k\}$. If $x \in \mathbf{R}$, 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, $(a,b] = \{p \mid p \in \mathbf{R}, a < p \leq b\}$.

If $E(k)$ is an equation containing the variable k, with k ranging over $\mathbf{N}$, then we write "$E(k)$ *a.e.* (k)" to indicate that the equation $E(k)$ is true ***almost everywhere***, or for all but finitely many values of k. We write "$E(k)$ *i.o.* (k)" to indicate that $E(k)$ is true ***infinitely often***, or for infinitely many values of k. The word "iff" means "if and only if".

Lower case letters ($i,j,k,...$) 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. $\mathbf{T}$ denotes the set of all total recursive functions. $f|_k$ is the restriction of the function f to the domain $\{x \mid x \leq k\}$. The set U will range over all subsets of $\mathbf{T}$. The graph of a function ϕ is the set of all pairs $<x, \phi(x)>$ for all x in the domain of ϕ. We write $\phi =^k f$ to indicate that $|\{x : \phi(x) \neq f(x)\}| \leq k$. Similarly, $\phi =^* f$ indicates that $\{x : \phi(x) \neq f(x)\}$ is finite. By convention, if $k \in \mathbf{N}$, then $k < *$.

1.3. Preliminary Definitions

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 [12]. Hence the numbers {0,1,2,...} 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 $<\phi_i>_{i \in \mathbf{N}}$ is an acceptable numbering of all and only the partial recursive functions [14]. If $\phi_i = f$, 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. If $\phi(x)$ is defined, and not equal to y, then we say that $\phi(x)$ converges $\neq y$.

If f is a total recursive function, then we define three sets, $GOOD_f$, $SLOW_f$, and $WRONG_f$ such that $GOOD_f \uplus SLOW_f \uplus WRONG_f = \mathbf{N}$, as follows:

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

$SLOW_f = \{i \mid \phi_i \neq f$, and for all x such that $\phi_i(x) \neq f(x)$, ϕ_i diverges on $x\}$.

$WRONG_f = \{i \mid \phi_i \neq f$, there exists a number x such that $\phi_i(x)$ converges $\neq f(x)\}$.

$GOOD_f$ is the set of "good" programs for f, $SLOW_f$ is the set of programs which are restrictions of f to some domain properly contained in $\mathbf{N}$, thus wherever they differ from f, they diverge, or are "slow", and $WRONG_f$ is the set of programs which converge to a value "wrong" for f on at least one argument.

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 [11]. We adopt the definition of L. Blum and M. Blum [4]:

> 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 $<x, y>$, 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].

If ϕ is a partial recursive function, then we say that M is fed the graph of ϕ iff each element input to M is either a pair $<x,\phi(x)>$, or a "*", and for every x in the domain of ϕ, $<x,\phi(x)>$ is input at least once to M.

Throughout this paper we will deal only with the inference of total recursive functions, but we note that most of our arguments need only minor modification to cover the case of inference of partial recursive functions. Note this assumption does not restrict IIMs to hypothesizing only indices of programs computing total functions.

Definition 1: M EX-identifies f iff when fed the graph of f in any order, M outputs infinitely many numbers, $g_1,g_2,g_3,\ldots$, and for some n, $g_n = g_{n+1} = g_{n+2} \cdots$, and g_n is (the encoding of) a program that computes the function f. (EX abbreviates "M EXplains f".)

There is another definition of EX-identification that requires M to output only finitely many numbers, the last of which is a program for f. It is easy to show that these definitions are equivalent. Furthermore, we assume without loss of generality, that every inductive inference machine outputs the guess g_n before receiving the input $f(n+1)$.

Definition 2: M BC-identifies f, iff when fed the graph of f in any order, M outputs infinitely many numbers $g_1,g_2,g_3,\ldots$, such that $\phi_{g_k} = f$ *a.e.* (k). (M eventually outputs only "Behaviorally Correct" programs.)

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.

Definition 3: Let M be an IIM. Then
$\mathrm{EX}(M) = \{f \mid M \text{ EX-identifies } f\}$.
M EX-identifies U if $U \subseteq \mathrm{EX}(M)$.
$\mathrm{EX} = \{U \mid \exists\, M \text{ such that } U \subseteq \mathrm{EX}(M)\}$.

Definition 4: Let M be an IIM. Then
$\mathrm{BC}(M) = \{f \mid M \text{ BC-identifies } f\}$.
M BC-identifies U if $U \subseteq \mathrm{BC}(M)$.
$\mathrm{BC} = \{U \mid \exists\, M \text{ such that } U \subseteq \mathrm{BC}(M)\}$.

EX is clearly contained in BC, and the containment is proper [2, 5].

We say that M is *order independent* if the sequence of guesses that M makes is independent of the particular order in which the graph of f is input to M. It is easily shown that if M is an IIM, then M can be effectively transformed into an order independent IIM M' such that EX(M) $\subseteq$ EX(M') and BC(M) $\subseteq$ BC(M') [5]. We assume without loss of generality, that all IIMs are order independent. We also assume without loss of generality, that the graph of any function to be identified by an IIM M will be presented to M in the canonical order "$<0,f(0)>$, $<1,f(1)>$, $<2,f(2)>$, ..." An order independence result for partial recursive functions is covered in [4].

C. Smith [20] introduces the notion of team inference. In this model, a team of IIM's $(M_1,M_2,...,M_n)$ identifies the function f if there is at least one i such that M_i identifies f. Each member of the team carries out a separate computation, and there is no 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. A fair portion of this paper will be devoted to relating the (yet undefined) notion of probabilistic inference to team inference. We will argue in section 7 that team inference seems to be the only natural definition of nondeterminism for inductive inference.

Definition 5: Let $\{M_1,M_2,...,M_n\}$ be a collection of IIMs. Then
EX($M_1,M_2,...,M_n$) = $\{f \mid \exists\, i$ such that M_i EX-identifies $f\}$.
The team $\{M_1,M_2,...,M_n\}$ EX-identifies U if $U \subseteq$ EX($M_1,M_2,...,M_n$).
$\mathrm{EX}_{\mathrm{team}}(n) = \{U \mid \exists\, M_1,M_2,...,M_n$ such that $U \subseteq$ EX($M_1,M_2,...,M_n$)$\}$.

Definition 6: Let $\{M_1,M_2,...,M_n\}$ be a collection of IIMs. Then
BC($M_1,M_2,...,M_n$) = $\{f \mid \exists\, i$ such that M_i BC-identifies $f\}$.
The team $\{M_1,M_2,...,M_n\}$ BC-identifies U if $U \subseteq$ BC($M_1,M_2,...,M_n$).
$\mathrm{BC}_{\mathrm{team}}(n) = \{U \mid \exists\, M_1,M_2,...,M_n$ such that $U \subseteq$ BC($M_1,M_2,...,M_n$)$\}$.

Smith shows 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 "inferrability":

Theorem 7: (Smith) For all integers $n \geq 1$, $\mathrm{EX}_{\mathrm{team}}(n) \subset \mathrm{EX}_{\mathrm{team}}(n+1)$, and $\mathrm{BC}_{\mathrm{team}}(n) \subset \mathrm{BC}_{\mathrm{team}}(n+1)$.

In the next few sections we focus mainly on probabilistic BC-identification. Somewhat more complicated arguments for probabilistic EX-identification will rely on some of the concepts introduced for BC-identification, hence we delay their introduction until section 3.2.

2. Probabilistic Inductive Inference Machines

A probabilistic IIM M is a deterministic IIM with a random 0-1 oracle called a *coin*. The IIM may "query" (or "flip") the coin from time to time, and receive the result of the flip (which is 0 or 1 equiprobably) on a special read-only coin tape.

Without loss of generality, we may assume that every probabilistic IIM outputs a guess for a program index of the function f infinitely often. Note that whether M identifies f (as defined for deterministic IIMs), depends on the sequence of coin flips that M receives from the coin oracle.

We may also modify M so that, in addition to guessing an index for f infinitely often, it follows each guess with a coin flip. Hence we require that a probabilistic IIM execute the following loop:

1. Receive a value of f
2. Guess a program index
3. Flip the coin
4. Execute a finite number of deterministic steps
5. go to 1.

Since we haven't defined what "the probability that M identifies f" means, it is not clear whether this "probability" will remain unchanged for all f after altering M to satisfy the above conventions. Moreover, for ease in presentation, we will only define this probability for machines which follow the above conventions. We claim that a suitable (similar) definition for "the probability that M identifies f" exists for all probabilistic IIMs such that for all M, there exists an M' which follows the above conventions, and if f is any function, then $\Pr(M'$ identifies $f)$ $= \Pr(M$ identifies $f)$.

2.1. Infinite Computation Trees

Given a probabilistic IIM M, we wish to talk of the probability of M exhibiting certain behaviors when given the values of the function f as input. We assume that the IIM executes the read-guess-flip-compute loop infinitely often. For a particular function f, M may follow different computations depending on the sequence of coin flips. We can represent all of M's possible computations for the function f as an infinite (complete) binary tree which we denote by $T_{M,f}$. The nodes of $T_{M,f}$ will correspond to *configurations* of M, and the edges will correspond to the results of coin flips. (A *configuration* is a structure which specifies the state of M, the contents of all of its tapes, and the positions of all of its read and write heads [12].)

In particular, the root node will correspond to the configuration of M, immediately after M

makes its first guess. According to our convention, *M*'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, *M* (according to our conventions) executes a finite number of transitions, receives the next value of f, then guesses again. The left child of the root node will correspond to the configuration that *M* reaches just after it makes its second guess, given that the first flip was heads. In general, a node of depth d in $T_{M,f}$ will correspond to the configuration of *M* reached if *M* were to run through d iterations of the read-guess-flip-compute loop, and the sequence of d coin flips that *M* received was exactly the sequence of heads and tails which lead to node n in the tree.

The nodes of $T_{M,f}$ 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 $T_{M,f}$ 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 $T_{M,f}$. 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:$\mathbf{N} \rightarrow \mathbf{N}$ on the nodes of $T_{M,f}$ by: $ind(n)$ = the guess that *M* 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 *M*'s guess for a program index for f. Note that for any probabilistic IIM *M*, 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 *M*, constructs $T_{M,f}$ through the k^{th} level.

A *path* p of $T_{M,f}$ is an infinite sequence of adjacent nodes $<t_0,t_1,t_2,t_3,...>$, 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 $T_{M,f}$.

Definition 8: Let $p = <t_0,t_1,t_2,t_3,...>$ be a path in $T_{M,f}$, and $A \subseteq \mathbf{N}$. The path p *BC-converges to A* iff $ind(t_k) \in A$ *a.e.* (k).

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

Path $p = <t_0,t_1,t_2,t_3,...>$ *BC-converges to A at node n*, iff

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

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 BC-convergence occurs.

If a path p BC-converges to the set $GOOD_f$, then p contains a sequence of coin flips which causes M to output a sequence of guesses corresponding to a single deterministic BC-identification of f. Our goal is to define $\Pr(M$ BC-identifies $f)$ as the percentage of paths of $T_{M,f}$ which BC-converge to $GOOD_f$. In the next section we do this formally. Readers unfamiliar with standard techniques for defining a probability measure (Borel sets) may wish to consult the appendix before continuing.

2.2. Probability on Infinite Computation Trees

We now precisely define probability with respect to an IIM's computation on a given input. The "experiment" for which the probability is defined is the running of M with input f, and the result is the particular infinite path that M follows which depends on the infinite sequence of results of the coin flips. Thus the set of events in which we have interest, is $\Omega = \{p \mid p$ is a path in $T_{M,f}\}$.

Following the standard methodology, we define a class of basic sets.

Definition 9: For each node $n \in T_{M,f}$, $P_n = \{$paths $p \in T_{M,f} \mid p$ contains node $n\}$.

We define the function $\Pr : \{P_n\}_{n \in T_{M,f}} \rightarrow [0,1]$ as follows:

Definition 10: $\Pr[P_n] = 2^{-d(n)}$

It is easy to see that this is what we want from our probability measure: The probability of a randomly chosen path passing through node n should be $2^{-d(n)}$ since every path must pass through exactly 1 node at depth $d(n)$ and we'd like these to be equiprobable.

Let $\mathcal{B}(\{P_n\})$ be the smallest Borel field containing $\{P_n\}_{n \in T_{M,f}}$

Lemma 11: Pr is a probability measure when extended to $\mathcal{B}(\{P_n\})$.

A sketch of the proof of Lemma 11 may be found in the appendix.

For any probabilistic IIM M, and any function f, we have defined a probability measure on the tree $T_{M,f}$. We must now define "the probability that M BC-identifies f" with respect to the measure Pr. Clearly the "probability that M BC-identifies f" ought to be $\Pr[\{$paths $p \mid p$ BC-converges to $GOOD_f\}]$.

It is not clear however, that the above set is measurable. We will show that it is by expressing it as countable unions and intersections of our basic sets.

Definition 12: $B(A) = \{p \mid p$ is a path in $T_{M,f}$, and p BC-converges to $A\}$.
$B_j(A) = \{p \mid p$ is a path in $T_{M,f}$, and p BC-converges to A at node $j\}$.

Note that for all $j \neq m$, and A, $B_j(A) \cap B_m(A) = \emptyset$: 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 $p = <t_0,t_1,t_2,t_3,...>$ is ***k-consistent*** with $B_j(A)$ iff the following two conditions hold:

1. For all i such that $j \leq i \leq k$, $ind(t_i) \in A$.
2. j is the root OR $ind(t_{j-1}) \notin A$.

Definition 13: $B_{j,k}(A) = \{p \mid p \text{ is } k\text{-consistent with } B_j(A)\}$.

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, $B(A) = \biguplus_{j \in N} B_j(A)$. 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 our basic sets $\{P_n\}$, but we will first need the following definition.

Definition 14: $N_{j,k}(A) = \{\text{nodes } n \mid d(n) = k, \text{ and } \exists \text{ path } p \in B_{j,k}(A) \text{ passing through node } n\}$.

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 15: For all j, $k \geq d(j)$, and $A \subseteq \mathbf{N}$, $B_{j,k}(A)$ is measurable, and $\Pr[B_{j,k}(A)] = |N_{j,k}(A)|/2^k$.

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 $P_x \cap P_y = \emptyset$.

($\subseteq$) 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

$$\Pr[B_{j,k}(A)] = \Pr[\biguplus_{x \in N_{j,k}(A)} P_x] = \sum_{x \in N_{j,k}(A)} \Pr[P_x] = \sum_{x \in N_{j,k}(A)} 2^{-d(x)} = \sum_{x \in N_{j,k}(A)} 2^{-k}$$
$$= |N_{j,k}(A)|/2^k. \quad \Box$$

We are finally ready to define what is meant by Pr[M BC-identifies f)].

Definition 16: Let M be a probabilistic IIM, and $T_{M,f}$ be the infinite computation tree of M on input f. Then Pr[M BC-identifies f] = Pr[$B(GOOD_f)$].

This defines the probability that M BC-identifies f as the fraction of paths of $T_{M,f}$ which BC-converge to a correct program index for f; or the fraction of M's possible computations which correspond to a single deterministic BC-identification of the function f.

Definition 17: Let M be an IIM. Then
$BC_p(M) = \{f \mid \Pr[M \text{ BC-identifies } f] \geq p\}$.
M BC-identifies f with probability p if $f \in BC_p(M)$.
M BC-identifies U with probability p if $U \subseteq BC_p(M)$.
$BC_{prob}(p) = \{U \mid \exists\, M \text{ such that } U \subseteq BC_p(M)\}$.

It is clear that if $p_1 \leq p_2$ then $BC_{prob}(p_2) \subseteq BC_{prob}(p_1)$. We will show under what circumstances the containment is strict.

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 18: For all nodes j, for all $A \subseteq \mathbf{N}$, and for all $T_{M,f}$,

1. For all $k \geq d(j)$, $B_{j,k}(A) \supseteq B_{j,k+1}(A)$.
2. For all $k \geq d(j)$, $\Pr[B_{j,k}(A)] \geq \Pr[B_j(A)]$.
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 $T_{M,f}$. 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 19: For all $A \subseteq \mathbf{N}$, for all $p \in \mathbf{R}$ 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: $B(A) = \biguplus_{j=1}^{\infty} B_j(A)$, so

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

By a simple property of limits there is a least v such that $\sum_{j=1}^{v} \Pr[B_j(A)] > p$. □

3. Relationship Between Probabilistic and Team Inference Strategies

In this section we examine the relationship between team and probabilistic inference strategies. We will demonstrate that there is an infinite hierarchy of probabilistic inference classes, and that this hierarchy is identical to the hierarchy of team inference shown in [20]. This is achieved by showing that probabilistic IIMs can "simulate" a team of IIMs, and vice versa. Section 3.1 proves the main results for the BC identification criterion. In section 3.2 we introduce the EX identification criterion for probabilistic IIMs, and prove that the same relationships hold. Finally, in section 4 we summarize the results and the main theorems of this section, and briefly discuss their consequences.

3.1. BC Probability and Teams

3.1.1. Probabilistic Simulation of Teams of IIMs

We show that

Theorem 20: For all integers $n \geq 1$, $\mathrm{BC}_{\mathrm{team}}(n) \subseteq \mathrm{BC}_{\mathrm{prob}}(1/n)$.

To show this, we must show that for any set of functions U, if there exists a team of n deterministic machines $M_1, M_2, \ldots, M_n$ which BC-identifies U, then there is a single probabilistic machine M which BC-identifies U with probability $\geq 1/n$.

If our definition of probabilistic machines allowed the machine to "flip" an n-sided coin for arbitrary integers n, then the following "proof" would suffice:

Let $U \in \mathrm{BC}_{\mathrm{team}}(n)$. Then there exists $M_1, M_2, \ldots, M_n$ such that $U \in \mathrm{BC}(M_1, M_2, \ldots, M_n)$. Let M be a probabilistic machine which flips an n-sided coin (with the possible results being $\{1,2,\ldots n\}$ equiprobably). If the result of the flip is i, then M simulates M_i on input $f \in U$.

With minor modification, essentially the same "proof" will work for probabilistic machines with only a 2-sided coin. The probabilistic machine will simulate a single n-sided coin flip by using the 2-sided coin to compute a binary fraction *frac* in the interval [0,1]. Depending on which interval $((i-1)/n,\ i/n)$ *frac* is in, M will simulate the i^{th} team member.

Proof: On input $f \in U$, M outputs the guess "0" while flipping coins and adding a bit to the binary fraction *frac* being built up. (The j^{th} flip will add 2^{-j} to the value of *frac* if the result is "heads", 0 is added if the result is tails). M also computes as many places in the binary representation of each element of the set of fractions $F = \{0,\ 1/n,\ 2/n,\ \ldots,\ n/n\}$. M determines whether *frac* differs from all of the binary representations of elements of F. If not, M continues flipping. If ever M is able to determine that *frac* differs in at least one bit position from each element of F, then M determines which interval *frac* falls into, say interval $((i-1)/n,\ i/n)$, and then M simulates the i^{th} team member.

There is a straightforward way to construct an IIM which does the above, and a simple argument which shows that for all numbers $s \in [0,1]$, Pr[{paths for which *frac* $< s$}] $= s$. This fact, together with the fact that Pr[{paths for which *frac* $= s$}] $= 0$, is used to easily show that the probability that *frac* falls into any one of the intervals is exactly $1/n$. A small technical point involves the possibility that the generation of *frac* is halted because it differs from the representations of elements of F whereas due to possible duplicate representations, some infinite bit extension of *frac* would represent the same value as i/n for some i. The probability associated with this event is 0, as there are only a finite number of paths for which this occurs.

Now for each i, the probability that M chooses to simulate M_i is exactly $1/n$. Hence the probability that M BC-identifies f is at least $1/n$ for all $f \in U$. Thus if $U \in \mathrm{BC}_{\mathrm{team}}(n)$, then $U \in \mathrm{BC}_{\mathrm{prob}}(1/n)$. This completes the proof of Theorem 20. □

3.1.2. Team Simulation of Probabilistic IIMs

In this section we show under what circumstances a team of IIMs may be used to simulate a probabilistic IIM.

Theorem 21: For all integers $n \geq 1$, for all $p \in \mathbf{R}$, if $1/(n+1) < p \leq 1$ then $\mathrm{BC}_{\mathrm{prob}}(p) \subseteq \mathrm{BC}_{\mathrm{team}}(n)$.

Corollary: $\mathrm{BC}_{\mathrm{prob}}(1/n) = \mathrm{BC}_{\mathrm{team}}(n)$.

The corollary follows from Theorems 21 and 20. Thus the probabilistic BC-identification hierarchy contains the team BC-identification hierarchy. Theorem 21 also implies that the probabilistic hierarchy is "no finer" than the team hierarchy. We shall discuss this more in section 4.

We first note that a special case of Theorem 21 has a very simple proof. If $n = 1$, then the theorem asserts that if there is a probabilistic IIM M which BC-identifies a set of functions U with probability $p > 1/2$, then there is a deterministic IIM M_1 which BC-identifies U. To prove this, we merely need to argue that since $\Pr[B(GOOD_f)] > 1/2$, the k^{th} level of $T_{M,f}$ consists of $> 1/2$ correct programs for all but finitely many levels k. The machine M_1 which will identify U deterministically will, given $f \in U$, construct $T_{M,f}$, and for its k^{th} guess output a program which does a dovetail and majority vote of the computations of the programs whose indices occur at the k^{th} level of $T_{M,f}$. A simple argument shows that M_1 BC-identifies U.

In order to prove Theorem 21, we will first need some definitions, and an important lemma.

Definition 22: For all $t \in \mathbf{Q}$, a multiset I of program indices is a *t-threshold* list for f iff $|I \cap WRONG_f| < t < |I \cap GOOD_f|$.

Consider IIMs which, rather than outputting a sequence of program indices, instead output a sequence of ordered pairs $<t_1, I_1>$, $<t_2, I_2>$, ..., where $t_i \in \mathbf{Q}$, and I_i are finite multisets of program indices.

Definition 23: M $\mathrm{BC}_{\mathrm{threshold}}$-identifies f ($f \in \mathrm{BC}_{\mathrm{threshold}}(M)$), iff when fed the graph of f, M outputs infinitely many ordered pairs $\{<t_i, I_i>\}$, and I_k is a t_k-threshold list for f *a.e.* (k).

$\mathrm{BC}_{\mathrm{threshold}} = \{U \mid \exists\, M \text{ such that } U \subseteq \mathrm{BC}_{\mathrm{threshold}}(M)\}$.

Lemma 24: $\mathrm{BC}_{\mathrm{threshold}} = \mathrm{BC}$.

The intuition behind the proof of Lemma 24 is that a "threshold-plurality" vote of the programs of the list I_k can be used to identify f. This is because each element of the list is in one of the sets $GOOD_f$, $WRONG_f$, or $SLOW_f$. If a program is in $SLOW_f$, it can never halt with an incorrect answer.

Proof: Let $U \subseteq \mathrm{BC}_{\mathrm{threshold}}(M)$. We construct M' which BC-identifies U. M''s k^{th} guess for a program for f is computed as follows: When fed the values of f, M' simulates M on f, and obtains the k^{th} ordered pair $<t_k, I_k>$ output by M. M' then outputs the program p_k, which on input x, dovetails the computations of $\{\phi_i(x)\}$ for $i \in I_k$, until $s > t_k$ elements $\{i_1, i_2, \ldots, i_s\}$ of I_k have been found such that all s computations $\{\phi_{i_n}(x) \mid 1 \leq n \leq s\}$ have been completed and the values of all s of these computations yield the same result y. Then p_k outputs the value $p_k(x) = y$.

To see that M' BC-identifies U, let $f \in U$. Since M outputs t_k-threshold lists *a.e.* (k), there is a number k_0 such that for all $k \geq k_0$, I_k is a t_k-threshold list for f. Let $k > k_0$. We show that p_k computes f. Note that $p_k(x)$ converges for all inputs x, since I_k is a t_k-threshold list for f, we have $|I_k \cap GOOD_f| > t_k$; *i.e.* the number of correct programs for f in the list I_k is greater than t_k, so after some finite number of simulation steps at least t_k values must have been computed.

Now let $S = \{i_1, i_2, \ldots, i_s\}$ be the elements of I_k which p_k finds. If $y \neq f(x)$, then all s elements of S are in $WRONG_f$, and $|I_k \cap WRONG_f| \geq s > t_k$, which contradicts the fact that I_k is a t_k-threshold list for f. Hence $p_k(x) = f(x)$ for all x. □.

In order to prove Theorem 21, we need only show that if $U \subseteq \mathrm{BC}_p(M)$ with $p > 1/(n+1)$, then there is a team $\{M_1, M_2, \ldots, M_n\}$ such that for every $f \in U$, there is an i with M_i $\mathrm{BC}_{\mathrm{threshold}}$-identifying f.

The intuition is as follows: Since the "weight" of paths which BC-converge to correct programs for f is $> 1/(n+1)$, we can show that the fraction of correct programs at each level of the computation tree $T_{M,f}$ is greater than $1/(n+1)$ 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 $\mathrm{BC}_{\mathrm{threshold}}$-identify f might simply output the multiset of program indices found at each level, since as we will show below, this list will contain greater than the fraction $1/(n+1)$ of $GOOD_f$ indices. The problem with this strategy is that each level might contain greater than the fraction $1/(n+1)$ 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 $BC_{threshold}$-identify, and hence 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 25: Let $T_{M,f}$ be a computation tree, and $A \subseteq \mathbf{N}$ be a set of program indices. Then

$L_k = \{n \mid n \text{ is a node at level } k \text{ of } T_{M,f}\}$.

$L_k(A) = \{n \in L_k \mid ind(n) \in A\}$.

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

We employ the following

Lemma 26: For all $A \subseteq \mathbf{N}$, for all f, and all probabilistic IIMs M, If $\Pr[B(A)] > p$ in the tree $T_{M,f}$, then $|L_k(A)| > p2^k$ *a.e.* (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 19 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 18, 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 15,

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

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 $N_{j,k}(A) \cap N_{m,k}(A) = \emptyset$ 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.* $|L_k(A)| > p2^k$ *a.e.* (k), proving the lemma. □

We are now ready to prove Theorem 21. Let $U \in \mathrm{BC}_{\mathrm{prob}}(p)$, with $p > 1/(n+1)$. Then there is a probabilistic IIM M which BC-identifies every $f \in U$ with probability $\geq p > 1/(n+1)$. We construct a team of n deterministic IIMs such that for all $f \in U$, there is a team member which $\mathrm{BC}_{\mathrm{threshold}}$-identifies f, hence BC-identifies f.

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

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

Lemma 26 asserts that

$|L_k(GOOD_f)| > 2^k/(n+1)$ *a.e.* (k).

Now since $|L_k(GOOD_f)| + |L_k(SLOW_f)| + |L_k(WRONG_f)| = 2^k$ for all k,

$|L_k(WRONG_f)| < 2^k n/(n+1)$ *a.e.* (k).

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

Possibility n: $|L_k(WRONG_f)| < 2^k n/(n+1)$ *a.e.* (k), and

$|L_k(WRONG_f)| \geq 2^k(n-1)/(n+1)$ *i.o.* (k).

⋮

Possibility i: $|L_k(WRONG_f)| < 2^k i/(n+1)$ *a.e.* (k), and

$|L_k(WRONG_f)| \geq 2^k(i-1)/(n+1)$ *i.o.* (k).

⋮

Possibility 1: $|L_k(WRONG_f)| < 2^k/(n+1)$ *a.e.* (k), and

$|L_k(WRONG_f)| \geq 0$ *i.o.* (k).

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 $\mathrm{BC}_{\mathrm{threshold}}$-identify f.

The idea behind the construction is fairly simple. If a machine M_i knows roughly what the fraction of "*WRONG*" 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 $T_{M,f}$ such that the fraction of *WRONG* guesses among those output is at least $(i-1)/(n+1)$, and then cancel these wrong guesses.

If it is also true that past some point, the fraction of *WRONG* guesses is bounded above by $i/(n+1)$, then M_i will be able to form (in the limit) sets of indices for which at least the fraction $1/(n+1)$ are correct, and strictly less than this are *WRONG* indices. Thus M_i will be able to $\mathrm{BC}_{\mathrm{threshold}}$-identify f.

We now exhibit the team of n machines.

Machine M_i

```
1. k_old ← 0
2. LOOP :
3.        Simulate M on input values received from f, and build T_{M,f}.
4.        DOVETAIL the computations of φ_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 ≥ 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 ← The multiset of indices of nodes in L_k − CANCEL_k
6.        OUTPUT the ordered pair <2^k/(n+1), I_k>
7.        k_old ← k
8. GO TO LOOP
```

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, $\phi_{ind(n)}(x)$ converges $\neq f(x)$. Thus $CANCEL_k$ contains only elements of $L_k(WRONG_f)$. Note that we have not specified exactly when M reads values from f, and what it outputs for each new element of the graph of f. M receives a new element of the graph of f between each of its steps (including each simulation step in the dovetail), and outputs $<t_{k_{old}}, I_{k_{old}}>$ (the last ordered pair computed) after each such input, until it has had enough time to find a new value of k satisfying the conditions specified.

Now let $U \subseteq \mathrm{BC}_p(M)$, with $p > 1/(n+1)$, and let $M_1, M_2, \ldots, M_n$ be defined as above. Then to prove Theorem 21 we only need to prove the following lemma:

> **Lemma 27:** 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 M_i $\mathrm{BC}_{\mathrm{threshold}}$-identifies f.

Proof:
To prove the lemma, we must show

1. M_i outputs infinitely many ordered pairs $<2^k/(n+1), I_k>$.
2. $|I_k \cap WRONG_f| < 2^k/(n+1)$ *a.e.* (k).
3. $|I_k \cap GOOD_f| > 2^k/(n+1)$ *a.e.* (k).

To show 1, we note that the only possible way for M_i to output only finitely many pairs $<2^k/(n+1), I_k>$, 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, $|L_k(WRONG_f)| \geq 2^k/(n+1)$ *i.o.* (k), therefore there is some $k > k_{\text{old}}$ with $|L_k(WRONG_f)| \geq 2^k/(n+1)$. 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 $\geq 2^k/(n+1)$ 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

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

By assumption on i, $|L_k(WRONG_f)| < i2^k/(n+1)$ *a.e.* (k),

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

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

Thus $|I_k \cap WRONG_f| < i2^k/(n+1) - (i-1)2^k/(n+1)$ *a.e.* (k),

and $|I_k \cap WRONG_f| < 2^k/(n+1)$ *a.e.* (k).

Finally, to see that $|I_k \cap GOOD_f| > 2^k/(n+1)$ *a.e.* (k), 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

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

That is, the number of $GOOD$ indices in I_k equals the number of nodes with $GOOD$ indices at level k. Now since $\Pr[B(GOOD_f)] \geq p > 1/(n+1)$, as noted earlier, Lemma 26 implies

$|I_k \cap GOOD_f| = |L_k(GOOD_f)| > 2^k/(n+1)$ *a.e.* (k).

This completes the proof of Lemma 27, and Theorem 21. □

3.2. EX Probability and Teams

In this section we define probabilistic EX-identification, and prove theorems analogous to the theorems 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 somewhat more complicated machinery.

3.2.1. Definitions for Probabilistic EX-identification

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

Definition 28: Let $p = <t_0,t_1,t_2,t_3,...>$ be a path in $T_{M,f}$, and j be a program index. The path p *converges to* j iff $ind(t_k) = j$ *a.e.* (k).

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

Definition 29: Path $p = <t_0,t_1,t_2,t_3,...>$ *converges at node* n, iff

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

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 30: $C(A) = \{p \mid$ path p in $T_{M,f}$, and $\exists\, a \in A$ such that p converges to $a\}$.

Let M be a probabilistic IIM. Then Pr[M EX-identifies f] $=$ Pr[$C(GOOD_f)$].

This defines the probability that M EX-identifies f as the fraction of paths of $T_{M,f}$ which converge to a correct program index for f; or the fraction of M's possible computations which correspond to a single deterministic EX-identification of the function f.

Definition 31: Let M be an IIM. Then
$EX_p(M) = \{f \mid Pr[M \text{ EX-identifies } f] \geq p\}$,
M EX-identifies f with probability p if $f \in EX_p(M)$,
M EX-identifies U with probability p if $U \subseteq EX_p(M)$,
$EX_{prob}(p) = \{U \mid \exists\, M \text{ such that } U \subseteq EX_p(M)\}$.

Clearly, if $p_1 \leq p_2$ then $EX_{prob}(p_2) \subseteq EX_{prob}(p_1)$. We will see in the next section when the containment is proper, and discuss the consequences in section 4.

As in the BC case, we must show that Pr is defined on $C(GOOD_f)$, *i.e.* that $C(GOOD_f)$ is a measurable set. We show that for all $A \subseteq \mathbf{N}$, $C(A)$ is measurable. We define[1]

$C_n = \{p \mid p$ is a path in $T_{M,f}$, and p converges at node $n\}$.

A path $p = <t_0,t_1,t_2,t_3,...>$ is ***k-consistent*** with C_n iff the following two conditions hold:

1. For all i such that $d(n) \leq i \leq k$, $ind(t_i) = ind(n)$.
2. n is the root OR $ind(t_{d(n)-1}) \neq ind(n)$.

$C_{n,k} = \{p \mid p$ is k-consistent with $C_n\}$.

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

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

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

Clearly $C(A) = \bigcup_{ind(n) \in A} C_n$. Also $C_n = \bigcap_{k=d(n)}^{\infty} C_{n,k}$.

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

We end this section with some important lemmas. The following two lemmas are analogues of Lemmas 18 and 19. The proofs are omitted.

Lemma 32: For all nodes n, and for all $T_{M,f}$,

1. For all $k \geq d(n)$, $C_{n,k} \supseteq C_{n,k+1}$
2. For all $k \geq d(n)$, $\Pr[C_{n,k}] \geq \Pr[C_n]$.
3. $\Pr[C_n] = \lim_{k \to \infty} \Pr[C_{n,k}]$.

Thus the sets $\{C_{n,k}\}$ are increasingly better estimates of the set C_n as k increases.

Lemma 33: For all $A \subseteq \mathbf{N}$, for all $p \in \mathbf{R}$ such that $0 \leq p \leq 1$, if $\Pr[C(A)] > p$, then there exists nodes $\{n_1,n_2,...,n_k\}$ such that for all i, $ind(n_i) \in A$, and $\Pr[\biguplus_{j=1}^{k} C_{n_j}] > p$.

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

[1]These definitions are independent of the set A.

converge at *at most* one node.

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

Lemma 34: $\Pr[C_{j,k}]$ is computable from the first k levels of $T_{M,f}$.

Proof: $\Pr[C_{j,k}] = \Pr[B_{j,k}(\{ind(j)\})] = |N_{j,k}(\{ind(j)\})|/2^k$, by Lemma 15. Thus we need only show that for all k, $N_{j,k}(\{ind(j)\})$ is computable from the first k levels of $T_{M,f}$. 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 $T_{M,f}$ 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 and Probabilistic Simulations

The following theorem is analogous to Theorem 20. The proof is identical.

Theorem 35: For all integers $n \geq 1$, $\mathrm{EX}_{\mathrm{team}}(n) \subseteq \mathrm{EX}_{\mathrm{prob}}(1/n)$.

We are surprised to find that analogues of Theorem 21 and its corollary exist for EX-identification, since the majority voting techniques do not seem to work in this case:

Theorem 36: For all integers $n \geq 1$, for all $p \in \mathbf{R}$, if $1/(n+1) < p \leq 1$ then $\mathrm{EX}_{\mathrm{prob}}(p) \subseteq \mathrm{EX}_{\mathrm{team}}(n)$.

Corollary: $\mathrm{EX}_{\mathrm{prob}}(1/n) = \mathrm{EX}_{\mathrm{team}}(n)$.

To prove Theorem 36, we will need the following definition and lemma.

We say that a finite list I of program indices is a ***correct list for f*** if I contains at least one element of $GOOD_f$.

The class OEX was introduced in [5][2]:

Definition 37: M OEX-identifies f (written $f \in \mathrm{OEX}(M)$), 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 L for f such that $I_k = L$ *a.e.* (k).

$\mathrm{OEX} = \{U \mid \exists M \text{ such that } U \subseteq \mathrm{OEX}(M)\}$.

Case and Smith [5] prove a generalization of the following lemma.

Lemma 38: OEX = EX.

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

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 L, M' can cancel (in the limit) every element of L which differs from f by converging $\neq f$ for some argument. Then M' can construct a program for f which depends only on the remaining elements of L, none of which can converge with an incorrect answer.

M', on input $f|_k$, simulates M and obtains the list I_k. M' tries to compute, allowing k steps for each computation, the values $\{\phi_i(j) \mid i \in I_k,\ 1 \leq j \leq k\}$. M' sets I_k' to $I_k - \{i \in I_k \mid \exists\, j \leq k$ and in $\leq k$ steps, $\phi_i(j)$ converges $\neq f(j)\}$. M' then outputs the index of the program $RACE_{I_k'}$, which on input x, dovetails the computations of $\{\phi_i(x) \mid i \in I_k'\}$, and outputs the first value computed.

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 $k > k_0$, $I_k = L$, a correct list for f; and for all $i \in I_k \cap WRONG_f$, M' cancels i within k steps.

No further cancellations occur once $k \geq k_0$, and M' "settles" on the list L'. Then M' converges to outputting the index of $RACE_{L'}$, and we show that $RACE_{L'}$ computes f. Since L' contains a correct program index for f (none ever get cancelled), $RACE_{L'}$ halts on all inputs. Furthermore, L' consists only of correct indices for f, and indices in $SLOW_f$ (indices of programs computing restrictions of f), hence the value output by $RACE_{L'}(x) = f(x)$. □

It is now clear that Theorem 36 follows from:

Lemma 39: If $U \in \mathrm{EX_{prob}}(p)$, $p > 1/(n+1)$, then there exists a team $\{M_1, M_2, \ldots, M_n\}$ of deterministic IIMs such that for all $f \in U$, there is some i such that M_i OEX-identifies (and hence EX-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 M is a probabilistic IIM which EX-identifies U with probability $p > 1/2$, then certainly a program which did a "majority vote" of the programs at each level of $T_{M,f}$ 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 M *EX*-identifies f with probability $> 1/2$, that there is some correct index of f, say i, such that in the limit $> 1/2$ of the nodes at each level of $T_{M,f}$ had index i. This is unfortunately not true, since probabilistic EX-identification was defined to capture the intuitive notion that "when you run M with input f, the probability that you get a correct EX-identification is at least p." This doesn't imply that there is some single index for f which M 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.

The idea behind the proof of Lemma 39 is that rather than look only at levels of the tree, each deterministic IIM of the team will have to scan the tree, and identify converging paths, and nodes at which this convergence occurs.

Proof of Lemma 39:

Let M be the probabilistic machine which identifies U with probability $p > 1/(n+1)$. For a particular $f \in U$, we'll informally use the term "weight" of a set of paths P in $T_{M,f}$ to mean $\Pr[P]$, as this term more accurately suggests the appropriate intuition. (The entire tree $T_{M,f}$ has weight 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 non-determinism 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 $1 \le i \le n$, M_i assumes that the total weight of all converging paths is in the half-open interval $(i/(n+1), (i+1)/(n+1)]$. 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 M be a probabilistic IIM with $U \subseteq \mathrm{EX}_p(\mathrm{M})$, and $p > 1/(n+1)$. Then let the team $\{M_1, M_2, \ldots, M_n\}$ be the following machines:

Machine M_i

1. On input $f|_k$, simulate M with input $f|_k$, and construct T_k = the finite tree consisting of the first k levels of $T_{M,f}$.

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
$$\sum_{j=1}^{c_k} \Pr[C_{j,k}] \ge i/(n+1)$$
(If no such c_k exists, then output $\emptyset$)

4. Output $\{ind(i) \mid 1 \le i \le c_k\}$

Note that each step of M_i is a simple, computable operation: Step 2 can be done by Lemma 34. We comment that c_k exists for all k, but this is not necessary for the proof.

We show that for all $f \in U$, there is an i such that M_i converges to 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:

$\Pr[C(GOOD_f)] \geq p > 1/(n+1)$ by the definition of "M EX-identifies f with probability p."

$C(\mathbf{N})$ = is the set of paths which converge to *any* index (good or bad), so clearly

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

and therefore

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

Let $m = max\ \{i \mid i/(n+1) < \Pr[C(\mathbf{N})]\}$. The value m is well defined, since $1/(n+1) < \Pr[C(\mathbf{N})] \leq (n+1)/(n+1)$. 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 converging paths is greater than $m/(n+1)$. By Lemma 33, there exists a finite set of nodes V, with the weight of the paths converging at a node in V greater than $m/(n+1)$. 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 32), 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 $m/(n+1)$. (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 $\geq m/(n+1)$, and the estimate of these weights are becoming better in the limit. More formally:

By the definition of m, $m/(n+1) < \Pr[C(\mathbf{N})] \leq (m+1)/(n+1)$. Since $\Pr[C(\mathbf{N})] > m/(n+1)$ Lemma 33 gives nodes $n_1,n_2,\ldots,n_v$ with $ind(n_i) \in \mathbf{N}$ and $\sum_{i=1}^{v} \Pr[C_{n_i}] > m/(n+1)$. Since *all* nodes j have $ind(j) \in \mathbf{N}$, this implies that there exists a smallest numbered node s, such that

$$\sum_{j=1}^{s} \Pr[C_j] > m/(n+1)$$

(choosing $s \geq max\ \{n_i\}$ will certainly satisfy the inequality).

Now for all $k \geq d(s)$, nodes 1,2,...s will be in T_k, and furthermore, by Lemma 32

$\sum_{j=1}^{s} \Pr[C_{j,k}] \geq \sum_{j=1}^{s} \Pr[C_j] \geq m/(n+1)$, hence $c_k \leq s$ *a.e.* (k) in step 3 of M_m.

Now Lemma 39 follows from:

Claim:

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 $c_k \leq s$ *a.e.* (k). Now, by Lemma 32, 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 (*a.e.* (k)), since c_k was chosen as the ***smallest*** value satisfying the inequality $\sum_{j=1}^{c_k} \Pr[C_{j,k}] \geq m/(n+1)$, 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, $\sum_{j=1}^{t} \Pr[C_{j,k}] \geq m/(n+1)$. This implies that $\sum_{j=1}^{t} \Pr[C_j] \geq m/(n+1)$, 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 $1/(n+1)$, and the weight of paths converging to any index is less than or equal to $(m+1)/(n+1)$. It follows that the weight of paths converging to an index which is not an index for f is strictly less than $m/(n+1)$. But M has found a list of indices with weight greater than or equal to $m/(n+1)$, hence not all of the indices on the list can be incorrect. More formally,

Define $BAD_f = \mathbf{N} - GOOD_f$, the set of "bad" guesses for f.

Then

$$C(\mathbf{N}) = C(GOOD_f) \uplus C(BAD_f)$$

so that

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

We also know that

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

and

$$\Pr[C(GOOD_f)] > 1/(n+1).$$

We conclude that $\Pr[C(BAD_f)] < m/(n+1)$.

Now observe that the set of indices $I = \{ind(1), ind(2), ...ind(s)\}$ has the property that $\sum_{j=1}^{s} \Pr[C_j] \geq m/(n+1)$, that is, $\Pr[C(I)] \geq m/(n+1)$, 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 $\Pr[C(BAD_f)] \geq m/(n+1)$. This completes the proof of the claim, Lemma 39, and Theorem 36. □

4. Relationship Between the Team and Probabilistic Inference Hierarchies

Let the symbol "ID" stand for each of the symbols "EX" and "BC". Theorems 20, 21, 35, 36, and their corollaries can then be generalized as

Theorem 40:

1. For all integers $n \geq 1$, $\text{ID}_{\text{team}}(n) \subseteq \text{ID}_{\text{prob}}(1/n)$.
2. For all integers $n \geq 1$, for all $p \in \mathbf{R}$, if $1/(n+1) < p \leq 1$ then $\text{ID}_{\text{prob}}(p) \subseteq \text{ID}_{\text{team}}(n)$.
3. For all integers $n \geq 1$, $\text{ID}_{\text{prob}}(1/n) = \text{ID}_{\text{team}}(n)$.

Part 3, together with the team hierarchy theorem (Theorem 7), implies that

Theorem 41: For all integers $n \geq 1$, $\text{ID}_{\text{prob}}(1/n) \subset \text{ID}_{\text{prob}}(1/(n+1))$.

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 $1/(n+1) < p \leq 1/n$;

Clearly $\text{ID}_{\text{prob}}(1/n) \subseteq \text{ID}_{\text{prob}}(p)$.

Parts 2 and 1 of Theorem 40 give the two containments

$\text{ID}_{\text{prob}}(p) \subseteq \text{ID}_{\text{team}}(n) \subseteq \text{ID}_{\text{prob}}(1/n)$,

and therefore $\text{ID}_{\text{prob}}(p) = \text{ID}_{\text{prob}}(1/n)$.

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

Corollary: For all $p \in \mathbf{R}$, and for all positive integers n,

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

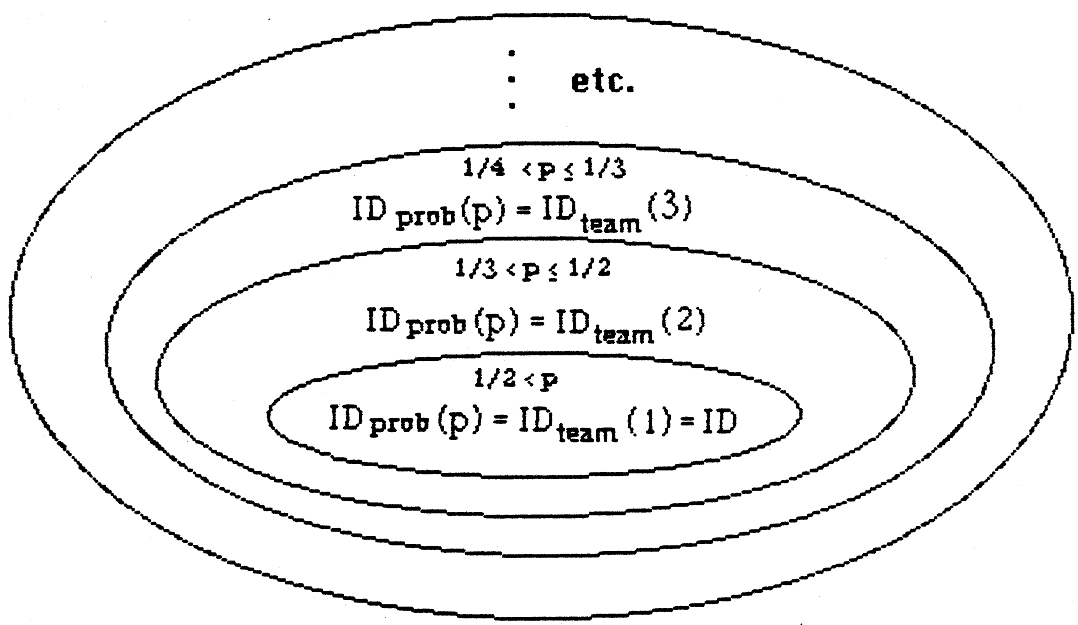
We conclude that the probabilistic hierarchy is *exactly* the team hierarchy.

Of particular interest is the following special case ($n = 1$) of our results:

If $p > 1/2$ then $ID_{prob}(p) = ID$ $(= ID_{team}(1))$.

That is, if we have a probabilistic IIM which EX- (BC-) identifies the set of functions U with probability $p > 1/2$, then there is a deterministic IIM which EX- (BC-) identifies U. This result is shown independently by Wiehagen, Freivald, and Kinber [23] for the EX case.

The following picture 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 [20]:

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 $\geq 3/11$. Since $3/11 > 1/4$, 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.

5. Frequency Identification

5.1. 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 [18], and capture this intuitive notion.

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

$$F_k = |\{i : \phi_{g_i} = f \text{ and } 1 \leq i \leq k\}| \; / \; k$$

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 42: M BC-identifies f with frequency p iff when fed the graph of f in any order[3] $\underline{\lim}_k F_k \geq p$.

[3]Podnieks' definition was that M need only have correct frequency behavior when fed the graph of f in the canonical order $<f(0), f(1), \ldots>$. 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.

M BC-identifies U with frequency p iff for all $f \in U$, M BC-identifies f with frequency p.

$BC_{freq}(p) = \{U \mid$ there exists an IIM M which BC-identifies U with frequency $p\}$.

In [18], Podnieks shows[4] that for all integers $n \geq 1$, $BC_{freq}(1/n) \subset BC_{freq}(1/(n+1))$. 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 "break points" for this hierarchy are at exactly the numbers of the form $1/n$. More specifically, we show that this "frequency" hierarchy is identical to the BC team hierarchy.

We begin by showing

Theorem 43: For all integers $n \geq 1$, $BC_{team}(n) \subseteq BC_{freq}(1/n)$.

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 $1/n$. To do this, it is sufficient to show that for every $f \in U$, and for all $\epsilon > 0$, $F_k > 1/n - \epsilon$ *a.e.* (k). 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 $i > c$, $ROUND_i$ contains at least one correct guess.

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

- $x > c$
- $1/(cn+xn) < \epsilon/2$
- $x/(cn+xn) > 1/n - \epsilon/2$

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

$F_k = (\#$ 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 $x+c^{th}$ round. Hence:

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

[4] Actually, his results include the stronger statement that for all $\epsilon > 0$, $BC_{freq}(1/n + \epsilon) \subset BC_{freq}(1/n)$.

We've shown that for all $\epsilon > 0$, and for every $f \in U$, M outputs a sequence such that $F_k > 1/n - \epsilon$ *a.e.* (k). It follows that M BC-identifies U with frequency $1/n$, which completes the proof of Theorem 43. □

Now we show that

Theorem 44: For all integers $n \geq 1$, for all $p \in \mathbf{R}$, if $1/(n+1) < p \leq 1$ then $\mathrm{BC}_{\mathrm{freq}}(p) \subseteq \mathrm{BC}_{\mathrm{team}}(n)$.

Theorem 44 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:

Proof: Let $U \in \mathrm{BC}_{\mathrm{freq}}(p)$, with $p > 1/(n+1)$, and let M BC-identify U with frequency p. To prove Theorem 44, we construct a team $\{M_1, M_2, ..., M_n\}$ such that for all $f \in U$, there is some i such that M_i $\mathrm{BC}_{\mathrm{threshold}}$-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 $p > 1/(n+1)$, $|I_k(GOOD_f)| > k/(n+1)$ *a.e.* (k); and therefore $|I_k(WRONG_f)| < kn/(n+1)$ *a.e.* (k).

As in the proof of Theorem 21, there are n distinct and mutually exclusive possibilities for how the sequence $I_k(WRONG_f)$ behaves in the limit; let $1 \leq i \leq n$:

Possibility i: $|I_k(WRONG_f)| < ik/(n+1)$ *a.e.* (k), and

$|I_k(WRONG_f)| \geq (i-1)k/(n+1)$ *i.o.* (k).

The construction and proof now follow that on page 17. We display M_i, and briefly sketch the proof.

Machine M_i

```
k_old ← 0
LOOP :
        Simulate M on input values received from f, and let
          g_1,g_2,g_3,... be the sequence of guesses output by M.
        DOVETAIL the computations of φ_{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 ≥ (i−1)k/(n+1)
          elements in the multiset CANCEL_k, the multiset of guesses
          among the first k guesses of M which have been observed
          to be in WRONG_f.
        S_k ← {g_i | 1 ≤ i ≤ k} − CANCEL_k
        OUTPUT the ordered pair <k/(n+1), S_k>
        k_old ← k
GO TO LOOP
```

Let M_i satisfy the i^{th} possibility. We've already observed that $|I_k(GOOD_f)| > k/(n+1)$ *a.e.* (k). 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 $|S_k \cap GOOD_f| > k/(n+1)$ *a.e.* (k). Also, by assumption on i, $|I_k(WRONG_f)| \geq (i-1)k/(n+1)$ *i.o.* (k), and hence M_i can find successively larger values of k for which it outputs $<k/(n+1), S_k>$, such that by assumption on i,

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

Thus M_i $\mathrm{BC}_{\text{threshold}}$-identifies f, completing the proof of Theorem 44. □

5.2. EX Frequency Identification

In this section we 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 $g_1,g_2,g_3,\ldots$.

Let $F_k(g_i) = |\{j : 1 \leq j \leq k \text{ and } g_i = g_j\}| \,/\, k$

Definition 45: M EX-identifies f with frequency p iff there exists a guess g_i such that $\underline{\lim}_k F_k(g_i) \geq p$, and $\phi_{g_i} = f$.
M EX-identifies U with frequency p iff for all $f \in U$, M EX-identifies f with frequency p.
$\mathrm{EX}_{\text{freq}}(p) = \{U \mid \text{there exists an IIM } M \text{ which identifies } U \text{ with frequency } p\}$.

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 $p_1 \le p_2$, then $EX_{freq}(p_1) \supseteq EX_{freq}(p_2)$. We now show that

Theorem 46:

1. For all integers $n \ge 1$, $EX_{team}(n) \subseteq EX_{freq}(1/n)$.
2. For all integers $n \ge 1$, for all $p \in \mathbf{R}$, if $1/(n+1) < p \le 1$ then $EX_{freq}(p) \subseteq EX_{team}(n)$.

Theorem 46 asserts that the relationship between frequency EX-identification and team EX-identification is the same as the relationship 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 43. If $U \in EX_{team}(n)$, and is EX-identified by the team $\{M_1, M_2, ..., M_n\}$, then we construct M which $EX_{freq}(1/n)$-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 43, that M EX-identifies f with frequency $1/n$.

We prove the second part of the theorem. Let $U \in EX_{freq}(p)$, with $p > 1/(n+1)$. Let M be an IIM which EX-identifies U with frequency p. To show $U \in EX_{team}(n)$, we construct a team $M_1, M_2, ..., M_n$ 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 $1/(n+1)$. 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 $1/(n+1)$ 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 $FREQ_k = \{g_i \mid F_k(g_i) > 1/(n+1)\}$. $FREQ_k$ 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.* $> k/(n+1)$ times). Clearly $|FREQ_k| \le n$, and we note that since f is EX-identified by M with frequency $p > 1/(n+1)$, there must exist a guess $g \in GOOD_f$ such that $g \in FREQ_k$ *a.e.* (k).

For each k, we define the function W_k which tells us for each $i \in FREQ_k$ where in the sequence i first occurred as a guess with $F_x(i) > 1/(n+1)$, and $F_x(i)$ has been greater than that value through $x = k$.

More precisely,

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

Clearly, for each i, $\{W_k(i)\}_{k\to\infty}$ is monotone nondecreasing.

Now we describe 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 there are $< i$ elements in $FREQ_k$, then output "0".
4. Otherwise, sort[5] 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 for each $f \in U$, there exists an i such that M_i EX-identifies f. If $f \in U$, then there exists a $g \in GOOD_f$ such that $g \in FREQ_k$ *a.e.* (k). We argue that g eventually occupies the same position in the ordered sets $FREQ_k$.

Since $g \in FREQ_k$ *a.e.* (k), there must be a number k_0 such that $W_k(g) \le k_0$ for all $k \ge k_0$, 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 $1 \le pos(k) \le n$ for all $k \ge k_0$. 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 $k \ge k_0$, $pos(k) = j$, 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 $W_{k+1}(h) = k+1 > k_0 \ge W_k(g) = W_{k+1}(g)$, 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, $1 \le j \le n$, and for all sufficiently large k, g will occupy the j^{th} position in the ordered set $FREQ_k$. It follows that M_j will converge to

[5] A simple argument shows that there can be no ties in this ordering, but this is unnecessary for the proof to follow, since ties can be broken by ordering on the actual value of the guess.

outputting "g" as a guess. Hence M_j EX-identifies f, which completes the proof of Theorem 46. □

5.3. Relationship Between Frequency, Probability, and Team Hierarchies

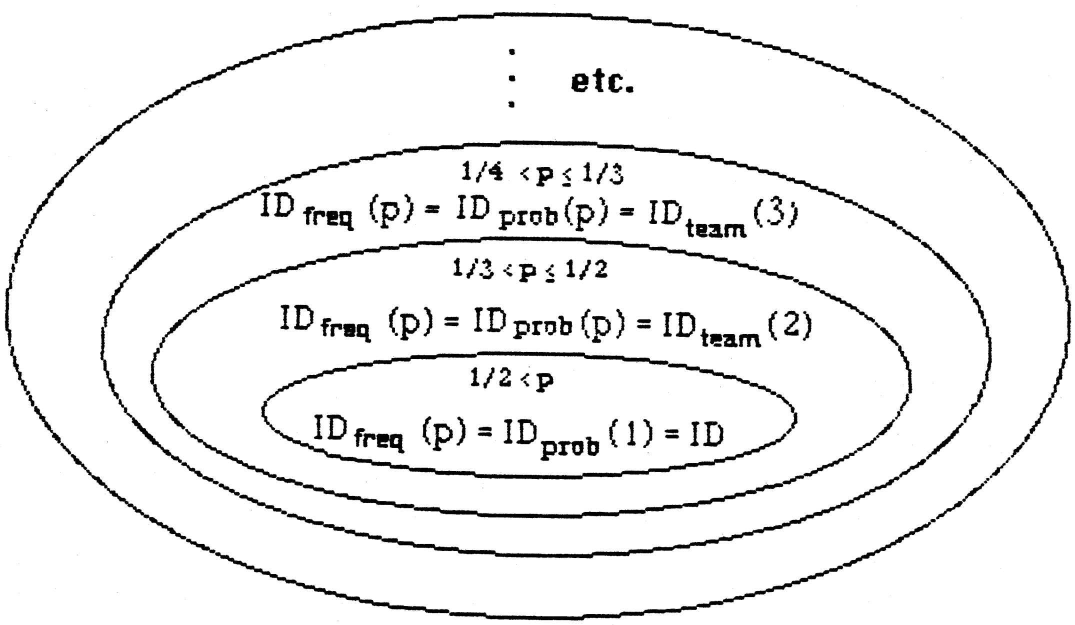
Theorems 43, 44, and 46, relating frequency identification to team identification for both the EX and BC criteria are easily assembled to show that the frequency hierarchies are identical to the team hierarchies (which are identical to the probabilistic hierarchies). In particular, Theorems 40, 41, and the corollary in section 4 are all true if "frequency" is substituted for "probability".

We conclude that if ID represents both of the symbols EX and BC, then

Theorem 47: For all integers $n \geq 1$, if $p \in \mathbf{R}$ and $1/(n+1) < p \leq 1/n$, then

$ID_{freq}(p) = ID_{prob}(p) = ID_{team}(n)$.

We illustrate this with the following diagram:



6. 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:

Definition 48: Let M be a deterministic IIM, and k an integer ≥ 1. Then M EX^k-identifies f iff when fed the graph of f in any order, M converges to outputting the program index i, and $\phi_i =^k f$. ("$=^k$" is defined at the end of section 1.2.)

$EX^k = \{U \mid \exists\, M$ such that for every $f \in U$, $M\, EX^k$-identifies $f\}$.

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

We might further allow any finite number of anomalies:

Definition 49: M EX^*-identifies f iff when fed the graph of f in any order, M converges to outputting the program index i, and $\phi_i =^* f$.

$EX^* = \{U \mid \exists\, M$ such that for every $f \in U$, $M\, EX^*$-identifies $f\}$.

The definition of EX^* is the same as that of *a.e. identification* introduced in [4], and *sub-identification* in [16].

Case and Smith [5] prove that

Theorem 50: For all $k \in N$, $EX^{k+1} - EX^k \neq \emptyset$, and $EX^* - \bigcup_{k \in N} EX^k \neq \emptyset$.

Smith [20] defines team inference with anomalies in the natural way; for $a \in \mathbf{N} \cup \{*\}$

$EX^a_{team}(n) = \{U \mid \exists\, M_1, M_2, \ldots, M_n$ such that for every $f \in U$, there is an M_i which EX^a-identifies $f\}$.

Smith shows that for all $a \in \mathbf{N} \cup \{*\}$, and for all integers $n \geq 1$,

$EX^a_{team}(n) \subset EX^a_{team}(n+1)$.

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 paper; the reader is encouraged to consult [20] for further details.

Let M be a probabilistic IIM, and let $a \in \mathbf{N} \cup \{*\}$. Then

M EX^a-identifies f with probability p iff $\Pr[\{$paths in $T_{M,f}$ which correspond to a single deterministic EX^a-identification of $f\}] \geq p$.

$\mathrm{EX}^a_{\mathrm{prob}}(p) = \{U \mid \exists\ M$ such that for all $f \in U$, M EX^a-identifies f with probability $p\}$.

Similarly, we define EX-frequency identification with anomalies:

If M is a deterministic IIM, and on input f, M outputs the sequence of guesses $g_1, g_2, g_3, \ldots$, then for all $a \in \mathbf{N} \cup \{*\}$,

M EX^a-identifies f with frequency p iff there exists a guess g_i such that $\underline{\lim}_k\ F_k(g_i) \geq p$, and $\phi_{g_i} =^a f$. (Recall $F_k(g_i)$ is the fraction of M's guesses among the first k which are "g_i".)

$\mathrm{EX}^a_{\mathrm{freq}}(p) = \{U \mid \exists\ M$ such that for all $f \in U$, M EX^a-identifies f with frequency $p\}$.

We now state

Theorem 51: For all $a \in \mathbf{N} \cup \{*\}$, for all integers $n \geq 1$, and for all $p \in \mathbf{R}$, if $1/(n+1) < p \leq 1/n$, then $\mathrm{EX}^a_{\mathrm{prob}}(p) = \mathrm{EX}^a_{\mathrm{freq}}(p) = \mathrm{EX}^a_{\mathrm{team}}(n)$.

Before we prove Theorem 51, we give the generalizations of the definition of the class OEX, and Lemma 38 which appear in [5].

For $a \in \mathbf{N} \cup \{*\}$,

M OEX^a-identifies f (written $f \in \mathrm{OEX}^a(M)$), 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 L such that $I_k = L$ *a.e.* (k), and for some $i \in L$, $\phi_i =^a f$.

$\mathrm{OEX}^a = \{U \mid \exists\ M$ such that $U \subseteq \mathrm{OEX}^a(M)\}$

Case and Smith [5] prove the following

Lemma 52:

1. For all $k \in \mathbf{N}$, $\mathrm{OEX}^k = \mathrm{EX}^k$.
2. $\mathrm{OEX}^* - \mathrm{EX}^* \neq \emptyset$.

The proof of part 1 of Lemma 52 is similar in spirit to the proof of Lemma 38.

To prove Theorem 51, we first note that the proof of Theorem 46 does not involve any simulation of the hypothesized programs, hence by simply inserting "EX^a" for "EX", "$\mathrm{EX}^a_{\mathrm{freq}}$" for "$\mathrm{EX}_{\mathrm{freq}}$", and "$\mathrm{EX}^a_{\mathrm{team}}$" for "$\mathrm{EX}_{\mathrm{team}}$", we have proved that if the hypothesis of Theorem 51 holds, then $\mathrm{EX}^a_{\mathrm{freq}}(p) = \mathrm{EX}^a_{\mathrm{team}}(n)$.

Now we show that $\mathrm{EX}^a_{\mathrm{prob}}(p) = \mathrm{EX}^a_{\mathrm{team}}(n)$ for $1/(n+1) < p \leq 1/n$ and $a \in \mathbf{N} \cup \{*\}$.

Consider the case that $a \in \mathbf{N}$. Then by Lemma 52, the analogues of Theorems 35, 36, and Lemma 39 with a anomalies all hold, and we have that $\mathrm{EX}^a_{\mathrm{prob}}(p) = \mathrm{EX}^a_{\mathrm{team}}(n)$ for $1/(n+1) < p \leq 1/n$.

These proofs do not work however, to show the corresponding result for any finite number of anomalies, since by part 2 of Lemma 52, 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 53:

1. For all integers $n \geq 1$, $\mathrm{EX}^*_{\mathrm{team}}(n) \subseteq \mathrm{EX}^*_{\mathrm{prob}}(1/n)$.
2. For all integers $n \geq 1$, for all $p \in \mathbf{R}$, if $1/(n+1) < p \leq 1$ then $\mathrm{EX}^*_{\mathrm{prob}}(p) \subseteq \mathrm{EX}^*_{\mathrm{team}}(n)$.

The proof of part 1 is identical to that of the proof of Theorem 35. We prove part 2.

Let $U \in \mathrm{EX}^*_{\mathrm{prob}}(p)$, and let M EX^*-identify U with probability $p > 1/(n+1)$. 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 M and constructs T_k, the finite subtree of $T_{M,f}$ 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 M, 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.

Description of phase k of M_i

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

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

   I_k is simply the indices of the smallest initial set of nodes, ordered by Q, which have total estimated probability $\geq i/(n+1)$.
5. Output the index of the program p_k, which on input x, dovetails the computations $\{\phi_j(x) \mid j \in I_k\}$, and outputs the first value computed.

We must show that for all $f \in U$, there is some i such that M_i EX^*-identifies f. We define

$GOOD^*_f = \{i \mid \phi_i =^* f\}$,

$WRONG^*_f = \{i \mid \{x : \phi_i(x)$ converges $\neq f(x)\}$ is infinite$\}$,

$SLOW^*_f = \mathbf{N} - (GOOD^*_f \cup WRONG^*_f)$.

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 $GS = GOOD^*_f \cup SLOW^*_f$. Then $\Pr[C(GS)]$ is in some half open interval $(j/(n+1), (j+1)/(n+1)]$ for some j, $1 \leq j \leq n$. Suppose that it falls in the interval $(i/(n+1), (i+1)/(n+1)]$. We show that M_i identifies f, proving Lemma 53.

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 $I \cap GOOD^*_f \neq \emptyset$. If this is the case, then M_i converges to outputting the index of a fixed program p which on input x, dovetails the indices of I on input x. Furthermore, $p =^* f$, 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 $\Pr[C(GS)] > i/(n+1)$ by assumption on i, by Lemma 33, there must be a *finite* collection of nodes V such that

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

Note that if $j \in GS$, then there are only finitely many arguments for which ϕ_j converges $\neq f$. 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 32,

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

and we have that $c_k \leq v$ *a.e.* (k).

Finally, by an argument similar to that in the proof of Lemma 39, the sequence $\{c_k\}$ converges to s, where s is the smallest value $1 \leq s \leq v$ such that

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

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 $I \cap GOOD^*_f = \emptyset$. Then

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

Also, since $\Pr[C(GOOD_f^*)] > 1/(n+1)$, it follows that $\Pr[C(GS)] > (i+1)/(n+1)$, which contradicts our assumption on i. Thus M_i converges to outputting the index of $p =^* f$, and Lemma 53 and Theorem 51 follow. □

Identification with anomalous hypotheses has also been studied for BC-identification.

Definition 54: Let M be a deterministic IIM, and $a \in \mathbf{N} \cup \{*\}$.

M BC^a-identifies f iff when fed the graph of f in any order, M outputs the infinite sequence $g_1, g_2, g_3, \ldots$, and $g_k =^a f$ *a.e.* (k).

BC^a = $\{U \mid \exists\ M$ such that for every $f \in U$, M BC^a-identifies $f\}$.

Case and Smith [5] show that $BC^{k+1} - BC^k \neq \emptyset$, and $BC^* - \bigcup_{k \in N} BC^k \neq \emptyset$.

L. Harrington has shown that BC* contains the class of partial recursive functions; this result also appears in [5]. In [20], Smith also gives definitions for BC-identification with anomalies by teams. It is shown that there is a proper hierarchy for all integers $k \geq 1$:

$$BC^k_{team}(n) \subset BC^k_{team}(n+1).$$

Daley [7] proves interesting tradeoffs, analogous to those shown for EX in [20], relating number of team members, number of anomalies, and number of mind changes required for BC-identification.

Now let M be a probabilistic IIM, and let $k \in \mathbf{N}$.

Definition 55: M BC^k-identifies f with probability p iff Pr[{paths in $T_{M,f}$ which correspond to a single deterministic BC^k-identification of f}] $\geq p$.

$BC^k_{prob}(p)$ = $\{U \mid \exists\ M$ such that for all $f \in U$, M BC^k-identifies f with probability $p\}$.

We similarly define BC-frequency identification with anomalies: 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 56: $BC^k_{freq}(p)$ = $\{U \mid \exists\ M$ such that for all $f \in U$, M BC^k-identifies f with frequency $p\}$.

It seems appropriate to form the following

Conjecture: For all $k \in \mathbf{N}$, for all integers $n \geq 1$, and for all $p \in \mathbf{R}$:
If $1/(n+1) < p \leq 1/n$, then $BC^k_{prob}(p) = BC^k_{freq}(p) = BC^k_{team}(n)$.

7. Nondeterministic Inference Strategies: An Observation

The notion of "team inference" which we have been using, at first seems to be an unnatural model of computation. After a moment of reflection however, we realize that a "team of n machines" may be thought of as a single nondeterministic machine, which is restricted to choosing from among n deterministic strategies. In this section we consider unrestricted nondeterministic IIMs, and give a simple argument showing why this model is too powerful to be interesting. We then consider a different type of restriction, that of "reliability" [4] , and show that reliable nondeterministic IIMs are no more powerful than deterministic IIMs.

A nondeterministic IIM (NIIM) is simply a deterministic IIM with a 0-1 oracle. The NIIM may query the oracle for a "nondeterministic bit" which it then receives on a special tape. Thus there is essentially no difference between a probabilistic IIM and an NIIM, except that in the latter case, the oracle isn't a coin, and there is no associated notion of probability. For a particular NIIM N, and a function f as input, there is a corresponding computation tree $T_{N,f}$, defined as for probabilistic IIMs.

The NIIM N EX- (BC-) identifies the function f iff there exists a path in $T_{N,f}$ which corresponds to a single deterministic EX- (BC-) identification of f. It is immediately clear that there is a single NIIM N which EX- (and hence BC-) identifies $\mathbf{T}$, 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 $f \in \mathbf{T}$ (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***.[6] An IIM M is ***reliable*** iff for all f, if M converges to some guess on input f, then the guess is a program index for f. Then M reliably EX-identifies U if M EX-identifies U, and M is reliable. Thus we may assume that whenever M converges, its answer is correct. Reliable inference strategies have been studied in [4], [16].

We consider the following question about Reliable Nondeterministic IIMs (RNIIMs): Are RNIIMs too powerful, as are NIIMs, or does the reliability restriction prohibit the type of unlimited guessing that allowed a single NIIM to identify all $f \in \mathbf{T}$? We are surprised to find

[6]Reliability is not a meaningful notion for BC-identification.

that

> **Theorem 57:** If U is EX-identifiable by a RNIIM, then U is EX-identifiable by a deterministic IIM.

Hence the class of RNIIM identifiable subsets of $\mathbf{T}$ is exactly the class EX, showing that reliability is too strong a restriction for NIIMs to yield interesting identifiability classes.

Proof: Let N be a RNIIM which EX-identifies the set U of functions. We construct a deterministic IIM M which EX-identifies U: M, given values from the graph of f, constructs $T_{N,f}$ level by level, making 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 34.) M then outputs the index of the least numbered node n such that $C_{n,k} \neq \emptyset$.

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 EX-identifies U, for every function $f \in U$, there is at least one path 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 for all $k > d(s)$, 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 less than 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 $n < s$, with $C_{n,k} \neq \emptyset$ for all $k \geq d(n)$. 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 $C_{n,k} \neq \emptyset$ for all $k \geq d(n)$. 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 $C_{n,d(m)} \cap P_m \neq \emptyset$. 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 for all $k \geq d(n)$, $C_{n,k} \neq \emptyset$. 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 $T_{M,f}$ induced by the red nodes is a tree. Since $T_{M,f}$ 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. Konig's Lemma asserts that there must be an infinite path in this red tree. But an infinite red path in $T_{M,f}$ corresponds to a path which is in C_n, hence C_n is not empty. Therefore, it cannot be the case that $C_{n,k} \neq \emptyset$ for all $k \geq d(n)$. This completes the proof of Theorem 57. □

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.

8. Other Properties of Probabilistic IIMs.

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

8.1. 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 $\mathbf{T}$ 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: $\mathbf{T} \rightarrow [0,1]$ be a probability distribution which assigns to every total recursive function f, a real number in $[0,1]$ such that $\sum_{f \in \mathbf{T}} D(f) = 1$.

Let M_{pr} be a probabilistic IIM, and M a deterministic IIM. Define $M(f)$ to be 1 if M identifies f, 0 otherwise, and $M_{pr}(f)$ to be the probability that M_{pr} identifies f.

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

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

and the average performance of M_{pr} is

$$A(M_{pr},D) = \sum_{f \in \mathbf{T}} D(f) \cdot M_{pr}(f).$$

Theorem 58: For all distributions D on $\mathbf{T}$, for all $\epsilon > 0$, and for all probabilistic IIMs

M_{pr}, there exists a deterministic IIM M such that $A(M,D) > A(M_{pr},D) - \epsilon$.

In other words, there are deterministic machines which have average performance arbitrarily close to that of any particular probabilistic IIM.

Proof: Since $\sum_{f \in \mathbf{T}} D(f) = 1$, there exists a finite number of distinct functions $\{f_1,f_2,...,f_k\}$ such that $\sum_{i=1}^{k} D(f_i) > 1 - \epsilon$. 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 $\{f_1,f_2,...,f_k\}$ 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 $\{f_1,f_2,...,f_k\}$, and it follows that $A(M,D) > 1 - \epsilon$. Since $A(M_{pr},D)$ is at most 1, the theorem follows. □

If the reader feels cheated by the above theorem, it is for a good reason: The proof relies solely on the countability of the function space over which the distribution D is defined, rather than reflecting any deep property of probabilistic computation. Since the class of all partial recursive functions is countable, it might be the case that the concept of average performance will not yield much insight into probabilistic inference.

While Theorem 58 states that for any probabilistic IIM M_{pr}, there are deterministic IIMs with average performance "within ϵ", it doesn't address the question of whether there are deterministic IIMs with $A(M,D) \geq A(M_{pr},D)$. We do not know the answer to this question, but the following theorem demonstrates that for certain types of probabilistic IIMs, the answer is "yes".

Theorem 59: Let D be a probability distribution on $\mathbf{T}$. Let M_{pr} be a probabilistic IIM which behaves as follows: M_{pr} randomly chooses to simulate one of the deterministic strategies $M_1,M_2,...,M_k$ with probabilities $p_1,p_2,...,p_k$, respectively. ($\sum_{i=1}^{k} p_i = 1$). Then there is a deterministic IIM M such that $A(M,D) \geq A(M_{pr},D)$.

Proof: Let $M_i(f) = 1$ if M_i identifies f, 0 otherwise.

We let w_i denote the "weight" (with respect to D) of the set of functions that M_i identifies:

$$w_i = \sum_{f \in \mathbf{T}} D(f) \cdot M_i(f).$$

Then we have

$$A(M_{pr},D) = \sum_{i=1}^{k} p_i \cdot w_i.$$

Now since $\sum_{i=1}^{k} p_i = 1$, then $max\ \{w_i \mid 1 \leq i \leq k\} \geq \sum_{i=1}^{k} p_i \cdot w_i$.

Let w_j be $max\ \{w_i\}$. Then M_j has average performance

$$A(M_j,D) = \sum_{f \in T} D(f) \cdot M_j(f) = w_j \geq \sum_{i=1}^{k} \pi \cdot w_i = A(M_{pr},D). \quad \square$$

Hence if we are to exhibit a probabilistic IIM which has average performance strictly better than any deterministic IIM, it must not be one which chooses probabilistically from among a finite number of deterministic strategies.

8.2. Restricted Choice Probabilistic IIMs

The discussion in the previous section motivates the following question: Is there a difference between probabilistic IIMs which choose randomly to simulate one of a finite collection of deterministic strategies, 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 M is any probabilistic IIM which identifies the set U of functions with probability p, then consider the least positive integer n such that $1/(n+1) < p \leq 1/n$. Then by part 2 of Theorem 40 there is a team of n deterministic IIMs identifying U, and by part 1 of Theorem 40 there is a restricted choice probabilistic IIM which identifies the set U with probability $1/n \geq p$.

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

Theorem 60: There exists an unrestricted choice probabilistic IIM M_u such that for any restricted choice probabilistic IIM M_r, there exists a total recursive function f such that Pr[M_r identifies f] < Pr[M_u identifies f].

Proof: M_u uses coin flips in such a way that M_u guesses the index "0" with probability 1/2, "1" with probability 1/4, ... , "n" with probability $1/2^{n+1}$. Thus the probability that M_u identifies any given total recursive function is greater than 0. Suppose there was a restricted choice probabilistic IIM M_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[M_r identifies f] $\geq$ Pr[M_u identifies f], it must be the case that for all f, Pr[M_r identifies f] > 0. It follows that $\mathbf{T} \subseteq \bigcup_{i=1}^{k} \mathrm{ID}(M_i)$ violating the team hierarchy theorem (Theorem 7). $\square$.

Acknowledgements

I am extremely grateful to Dana Angluin for suggesting the problem area, for supplying much encouragement, and for a countless number of very fruitful discussions, without which this paper would not have been possible. I also thank Bob Daley for pointing out Podnieks' frequency results; and Carl Smith, for reading an early abstract, and for bringing the excellent paper [5] to my attention.

APPENDIX

Elementary Probability Theory

We outline some of the key steps involved in defining a probability measure on an infinite set of possible outcomes, Ω. For a more detailed discussion of some of these issues, we refer the reader to [17], from which much of this section was taken.

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 61: A set of events Ω, together with a family of subsets $\mathcal{B}$ of Ω, is a Borel field iff the following three conditions hold:

1. $\Omega \in \mathcal{B}$.
2. If $A \in \mathcal{B}$, then $\Omega - A \in \mathcal{B}$.
3. If $\{A_i\}_{i \in I}$ is a finite or countable collection of elements of $\mathcal{B}$, then $\bigcup_{i \in I} A_i \in \mathcal{B}$.

It follows that any Borel field is closed under complementation, and countable unions and intersections. The elements of $\mathcal{B}$ are commonly called the Borel *sets* of Ω.

Before continuing our discussion of Borel fields, we review some set theory.

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

$$\overline{\lim}_k A_k = \bigcap_{k=0}^{\infty} \bigcup_{i=k}^{\infty} A_i$$

$\underline{\lim}_k A_k = \bigcup_{k=0}^{\infty} \bigcap_{i=k}^{\infty} A_i$

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 we call this the limit of the sequence, *i.e.*

$\lim_{k\to\infty} A_k = \underline{\lim}_k A_k = \overline{\lim}_k A_k$

A sequence of sets is *monotone*, if for all k, $A_k \subseteq A_{k+1}$, or for all k, $A_{k+1} \subseteq A_k$. Every monotone sequence of sets has a limit, and every Borel field is closed under $\underline{\lim}_k$ and $\overline{\lim}_k$.

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

For example, consider the the real line $\mathbf{R}$, and let $\mathcal{I}$ be the family of all of the open intervals of the form $(-\infty,w)$, for $w \in \mathbf{R}$. 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 $\mathbf{R}$ 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 62: If Ω and $\mathcal{B}$ together are a Borel field, then the function $\Pr:\mathcal{B} \to \mathbf{R}$ is a *probability measure* iff

1. $\Pr[\Omega] = 1$
2. For all $A \in \mathcal{B}$, $\Pr[A] \geq 0$
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 $\Pr[A] \leq \Pr[B]$. Also, if $\{A_i\}$ is a sequence of Borel sets for which the limit is defined, then

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

Now if we wish to define a probability measure Pr for some application, then we do the following: We begin with some collection of *basic sets* C, for which we want Pr to be defined. (For the real numbers, the basic sets were the half infinite intervals; for $\Omega = [0,1]$, we might want C to be all intervals of the form (a,b), $0 \leq a \leq b \leq 1$). Next we construct $\mathcal{A}(C)$ = the smallest field of subsets of Ω which contains $\mathcal{C}$, (*i.e.* the smallest family of subsets of Ω which are closed under *finitely* many applications of intersection, union, and complementation), and we extend Pr to $\mathcal{A}(\mathcal{C})$. If this can be done so that Pr satisfies the three axioms for a probability measure (given above) on $\mathcal{A}(\mathcal{C})$, then Caratheodory's Extension Theorem [17] guarantees that Pr extends uniquely to $\mathcal{B}(\mathcal{C})$ = the smallest Borel field containing $\mathcal{C}$, and the extension is a probability measure.

Once we have $\mathcal{B}(\mathcal{C})$ to which Pr extends uniquely, then given $A \subseteq \Omega$, to find Pr[A], we need only show that A is "measurable" by showing it is in $\mathcal{B}(\mathcal{C})$, and then we can find Pr[A] by expressing A as a countable union and intersection of basic sets and their complements, and applying properties of the probability measure. We can then be sure that regardless of how A was expressed in terms of the basic sets, the value Pr[A] computed is correct.

Proof of Lemma 11

Let P denote the class of sets $\{P_n\}_{n \in T_{M,f}}$, where P_n is the set of paths passing through node n. We first show that Pr satisfies the probability axioms on P.

Clearly, $\Omega = P_1$ = all paths in $T_{M,f}$, and $\Pr[\Omega] = \Pr[P_1] = 2^{-d(1)} = 2^0 = 1$. Also by definition of Pr, $\Pr[P_n] \geq 0$ for all $n \geq 1$. We show that for all $I \subseteq \mathbf{N}$,

$$\text{if } P_n = \biguplus_{i \in I} P_i, \text{ then } \Pr[P_n] = \sum_{i \in I} \Pr[P_i].$$

First we note that if $P_n = \biguplus_{i \in I} P_i$, then I is finite: Suppose not, then we construct a path in P_n, but not in P_i for any $i \in I$. Clearly every $i \in I$ is a descendant of n. Color red every node x such that x is a descendant of n, and the partial path from n to x does not contain any $i \in I$. There is a red node at every level of $T_{M,f}$, otherwise we would have a *finite* subcollection $S \subset I$ such that $P_n = \biguplus_{i \in S} P_i$. Therefore there are infinitely many red nodes, and the subgraph induced by red nodes is a connected tree with root n. By Konig's Lemma, it contains an infinite path $p \in P_n - \biguplus_{i \in I} P_i$.

Hence we only need show that for all *finite* sets $I \subseteq \mathbf{N}$,

$$\text{if } P_n = \biguplus_{i \in I} P_i, \text{ then } \Pr[P_n] = \sum_{i \in I} \Pr[P_i].$$

To see this, let l be the deepest level which contains some $i \in I$. Note that if a node x has depth $< l$, then it has $2^{l-d(x)}$ descendants at level l. Since $P_n = \biguplus_{i \in I} P_i$, then the number of descendants of n at level l is exactly the sum of the descendants at level l of each $i \in I$. We have

$$2^{l-d(n)} = \sum_{i \in I} 2^{l-d(i)} \quad \text{which implies that}$$

$$2^{-d(n)} = \sum_{i \in I} 2^{-d(i)}, \quad \text{or}$$

$$\Pr[P_n] = \sum_{i \in I} \Pr[P_i].$$

We've thus shown that Pr satisfies the axioms of a probability measure on the class of basic sets P. Now let $\mathcal{A}(P)$ be the smallest family of subsets of Ω containing P which is closed under finitely many applications of intersection, union, and complement. As outlined in the first part of this appendix, if we show that Pr satisfies the probability axioms on $\mathcal{A}(P)$, then the Extension Theorem of Caratheodory asserts that Pr is well defined on $\mathcal{B}(P)$, and Lemma 11 is proved.

It is clear that Pr satisfies the axioms on $\mathcal{A}(P)$ from the following

Claim: $\mathcal{A}(P) = \{A \mid A$ is the finite disjoint union of elements of $P\}$.

We sketch an inductive proof of the claim. Certainly each element of P is a finite disjoint union of elements of P. Any element $A \in \mathcal{A}(P)$ is obtained by finitely many applications of the operations intersection, union, and complement. If the last operation is union, then $A = B \cup C$, where by induction, B and C are finite disjoint unions of elements of P. Let $P_{BC} \subseteq P$ denote the finite collection of sets $\{P_n\}$ whose union is $B \cup C$.

If the elements of B are not mutually disjoint from the elements of C, then we can rewrite B and C as disjoint unions of sets $\{P_i\}$ of P, each i with depth $d(i) = m$, where $m = max\ \{d(n) \mid P_n \in P_{BC}\}$. Thus A is a finite disjoint union of elements of P.

If the last operation in the construction of A is a complement, then $A = B^c$, where B is a finite disjoint union of elements of P by induction. Then for some l, we can rewrite B as a finite disjoint union of elements of P which are all at the same level l of $T_{M,f}$, and A is simply the finite disjoint union of all other elements at level l.

Finally, if $A = B \cap C$, then by DeMorgan's Law, $A = (B^c \cup C^c)^c$, and by induction on the previous arguments for union and complement, A is the finite disjoint union of elements of P. □

References

[1] D. Angluin and C. Smith.
Inductive inference: theory and methods.
Computing Surveys 15(3):237-269, September, 1983.

[2] J. M. Barzdin.
Two theorems on the limiting synthesis of functions.
Latvii gosudarst. Univ. Ucenye Zapiski 210:82-88, 1974.
(in Russian).

[3] J. M. Barzdin and R. V. Freivald.
On the prediction of general recursive functions.
Soviet Math. Dokl. 13:1224-1228, 1972.

[4] L. Blum and M. Blum.
Toward a mathematical theory of inductive inference.
Inform. Contr. 28:125-155, 1975.

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

[6] C. M. Cook, A. Rosenfeld, and A. R. Aronson.
Grammatical inference by hill-climbing.
Information Sciences 10:59-80, 1976.

[7] R. Daley.
On the error correcting power of pluralism in BC-type inductive inference.
Theoretical Computer Science 24:95-104, 1983.

[8] J. A. Feldman.
Some decidability results in grammatical inference and complexity.
Inform. Contr. 20:244-262, 1972.

[9] 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. Akademie-Verlag, 1979.

[10] B. R. Gaines.
Maryanski's grammatical inferencer.
IEEE Trans. on Computers C-28:62-64, 1978.

[11] E. M. Gold.
Language identification in the limit.
Inform. Contr. 10:447-474, 1967.

[12] J. E. Hopcroft and J. D. Ullman.
Introduction to Automata Theory, Languages, and Computation.
Addison-Wesley, Reading, MA, 1979.

[13] J. J. Horning.
A study of grammatical inference.
PhD thesis, Stanford University, Computer Science Dept., 1969.

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

[15] F. J. Maryanski and T. L. Booth.
Inference of finite state probabilistic grammars.
IEEE Trans. on Computers C-26:521-536, 1977.

[16] E. Minicozzi.
Some natural properties of strong identification in inductive inference.
Theoretical Computer Science 2:345-360, 1976.

[17] M. F. Neuts.
Probability.
Allyn and Bacon, Boston, MA, 1973.

[18] K. M. Podnieks.
Comparing various concepts of function prediction.
Latvii gosudarst. Univ. Ucenye Zapiski 210:68-81, 1974.
(in Russian).

[19] K. M. Podnieks.
Probabilistic synthesis of enumerated classes of functions.
Soviet Math. Dokl. 16:1042-1045, 1975.

[20] C. Smith.
The power of pluralism for automatic program synthesis.
J. ACM 29(4):1144-1165, 1982.

[21] 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.

[22] A. Van der Mude and A. Walker.
On the inference of stochastic regular grammars.
Inform. and Contr. 38:310-329, 1978.

[23] R. Wiehagen, R. Freivald, and E.B. Kinber.
On the power of probabilistic strategies in inductive inference.
Theoretical Computer Science 28:111-133, 1984.