

On Identification by Teams and Probabilistic Machines

Sanjay Jain

Institute of Systems Science
National University of Singapore
Singapore 0511, Republic of Singapore
Email: sanjay@iss.nus.sg

Arun Sharma

School of Computer Science and Engineering
The University of New South Wales
Sydney, NSW 2052, Australia
Email: arun@cse.unsw.edu.au

March 11, 2007

1 Introduction

Inductive inference in the scientific domain is seldom an individual enterprise. Many a scientific breakthrough are result of the efforts of several scientists investigating a problem; scientific success is achieved if any one or more of members of the scientific community are successful. This observation about the practice of science can be partially incorporated in a model of computational learning that employs a ‘team’ of algorithmic machines instead of a single algorithmic machine. The team is said to be successful just in case one or more members in the team are successful.

Another variation on the notion of an algorithmic learning machine is obtained by considering devices that in addition to being algorithmic are also capable of basing their actions on the outcomes of random events. Such learning agents can be modeled using probabilistic Turing machines.

The present paper surveys the work on both team and probabilistic learning. The notion of team learning in the context of identification in the limit of functions was first investigated by Smith [36]. The study of probabilistic learning machines was initiated by Freivalds [15] and Pitt [32]. Freivalds’ study is in the context of finite identification of functions and Pitt’s study is for identification in the limit of functions.

Like most investigations in Learning Theory, the work on team and probabilistic learning can be classified under two dimensions:

- *Concepts being learned*: computable functions and recursively enumerable languages.
- *Criteria of Success*: learning in the limit, learning with bounded number of mind changes and finite learning, vacillatory learning, and behaviorally correct learning.

Most progress has been reported on team and probabilistic learning in the limit of computable functions. Partial results have been reported on identification in the limit of r.e. languages by teams and probabilistic machines and on finite identification of computable functions by teams and probabilistic machines. Only preliminary results are known for other cases. Given this state of

affairs, we begin with a extensive treatment of identification in the limit by teams and identification in the limit by probabilistic machines of computable functions. After showing that these two notions turn out to be equivalent, we turn our attention to language identification in the limit by both teams and probabilistic machines. This is followed by a discussion of results about team finite identification.¹ Finally, we present pointers to other investigations about team and probabilistic learners.

In what follows, Section 2 introduces the preliminaries, Section 3 describes team identification of functions, Section 4 introduces probabilistic identification of functions, Section 5 is devoted to both team and probabilistic identification of languages, Section 6 discusses results about finite identification of functions by teams, and finally in Section 7 we present pointers to the literature for additional results.

2 Preliminaries

In this section, we introduce our notation and describe the fundamental learning paradigms.

2.1 Notation

Any unexplained recursion theoretic notation is from [35]. The symbol N denotes the set of natural numbers, $\{0, 1, 2, 3, \dots\}$. The symbol N^+ denotes the set of positive natural numbers, $\{1, 2, 3, \dots\}$. Unless otherwise specified, $i, j, k, l, m, n, q, r, s, t, x, y$, with or without decorations², range over N . Symbols $\emptyset, \subseteq, \subset, \supseteq, \supset$ denote empty set, subset, proper subset, superset, and proper superset, respectively. Symbols A and S , with or without decorations, range over sets. P, Q , and X , with or without decorations, range over finite sets. D_0, D_1, \dots , denotes a canonical (recursive) indexing of finite sets [35]. Cardinality of a set S is denoted by $\text{card}(S)$. We say that $\text{card}(A) \leq *$ to mean that $\text{card}(A)$ is finite. Intuitively, the symbol, $*$, denotes ‘finite without any prespecified bound.’ The letters a and b , with or without decorations, range over $N \cup \{*\}$. The maximum and minimum of a set are denoted by $\max(\cdot), \min(\cdot)$, respectively, where $\max(\emptyset) = 0$ and $\min(\emptyset) = \uparrow$. We also order pairs in the following manner. We say that a pair (i, j) is $< (k, l)$ just in case $[(i < k) \vee (i = k \wedge j < l)]$.

Letters f, g, h and G , with or without decorations, range over *total* functions with arguments and values from N . Symbol \mathcal{R} denotes the set of all total computable functions. \mathcal{C} and \mathcal{S} , with or without decorations, range over subsets of \mathcal{R} .

We let ψ , with or without decorations, range over partial functions. For $a \in (N \cup \{*\})$, $\psi_1 =^a \psi_2$ means that $\text{card}(\{x \mid \psi_1(x) \neq \psi_2(x)\}) \leq a$.

A pair $\langle i, j \rangle$ stands for an arbitrary, computable, one-to-one encoding of all pairs of natural numbers onto N [35]. $\pi_1(\langle x, y \rangle) = x$ and $\pi_2(\langle x, y \rangle) = y$. Similarly, we can define $\langle \cdot, \dots, \cdot \rangle$ for encoding multiple tuples of natural numbers onto N .

By φ we denote a fixed *acceptable* programming system for the partial computable functions:

¹This subject is addressed in detail by Daley and Kalyanasundaram elsewhere in this volume.

²Decorations are subscripts, superscripts and the like.

$N \rightarrow N$ [34, 35, 27]. By φ_i we denote the partial computable function computed by program i in the φ -system. The letter, p , in some contexts, with or without decorations, ranges over programs; in other contexts p ranges over total functions with its range being construed as programs. By Φ we denote an arbitrary fixed Blum complexity measure [2, 19] for the φ -system. By W_i we denote $\text{domain}(\varphi_i)$. W_i is, then, the r.e. set/language ($\subseteq N$) accepted (or equivalently, generated) by the φ -program i . Symbol \mathcal{E} will denote the set of all r.e. languages. Symbol L , with or without decorations, ranges over \mathcal{E} . Symbol \mathcal{L} , with or without decorations, ranges over subsets of \mathcal{E} . We denote by $W_{i,s}$ the set $\{x \leq s \mid \Phi_i(x) < s\}$. $L_1 \triangle L_2$ denotes $(L_1 - L_2) \cup (L_2 - L_1)$, the symmetric difference of L_1 and L_2 . For $a \in (N \cup \{*\})$, $L_1 =^a L_2$ means that $\text{card}(L_1 \triangle L_2) \leq a$.

$[i \dots j]$ denotes the set of real numbers $\geq i$ and $\leq j$.

2.2 Learning Machines

We first describe function learning machines.

We assume, without loss of generality, that the graph of a function is fed to a machine in canonical order. For $f \in \mathcal{R}$ and $n \in N$, we let $f[n]$ denote the finite initial segment $\{(x, f(x)) \mid x < n\}$. Clearly, $f[0]$ denotes the empty segment. SEG denotes the set of all finite initial segments, $\{f[n] \mid f \in \mathcal{R} \wedge n \in N\}$.

Definition 1 [18] A *function learning machine* is an algorithmic device which computes a mapping from SEG into N .

We now consider language learning machines. A *sequence* σ is a mapping from an initial segment of N into $(N \cup \{\#\})$. The *content* of a sequence σ , denoted $\text{content}(\sigma)$, is the set of natural numbers in the range of σ . The *length* of σ , denoted by $|\sigma|$, is the number of elements in σ . For $n \leq |\sigma|$, the initial sequence of σ of length n is denoted by $\sigma[n]$. Intuitively, $\#$'s represent pauses in the presentation of data. We let σ, τ , and γ , with or without decorations, range over finite sequences. SEQ denotes the set of all finite sequences. We let $\sigma \diamond x$ denote the concatenation of x at the end of σ . Thus $\tau = \sigma \diamond x$ may be defined as follows:

$$\tau(z) = \begin{cases} \sigma(z), & z < |\sigma|; \\ x, & z = |\sigma|; \\ \uparrow, & \text{otherwise.} \end{cases}$$

A text T is a mapping from N into $(N \cup \{\#\})$. The content of a text T , denoted by $\text{content}(T)$, is the set of natural numbers in the range of T . A text T is for L iff $\text{content}(T) = L$. $T[n]$ denotes the initial segment of T with length n .

Definition 2 A *language learning machine* is an algorithmic device which computes a mapping from SEQ into N .

The set of all finite initial segments, SEG, can be coded onto N . Also, the set of all finite sequences of natural numbers and $\#$'s, SEQ, can be coded onto N . Thus, in both Definitions 1 and

2, we can view these machines as taking natural numbers as input and emitting natural numbers as output. Henceforth, we will refer to both function-learning machines and language-learning machines as just learning machines, or simply as machines. We let \mathbf{M} , with or without decorations, range over learning machines.

It should be noted that for all the identification criteria surveyed in this paper, we are assuming, without loss of generality, that the learning machines are total.

2.3 Function Identification in the Limit

The next definition describes identification in the limit of functions. We also consider the case in which the final program is allowed to have anomalies.

Definition 3 [18, 1, 5] Let $a \in N \cup \{*\}$. \mathbf{M} \mathbf{Ex}^a -identifies f (read: $f \in \mathbf{Ex}(\mathbf{M})$) $\iff (\exists i \mid \varphi_i =^a f) (\forall^\infty n)[\mathbf{M}(f[n]) = i]$. We define the class $\mathbf{Ex}^a = \{\mathcal{S} \subseteq \mathcal{R} \mid (\exists \mathbf{M})[\mathcal{S} \subseteq \mathbf{Ex}^a(\mathbf{M})]\}$.

The relationship between the above criteria is summarized in the following theorem.

Theorem 1 [5, 1] $\mathbf{Ex} = \mathbf{Ex}^0 \subset \mathbf{Ex}^1 \subset \mathbf{Ex}^2 \subset \dots \subset \mathbf{Ex}^*$.

A variant on the above criterion, described next, was introduced by Case and Smith [5] and is useful in the proof of some of the results in the present chapter.

Definition 4 [5] Let $a \in N \cup \{*\}$. \mathbf{M} \mathbf{Oex}^a -identifies f (written: $f \in \mathbf{Oex}^a(\mathbf{M})$) just in case there exists a nonempty finite set D such that the following hold:

1. $(\exists i \in D)[\varphi_i =^a f]$;
2. $(\forall^\infty n)[\mathbf{M}(f[n]) \in D]$;
3. $(\forall i \in D)(\exists^\infty n)[\mathbf{M}(f[n]) = i]$.

We define the class $\mathbf{Oex}^a = \{\mathcal{S} \mid (\exists \mathbf{M})[\mathcal{S} \subseteq \mathbf{Oex}^a(\mathbf{M})]\}$.

Thus, \mathbf{M} \mathbf{Oex}^a identifies a function f just in case \mathbf{M} , fed the graph of f , vacillates among a nonempty finite set D of indexes such that there is at least one a -error index for f in the set D and each index in D is conjectured infinitely often by \mathbf{M} .

The following formulation of \mathbf{Oex}^a is equivalent to the above definition. We often employ the following variant in our applications of \mathbf{Oex}^a .

Definition 5 [5] Let $a \in N \cup \{*\}$. \mathbf{M} \mathbf{Oex}^a -identifies f (written: $f \in \mathbf{Oex}^a(\mathbf{M})$) just in case there exists an i such that

1. $(\exists j \in D_i)[\varphi_j =^a f]$;
2. $(\forall^\infty n)[\mathbf{M}(f[n]) = i]$;

The following is true about \mathbf{Oex}^a criteria.

Theorem 2 [5]

1. For $m \in N$, $\mathbf{Oex}^m = \mathbf{Ex}^m$.
2. $\mathbf{Oex}^* - \mathbf{Ex}^* \neq \emptyset$.

2.4 Language Identification in the Limit

We now introduce language identification in the limit.

Definition 6 [18, 4, 30] Let $a \in N \cup \{*\}$. \mathbf{M} \mathbf{TxtEx}^a -identifies L (read: $L \in \mathbf{TxtEx}(\mathbf{M})$) \iff $(\forall \text{ texts } T \text{ for } L) (\exists i \mid W_i =^a L) (\forall^\infty n)[\mathbf{M}(T[n]) = i]$. We define the class $\mathbf{TxtEx}^a = \{\mathcal{L} \subseteq \mathcal{E} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{TxtEx}^a(\mathbf{M})]\}$.

The relationship between the above criteria are summarized in the following theorem.

Theorem 3 [4, 30] $\mathbf{TxtEx} = \mathbf{TxtEx}^0 \subset \mathbf{TxtEx}^1 \subset \mathbf{TxtEx}^2 \subset \dots \subset \mathbf{TxtEx}^*$.

3 Limiting Identification of Functions by Teams

Consider the following well known result from Learning Theory.

Theorem 4 [1] *Let*

$$\mathcal{S}_1 = \{f \in \mathcal{R} \mid \varphi_{f(0)} = f\} \text{ and}$$

$$\mathcal{S}_2 = \{f \in \mathcal{R} \mid (\forall^\infty n)[f(n) = 0]\}.$$

Then, $\mathcal{S}_1 \in \mathbf{Ex}$, $\mathcal{S}_2 \in \mathbf{Ex}$, but $\mathcal{S}_1 \cup \mathcal{S}_2 \notin \mathbf{Ex}$.

The above result, popularly referred to as ‘non-union theorem,’ says that the class \mathbf{Ex} is not closed under union. In other words, there are collections of functions which are identifiable, but the union of these collections is not identifiable. This result may be viewed as a fundamental limitation on building a general purpose device for machine learning, and, to an extent, justifies the use of heuristic methods in Artificial Intelligence. However, this result also suggests a more general criterion of identification in which a team of scientists is employed and success of the team is the success of any member in the team. We illustrate this idea next.

Consider the collections of functions \mathcal{S}_1 and \mathcal{S}_2 in Theorem 4. Let \mathbf{M}_1 **Ex**-identify \mathcal{S}_1 and \mathbf{M}_2 **Ex**-identify \mathcal{S}_2 . Now, if we employed a team of \mathbf{M}_1 and \mathbf{M}_2 to identify $\mathcal{S}_1 \cup \mathcal{S}_2$ and weakened the criterion of success to the requirement that success is achieved just in case any one member in the team is successful, then the collection $\mathcal{S}_1 \cup \mathcal{S}_2$ becomes identifiable by the team consisting of \mathbf{M}_1 and \mathbf{M}_2 under this new criterion of success. This idea can be extended to teams of n machines out of which at least m ($m \leq n$) are required to be successful. The formal definitions for team function identification and team language identification are presented next. J. Case first suggested the notion of team function identification based on the non-union theorem of the Blums, and it was extensively investigated by C. Smith. The general case of m out of n teams is due to Osherson, Stob, and Wienstein [28].

We now formally define team identification for functions.

A team of learning machines is a multiset of learning machines.

Definition 7 [36, 28] Let $a \in N \cup \{*\}$ and let $m, n \in N^+$.

(a) Let $f \in \mathcal{R}$. A team of n machines $\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_n$ is said to **Team** $_n^m \mathbf{Ex}^a$ -identify f (written: $f \in \mathbf{Team}_n^m \mathbf{Ex}^a(\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_n)$) just in case there exist m distinct numbers i_1, i_2, \dots, i_m , $1 \leq i_1 < i_2 < \dots < i_m \leq n$, such that each of $\mathbf{M}_{i_1}, \mathbf{M}_{i_2}, \dots, \mathbf{M}_{i_m}$ **Ex** a -identifies f .

(b) **Team** $_n^m \mathbf{Ex}^a$ is defined to be the class of sets \mathcal{S} of computable functions such that some team of n machines **Team** $_n^m \mathbf{Ex}^a$ -identifies each function in \mathcal{S} .

For the criterion **Team** $_n^m \mathbf{Ex}^a$ -identification, we refer to the fraction $\frac{m}{n}$ as the *success ratio* of the criterion.

We now turn our attention to results. We first describe results about team function criteria in which success of the team requires only one member in the team to be successful, namely, **Team** $_n^1 \mathbf{Ex}$ -identification. Section 4 describes identification by probabilistic machines. These results illustrate an intimate relationship between **Team** $_n^1 \mathbf{Ex}$ -identification and identification by probabilistic machines. This relationship is then used to obtain results about the general team function criteria, **Team** $_n^m \mathbf{Ex}$ -identification.

All the results to follow in the present section are due to Smith [36].

The following result says that there are collections of functions for which a correct program can be synthesized by a team of $n + 1$ machines, at least one of which is successful, but for which even a finite variant program cannot be synthesized by a team of n machines with the requirement that at least one of them be successful.

Theorem 5 [36] $(\forall n \geq 1)[\mathbf{Team}_{n+1}^1 \mathbf{Ex} - \mathbf{Team}_n^1 \mathbf{Ex}^* \neq \emptyset]$.

PROOF. For $n \in N^+$, let $\mathcal{C}_n = \{f \in \mathcal{R} \mid (\exists x \leq n)[\text{card}(W_{f(x)}) < \infty \wedge \varphi_{\max(W_{f(x)})} = f]\}$.

It is easy to verify that $\mathcal{C}_n \in \mathbf{Team}_{n+1}^1 \mathbf{Ex}$. Below, we show using a diagonalization argument that $\mathcal{C}_2 \notin \mathbf{Team}_2^1 \mathbf{Ex}^*$. The following argument can easily be generalized to show that $\mathcal{C}_n \notin \mathbf{Team}_n^1 \mathbf{Ex}^*$.

Suppose by way of contradiction, there exist machines \mathbf{M}_0 and \mathbf{M}_1 such that $\mathcal{C}_2 \subseteq \mathbf{Team}_2^1(\mathbf{M}_0, \mathbf{M}_1)$. Then, by the implicit use of the operator recursion theorem, there exists a one to one, monotone increasing, recursive function p such that $\varphi_{p(\cdot)}$ ($W_{p(\cdot)}$) can be described in stages below.

We initialize $\varphi_{p(3)}(0) = p(0)$, $\varphi_{p(3)}(1) = p(1)$, and $\varphi_{p(3)}(2) = p(2)$. Enumerate $p(3)$ in to $W_{p(0)}$.

Let $\text{avail} = 3$. Intuitively, avail denotes the least number such that, for all $i > \text{avail}$, $p(i)$ has not been used in the diagonalization before. Let x_s denote the least x such that $\varphi_{p(3)}(x)$ has not been defined before stage s . Go to stage 0.

Begin {stage s }

1. Let $\text{avail} = \text{avail} + 1$.
Let $\text{cur} = \text{avail}$.
Enumerate $p(\text{cur})$ into $W_{p(1)}$.
For $x < x_s$, let $\varphi_{p(\text{cur})}(x) = \varphi_{p(3)}(x)$.
Let $r = s \bmod 2$.
2. Dovetail steps 3 and 4 until, if ever, step 3 succeeds. If and when step 3 succeeds go to step 5.
(* Intuitively, if step 3 succeeds in each stage, then $\varphi_{p(3)} \in \mathcal{C}_2$ and both \mathbf{M}_0 and \mathbf{M}_1 do not **Ex***-identify $\varphi_{p(3)}$.)
3. Search for an extension $\sigma \in \text{SEQ}$ of $\varphi_{p(3)}[x_s]$ such that $\varphi_{\mathbf{M}_r(\sigma)}(|\sigma|) \downarrow$.
4. Let $x_{s,s'}$ denote the least x such that $\varphi_{p(\text{cur})}(x)$ has not been defined before substage s' of stage s .

Go to substage 0.

Begin {substage s' }

- 4.1. Let $\text{avail} = \text{avail} + 1$.
 - 4.2. Enumerate $p(\text{avail})$ into $W_{p(2)}$.
 - 4.3. For $x < x_{s,s'}$, let $\varphi_{p(\text{avail})}(x) = \varphi_{p(\text{cur})}(x)$.
 - 4.4. Dovetail steps 4.5 and 4.6 until, if ever, step 4.5 succeeds. If and when step 4.5 succeeds, go to step 4.7.
 - 4.5. Search for an extension $\tau \in \text{SEQ}$ of $\varphi_{p(\text{cur})}[x_{s,s'}]$ such that $\varphi_{\mathbf{M}_{1-r}(\tau)}(|\tau|) \downarrow$.
 - 4.6. Let $x = x_{s,s'}$.
repeat
 Let $\varphi_{p(\text{avail})}(x) = 0$.
 Let $x = x + 1$.
forever
 - 4.7. Let τ be as found in step 4.5.
For $x < |\tau|$, let $\varphi_{p(\text{cur})}(x) = y$ such that $(x, y) \in \text{content}(\tau)$.
Let $\varphi_{p(\text{cur})}(|\tau|) = \varphi_{\mathbf{M}_{1-r}(\tau)}(|\tau|) + 1$.
 - 4.8. Go to substage $s' + 1$.
End {substage s' }
 5. Let σ be as found in step 3.
5.1. For $x < |\sigma|$, let $\varphi_{p(3)}(x) = y$ such that $(x, y) \in \text{content}(\sigma)$.
5.2. Let $\varphi_{p(3)}(|\sigma|) = \varphi_{\mathbf{M}_r(\sigma)}(|\sigma| + 1)$.
 6. Go to stage $s + 1$.
- End {stage s }

Now, consider the following cases.

Case 1: Each stage terminates.

Let $f = \varphi_{p(3)}$. Clearly, f is computable and a member of \mathcal{C}_2 (since $f(0) = p(0)$, and $W_{p(0)} = \{p(3)\}$). Also, because of the success of step 3 and the diagonalization at step 5.2 in each even (odd) stage, \mathbf{M}_0 (\mathbf{M}_1) either diverges on f or the last program output by \mathbf{M}_0 (\mathbf{M}_1) on f commits infinitely many convergent errors.

Case 2: Some stage s starts but does not terminate.

Let r and cur be as defined in step 1 of stage s . Since step 3 does not succeed, \mathbf{M}_r does not \mathbf{Ex}^* -identify any extension of $\varphi_{p(3)}$, and thus it does not \mathbf{Ex}^* -identify any extension of $\varphi_{p(\text{cur})}$.

Case 2.1: Each substage in stage s terminates.

Let $f = \varphi_{p(\text{cur})}$. Clearly, f is computable and a member of \mathcal{C}_2 , since $f(1) = p(1)$ and $\max(W_{p(1)}) = p(\text{cur})$. Also, by the success of step 4.5 and the diagonalization at step 4.7 in each substage, \mathbf{M}_{1-r} either diverges on f , or the last program output by \mathbf{M}_{1-r} on f commits infinitely many convergent errors.

Case 2.2: Some substage s' in stage s starts but does not terminate.

In this case let avail be as at step 4.2 in substage s' of stage s . Let $f = \varphi_{p(\text{avail})}$. Clearly, f is recursive and an extension of $\varphi_{p(\text{cur})}$. Also, f is a member of \mathcal{C}_2 , since $f(2) = p(2)$ and $\max(W_{p(2)}) = p(\text{avail})$. However, since step 4.5 does not succeed, \mathbf{M}_{1-r} , does not \mathbf{Ex}^* -identify any extension of $\varphi_{p(\text{cur})}$ and thus does not \mathbf{Ex}^* -identify f .

From the above cases it follows that $\mathcal{C}_2 \notin \mathbf{Team}_2^1(\mathbf{M}_0, \mathbf{M}_1)$. ■

The following corollary to the above theorem says that increasing the size of team renders larger collections of functions identifiable.

Corollary 1 [36] *Let $a \in N \cup \{*\}$. Then,*
 $\mathbf{Ex}^a = \mathbf{Team}_1^1 \mathbf{Ex}^a \subset \mathbf{Team}_2^1 \mathbf{Ex}^a \subset \mathbf{Team}_3^1 \mathbf{Ex}^a \subset \dots$

Now, if $m, n \in N^+$ such that $m \leq n$, then we would like to know for which values of $i, j \in N$, $\mathbf{Team}_m^1 \mathbf{Ex}^i \subseteq \mathbf{Team}_n^1 \mathbf{Ex}^j$. From the above corollary, it is clear that the \subseteq relationship holds if $i \leq j$, but we wish to determine for a given n , by how much can the value of j be reduced so that the \subseteq relationship still holds. In other words, we would like to know how anomalies in the final program can be traded for extra team members. This is the subject of next result; we omit the proof.

Theorem 6 [36] *For all $i, j \in N$ and $m \in N^+$, $\mathbf{Team}_m^1 \mathbf{Ex}^i \subseteq \mathbf{Team}_n^1 \mathbf{Ex}^j$, where $n = m(1 + \lfloor \frac{i}{j+1} \rfloor)$.*

The next two results show that above Theorem 6 is optimal. For Theorems 7 and 8 below, define $\mathcal{C}_{r,l}$, for $r, l \geq 1$, to be the collection of functions $\{f \in \mathcal{R} \mid \varphi_{f(0)} = {}^{r \cdot l} f \wedge (\exists i \leq r)[\text{card}(\{x \mid \varphi_{f(0)}(x) \neq f(x)\}) = i \cdot l]\}$.

Theorem 7 [36] *For all k , $\mathcal{C}_{r,l} \in \mathbf{Team}_m^1 \mathbf{Ex}^{k \cdot l}$, where $m = 1 + \lfloor \frac{r}{k+1} \rfloor$.*

Theorem 8 [36] *$\mathcal{C}_{r,l} \notin \mathbf{Team}_r^1 \mathbf{Ex}^{l-1}$.*

PROOF. The proof of the above theorem involves a complicated priority construction. Fix $r, l \geq 1$. Suppose by way of contradiction there exist machines $\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_r$ such that $\mathcal{C}_{r,l} \in \mathbf{Team}_r^1 \mathbf{Ex}^{l-1}(\mathbf{M}_1, \mathbf{M}_1, \dots, \mathbf{M}_r)$. Then, by implicit use of Kleene's recursion theorem, there exists an e such that the (partial) function φ_e may be described as follows.

φ_e employs r moving anomaly markers, each marking l consecutive numbers, which we are temporarily trying to keep out of the domain of φ_e . We name the r markers $\alpha_0, \dots, \alpha_{r-1}$. At any time, we let the set variable A_i denote the set of numbers marked by α_i at that time. A_i 's will be pairwise disjoint at all times. A_i^s denotes the value of A_i at the beginning of stage s .

In the construction, we will assign tasks, numbered 0 to ∞ , to the markers. Priority of a task numbered t is t . Lower number means higher priority. The tasks may get "done" at some stage. However, a higher priority task may undo a lower priority task which was completed earlier. Priority of a marker, at any time, is the priority of the highest priority task assigned to it which has not been done.

We now proceed to give an informal description of φ_e . Let $\varphi_e(0) = e$. For $i < r$, let $A_i^0 = \{l \cdot i + 1 + x \mid x < l\}$. Let φ_e^s denote the part of φ_e defined before stage s . We let x_s denote the least element not in $\text{domain}(\varphi_e^s) \cup \bigcup_{i < r} A_i^s$. It will be the case that $\text{domain}(\varphi_e^s) \cup \bigcup_{i < r} A_i^s$ is an initial segment of N . Let τ^s denote the sequence such that $\text{content}(\tau^s) = \varphi_e^s \cup \{(x, 0) \mid x \in \bigcup_{i < r} A_i^s\}$. Let σ_i^s denote the initial segment of τ^s with length $\min(A_i^s)$.

Initially assign task i to α_i . Let nexttask denote, at any time, the least task number which has not been assigned to any marker till that time. Thus, initially $\text{nexttask} = r$. Go to stage 0.

Begin {stage s }

1. We say that α_i requires attention at this stage just in case

$(\exists \tau' \mid \sigma_i^s \subseteq \tau' \subseteq \tau^s)[\mathbf{M}_i(\sigma_i^s) \neq \mathbf{M}_i(\tau')]$ or
 there exists an $x \in A_i^s$ such that $\Phi_{\mathbf{M}_i(\sigma_i^s)} \leq s$.

2. **if** no α_i requires attention, then let $\varphi_e(x_s) = 0$ and go to stage $s + 1$.

endif

3. (* *Some α_i requires attention.* *)

Let α_i be the highest priority marker which requires attention.

Let t be the priority of highest priority marker which requires attention.

4. "undo" all tasks with lower priority than t .

5. (* *In steps 5 and 6 task t gets "done".* *)

Let $y = x_s$.

for $j = 0$ **to** $r - 1$ **do**
 if priority of α_j is lower than that of α_i ,
 then
 For $x \in A_j^s$, let $\varphi_e(x) = 0$.
 Let $A_j = \{y + x \mid x < l\}$.
 Let $y = y + l$.
 (* Note that in this step A_j changes. Thus, the marker α_j moves. *)
 endif
endfor
6. **if** there exists an $x \in A_i^s$ such that $\Phi_{\mathbf{M}_i(\sigma_i^s)} \leq s$
 then
 Let $x' \in A_i^s$ be such that $\Phi_{\mathbf{M}_i(\sigma_i^s)}(x') \leq s$.
 Let $\varphi_e(x') = \varphi_{\mathbf{M}_i(\sigma_i^s)}(x') + 1$.
 For $x \in A_i^s - \{x'\}$, let $\varphi_e(x) = 0$.
 Let $A_i^s = \{y + x \mid x < l\}$.
 Let $y = y + l$.
 else
 For $x \in A_i^s$, let $\varphi_e(x) = 0$.
 Let $A_i = \{y + x \mid x < l\}$.
 Let $y = y + l$.
 endif
 (* Note that in this step A_i changes. Thus, the marker α_i “moves”. *)
 Mark task t as “done”.
7. Assign task nexttask to marker α_i .
 Let nexttask = nexttask + 1.
8. Go to stage $s + 1$.
End {stage s }

It is easy to see that every stage terminates.

Claim 1 *Each task gets “done” or “undone” only finitely often.*

PROOF. A proof by induction suffices. Suppose all tasks numbered $< t$, get done or undone only finitely often. Let s be a stage such that no task numbered $< t$, gets done or undone beyond stage s . Then beyond stage s task t can never get undone, and thus it can get done at most once more beyond stage s . \square

Claim 2 *A marker can move due to step 5 at most finitely many times before it moves (afresh) due to step 6.*

PROOF. A marker can move due to step 5 in some stage s , only if it has lower priority than some marker which requires attention and gets done at stage s . It is now easy to prove this claim using Claim 1. We leave the details to the reader. \square

Claim 3 *If a marker requires attention only finitely often, then*

- (a) *the set of numbers assigned to it stabilizes in the limit, and*
- (b) *only finitely many tasks are assigned to it.*

PROOF. A marker can move at stage s only if the highest priority task assigned to it, which is not done at the beginning of stage s , gets done at stage s , or a higher priority task gets done at stage s . Part (a) now follows using Claims 1 and 2.

Part (b) is immediate because a task can be assigned to a marker at stage s only if it requires attention at stage s . \square

We say that a task is completed (at stage s), if it gets done at some stage s and never gets undone thereafter.

Claim 4 *If a marker requires attention infinitely often, then each task assigned to it gets completed, and thus*

- (a) *it moves infinitely often, and*
- (b) *infinitely many tasks are assigned to it.*

PROOF. Let t be the least numbered task such that t never gets completed, and the marker to which t is assigned gets attention infinitely often. Let s be a stage such that all tasks with priority higher than t which ever get completed are completed before stage s and no marker, which requires attention only finitely often, requires attention beyond stage s . Let $s' > s$ be such that the marker to which t is assigned gets attention at stage s' . Then, task t gets done at stage s' and never gets undone after stage s' —a contradiction. \square

Let f be the zero extension of φ_e . Clearly, $\varphi_e(x) \uparrow$ iff there exists an $i < r$ such that $\lim_s A_i^s$ exists and $x \in \lim_s A_i^s$. Thus, $f \in \mathcal{C}_{r,l}$.

Now for each machine \mathbf{M}_i , $i < r$, we show that \mathbf{M}_i does not \mathbf{Ex}^{l-1} -identify f .

Let lastfin be such that (a) no marker, which requires attention only finitely often, requires attention beyond stage lastfin , and (b) no marker which moves only finitely often, moves after stage lastfin .

For each $i < r$, consider the following cases.

Case 1: α_i requires attention finitely often.

In this case, $A_i^{\text{lastfin}} = \lim_s A_i^s$ and $\mathbf{M}_i(f) = \mathbf{M}(\sigma_i^{\text{lastfin}})$ and $A_i^{\text{lastfin}} \cap \text{domain}(\varphi_{\mathbf{M}(f)}) = \emptyset$ (otherwise, α_i would require attention beyond stage lastfin). Thus, \mathbf{M}_i does not \mathbf{Ex}^{l-1} -identify f .

Case 2: α_i requires attention infinitely often.

By Claim 4, infinitely many tasks are assigned to α_i . Note that if a task assigned to α_i gets completed at stage $s > \text{lastfin}$, then $\sigma_i^s \subseteq f$ and either \mathbf{M}_i is forced to change its mind on f after σ_i^s , or $\varphi_{\mathbf{M}_i(f)}$ commits an error in A_i^s . Thus, either \mathbf{M}_i changes mind infinitely often on f , or $\mathbf{M}(f)$ commits infinitely many convergent errors. ■

Theorem 9 [36] $\mathbf{Team}_m^1 \mathbf{Ex}^a \subseteq \mathbf{Team}_n^1 \mathbf{Ex}^b$ iff

- (a) $m \leq n$, and
- (b) $b = *$ or $n \geq m \cdot (1 + \lfloor a/(b+1) \rfloor)$.

PROOF. If part follows from Theorem 6.

For the only if part, observe that by Theorem 5, n must be $\geq m$. Suppose $b \neq *$. Now, let $l = b + 1$, and $r = m \cdot (1 + \lfloor a/(b+1) \rfloor) - 1$. Let $\mathcal{C}_{r,l}$ be as defined just before Theorem 7. Now, by Theorem 7, $\mathcal{C}_{r,l} \in \mathbf{Team}_m^1 \mathbf{Ex}^{(b+1) \cdot \lfloor a/(b+1) \rfloor} \subseteq \mathbf{Team}_m^1 \mathbf{Ex}^a$. Also, by Theorem 8, $\mathcal{C}_{r,l} \notin \mathbf{Team}_r^1 \mathbf{Ex}^b$. ■

4 Limiting Identification of Functions by Probabilistic Machines

The present section considers machines whose actions may be determined by the outcome of random events. These devices, referred to as probabilistic machines, behave very much like algorithmic machines except that every now and then they have the ability to base their actions on the outcome of a random event like a coin flip.

More precisely, let t be a positive integer greater than 1. Then, a probabilistic machine \mathbf{P} may be construed as a algorithmic machine that is equipped with a t -sided coin. The response of \mathbf{P} to an evidential situation σ not only depends upon σ but also on the outcomes of coin flips performed by \mathbf{P} till that point. It is useful to make the notion of sequence of outcomes of a t -ary coin precise.

Definition 8 Let $t > 1$.

- (a) N_t denotes the set $\{0, 1, 2, \dots, t-1\}$.
- (b) An oracle for a t -sided coin, also referred to as a t -ary oracle is an infinite sequence of integers i_1, i_2, i_3, \dots such that for each j , $i_j \in N_t$. (A typical variable for oracles is O).

Clearly, N_t^∞ , the infinite cartesian product of N_t with itself, denotes the collection of all t -sided coin oracles. Observe that a t -ary oracle is somewhat like a text for the finite language N_t , and notations for texts carry over to oracles; the next definition records these conventions.

Definition 9 Let $t > 1$.

- (a) Let O be a t -ary oracle. Then, the n^{th} member of O is denoted O_n . The initial finite sequence of O of length n is denoted $O[n]$.

(b) The set $\{O[n] \mid O \text{ is a } t\text{-ary oracle and } n \in \mathbb{N}\}$ is the collection of all finite t -ary sequences. (A typical variable for finite t -ary sequences is ρ).

(c) Let ρ be a finite t -ary sequence. The length of ρ is denoted by $|\rho|$. For $n < |\rho|$, the n^{th} member of ρ is denoted by ρ_n , and the initial sequence of length n in ρ is denoted by $\rho[n]$.

Let ρ be a finite t -ary sequence and \mathbf{P} be a probabilistic machine equipped with a t -sided coin. Let $\sigma \in \text{SEG}$. Then, $\mathbf{P}^\rho(\sigma)$ denotes the output of \mathbf{P} on σ such that the result of any coin flip performed by \mathbf{P} are ‘read’ from ρ , that is, the outcome of the first coin flip is ρ_0 , the outcome of the second coin flip is ρ_1 , and so on and so forth. If \mathbf{P} performs more coin flips than $|\rho|$ in responding to the evidential state σ , then $\mathbf{P}^\rho(\sigma)$ is undefined.

Similarly, we can describe the behavior of \mathbf{P} for a given t -ary oracle O . \mathbf{P}^O behaves like \mathbf{P} except whenever \mathbf{P} flips its coin, \mathbf{P}^O reads the result of the coin flip from the oracle O , that is, the result of the first coin flip is O_0 , the result of the second coin flip is O_1 , and so on and so forth. Now, if the sequence of hypotheses issued by \mathbf{P}^O on the graph of a computable function f corresponds to an **Ex**-identification of f , then \mathbf{P}^O is said to **Ex**-identify f .

Our first task is to define the probability of a probabilistic machine \mathbf{P} **Ex**-identifying f . The subject of identification by probabilistic machine was first investigated by R. Freivalds [16] and L. Pitt [31, 32]. Our presentation closely follows that of Pitt. We first review some necessary probability theory.

4.1 Background Probability Theory

Let the outcomes of an experiment be elements of some universal set Ω . A probability measure may then be thought of as a function that assigns real values between 0 and 1 to outcomes of an experiment. In practice it is useful to define a probability measure on subsets of Ω . However, defining a probability measure on the power set of Ω poses technical difficulties, and hence it is defined only on those collections of subsets of Ω that satisfy certain properties stated in the following definition.

Definition 10 $\mathcal{B} \subseteq 2^\Omega$ is a Borel field just in case the following conditions hold:

(a) $\Omega \in \mathcal{B}$,

(b) $A \in \mathcal{B} \Rightarrow \Omega - A \in \mathcal{B}$,

(c) \mathcal{B} is closed under countable unions and intersections, i.e., if $\{A_i\}_{i \in I}$ is a finite or countable collection of elements of \mathcal{B} , then $\bigcup_{i \in I} A_i \in \mathcal{B}$ and $\bigcap_{i \in I} A_i \in \mathcal{B}$.

Given any collection \mathcal{C} of subsets of Ω , there is a unique smallest (with respect to containment) Borel field containing \mathcal{C} . The next definition introduces the notion of a probability measure on a Borel field \mathcal{B} on subsets of Ω .

Definition 11 A probability measure pr on a Borel field \mathcal{B} of subsets of Ω is a function $pr : \mathcal{B} \rightarrow [0 \dots 1]$ such that

(a) $pr(\Omega) = 1,$

(b) $(\forall A)[A \in \mathcal{B} \Rightarrow pr(A) \geq 0],$

(3) If $\{A_i\}$ is a finite or countable collection of mutually disjoint elements of \mathcal{B} , then $pr(\bigcup_i A_i) = \sum_i pr(A_i).$

Elements of \mathcal{B} are called *measurable* sets. We note some properties of measurable sets.

If $\{A_i\}$ is a countable collection of sets, then $\lim_{k \rightarrow \infty} \sup A_k = \bigcap_{k=0}^{\infty} \bigcup_{i=k}^{\infty} A_i$ and $\lim_{k \rightarrow \infty} \inf A_k = \bigcup_{k=0}^{\infty} \bigcap_{i=k}^{\infty} A_i$. If the limit supremum and limit infimum of a sequence of sets $\{A_i\}$ are equal, then this is the limit of the sequence. A sequence of sets $\{A_i\}$ is *monotone* if either $(\forall k)[A_k \subseteq A_{k+1}]$ or $(\forall k)[A_{k+1} \subseteq A_k]$. Every monotone sequence of sets has a limit, and every Borel field is closed under $\lim \inf$ and $\lim \sup$. If $\{A_i\}$ is a sequence of measurable sets for which the limit is defined, then $pr(\lim_{k \rightarrow \infty} A_k) = \lim_{k \rightarrow \infty} pr(A_k).$

Definition 12 A probability space is a triple $(\Omega, \mathcal{B}, pr)$ of a sample space Ω , a Borel field \mathcal{B} on subsets of Ω , and a probability measure pr on \mathcal{B} .

Given a probability space $(\Omega, \mathcal{B}, pr)$ and a set $A \subseteq \Omega$, A can be shown to be measurable by expressing A in terms of countable intersections, unions, and complements of known measurable sets. Similarly, $pr(A)$ can be computed by using properties of probability measures on the values of these known measurable sets.

The aim of all this machinery is to eventually define a probability space on oracle sequences which in turn is used to calculate the probability of a machine to **Ex**-identify a function. To this end, we first introduce a probability measure on a single coin flip. For a t -sided coin, let $(N_t, \mathcal{B}_t, pr_t)$ be a probability space on the sample space N_t , where $\mathcal{B}_t = \{S \mid S \subseteq N_t\}$ and $pr_t = \text{card}(S)/t$. Intuitively, this measure simply says that the probability of the outcome of flipping a t -sided coin belonging to a set $S \subseteq N_t$ is $\text{card}(S)/t$. We use this measure next to describe a probability measure on t -ary oracles.

Now, identification by a probabilistic machine may be viewed as an ongoing process in which a machine receives data, flips coin, and issues hypotheses. During identification, each of these activities may occur infinitely often. We would like to introduce a reasonable probability measure on an infinite sequence of coin flips. As already mentioned, the act of recording the outcomes of an infinite sequence of coin flips can be viewed as reading values off an infinite t -ary oracle. Thus, the sample space of events for oracles of a t -sided coin is N_t^∞ —the set of all infinite sequences of numbers less than t . Let \mathcal{B}_t^∞ be the smallest Borel field of subsets of N_t^∞ containing all the sets $N_t^{j-1} \times A_j \times N_t^\infty$, where for each j , $A_j \in \mathcal{B}_t$. Then, let $(N_t^\infty, \mathcal{B}_t^\infty, pr_t^\infty)$ be a probability space where pr_t^∞ is defined as follows:

Given a nonempty set of n integers, $i_1, i_2, i_3, \dots, i_n$, such that $0 < i_1 < i_2 < i_3 < \dots < i_n$, let $A_{i_1, i_2, i_3, \dots, i_n}$ denote the set $N_t^{i_1-1} \times A_{i_1} \times N_t^{i_2-i_1-1} \times A_{i_2} \times N_t^{i_3-i_2-1} \times A_{i_3} \times \dots \times A_{i_n} \times N_t^\infty$, where each $A_{i_j} \in \mathcal{B}_t$. Then, pr_t^∞ is defined on \mathcal{B}_t^∞ such that $pr_t^\infty(A_{i_1, i_2, \dots, i_n}) = \prod_{j=1}^n pr_t(A_{i_j})$, for each choice of n integers i_1, i_2, \dots, i_n .

Clearly, sets $A_{i_1, i_2, i_3, \dots, i_n}$ are measurable. Let us now look at some examples of measurable sets of oracles.

Consider a t -sided coin and j such that $0 \leq j \leq t - 1$. Then the set $\{O \mid O_n = j\}$ consists of all such t -ary oracles that have j as their n th value. As an immediate consequence of the forgoing discussion, $\{O \mid O_n = j\}$ is measurable, and $\text{pr}_t^\infty(\{O \mid O_n = j\}) = 1/t$.

As another example consider a finite t -ary sequence ρ of length k . Then, a t -ary oracle O *extends* ρ just in case $O[k] = \rho$. Again, as an immediate consequence of the forgoing discussion, the set of oracles $\{O \mid O \text{ extends } \rho\}$ is measurable, and $\text{pr}_t^\infty(\{O \mid O \text{ extends } \rho\}) = 1/t^k$. This fact is used below in showing that the set of oracles that correspond to a successful identification of a function is measurable.

4.2 Probability of Function Identification

Let \mathbf{P} be a probabilistic machine equipped with a t -sided coin and let $f \in \mathcal{R}$. Then, the probability of \mathbf{P} **Ex**-identifying f is taken to be $\text{pr}_t^\infty(\{O \mid \mathbf{P}^O \mathbf{Ex}\text{-identifies } f\})$. However, to be able to compute such a probability, it needs to be established that the set $\{O \mid \mathbf{P}^O \mathbf{Ex}\text{-identifies } f\}$ is measurable. This is the subject of next lemma.

Lemma 1 [31, 32] *Let \mathbf{P} be a probabilistic machine and let $f \in \mathcal{R}$. Then $\{O \mid \mathbf{P}^O \mathbf{Ex}\text{-identifies } f\}$ is measurable.*

PROOF. Let $\text{plausible}(\mathbf{P}, f, j, \varrho)$ be a boolean predicate, where \mathbf{P} ranges over probabilistic machines equipped with a t -ary oracle, f ranges over \mathcal{R} , j ranges over N^+ , and ϱ ranges over $N_t^\infty \cup \bigcup_{k=j}^\infty N_t^k$ (i.e., ϱ can be any t -ary sequence of length $\geq j$). Then, $\text{plausible}(\mathbf{P}, f, j, \varrho) \iff \mathbf{P}^\varrho(f[j]) = \mathbf{P}^\varrho(f[j-1]) \wedge \varphi_{\mathbf{P}^\varrho(f[j])} = f$. Then,

$$\begin{aligned} & \{O \mid \mathbf{P}^O \mathbf{Ex}\text{-identifies } f\} \\ &= \{O \mid (\exists k)(\forall j \geq k)[\text{plausible}(\mathbf{P}, f, j, O)]\} \\ &= \bigcup_{k=1}^\infty \{O \mid (\forall j \geq k)[\text{plausible}(\mathbf{P}, f, j, O)]\} \\ &= \bigcup_{k=1}^\infty \bigcap_{j=k}^\infty \{O \mid \text{plausible}(\mathbf{P}, f, j, O)\} \\ &= \bigcup_{k=1}^\infty \bigcap_{j=k}^\infty \bigcup_{\rho \in N_t^j \wedge \text{plausible}(\mathbf{P}, f, j, \rho)} \{O \mid O \text{ extends } \rho\}. \end{aligned}$$

But, it has already been shown that the set of oracles $\{O \mid O \text{ extends } \rho\}$ is measurable. Thus, $\{O \mid \mathbf{P}^O \mathbf{Ex}\text{-identifies } f\}$ is measurable because it can be expressed as countable unions and intersections of measurable sets. ■

The following definition is motivated by the above lemma.

Definition 13 [31, 32] *Let $f \in \mathcal{R}$ and \mathbf{P} be a probabilistic machine equipped with a t -sided coin ($t \geq 2$). Then, $\text{pr}_t^\infty(\mathbf{P} \mathbf{Ex}\text{-identifies } f) = \text{pr}_t^\infty(\{O \mid \mathbf{P}^O \mathbf{Ex}\text{-identifies } f\})$.*

The next lemma says that we do not sacrifice any learning power by restricting our attention to the investigation of identification by probabilistic machines equipped with only a two-sided coin. The proof of the lemma follows from a result in probability theory and we omit the details (the reader is directed to Pitt [32]) for a proof).

Lemma 2 [31, 32] *Let $t > 2$. Let \mathbf{P} be a probabilistic machine with a t -sided coin. Then, there exists a probabilistic machines \mathbf{P}' with a two-sided coin such that for each $f \in \mathcal{R}$, $pr_2^\infty(\mathbf{P}' \mathbf{Ex}\text{-identifies } f) = pr_t^\infty(\mathbf{P} \mathbf{Ex}\text{-identifies } f)$.*

We now present identification by probabilistic machines as a paradigm. The above lemma frees us from specifying the number of sides of the coin, thereby allowing us to talk about probability function pr_t^∞ without specifying t . For this reason, we will refer to pr_t^∞ as simply pr in the sequel. Also, we are at liberty to use whatever value of the number of sides of a coin that is convenient for the presentation at hand.

Definition 14 [31, 32] Let $p \in [0 \dots 1]$.

- (a) $\mathbf{P} \mathbf{Prob}^p \mathbf{Ex}\text{-identifies } f$ (written: $f \in \mathbf{Prob}^p \mathbf{Ex}(\mathbf{P})$) just in case $pr(\mathbf{P} \mathbf{Ex}\text{-identifies } f) \geq p$.
- (b) $\mathbf{Prob}^p \mathbf{Ex} = \{\mathcal{S} \subseteq \mathcal{R} \mid (\exists \mathbf{P})[\mathcal{S} \subseteq \mathbf{Prob}^p \mathbf{Ex}(\mathbf{P})]\}$.

An immediate result is the following theorem which links identification by probabilistic machines and team identification. The result shows that any collection of functions which can be identified by a team of n machines, at least one of which is required to be successful, can be identified by a probabilistic machine with probability $\geq 1/n$.

Theorem 10 [31, 32] $(\forall n \geq 1)[\mathbf{Team}_n^1 \mathbf{Ex} \subseteq \mathbf{Prob}^{\frac{1}{n}} \mathbf{Ex}]$.

PROOF. Let $\mathcal{S} \in \mathbf{Team}_n^1 \mathbf{Ex}$. Then, there exists a team of n machines $\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_n$ that $\mathbf{Team}_n^1 \mathbf{Ex}\text{-identifies}$ each $f \in \mathcal{S}$. Let \mathbf{P} be a probabilistic machine equipped with an n -sided coin. The behavior of \mathbf{P} is described thus. \mathbf{P} flips its coin once and obtains with probability $1/n$ a number $i \in N_n$. \mathbf{P} then simulates machine \mathbf{M}_{i+1} . Clearly, for each $f \in \mathcal{S}$, $pr(\mathbf{P} \mathbf{Ex}\text{-identifies } f) \geq 1/n$. ■

Pitt [31, 32] also established the converse of the above result, $(\forall n \geq 1)[\mathbf{Prob}^{\frac{1}{n}} \mathbf{Ex} \subseteq \mathbf{Team}_n^1 \mathbf{Ex}]$, thereby showing that probabilistic identification and team identification are successful on essentially the same collections on functions. This result will be an immediate corollary of the main result of this section (Theorem 11). In order to prove this main result, Pitt used a technique of calculating probabilities on “infinite computation tress,” which we describe next. To facilitate the description, it is expedient to place some “harmless” restrictions on our probabilistic machines. The next definition describes these restrictions.

Definition 15 [31, 32] A probabilistic machine \mathbf{P} is *nice* just in case \mathbf{P} is equipped with a two-sided coin and the performance of \mathbf{P} on a function f follows the following sequence in an infinite loop:

- (a) receive an element of the graph of f ;
- (b) issue a hypothesis;
- (c) flip its coin.

The following lemma allows us to restrict our attention to only nice probabilistic machines; its proof is left to the reader.

Lemma 3 [31, 32] *Let \mathbf{P} be a probabilistic machine. Then there exists a nice \mathbf{P}' such that $(\forall f \in \mathcal{R})[pr(\mathbf{P} \text{ Ex-identifies } f) \leq pr(\mathbf{P}' \text{ Ex-identifies } f)]$.*

4.3 Infinite Computation Trees

Let \mathbf{P} be a nice probabilistic machine and $f \in \mathcal{R}$. An infinite computation tree for \mathbf{P} on f , denoted $\mathcal{T}_{\mathbf{P},f}$, is simply a description of \mathbf{P} 's behavior on all possible 2-ary oracles when \mathbf{P} is fed the graph of f in canonical order. $\mathcal{T}_{\mathbf{P},f}$ is an infinite complete binary tree whose nodes represent the state just after \mathbf{P} has performed the following two actions:

- (1) received an element of the graph (in canonical order) of f ;
- (2) issued an hypothesis.

Since \mathbf{P} is nice, the next action of \mathbf{P} is a coin flip. The two edges emanating from a node correspond to the two possible directions that the computation of \mathbf{P} can take as determined by the result of the coin flip.

The nodes of $\mathcal{T}_{\mathbf{P},f}$ are numbered in breadth first search order starting with the root node which is numbered '1'. Observe that the root node represents the state of \mathbf{P} just after it has received $(0, f(0))$ and issued its first hypothesis. At this stage, \mathbf{P} , being nice, flips its coin and if the outcome is '0' the computation follows the left child and if the outcome is '1' the computation follows the right child. Thus, node 2 represents the state just after the following events have taken place in sequence:

- (1) \mathbf{P} flips its coin for the first time;
- (2) the outcome of the coin flip in (1) is '0';
- (3) \mathbf{P} receives $(1, f(1))$; and
- (4) \mathbf{P} issues its second hypothesis.

The depth of a node in $\mathcal{T}_{\mathbf{P},f}$ is denoted by $\text{depth}(n)$, where $\text{depth}(n) = \lfloor \log_2(n) \rfloor$. Note that the depth of the root node is 0. To summarize, a node n of depth d in $\mathcal{T}_{\mathbf{P},f}$ corresponds to the state of \mathbf{P} reached if \mathbf{P} has received $d + 1$ data points, issued $d + 1$ hypotheses, and the outcomes of the sequence of d coin flips performed by \mathbf{P} were exactly the sequence of 0's and 1's that lead to the node n in the tree $\mathcal{T}_{\mathbf{P},f}$.

We now define two useful functions on nodes of the tree $\mathcal{T}_{\mathbf{P},f}$. First, for any node n , the parent of n , denoted $\text{parent}(n)$, is the immediate ancestor of node n in $\mathcal{T}_{\mathbf{P},f}$; $\text{parent}(1)$ is undefined. Second, $\text{guess}(n)$ denotes the hypothesis that \mathbf{P} has just issued when it is in the state corresponding to node n . A path P in $\mathcal{T}_{\mathbf{P},f}$ is an infinite sequence of nodes $P_0, P_1, P_2, P_3, \dots$, such that $P_0 = 1$ (the root node) and for each i , $P_i = \text{parent}(P_{i+1})$. Observe that for each i , the i th node P_i in P occurs at depth i of $\mathcal{T}_{\mathbf{P},f}$. Also observe that each path in $\mathcal{T}_{\mathbf{P},f}$ corresponds to a unique 2-ary oracle and for each 2-ary oracle there is a unique path in $\mathcal{T}_{\mathbf{P},f}$. This isomorphism between the set of coin oracles

and the set of paths allows us to extend the function pr to sets of paths in $\mathcal{T}_{\mathbf{P},f}$ as follows:

Let C denote a collection of paths in $\mathcal{T}_{\mathbf{P},f}$. Then $\text{pr}(C) = \text{pr}(\{O \mid O \text{ corresponds to a path } P \in C\})$. An especially useful collection of paths is introduced in the following definition.

Definition 16 For each node n in $\mathcal{T}_{\mathbf{P},f}$, $P\langle n \rangle = \{P \mid P \text{ is a path in } \mathcal{T}_{\mathbf{P},f} \text{ and } P \text{ contains node } n\}$.

It is easy to verify that $\text{pr}(P\langle n \rangle) = 2^{-\text{depth}(n)}$, since $P\langle n \rangle$ corresponds to $\{O \mid O \text{ extends } \rho\}$, where ρ corresponds to the finite path segment starting from the root and leading to node n . The measurable sets $P\langle n \rangle$ will be used in computing probabilities of more interesting collections of paths. But, first we develop some more machinery about paths in $\mathcal{T}_{\mathbf{P},f}$. The next definition describes what it means for paths in $\mathcal{T}_{\mathbf{P},f}$ to converge.

Definition 17 Let $P = P_0, P_1, P_2, \dots$ be a path in $\mathcal{T}_{\mathbf{P},f}$.

- (a) P converges to j just in case $(\forall k)[\text{guess}(P_k) = j]$.
- (b) P converges at node n just in case the following hold:

- (i) P passes through node n (that is, $P_{\text{depth}(n)} = n$);
- (ii) $(\forall k \geq \text{depth}(n))[\text{guess}(P_k) = \text{guess}(P_n)]$;
- (iii) $\neg(\exists k < \text{depth}(n))(\forall m \geq k)[\text{guess}(P_m) = \text{guess}(P_n)]$.

Definition 18

$C(A) = \{P \mid P \text{ is a path in } \mathcal{T}_{\mathbf{P},f} \text{ and there exists an } a \in A \text{ such that } P \text{ converges to } a\}$.

$C(A)$ is the collection of all such paths in $\mathcal{T}_{\mathbf{P},f}$ that converge to some index in the set A . Let good_f denote the collection of all φ -indices for f , i.e., $\text{good}_f = \{i \mid \varphi_i = f\}$. Then, $C(\text{good}_f)$ is the collection of all such paths (oracles) which result in successful **Ex**-identification of f by \mathbf{P} . Hence, we write $\text{pr}(\mathbf{P} \text{ Ex-identifies } f) = \text{pr}(C(\text{good}_f))$.

One of the aims of developing this machinery is to be able to compute $\text{pr}(C(A))$. To this end, the next definition introduces further refinements on the collection of paths $C(A)$.

Definition 19 (a) $C_j = \{P \mid P \text{ is a path in } \mathcal{T}_{\mathbf{P},f} \text{ and } P \text{ converges at node } j\}$.

(b) A path $P = P_0, P_1, P_2, \dots$ k -agrees with C_j just in case the following hold:

- (i) $P_{\text{depth}(j)} = j$;
- (ii) $(\forall i \mid \text{depth}(j) \leq i \leq k)[\text{guess}(P_i) = \text{guess}(P_j)]$;
- (iii) $[j \text{ is the root}] \vee [\text{guess}(P_{\text{depth}(j)-1}) \neq \text{guess}(j)]$.

(c) $C_{j,k} = \{P \mid P \text{ is a path in } \mathcal{T}_{\mathbf{P},f} \wedge P \text{ } k\text{-agrees with } C_j\}$.

Observe that C_j is the collection of paths that converge at node j (to $\text{guess}(j)$), and $C_{j,k}$ is the collection of paths that appear to converge to $\text{guess}(j)$ up to level k in $\mathcal{T}_{\mathbf{P},f}$. Also, observe that $C_{j,k}$ provides a better estimate of C_j with increasing k , and $C_j = \bigcap_{k=\text{depth}(j)}^{\infty} C_{j,k}$. These and some other properties of these collections of paths are summarized in the next lemma, a proof of which is left to the reader.

Lemma 4 [31, 32] (a) $(\forall k \geq \text{depth}(j))[C_{j,k} \supseteq C_{j,k+1}]$.

(b) $(\forall k \geq \text{depth}(j))[pr(C_{j,k}) \geq pr(C_j)]$.

(c) $pr(C_j) = \lim_{k \rightarrow \infty} pr(C_{j,k})$.

Now, observe that $C(A) = \bigcup_{\text{guess}(j) \in A} C_j$, where the C_j 's are mutually disjoint because any path converges at a single node and $j_1 \neq j_2$ implies $C_{j_1} \cap C_{j_2} = \emptyset$. Thus, if we can show that $C_{j,k}$'s are measurable, then so will be C_j and $C(A)$. This is the subject of the next lemma. But, first a technical definition.

Definition 20 Let $j \in N$ and let $k \geq \text{depth}(j)$. Then, $N_{j,k} = \{n \mid \text{depth}(n) = k \wedge (\exists P \in C_{j,k})[P_k = n]\}$.

Lemma 5 [31, 32] For each j , for each $k \geq \text{depth}(j)$, the following hold:

(a) $C_{j,k}$ is measurable;

(b) $pr(C_{j,k}) = \text{card}(N_{j,k})/2^k$;

(c) $pr(C_{j,k})$ can be computed by looking at only the first k levels of $\mathcal{T}_{\mathbf{P},f}$.

PROOF. (a) Recall that $P\langle x \rangle$, the collection of paths in $\mathcal{T}_{\mathbf{P},f}$ passing through node x , is measurable. $C_{j,k}$ is shown to be measurable by expressing it as union of mutually disjoint $P\langle x \rangle$'s.

Observe that for each $y_1, y_2 \in N_{j,k}$, $y_1 \neq y_2 \Rightarrow P\langle y_1 \rangle \cap P\langle y_2 \rangle = \emptyset$. This is because $\text{depth}(y_1) = \text{depth}(y_2) = k$, and every path must pass through exactly one node at each level.

It is now claimed that $C_{j,k} = \bigcup_{x \in N_{j,k}} P\langle x \rangle$.

To see that $C_{j,k} \subseteq \bigcup_{x \in N_{j,k}} P\langle x \rangle$, let $P \in C_{j,k}$. Then P passes through some node y at level k , and $y \in N_{j,k}$. Therefore, $P \in P\langle y \rangle$ and $P\langle y \rangle \subseteq \bigcup_{x \in N_{j,k}} P\langle x \rangle$.

Similarly, to see that $\bigcup_{x \in N_{j,k}} P\langle x \rangle \subseteq C_{j,k}$, let $P \in \bigcup_{x \in N_{j,k}} P\langle x \rangle$, and let y be the node at depth k on P . Now, since the definition of $N_{j,k}$ doesn't depend on nodes deeper than level k , all paths passing through y must be in $C_{j,k}$.

Thus, $C_{j,k}$ is measurable.

(b) The proof of part (a) implies that $pr(C_{j,k}) = \sum_{x \in N_{j,k}} pr(P\langle x \rangle) = \sum_{x \in N_{j,k}} 2^{-\text{depth}(x)} = \sum_{x \in N_{j,k}} 2^{-k} = \text{card}(N_{j,k})/2^k$.

(c) It is easy to see that $N_{j,k}$ can be computed by observing only the first k levels of $\mathcal{T}_{\mathbf{P},f}$. Thus, $\text{pr}(C_{j,k})$ can be computed from the first k levels of $\mathcal{T}_{\mathbf{P},f}$. ■

We now present a lemma that is crucial to the proof of the main result of this section.

Lemma 6 [31, 32] *For all $A \subseteq N$ and for all $p \in [0 \dots 1]$, if $\text{pr}(C(A)) > p$, then there exist nodes $\{n_1, n_2, \dots, n_k\}$ such that for each i , $1 \leq i \leq k$, $\text{guess}(n_i) \in A$ and $\text{pr}(\cup_{j=1}^k C_{n_j}) > p$.*

PROOF. Observe that for j_1 and j_2 , $j_1 \neq j_2 \Rightarrow C_{j_1} \cap C_{j_2} = \emptyset$. Thus,

$$\text{pr}(C(A)) = \sum_{j \in \{m \mid \text{guess}(m) \in A\}} \text{pr}(C_j) > p.$$

But, since a path can converge at at most one node, there is a finite collection of nodes $\{n_1, n_2, \dots, n_k\}$ such that $\text{pr}(\cup_{j=1}^k C_{n_j}) = \sum_{j=1}^k \text{pr}(C_{n_j}) > p$. ■

Theorem 11 [31, 32] $(\forall n \geq 1)(\forall p)[1/(n+1) < p \leq 1 \Rightarrow \mathbf{Prob}^p \mathbf{Ex} \subseteq \mathbf{Team}_n^1 \mathbf{Ex}]$.

Theorem 11 immediately follows from the following theorem and the fact that $\mathbf{Oex} = \mathbf{Ex}$ (see Theorem 2).

Theorem 12 [31, 32] *Let $p > 1/(n+1)$. Let $\mathcal{S} \in \mathbf{Prob}^p \mathbf{Ex}$. Then, $(\exists \mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_n)$ such that $(\forall f \in \mathcal{S})(\exists i \mid 1 \leq i \leq n)[\mathbf{M}_i \mathbf{Oex}$ -identifies $f]$.*

PROOF. Let \mathbf{P} be a probabilistic machine such that $\mathcal{S} \subseteq \mathbf{Prob}^p \mathbf{Ex}(\mathbf{P})$. Using the description of \mathbf{P} , a team of n deterministic machines, $\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_n$, is described such that for each $f \in \mathcal{S}$, some member of the team \mathbf{Oex} -identifies f . This is achieved by constructing the machines in such a way that each machine guesses a different range of converging paths in $\mathcal{T}_{\mathbf{P},f}$ and for each f , one of the machines has the correct guess. This will also be the machine which \mathbf{Oex} -identifies f . The machines are described below.

Begin $\{\mathbf{M}_i(f[k])\}$

1. Construct \mathcal{T}_k , the first k levels of $\mathcal{T}_{\mathbf{P},f}$, by simulating \mathbf{P} with input $f[k]$ and all 2-ary sequences of length k .
2. For each node j in \mathcal{T}_k , compute $\text{pr}(C_{j,k})$.
3. Let c_k be the least numbered node in \mathcal{T}_k such that $\sum_{j=1}^{c_k} \text{pr}(C_{j,k}) > i/(n+1)$.
4. **if** c_k found in Step 3,
then output q such that $D_q = \{\text{guess}(i) \mid 1 \leq i \leq c_k\}$
else output 0
endif

End $\{\mathbf{M}_i(f[k])\}$

Now, $f \in \mathbf{Prob}^p \mathbf{Ex}(\mathbf{P})$ implies that $\text{pr}(C(\text{good}_f)) \geq p$. Recall that $C(\text{good}_f)$ is the collection of paths that converge to a φ -index of f and $C(N)$ is the collection of all converging paths in $\mathcal{T}_{\mathbf{P},f}$. Hence, $C(\text{good}_f) \subseteq C(N)$ and $\text{pr}(C(N)) \geq \text{pr}(C(\text{good}_f)) > 1/(n+1)$. Since $1/(n+1) < \text{pr}(C(N)) \leq 1$, there exists an m , $1 \leq m \leq n$, such that $m = \max(\{i \mid i/(n+1) < \text{pr}(C(N))\})$.

It will be shown that the machine \mathbf{M}_m **Oex**-identifies f . Let us focus on the behavior of \mathbf{M}_m on $f[k]$. \mathbf{M}_m correctly assumes that the probability of converging paths in $\mathcal{T}_{\mathbf{P},f}$ is greater than $m/(n+1)$ and attempts to find, in the limit, a finite collection of nodes where most paths converge. To this end, \mathbf{M}_m first finds the smallest number c_k such that $\sum_{j=1}^{c_k} \text{pr}(C_{j,k}) > m/(n+1)$, and then outputs the canonical index for the finite set $\{\text{guess}(x) \mid 1 \leq x \leq c_k\}$.

Now, by Lemma 6, there exists a finite collection of nodes $\{n_1, n_2, \dots, n_v\}$ in $\mathcal{T}_{\mathbf{P},f}$ such that $\sum_{i=1}^v \text{pr}(C_{n_i}) > m/(n+1)$. This implies that there exists a smallest numbered node s , such that $\sum_{j=1}^s \text{pr}(C_j) > m/(n+1)$. (Choosing any $s \geq \max(\{n_1, n_2, \dots, n_v\})$ satisfies the inequality.)

Now the theorem follows from the following claim.

Claim 5 (a) $(\forall k)[c_k = s]$.

(b) $\{\text{guess}(x) \mid 1 \leq x \leq s\}$ contains a φ -index for f .

It is easy to see that the theorem follows from the above claim. Since \mathbf{M}_m , fed $f[k]$, outputs the canonical index for the finite set $\{\text{guess}(x) \mid 1 \leq x \leq c_k\}$, (a) implies that \mathbf{M}_m converges to the canonical index for $\{\text{guess}(x) \mid 1 \leq x \leq s\}$. And, according to (b) $\{\text{guess}(x) \mid 1 \leq x \leq c_k\}$ contains a φ -index for f , thereby implying that \mathbf{M}_m **Oex**-identifies f . We now prove the claim.

(a) Observe that the choice of s implies that for each $k \geq \text{depth}(s)$, nodes $1, 2, \dots, s$ will be in the partial tree \mathcal{T}_k constructed in Step 1 of $\mathbf{M}_m(f[k])$. Moreover, Lemma 4 implies that $\sum_{j=1}^s \text{pr}(C_{j,k}) \geq \sum_{j=1}^s \text{pr}(C_j) > m/(n+1)$. Hence, $(\forall k)[c_k \leq s]$.

Again, Lemma 4 implies that, for all j , and, for all $k \geq \text{depth}(j)$, $\text{pr}(C_{j,k}) \geq \text{pr}(C_{j,k+1})$. Thus, for all but finitely many k , the sequence $\{c_k\}$ is a nondecreasing one, as c_k is chosen as the smallest value satisfying the inequality $\sum_{j=1}^{c_k} \text{pr}(C_{j,k}) > m/(n+1)$. Now, since $\{c_k\}$ is a nondecreasing sequence bounded above by s , it converges. If $\{c_k\}$ converges to s , we are done. Therefore, let $\{c_k\}$ converge to $s' < s$. Then, for all but finitely many k , $\sum_{j=1}^{s'} \text{pr}(C_{j,k}) > m/(n+1)$. But, then $\sum_{j=1}^{s'} \text{pr}(C_j) > m/(n+1)$ because $\sum_{j=1}^{s'} \text{pr}(C_j)$ is $\lim_{k \rightarrow \infty} \sum_{j=1}^{s'} \text{pr}(C_{j,k})$. This is a contradiction because s is the least integer such that $\sum_{j=1}^s \text{pr}(C_j) > m/(n+1)$. Therefore, $\{c_k\}$ converges to s .

(b) Since good_f is the set of all φ -indexes for f , $N - \text{good}_f$ is the collection of all such φ -indexes that are not for f . Now, observe that since $C(\text{good}_f)$ and $C(N - \text{good}_f)$ are mutually disjoint, $\text{pr}(C(N)) = \text{pr}(C(\text{good}_f)) + \text{pr}(C(N - \text{good}_f))$. We also know by the hypothesis of the theorem and the choice of m that $\text{pr}(C(N)) \leq (m+1)/(n+1)$ and $\text{pr}(C(\text{good}_f)) > 1/(n+1)$. Thus, $\text{pr}(C(N - \text{good}_f)) < m/(n+1)$.

Let I denote the set $\{\text{guess}(x) \mid 1 \leq x \leq s\}$. Observe that $\text{pr}(C(I)) \geq m/(n+1)$. Therefore, at least one element in I must be a correct φ -index for f , because otherwise $I \subseteq N - \text{good}_f$, $C(I) \subseteq C(N - \text{good}_f)$, and $\text{pr}(C(N - \text{good}_f)) \geq m/(n+1)$ — a contradiction. Hence, I contains a φ -index for f .

This completes the proof of the theorem. ■

As an immediate corollary to Theorems 10 and 11, we have the following:

Corollary 2 [31, 32] $(\forall n \geq 1)[\mathbf{Prob}^{\frac{1}{n}}\mathbf{Ex} = \mathbf{Team}_n^1\mathbf{Ex}]$.

The above corollary, together with Corollary 1 implies the following:

Corollary 3 [31, 32] $(\forall n \geq 1)[\mathbf{Prob}^{\frac{1}{n}}\mathbf{Ex} \subseteq \mathbf{Prob}^{\frac{1}{n+1}}\mathbf{Ex}]$.

Thus, the team hierarchy is contained in the probabilistic hierarchy. However, it turns out that the probabilistic hierarchy is no finer than the team hierarchy. To see this, let $1/(n+1) < p \leq 1/n$. Clearly, $\mathbf{Prob}^{\frac{1}{n}}\mathbf{Ex} \subseteq \mathbf{Prob}^p\mathbf{Ex}$. Now, by Theorem 11, $\mathbf{Prob}^p\mathbf{Ex} \subseteq \mathbf{Team}_n^1\mathbf{Ex}$. But, Corollary 2 implies that $\mathbf{Team}_n^1\mathbf{Ex} = \mathbf{Prob}^{\frac{1}{n}}\mathbf{Ex}$. Thus, $\mathbf{Prob}^{\frac{1}{n}}\mathbf{Ex} = \mathbf{Prob}^p\mathbf{Ex}$. We have essentially shown the following corollary which says that the probabilistic hierarchy is exactly the same as the team hierarchy.

Corollary 4 [31, 32] $(\forall n \geq 1)(\forall p)[1/(n+1) < p \leq 1/n \Rightarrow \mathbf{Prob}^p\mathbf{Ex} = \mathbf{Team}_n^1\mathbf{Ex}]$.

We would like to note that counterpart of Corollary 4 is also true when anomalies are allowed in the final program; we direct the reader to Pitt [32] for details.

Corollary 4 can be used to characterize generalized team identification paradigms in which more than one member of the team is required to be successful. In Section 3, results about $\mathbf{Team}_n^1\mathbf{Ex}$ -identification were presented; the rest of the present section is devoted to results about $\mathbf{Team}_n^m\mathbf{Ex}$ -identification.

We start by stating the following notion of interval.

Definition 21 *Let $p \in (0 \dots 1]$. Then $\text{IN}(p)$ is defined to be $1/n$, where n is such that $1/(n+1) < p \leq 1/n$.*

It is easy to verify that for $p \in (0 \dots 1]$, $\text{IN}(p) = 1/\lfloor \frac{1}{p} \rfloor$. The next result is simply a restatement of Corollary 4 using this notion of interval.

Corollary 5 [33] $(\forall p \in (0 \dots 1))[\mathbf{Prob}^p\mathbf{Ex} = \mathbf{Team}_{\text{IN}(p)}^1\mathbf{Ex}]$.

The following result says that all such collections of functions that can be identified by a team of n machines with the requirement that at least m out of n are correct can also be identified by a single probabilistic machine with probability $\frac{m}{n}$.

Theorem 13 [33] $(\forall m, n \in N^+ \mid m \leq n)(\forall p \in (0 \dots 1))[\mathbf{Team}_n^m\mathbf{Ex} \subseteq \mathbf{Prob}^{\frac{m}{n}}\mathbf{Ex}]$.

PROOF. Let $\mathcal{S} \subseteq \mathcal{R}$. Let $\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_n$ be (deterministic) machines witnessing $\mathcal{S} \in \mathbf{Team}_n^m \mathbf{Ex}$. Let \mathbf{P} be a probabilistic machine equipped with an n -sided coin. The behavior of \mathbf{P} can be described thus: \mathbf{P} , before receiving any input, flips its n -sided coin and obtains a number $i \in N_n$. \mathbf{P} then simulates the deterministic machine \mathbf{M}_{i+1} . Clearly, for each $f \in \mathcal{S}$, $\text{pr}(\mathbf{P} \text{ Ex-identifies } f) \geq m/n$. Hence, $\mathcal{S} \in \mathbf{Prob}_n^{\frac{m}{n}} \mathbf{Ex}$. \blacksquare

The next result completely characterizes $\mathbf{Team}_n^m \mathbf{Ex}$ -identification in terms of probabilistic identification.

Theorem 14 [33] $(\forall m, n \in N^+ \mid m \leq n)[\mathbf{Team}_n^m \mathbf{Ex} = \mathbf{Team}_{\lfloor \frac{n}{m} \rfloor}^1 \mathbf{Ex} = \mathbf{Prob}^{\text{IN}(\frac{m}{n})} \mathbf{Ex}]$.

PROOF. We first show that $\mathbf{Team}_{\lfloor \frac{n}{m} \rfloor}^1 \mathbf{Ex} = \mathbf{Prob}^{\text{IN}(\frac{m}{n})} \mathbf{Ex}$. The definition of IN implies that for all $p \in (0 \dots 1]$, $\text{IN}(\text{IN}(p)) = \text{IN}(p)$. Now, by Corollary 5, we have $\mathbf{Prob}^{\text{IN}(\frac{m}{n})} \mathbf{Ex} = \mathbf{Team}_{\frac{1}{\text{IN}(\frac{m}{n})}}^1 \mathbf{Ex} = \mathbf{Team}_{\frac{1}{\text{IN}(\frac{m}{n})}}^1 \mathbf{Ex} = \mathbf{Team}_{\frac{1}{\lfloor \frac{n}{m} \rfloor}}^1 \mathbf{Ex} = \mathbf{Team}_{\lfloor \frac{n}{m} \rfloor}^1 \mathbf{Ex}$.

We now show that $\mathbf{Team}_n^m \mathbf{Ex} = \mathbf{Team}_{\lfloor \frac{n}{m} \rfloor}^1 \mathbf{Ex}$. Since $m \leq n$, Theorem 13 implies that $\mathbf{Team}_n^m \mathbf{Ex} \subseteq \mathbf{Prob}_n^{\frac{m}{n}} \mathbf{Ex}$. Now, observe that $1/(\lfloor \frac{n}{m} \rfloor + 1) < \frac{m}{n} \leq 1/\lfloor \frac{n}{m} \rfloor$. Thus, by Corollary 4, we have $\mathbf{Prob}_n^{\frac{m}{n}} \mathbf{Ex} = \mathbf{Team}_{\lfloor \frac{n}{m} \rfloor}^1 \mathbf{Ex}$. Hence, $\mathbf{Team}_n^m \mathbf{Ex} \subseteq \mathbf{Team}_{\lfloor \frac{n}{m} \rfloor}^1 \mathbf{Ex}$. Now, we only need show that $\mathbf{Team}_{\lfloor \frac{n}{m} \rfloor}^1 \mathbf{Ex} \subseteq \mathbf{Team}_n^m \mathbf{Ex}$. Now observe that for any $c \in N^+$, $\mathbf{Team}_k^1 \mathbf{Ex} \subseteq \mathbf{Team}_{c \cdot k}^c \mathbf{Ex}$. Thus, $\mathbf{Team}_{\lfloor \frac{n}{m} \rfloor}^1 \mathbf{Ex} \subseteq \mathbf{Team}_{m \cdot \lfloor \frac{n}{m} \rfloor}^m \mathbf{Ex}$. Now, since $m \cdot \lfloor \frac{n}{m} \rfloor \leq n$, we have $\mathbf{Team}_{m \cdot \lfloor \frac{n}{m} \rfloor}^m \mathbf{Ex} \subseteq \mathbf{Team}_n^m \mathbf{Ex}$. Therefore, $\mathbf{Team}_{\lfloor \frac{n}{m} \rfloor}^1 \mathbf{Ex} \subseteq \mathbf{Team}_n^m \mathbf{Ex}$. \blacksquare

5 Team and Probabilistic Identification of Languages

Consider the following result about language identification in the limit.

Theorem 15 [18] *Let*

$$\mathcal{L}_1 = \{L \in \mathcal{E} \mid \text{card}(L) < \infty\} \text{ and}$$

$$\mathcal{L}_2 = \{N\}.$$

Then, $\mathcal{L}_1 \in \mathbf{TxtEx}$, $\mathcal{L}_2 \in \mathbf{TxtEx}$, but $\mathcal{L}_1 \cup \mathcal{L}_2 \notin \mathbf{TxtEx}$.

Clearly, the above non-union theorem for \mathbf{TxtEx} motivates the notion of teams for language identification. We introduce team identification and probabilistic identification for languages next.

5.1 Team Identification of Languages

We define team identification of languages.

Definition 22 Let $m, n \in N^+$ and $a \in N \cup \{*\}$.

(a) Let $L \in \mathcal{E}$. A team of n machines $\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_n$ is said to **Team m TxtEx a -identify** L (written: $L \in \mathbf{Team}^m \mathbf{TxtEx}^a(\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_n)$) just in case there exist m distinct numbers $i_1, i_2, \dots, i_m, 1 \leq i_1 < i_2 < \dots < i_m \leq n$, such that each of $\mathbf{M}_{i_1}, \mathbf{M}_{i_2}, \dots, \mathbf{M}_{i_m}$ **TxtEx a -identifies** L .

(b) **Team m TxtEx a** is defined to be the class of sets \mathcal{L} of recursively enumerable languages such that some team of n machines **Team m TxtEx a -identifies** each language in \mathcal{L} .

For the criterion **Team m TxtEx a -identification**, we refer to the fraction $\frac{m}{n}$ as the *success ratio* of the criterion.

5.1.1 Probabilistic Language Identification

Let \mathbf{P} be a probabilistic machine equipped with a t -sided coin and let T be a text for some language $L \in \mathcal{E}$. Then, the probability of \mathbf{P} **TxtEx a -identifying** T is taken to be $\text{pr}_t^\infty(\{O \mid \mathbf{P}^O \mathbf{TxtEx}^a\text{-identifies } T\})$. The next lemma establishes that the set $\{O \mid \mathbf{P}^O \mathbf{TxtEx}^a\text{-identifies } T\}$ is measurable.

Lemma 7 [31] *Let \mathbf{P} be a probabilistic machine and let T be a text. Then $\{O \mid \mathbf{P}^O \mathbf{TxtEx}^a\text{-identifies } T\}$ is measurable.*

The following definition, motivated by the above lemma, introduces probability of identification of a text.

Definition 23 [31] Let T be a text and \mathbf{P} be a probabilistic machine equipped with a t -sided coin ($t \geq 2$). Then, $\text{pr}_t^\infty(\mathbf{P} \mathbf{TxtEx}^a\text{-identifies } T) = \text{pr}_t^\infty(\{O \mid \mathbf{P}^O \mathbf{TxtEx}^a\text{-identifies } T\})$.

As in the case of function identification, there is no loss of generality in assuming a two sided coin.

Lemma 8 (Adopted from [31, 32]) *Let $t, t' > 2$. Let \mathbf{P} be a probabilistic machine with a t -sided coin. Then, there exists a probabilistic machine \mathbf{P}' with a t' -sided coin such that for each text T , $\text{pr}_{t'}^\infty(\mathbf{P}' \mathbf{TxtEx}^a\text{-identifies } T) = \text{pr}_t^\infty(\mathbf{P} \mathbf{TxtEx}^a\text{-identifies } T)$.*

The next definition describes language identification by probabilistic machines. As in the function case, the above lemma frees us from specifying the number of sides of the coin, thereby allowing us to talk about probability function pr_t^∞ without specifying t . For this reason, we will refer to pr_t^∞ as simply pr in the sequel.

Definition 24 [31] Let $0 \leq p \leq 1$.

(a) $\mathbf{P} \mathbf{Prob}^p \mathbf{TxtEx}^a\text{-identifies } L$ (written: $L \in \mathbf{Prob}^p \mathbf{TxtEx}^a(\mathbf{P})$) just in case for each text T for L $\text{pr}(\mathbf{P} \mathbf{TxtEx}^a\text{-identifies } T) \geq p$.

(b) $\mathbf{Prob}^p \mathbf{TxtEx}^a = \{\mathcal{L} \subseteq \mathcal{E} \mid (\exists \mathbf{P})[\mathcal{L} \subseteq \mathbf{Prob}^p \mathbf{TxtEx}^a(\mathbf{P})]\}$.

5.2 Results

In the context of functions, the reader can verify that as a simple consequence of the equivalence of team and probabilistic identification if the success ratio of a team is greater than $\frac{1}{2}$, then the team can be simulated by a single learning machine without any loss in learning power. Such a cut-off ratio is referred to as the *aggregation ratio* of the learning criterion. It is also clear that the only success ratios of interest are of the form $\frac{1}{k}$, $k > 1$. However, the story is completely different for language identification in the limit. First, the aggregation ratio for language identification in the limit turns out to be $\frac{2}{3}$. Second, the notion of team and probabilistic identification are different for languages. In fact, probabilistic identification turns out to be strictly more powerful than team identification. Finally, the results for languages are more difficult to obtain. In what follows, we first present results (with proofs) for team identification of languages with success ratios $\geq \frac{2}{3}$. This is followed by presentation of results for success ratios of the form $\frac{1}{k}$, $k > 2$.

5.2.1 Team Language Identification with Success Ratio $\geq \frac{2}{3}$

We first consider the problem of when can a team of learning machines be simulated by a single learning machine.

As noted above, in the context of function identification, Osherson, Stob, and Weinstein [28] and Pitt and Smith [33] have shown that the collections of functions that can be identified by teams with success ratio *greater than one-half* (that is, a majority of members in the team are required to be successful) are the same as those collections of functions that can be identified by a single machine.

Theorem 16 [28, 33] $(\forall j, k \mid \frac{j}{k} > \frac{1}{2})(\forall a)[\mathbf{Team}_k^j \mathbf{Ex}^a = \mathbf{Ex}^a]$.

Surprisingly, an analog of Theorem 16 for language identification holds for success ratio $\frac{2}{3}$ as opposed to success ratio $\frac{1}{2}$ for function identification. Corollary 6 to Theorem 17 below says that the collections of languages that can be identified by teams with success ratio greater than $\frac{2}{3}$ (that is, more than two-thirds of the members in the team are required to be successful) are the same as those collections of languages which can be identified by a single machine.³ Corollary 7 is a similar result about \mathbf{TxtEx}^* -identification.

Theorem 17 $(\forall j, k \mid \frac{j}{k} > \frac{2}{3})(\forall a)[\mathbf{Team}_k^j \mathbf{TxtEx}^a \subseteq \mathbf{TxtEx}^{\lceil (j+1)/2 \rceil \cdot a}]$.

Corollary 6 $(\forall j, k \mid \frac{j}{k} > \frac{2}{3})[\mathbf{Team}_k^j \mathbf{TxtEx} = \mathbf{TxtEx}]$.

Corollary 7 $(\forall j, k \mid \frac{j}{k} > \frac{2}{3})[\mathbf{Team}_k^j \mathbf{TxtEx}^* = \mathbf{TxtEx}^*]$.

To facilitate the proof of Theorem 17 and other simulation results, we define the following technical notion:

³Corollary 6 also appears in Osherson, Stob, and Weinstein [28], and may also be shown using an argument from Pitt [31] about probabilistic language learning.

Let A^m be a *nonempty finite* multiset of grammars. We define grammar majority(A^m) as follows:

$$W_{\text{majority}(A^m)} = \{x \mid \text{for majority of } g \in A^m, x \in W_g\}.$$

Clearly, majority(A^m) can be defined using the *s-m-n* theorem [35]. Intuitively, majority(A^m) is a grammar for a language that consists of all such elements that are enumerated by a majority of grammars in A^m . Below, whenever we use a set as an argument to majority we assume the argument to be a multiset.

PROOF OF THEOREM 17. Let j, k , and a be as given in the hypothesis of the theorem. Let \mathcal{L} be $\mathbf{Team}_k^j \mathbf{TxtEx}^a$ -identified by the team of machines $\{\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_k\}$. We define a machine \mathbf{M} that $\mathbf{TxtEx}^{\lceil (j+1)/2 \rceil \cdot a}$ -identifies \mathcal{L} .

Let $\text{conv}(\mathbf{M}', \sigma) = \max(\{|\tau| \mid \tau \subseteq \sigma \wedge \mathbf{M}'(\tau) \neq \mathbf{M}'(\sigma)\})$. Let $m_1^\sigma, m_2^\sigma, \dots, m_k^\sigma$ be a permutation of $1, 2, \dots, k$, such that, for $1 \leq r < k$, $[(\text{conv}(\mathbf{M}_{m_r^\sigma}, \sigma), m_r^\sigma) < (\text{conv}(\mathbf{M}_{m_{r+1}^\sigma}, \sigma), m_{r+1}^\sigma)]$.

$$\text{Let } \mathbf{M}(\sigma) = \text{majority}(\{\mathbf{M}_{m_1^\sigma}(\sigma), \mathbf{M}_{m_2^\sigma}(\sigma), \dots, \mathbf{M}_{m_j^\sigma}(\sigma)\}).$$

It is easy to verify that if $\{\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_k\}$ $\mathbf{Team}_k^j \mathbf{TxtEx}^a$ -identify $L \in \mathcal{L}$, then \mathbf{M} $\mathbf{TxtEx}^{\lceil (j+1)/2 \rceil \cdot a}$ -identifies L . ■

The reader is directed to [22] for a better analysis of the errors in the above simulation.

Corollary 8 to Theorem 18 below says that the collections of languages that can be identified by a team with success ratio $2/3$ (that is, at least two-thirds of the members in the team are required to be successful) are the same as those collections of languages that can be identified by a team of three machines at least two of which are required to be successful. Corollary 9 is a similar result about \mathbf{TxtEx}^* -identification with success ratio exactly $2/3$.

Theorem 18 $(\forall j > 0)(\forall a)[\mathbf{Team}_{3j}^{2j} \mathbf{TxtEx}^a \subseteq \mathbf{Team}_3^2 \mathbf{TxtEx}^{(j+1) \cdot a}]$.

Corollary 8 $(\forall j > 0)[\mathbf{Team}_{3j}^{2j} \mathbf{TxtEx} = \mathbf{Team}_3^2 \mathbf{TxtEx}]$.

Corollary 9 $(\forall j > 0)[\mathbf{Team}_{3j}^{2j} \mathbf{TxtEx}^* = \mathbf{Team}_3^2 \mathbf{TxtEx}^*]$.

PROOF OF THEOREM 18. Let j and a be as given in the hypothesis of the theorem. Suppose $\{\mathbf{M}_1, \dots, \mathbf{M}_{3j}\}$ $\mathbf{Team}_{3j}^{2j} \mathbf{TxtEx}^a$ -identify \mathcal{L} . We describe machines $\mathbf{M}'_1, \mathbf{M}'_2$, and \mathbf{M}'_3 such that $\mathcal{L} \subseteq \mathbf{Team}_3^2 \mathbf{TxtEx}^{(j+1) \cdot a}(\{\mathbf{M}'_1, \mathbf{M}'_2, \mathbf{M}'_3\})$.

Let conv be as defined in the proof of Theorem 17. Let $m_1^\sigma, m_2^\sigma, \dots, m_{3j}^\sigma$ be a permutation of $1, 2, \dots, 3j$, such that, for $1 \leq r < 3j$, $[(\text{conv}(\mathbf{M}_{m_r^\sigma}, \sigma), m_r^\sigma) < (\text{conv}(\mathbf{M}_{m_{r+1}^\sigma}, \sigma), m_{r+1}^\sigma)]$.

$$\mathbf{M}'_1(\sigma) = \mathbf{M}_{m_1^\sigma}(\sigma).$$

$$\mathbf{M}'_2(\sigma) = \text{majority}(\{\mathbf{M}_{m_2^\sigma}(\sigma), \mathbf{M}_{m_3^\sigma}(\sigma), \dots, \mathbf{M}_{m_{2j}^\sigma}(\sigma)\}).$$

$$\mathbf{M}'_3(\sigma) = \text{majority}(\{\mathbf{M}_{m_{2j+1}^\sigma}(\sigma), \mathbf{M}_{m_{2j+2}^\sigma}(\sigma), \dots, \mathbf{M}_{m_{3j}^\sigma}(\sigma)\}).$$

Now suppose T is a text for $L \in \mathcal{L}$. Consider the following two cases.

Case 1: At least $2j + 1$ of the machines in $\{\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_{3j}\}$ converge on T .

In this case clearly, $\mathbf{M}'_3 \mathbf{TxtEx}^{(j+1) \cdot a}$ -identifies T . Moreover, $\mathbf{M}'_1 (\mathbf{M}'_2) \mathbf{TxtEx}^{(j+1) \cdot a}$ -identifies T if $\mathbf{M}_{\lim_{s \rightarrow \infty} m_1^{T[s]}} \mathbf{TxtEx}^a$ -identifies T (does not \mathbf{TxtEx}^a -identify T).

Case 2: Not case 1.

In this case clearly, \mathbf{M}'_1 and $\mathbf{M}'_2 \mathbf{TxtEx}^{(j+1) \cdot a}$ identify T . ■

Above proof can be modified to show the following result which says that probabilistic identification of languages with probability of success at least $2/3$ is the same as team identification of languages with success ratio $2/3$.

Theorem 19 $\mathbf{Prob}^{2/3} \mathbf{TxtEx} = \mathbf{Team}_3^2 \mathbf{TxtEx}$.

Theorem 20 below establishes that $2/3$ is indeed the cut-off point at which team identification of languages becomes more powerful than identification by a single machine.

Theorem 20 $\mathbf{Team}_3^2 \mathbf{TxtEx} - \mathbf{TxtEx}^* \neq \emptyset$.

PROOF OF THEOREM 20.

Let $\mathcal{L} = \{L \mid (\exists \text{ distinct } x_1, x_2 \in \{0, 1, 2\})(\text{for } i = 1, 2)[\{y \mid \langle x_i, y \rangle \in L\} \text{ is non-empty and finite and } W_{\max(\{y \mid \langle x_i, y \rangle \in L\})} = L]\}$.

Clearly, $\mathcal{L} \in \mathbf{Team}_3^2 \mathbf{TxtEx}$. Suppose by way of contradiction some machine $\mathbf{M} \mathbf{TxtEx}^*$ -identifies \mathcal{L} . Without loss of generality, assume that \mathbf{M} is order independent [1]. Then, by the operator recursion theorem [3], there exists a 1-1 increasing, nowhere 0, recursive function p such that $W_{p(i)}$'s can be described as follows.

Enumerate $\langle 0, p(0) \rangle$ and $\langle 1, p(1) \rangle$ in both $W_{p(0)}$ and $W_{p(1)}$. Let σ_0 be such that $\text{content}(\sigma_0) = \{\langle 0, p(0) \rangle, \langle 1, p(1) \rangle\}$. Let W_i^s denote W_i enumerated before stage s . Go to stage 1.

Begin {stage s }

1. Enumerate $W_{p(0)}^s \cup W_{p(1)}^s$ in $W_{p(0)}, W_{p(1)}, W_{p(2s)}$, and $W_{p(2s+1)}$.
 Enumerate $\langle 2, p(2s) \rangle$ in $W_{p(0)}, W_{p(2s)}$.
 Enumerate $\langle 2, p(2s+1) \rangle$ in $W_{p(1)}, W_{p(2s+1)}$.
 Let τ_0 be an extension of σ_s such that $\text{content}(\tau_0) = W_{p(0)}$ enumerated till now.
 Let τ_1 be an extension of σ_s such that $\text{content}(\tau_1) = W_{p(1)}$ enumerated till now.
2. Let $x = 0$. Dovetail steps 2a and 2b until, if ever, step 2b succeeds. If and when step 2b succeeds, go to step 3.
 - 2a. Go to substage 0.

- Begin {substage s' }
- Enumerate $\langle 4, x \rangle$ in $W_{p(0)}, W_{p(2s)}$.
 - Enumerate $\langle 5, x \rangle$ in $W_{p(1)}, W_{p(2s+1)}$.
 - Let $x = x + 1$.
 - Go to substage $s' + 1$.
- End {substage s' }
- 2b. Search for $i \in \{0, 1\}$ and $n \in N$ such that $\mathbf{M}(\tau_i \diamond \langle 4+i, 0 \rangle \diamond \langle 4+i, 1 \rangle, \dots, \langle 4+i, n \rangle) \neq \mathbf{M}(\sigma_s)$.
3. If and when 2b succeeds, let i, n be as found in step 2b.
- Let $S =$
- $W_{p(0)}$ enumerated till now
 - $\cup W_{p(1)}$ enumerated till now
 - $\cup \{\langle 4+i, 0 \rangle, \langle 4+i, 1 \rangle, \dots, \langle 4+i, n \rangle\}$.
4. Let $\sigma_{s+1} =$ an extension of $\tau_i \diamond \langle 4+i, 0 \rangle \diamond \langle 4+i, 1 \rangle \diamond \dots \diamond \langle 4+i, n \rangle$ such that $\text{content}(\sigma_{s+1}) = S$.
- Enumerate S in $W_{p(0)}$.
- Go to stage $s + 1$.
- End {stage s }

Consider the following cases:

Case 1: All stages terminate.

In this case, let $L = W_{p(0)} = W_{p(1)} \in \mathcal{L}$. Let $T = \bigcup_s \sigma_s$. Clearly, T is a text for L . But, \mathbf{M} on T makes infinitely many mind changes (since the only way in which infinitely many stages can be completed is by the success of step 2b infinitely often). Thus, \mathbf{M} does not **TextEx***-identify \mathcal{L} .

Case 2: Some stage s starts but does not terminate.

In this case, let $L_1 = W_{p(0)} = W_{p(2s)} \in \mathcal{L}$ and $L_2 = W_{p(1)} = W_{p(2s+1)} \in \mathcal{L}$. Also, L_1, L_2 are infinitely different from each other. Let $T_i = \tau_i \diamond \langle 4+i, 0 \rangle \diamond \langle 4+i, 1 \rangle \diamond \dots \diamond \langle 4+i, n \rangle$, where $i \in \{0, 1\}$ and τ_i is as defined in stage s . Now, \mathbf{M} converges to $\mathbf{M}(\sigma_s)$ for both T_1 and T_2 . Since L_1, L_2 are infinitely different from each other, $W_{\mathbf{M}(\sigma_s)}$ is infinitely different from at least one of L_1 and L_2 . Hence, \mathbf{M} does not **TextEx***-identify at least one of L_1 and L_2 .

From the above cases we have that \mathbf{M} does not **TextEx***-identify \mathcal{L} . ■

5.2.2 Team Language Identification for Success Ratios $\frac{1}{k}$

We first present results for success ratio $\frac{1}{2}$. In the context of functions, the following result immediately follows from Theorem 14.

Theorem 21 [31, 33] $(\forall j > 0)[\mathbf{Team}_{2j}^j \mathbf{Ex} = \mathbf{Team}_2^1 \mathbf{Ex}]$.

This result says that the collections of functions that can be identified by a team with success ratio $1/2$ are the same as those collections of functions that can be identified by a team employing 2 machines and requiring at least 1 to be successful. Consequently, $\mathbf{Team}_2^1 \mathbf{Ex} = \mathbf{Team}_4^2 \mathbf{Ex} = \mathbf{Team}_6^3 \mathbf{Ex} = \dots$, etc.

Surprisingly, in the context of language identification, Theorem 22 below implies that there are collections of languages that can be identified by a team employing 4 machines and requiring at least 2 to be successful, but cannot be identified by any team employing 2 machines and requiring at least 1 to be successful. As a consequence of this result, a direct analog of Pitt's connection for function inference does *not* lift to language learning. A proof of this result can be obtained by a complicated adaptation of the proof of Theorem 20; we direct the reader to [21, 22] for the details.

Theorem 22 $\mathbf{Team}_4^2 \mathbf{TxtEx} - \mathbf{Team}_2^1 \mathbf{TxtEx}^* \neq \emptyset$.

Even more surprising is Corollary 10 to Theorem 23 below which implies that the collections of languages that can be identified by teams employing 6 machines and requiring at least 3 to be successful are exactly the same as those collections of languages that can be identified by teams employing 2 machines and requiring at least 1 to be successful!

Theorem 23 $(\forall j)(\forall i)[\mathbf{Team}_{4j+2}^{2j+1} \mathbf{TxtEx}^i \subseteq \mathbf{Team}_2^1 \mathbf{TxtEx}^{i \cdot (j+1)}]$.

Corollary 10 $(\forall j)[\mathbf{Team}_{4j+2}^{2j+1} \mathbf{TxtEx} = \mathbf{Team}_2^1 \mathbf{TxtEx}]$.

PROOF OF THEOREM 23. Suppose $\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_{4j+2}$ $\mathbf{Team}_{4j+2}^{2j+1} \mathbf{TxtEx}^i$ -identify \mathcal{L} . Let \mathbf{M}'_1 and \mathbf{M}'_2 be defined as follows.

Let conv be as defined in the proof of Theorem 17. Let $m_1^\sigma, m_2^\sigma, \dots, m_{4j+2}^\sigma$ be a permutation of $1, 2, \dots, 4j+2$, such that, for $1 \leq r < 4j+2$, $[(\text{conv}(\mathbf{M}_{m_r^\sigma}, \sigma), m_r^\sigma) < (\text{conv}(\mathbf{M}_{m_{r+1}^\sigma}, \sigma), m_{r+1}^\sigma)]$.

Let $\text{match}(r, \sigma) = \max(\{n \leq |\sigma| \mid \text{card}((\text{content}(\sigma[n]) - W_{r,|\sigma|}) \cup (W_{r,n} - \text{content}(\sigma))) \leq i\})$.

Let $S_\sigma \subseteq [1 \dots 2j+1]$ be the (lexicographically least) set of cardinality j such that, for $1 \leq r, k \leq 2j+1$, $[r \in S_\sigma \wedge k \notin S_\sigma] \Rightarrow [\text{match}(\mathbf{M}_{m_r^\sigma}(\sigma), \sigma) \geq \text{match}(\mathbf{M}_{m_k^\sigma}(\sigma), \sigma)]$.

$$\mathbf{M}'_1(\sigma) = \text{majority}(\{\mathbf{M}_{m_1^\sigma}(\sigma), \mathbf{M}_{m_2^\sigma}(\sigma), \dots, \mathbf{M}_{m_{2j+1}^\sigma}(\sigma)\}).$$

$$\mathbf{M}'_2(\sigma) = \text{majority}(\{\mathbf{M}_{m_{2j+2}^\sigma}(\sigma), \mathbf{M}_{m_{2j+3}^\sigma}(\sigma), \dots, \mathbf{M}_{m_{3j+2}^\sigma}(\sigma)\} \cup \{\mathbf{M}_{m_r^\sigma}(\sigma) \mid r \in S_\sigma\}).$$

It is easy to see that the team $\{\mathbf{M}'_1, \mathbf{M}'_2\}$ witness that $\mathcal{L} \in \mathbf{Team}_2^1 \mathbf{TxtEx}^{i \cdot (j+1)}$. ■

The scenario for team success ratio 1/2 is completely settled by Theorems 24 below. A proof of this result turns out to be very complicated and the reader is directed to [22] for the details.

Theorem 24 $(\forall n, m \in N^+ \mid 2n \text{ does not divide } m)[\mathbf{Team}_{4n}^{2n} \mathbf{TxtEx} - \mathbf{Team}_{2m}^m \mathbf{TxtEx} \neq \emptyset]$.

The following corollary of the above theorem is evident.

Corollary 11 $(\forall m, n \in N^+)[\mathbf{Team}_{2m}^m \mathbf{TxtEx} \subseteq \mathbf{Team}_{2n}^n \mathbf{TxtEx} \Leftrightarrow [m \text{ divides } n \vee m \text{ is odd}]]$.

Theorem 24 can also be used to show the following result which establishes that probabilistic identification of languages with probability of success at least $1/2$ is strictly more powerful than team identification of languages with success ratio $1/2$ (see [24]).

Theorem 25 $\text{Prob}^{1/2}\mathbf{TxtEx} - \bigcup_m \mathbf{Team}_{2^m}^m \mathbf{TxtEx} \neq \emptyset$.

A similar result can be shown for the ratio $1/k, k > 2$ as noted in the following three results. Again we direct the reader to [22] for the details

Theorem 26 $(\forall k \geq 2)(\forall \text{ even } j > 1)(\forall i \mid j \text{ does not divide } i)[\mathbf{Team}_{j,k}^j \mathbf{TxtEx} - \mathbf{Team}_{i,k}^i \mathbf{TxtEx} \neq \emptyset]$.

Corollary 12 $(\forall a \in \mathbb{N})(\forall k \geq 2)(\forall \text{ even } j > 1)(\forall i \mid j \text{ does not divide } i)[\mathbf{Team}_{j,k}^j \mathbf{TxtEx} - \mathbf{Team}_{i,k}^i \mathbf{TxtEx}^a \neq \emptyset]$.

Theorem 27 $(\forall k \geq 2)[\text{Prob}^{1/k}\mathbf{TxtEx} - \bigcup_j \mathbf{Team}_{j,k}^j \mathbf{TxtEx} \neq \emptyset]$.

6 Finite Identification of Functions by Teams

The results considered so far have been about identification in the limit criterion. A more practical learning criterion than learning in the limit is *finite identification* in which a machine, fed the graph of a recursive function f , outputs a program for f as its first conjecture and never abandons this conjecture. Finite identification by a probabilistic machine was first studied by Freivalds [16]. A lot of research activity has taken place in the last few years on finite team identification of functions. For example, Pitt's equivalence between team and probabilistic machines for function identification in the limit does not hold for finite function identification. In the present section, we survey a few preliminary results and refer the reader to a more comprehensive survey by Daley and Kalyanasundaram in the current volume.

We first formally introduce finite identification of functions.

6.1 Finite Function Identification

It is useful to extend the definition of a function learning machine as follows.

Definition 25 [18] A *learning machine* is an algorithmic device that computes a mapping from SEG into $N \cup \{\perp\}$. We further assume that for all $\sigma \subseteq \tau$, if $\mathbf{M}(\sigma) \neq \perp$, then $\mathbf{M}(\tau) \neq \perp$.

Intuitively, \perp is a nonnumeric element that a machine issues to say that it does not wish to conjecture a hypothesis, that is, $\mathbf{M}(\sigma) = \perp$ denotes that \mathbf{M} on σ does not output a conjecture.

We now describe what it means for a learning machine to finite identify a function.

Definition 26 A learning machine \mathbf{M} is said to **Fin-identify** a function f just in case there exists n_0 such that the following hold:

1. for all $n < n_0$, $\mathbf{M}(f[n]) = \perp$;
2. $\varphi_{\mathbf{M}(f[n_0])} = f$; and
3. for all $n \geq n_0$, $\mathbf{M}(f[n]) = \mathbf{M}(f[n_0])$.

If \mathbf{M} **Fin-identifies** f , then we write $f \in \mathbf{Fin}(\mathbf{M})$.

Definition 27 **Fin** denotes the class of all sets \mathcal{S} of recursive functions such that some learning machine **Fin-identifies** each function in \mathcal{S} .

6.2 Finite Function Identification by Teams

Based on the definitions in Smith [36], Pitt and Smith [33], and Freivalds, Smith, and Velauthapillai [17], we have the following definition of finite identification of functions by teams.

Definition 28 Let $m, n \in \mathbb{N}^+$, $m \leq n$. A team of machines $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3, \dots, \mathbf{M}_n$ **Team^mFin-identifies** f (written: $f \in \mathbf{Team}_n^m \mathbf{Fin}(\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3, \dots, \mathbf{M}_n)$) $\Leftrightarrow \text{card}(\{l \mid 1 \leq l \leq n \wedge f \in \mathbf{Fin}(\mathbf{M}_l)\}) \geq m$.

$$\mathbf{Team}_n^m \mathbf{Fin} = \{\mathcal{S} \mid (\exists \mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_n)[\mathcal{S} \subseteq \mathbf{Team}_n^m \mathbf{Fin}(\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_n)]\}.$$

6.3 Results

The results presented in the sequel can be divided into two groups. We refer the reader to Daley and Kalyanasundaram paper in the current volume for additional results. The first group of results is about success ratios greater than $\frac{1}{2}$ and the second group is about success ratio equal to $\frac{1}{2}$.

The following theorem shows that for success ratios greater than two-thirds, there is no advantage in using a team over a single machine. This result, together with the next theorem, implies that the aggregation ratio for finite function identification is $\frac{2}{3}$. This ratio is the same as the aggregation ratio for language identification in the limit and in fact the following two results can be established using techniques similar to those used to prove the corresponding language identification results.

Theorem 28 For all $j, k \in \mathbb{N}$ such that $j > 2k/3$, $\mathbf{Team}_k^j \mathbf{Fin} = \mathbf{Fin}$.

Theorem 29 $\mathbf{Team}_3^2 \mathbf{Fin} - \mathbf{Fin} \neq \emptyset$.

The above result can be generalized to the following; we direct the reader to [26] for details of the proof.

Theorem 30 ($\forall n \in \mathbb{N}$) [$\mathbf{Team}_{2n+3}^{n+2} \mathbf{Fin} - \mathbf{Team}_{2n+1}^{n+1} \mathbf{Fin} \neq \emptyset$].

Daley, Pitt, Velauthapillai, and Will [13] used techniques from Freivalds [16] to prove the following result which shows that Theorem 30 is tight.

Theorem 31 [13] *For all r, s such that $\frac{n+2}{2n+3} < \frac{r}{s} \leq \frac{n+1}{2n+1}$, $\mathbf{Team}_s^r \mathbf{Fin} = \mathbf{Team}_{2n+1}^{n+1} \mathbf{Fin}$.*

We refer the reader to [16, 13] for related results about probabilistic and team finite identification of functions. We now turn our attention to the success ratio $\frac{1}{2}$. Velauthapillai [37] showed that the Pitt's equivalence between probabilistic and team identification in the limit does not hold for finite identification of functions.

Theorem 32 [37] $\mathbf{Team}_4^2 \mathbf{Fin} - \mathbf{Team}_2^1 \mathbf{Fin} \neq \emptyset$.

We refer the reader to [37, 26] for a proof. The scenario at success ratio $\frac{1}{2}$ is settled by the following two results; the details can be found in [20, 26].

Theorem 33 $\mathbf{Team}_{4j+2}^{2j+1} \mathbf{Fin} \subseteq \mathbf{Team}_2^1 \mathbf{Fin}$.

Theorem 34 $\mathbf{Team}_{4j}^{2j} \mathbf{Fin} \subseteq \mathbf{Team}_4^2 \mathbf{Fin}$.

Many additional results have been obtained for finite function identification by teams. We direct the reader to Daley, Pitt, Velauthapillai, and Will [13], and Daley, Kalyanasundaram, and Velauthapillai [10].

The problem of teams for Popperian finite identification of functions is addressed by Daley, Kalyanasundaram, and Velauthapillai [11] and Daley and Kalyanasundaram [9].

Allowing teams of finite learners to make up to a finite number of errors in the hypothesis conjectured has been addressed by Daley, Kalyanasundaram, and Velauthapillai [12].

7 Team and Probabilistic Identification for Other Criteria

The problem of teams and probabilistic machines for identification in the limit with bounded number of mind changes has been addressed by Wiehagen, Freivalds, and Kinber [38] and Daley and Kalyanasundaram [8].

Behaviorally correct function identification has been studied by Daley [6, 7, 14].

In the context of language identification, work has hardly begun on other criteria. We direct the reader to [23, 25] for results on finite, vacillatory, and behaviorally correct identification of languages by teams.

References

- [1] L. Blum and M. Blum. Toward a mathematical theory of inductive inference. *Information and Control*, 28:125–155, 1975.
- [2] M. Blum. A machine independent theory of the complexity of recursive functions. *Journal of the ACM*, 14:322–336, 1967.
- [3] J. Case. Periodicity in generations of automata. *Mathematical Systems Theory*, 8:15–32, 1974.
- [4] J. Case and C. Lynes. Machine inductive inference and language identification. In M. Nielsen and E. M. Schmidt, editors, *Proceedings of the 9th International Colloquium on Automata, Languages and Programming*, volume 140, pages 107–115. Springer-Verlag, Berlin, 1982.
- [5] J. Case and C. Smith. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science*, 25:193–220, 1983.
- [6] R. Daley. On the error correcting power of pluralism in BC-type inductive inference. *Theoretical Computer Science*, 24:95–104, 1983.
- [7] R. P. Daley. Inductive inference hierarchies: Probabilistic vs pluralistic. *Lecture Notes in Computer Science*, 215:73–82, 1986.
- [8] R. P. Daley and B. Kalyanasundaram. Capabilities of probabilistic learners with bounded mind changes. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory, Santa Cruz, California*, pages 182–191. A. C. M. Press, 1993.
- [9] R. P. Daley and B. Kalyanasundaram. Use of reduction arguments in determining popperian fin-type learning capabilities. In K.P. Jantke, S. Kobayashi, E. Tomita, and T. Yokomori, editors, *Proceedings of the Fourth International Workshop on Algorithmic Learning Theory, Tokyo, Japan, Lecture Notes in Artificial Intelligence, No. 744*, pages 173–186. Springer-Verlag, November 1993.
- [10] R. P. Daley, B. Kalyanasundaram, and M. Velauthapillai. Breaking the probability 1/2 barrier in fin-type learning. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory, Pittsburgh, Pennsylvania*, pages 203–217. A. C. M. Press, 1992.
- [11] R. P. Daley, B. Kalyanasundaram, and M. Velauthapillai. The power of probabilism in popperian finite learning. In *Proceedings of the Third International Workshop on Analogical and Inductive Inference, Dagstuhl Castle, Germany*, pages 151–169, October 1992.
- [12] R. P. Daley, B. Kalyanasundaram, and M. Velauthapillai. Capabilities of fallible finite learning. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory, Santa Cruz, California*, pages 199–208. A. C. M. Press, 1993.
- [13] R. P. Daley, L. Pitt, M. Velauthapillai, and T. Will. Relations between probabilistic and team one-shot learners. In L. Valiant and M. Warmuth, editors, *Proceedings of the Workshop on Computational Learning Theory*, pages 228–239. Morgan Kaufmann Publishers, Inc., 1991.
- [14] R.P. Daley. Transformation of probabilistic learning strategies into deterministic learning strategies. In D. Haussler and L. Pitt, editors, *Proceedings of the Workshop on Computational Learning Theory*, pages 157–163. Morgan Kaufmann Publishers, Inc., 1988.

- [15] R. Freivalds. Functions computable in the limit by probabilistic machines. *Mathematical Foundations of Computer Science*, 1975.
- [16] R. Freivalds. Finite identification of general recursive functions by probabilistic strategies. In *Proceedings of the Conference on Algebraic, Arithmetic and Categorical Methods in Computation Theory*, pages 138–145. Akedemie-Verlag, Berlin, 1979.
- [17] R. Freivalds, C. H. Smith, and M. Velauthapillai. Trade-off among parameters affecting inductive inference. *Information and Computation*, 82:323–349, 1989.
- [18] E. M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.
- [19] J. Hopcroft and J. Ullman. *Introduction to Automata Theory Languages and Computation*. Addison-Wesley Publishing Company, 1979.
- [20] S. Jain and A. Sharma. Finite learning by a team. In M. Fulk and J. Case, editors, *Proceedings of the Third Annual Workshop on Computational Learning Theory, Rochester, New York*, pages 163–177. Morgan Kaufmann Publishers, Inc., August 1990.
- [21] S. Jain and A. Sharma. Language learning by a team. In M. S. Paterson, editor, *Proceedings of the 17th International Colloquium on Automata, Languages and Programming*, pages 153–166. Springer-Verlag, July 1990.
- [22] S. Jain and A. Sharma. Computational limits on team identification of languages. Technical Report 9301, School of Computer Science and Engineering; University of New South Wales, 1993.
- [23] S. Jain and A. Sharma. On aggregating teams of learning machines. In K.P. Jantke, S. Kobayashi, E. Tomita, and T. Yokomori, editors, *Proceedings of the Fourth International Workshop on Algorithmic Learning Theory, Tokyo, Japan, Lecture Notes in Artificial Intelligence, No. 744*, pages 150–163. Springer-Verlag, November 1993.
- [24] S. Jain and A. Sharma. Probability is more powerful than team for language identification. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory, Santa Cruz, California*, pages 192–198. ACM Press, July 1993.
- [25] S. Jain and A. Sharma. On aggregating teams of learning machines. Technical Report 9405, School of Computer Science and Engineering; University of New South Wales, 1994.
- [26] S. Jain, A. Sharma, and M. Velauthapillai. Finite identification of function by teams with success ration $1/2$ and above. In preparation, 1994.
- [27] M. Machtey and P. Young. *An Introduction to the General Theory of Algorithms*. North Holland, New York, 1978.
- [28] D. Osherson, M. Stob, and S. Weinstein. Aggregating inductive expertise. *Information and Control*, 70:69–95, 1986.
- [29] D. Osherson, M. Stob, and S. Weinstein. *Systems that Learn, An Introduction to Learning Theory for Cognitive and Computer Scientists*. MIT Press, Cambridge, Mass., 1986.
- [30] D. Osherson and S. Weinstein. Criteria of language learning. *Information and Control*, 52:123–138, 1982.

- [31] L. Pitt. *A characterization of probabilistic inference*. PhD thesis, Yale University, 1984.
- [32] L. Pitt. Probabilistic inductive inference. *Journal of the ACM*, 36:383–433, 1989.
- [33] L. Pitt and C. Smith. Probability and plurality for aggregations of learning machines. *Information and Computation*, 77:77–92, 1988.
- [34] H. Rogers. Gödel numberings of partial recursive functions. *Journal of Symbolic Logic*, 23:331–341, 1958.
- [35] H. Rogers. *Theory of Recursive Functions and Effective Computability*. McGraw Hill, New York, 1967. Reprinted, MIT Press 1987.
- [36] C. Smith. The power of pluralism for automatic program synthesis. *Journal of the ACM*, 29:1144–1165, 1982.
- [37] M. Velauthapillai. Inductive inference with bounded number of mind changes. In *Proceedings of the Workshop on Computational Learning Theory*, pages 200–213, 1989.
- [38] R. Wiehagen, R. Freivalds, and E. B. Kinber. On the power of probabilistic strategies in inductive inference. *Theoretical Computer Science*, 28:111–133, 1984.