Computational Limits on Team Identification of Languages *

Sanjay Jain
Department of Information Systems and Computer Science
National University of Singapore
Singapore 0511, Republic of Singapore
Email: sanjay@iscs.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

Abstract

A team of learning machines is a multiset of learning machines. A team is said to successfully identify a concept just in case each member of some nonempty subset, of predetermined size, of the team identifies the concept. Team identification of programs for computable functions from their graphs has been investigated by Smith. Pitt showed that this notion is essentially equivalent to function identification by a single probabilistic machine.

The present paper introduces, motivates, and studies the more difficult subject of team identification of grammars for languages from positive data. It is shown that an analog of Pitt's result about equivalence of team function identification and probabilistic function identification does not hold for language identification, and the results in the present paper reveal a very complex structure for team language identification. It is also shown that for certain cases probabilistic language identification is strictly more powerful than team language identification.

Proofs of many results in the present paper involve very sophisticated diagonalization arguments. Two very general tools are presented that yield proofs of new results from simple arithmetic manipulation of the parameters of known ones.

Categories and Subject Descriptors: F.1.m [Computation by Abstract Devices]: Miscellaneous; I.2.2 [Artificial Intelligence]: Automatic Programming – program synthesis; I.2.6 [Artificial Intelligence]: Learning – induction
General Terms: Theory
Additional Key Words and Phrases: Inductive Inference

1 Introduction

Identification of grammars (acceptors) for recursively enumerable languages from positive data by a (single) algorithmic device is a well studied problem in Learning Theory. The present paper investigates the computational limits on language identification by a 'team' of (deterministic) machines. A team of machines is a multiset of machines. A team is said to identify a language

*Some preliminary results were reported at the *17th International Colloquium on Automata, Languages and Programming*, Warwick University, July 1990 [18] and at the *Sixth Annual ACM Conference on Computational Learning Theory*, Santa Cruz, July 1993 [19].

if each member of some nonempty subset, of predetermined size, of the team identifies the language.

Identification of programs for functions from their graph is another extensively studied area in Learning Theory. For this related problem, L. Pitt [23, 25] established that team identification is essentially equivalent to identification by a single probabilistic machine. He showed that for any positive integer n and any probability p, if $1/(n+1) < p \leq 1/n$, then the collections of computable functions that can be identified by a single probabilistic machine with probability at least p are exactly the same as the collections of computable functions that can be identified by a team of n (deterministic) machines requiring at least one to be successful.

The present paper makes the following contributions to the study of team identification of languages.

(a) It is shown that an analog of Pitt's connection between probabilistic function and team function identification does not hold for languages. In fact our results show that the structure of team language identification is far more complex than the simple structure of team function identification.

(b) For $k \geq 2$, the relationship between probabilistic language identification with probabilities of the form $1/k$ and team language identification requiring at least $1/k$ of the machines to be successful is established.

(c) Techniques to simplify complicated diagonalization arguments are presented.

(a) follows from our results (for example, Theorem 12 and Theorem 14). Results in Section 5.5 illustrate the complexity of team language identification. We achieve (b) by showing that for $k \geq 2$, probabilistic identification of languages with probability at least $1/k$ is strictly more powerful than team language identification where at least $1/k$ of the members in the team are required to be successful. Proofs of results leading to this answer require very sophisticated diagonalization arguments. Two very general results (Theorems 7 and 8) are presented which allow us to prove new diagonalization theorems by simple arithmetic manipulation of the parameters of known results.

We also suggest that a plausible reason for Pitt's connection not holding for language identification may be the unavailability of negative data (information about what is not in the language) to the learning agent. We argue this by showing that an analog of Pitt's connection does hold for language learning if the learning agent is also given negative information. It should be noted that in the context of function identification, where Pitt's connection holds, negative information is implicitly available to the learning agent because it can eventually determine if a given ordered pair doesn't belong to the graph of a function.

Rest of the paper is organized as follows. Section 2 informally discusses our main results and motivates the study by describing scenarios which are partly modeled by team language learning. Some identification criteria are informally introduced in this section. Section 3 introduces the notation and Section 4 describes the identification criteria formally. Section 5 contains proofs of our results.

2 Discussion

In the present section we informally introduce the definitions and discuss some of our findings. The main subject of our investigation is identification of languages. However, with a view to compare and contrast our results with analogous investigations in the context of function identification, we will present notions from both function identification and language identification. Usually, we will first describe a notion in the context of function identification followed by the description of an analogous notion for language identification.

Learning machines may be thought of as Turing machines computing a mapping from 'finite sequences of data' into computer programs. A typical variable for learning machines is $\mathbf{M}$. At any given time, the input to a learning machine $\mathbf{M}$ is to be construed as a code for the data available to $\mathbf{M}$ until that time. The output of $\mathbf{M}$ is taken to be a hypothesis conjectured by

$\mathbf{M}$ in response to the data available to it. For example, in the context of function learning, the input is an initial segment of the graph of a function and the output is the index of a program in some fixed acceptable programming system. We now describe what it means for a machine to learn a function.

Let N denote the set of natural numbers. Let f be a total function and let $n \in N$. Then, the initial segment of f of length n is denoted $f[n]$. The set of all initial segments of total functions, $\{f[n] \mid f$ is a total function and $n \in N\}$, is denoted SEG. It is easy to see that there exists a computable bijection between SEG and N. Members of SEG are inputs to machines that learn programs for functions, and we avoid notational clutter by using $f[n]$ to denote the code for the initial segment $f[n]$. We also fix an acceptable programming system and the output of a learning machine is interpreted as the index of a program in this system. We say that $\mathbf{M}$ converges on f to i just in case, for all but finitely many n, $\mathbf{M}(f[n]) = i$. The following definition is Gold's criterion for successful identification of functions by learning machines.

Definition 1 [15] (a) $\mathbf{M}$ $\mathbf{Ex}$-identifies f just in case $\mathbf{M}$, fed the graph of f, converges to an index of a program for f. In this case we say that $f \in \mathbf{Ex}(\mathbf{M})$.

(b) $\mathbf{Ex}$ denotes all such collections $\mathcal{S}$ of computable functions such that some machine $\mathbf{Ex}$-identifies each function in $\mathcal{S}$.

The class $\mathbf{Ex}$ is a set theoretic summary of the capability of single machines to $\mathbf{Ex}$-identify collections of functions.

L. Blum and M. Blum [3] and Barzdin [1] showed that the class $\mathbf{Ex}$ is not closed under union. This result may be viewed as a fundamental limitation on building general purpose devices for learning functions, and, to an extent, justifies the use of heuristic methods in Artificial Intelligence. However, this result also suggests a more general criteria of successful learning of functions in which a team of machines is employed and success of the team is the success of any one or more members in the team. The idea of team identification for functions was first suggested by Case and extensively studied by Smith [31, 32]. The next definition describes team identification of functions. Recall that a team of machines is a multiset of machines.

Definition 2 (a) A team of n machines, $\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n\}$, is said to $\mathbf{Team}_n^m\mathbf{Ex}$-identify a function f just in case at least m members in the team $\mathbf{Ex}$-identify f. In this case we say that $f \in \mathbf{Team}_n^m\mathbf{Ex}(\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n\})$.

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

$\mathbf{Team}_n^1\mathbf{Ex}$-identification was investigated by Smith [31, 32] and $\mathbf{Team}_n^m\mathbf{Ex}$-identification was studied by Osherson, Stob, and Weinstein [21]. Pitt [23] noticed an interesting connection between $\mathbf{Team}_n^1\mathbf{Ex}$-identification and function identification by a single probabilistic machine. Probabilistic machines behave very much like computable 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. (For a discussion of probabilistic Turing machines see Gill [14].) The next definition informally describes probabilistic identification of functions; we refer the reader to [25] for detailed discussion on probabilistic identification of functions. Below, $\mathbf{P}$ ranges over probabilistic machines.

Definition 3 [23, 25] Let p be such that $0 \leq p \leq 1$.

(a) $\mathbf{P}$ $\mathbf{Prob}^p\mathbf{Ex}$-*identifies* f just in case $\mathbf{P}$ $\mathbf{Ex}$-identifies f with probability at least p. In this case we say that $f \in \mathbf{Prob}^p\mathbf{Ex}(\mathbf{P})$.

(b) $\mathbf{Prob}^p\mathbf{Ex} = \{\mathcal{S} \mid (\exists \mathbf{P})[\mathcal{S} \subseteq \mathbf{Prob}^p\mathbf{Ex}(\mathbf{P})]\}$.

Pitt [23, 25]showed that if $1/(n+1) < p \leq 1/n$, then $\mathbf{Team}_n^1\mathbf{Ex} = \mathbf{Prob}^p\mathbf{Ex}$. In other words, the collections of computable functions that can be identified by a single probabilistic machine with probability at least p are exactly the same as the collections of computable functions that can be identified by teams of n deterministic machines requiring at least one to be successful.

Using the above connection, Pitt and Smith [26, 27] studied the general case of $\mathbf{Team}_n^m\mathbf{Ex}$-identification[1] in which the criterion of success requires at least m out of n machines to be successful. They showed that for each $m, n > 0$ such that $m \leq n$, $\mathbf{Team}_n^m\mathbf{Ex} = \mathbf{Team}_{\lfloor \frac{n}{m} \rfloor}^1\mathbf{Ex}$.

However, the story is completely different for languages. We next describe preliminary notions about language identification.

Definition 4 A *sequence* σ is a mapping from an initial segment of N into $(N \cup \{\#\})$. The *content* of a sequence σ, denoted $\mathrm{content}(\sigma)$, 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 segment of σ of length n is denoted by $\sigma[n]$.

Intuitively, #'s represent pauses in the presentation of data. SEQ denotes the set of all finite sequences.

We now consider language learning machines.

Definition 5 A *language learning machine* is an algorithmic device that computes a mapping from SEQ into N.

The output of a language learning machine $\mathbf{M}$ on finite sequence σ, denoted $\mathbf{M}(\sigma)$, is interpreted as the index of a program (a grammar) in our fixed acceptable programming system φ.

The set of all finite sequences of natural numbers and #'s, SEQ, can be coded onto N. Thus, we can view these machines as taking natural numbers as input and emitting natural numbers as output. Henceforth, we will refer to language-learning machines as just learning machines, or simply as machines. We let $\mathbf{M}$, with or without decorations, range over learning machines.

Definition 6 A *text* T for a language L is a mapping from N into $(N \cup \{\#\})$ such that L is the set of natural numbers in the range of T. The *content* of a text T, denoted $\mathrm{content}(T)$, is the set of natural numbers in the range of T.

Intuitively, a text for a language is an enumeration or sequential presentation of all the objects in the language with the #'s representing pauses in the listing or presentation of such objects. For example, the only text for the empty language is just an infinite sequence of #'s.

We let T, with or without decorations, range over texts. $T[n]$ denotes the finite initial sequence of T with length n. Hence, $\mathrm{domain}(T[n]) = \{x \mid x < n\}$.

Initial sequences of texts are inputs to machines that learn grammars (acceptors) for r.e. languages. In Definition 7 below we spell out what it means for a learning machine on a text to converge in the limit.

Definition 7 Suppose $\mathbf{M}$ is a learning machine and T is a text. $\mathbf{M}(T)\downarrow$ (read: $\mathbf{M}(T)$ *converges*) $\iff$ $(\exists i)(\overset{\infty}{\forall}\ n)\ [\mathbf{M}(T[n]) = i]$. If $\mathbf{M}(T)\downarrow$, then $\mathbf{M}(T)$ is defined $=$ the unique i such that $(\overset{\infty}{\forall}\ n)[\mathbf{M}(T[n]) = i]$, otherwise we say that $\mathbf{M}(T)$ diverges (written: $\mathbf{M}(T)\uparrow$).

The following definition introduces Gold's criterion for successful identification of languages.

Definition 8 [15]

(a) $\mathbf{M}$ **TxtEx**-identifies a text T just in case $\mathbf{M}$, fed T, converges to a grammar for $\mathrm{content}(T)$.

(b) $\mathbf{M}$ **TxtEx**-identifies an r.e. language L just in case $\mathbf{M}$ **TxtEx**-identifies each text for L. In this case we say that $L \in \mathbf{TxtEx}(\mathbf{M})$.

(c) **TxtEx** denotes all such collections $\mathcal{L}$ of r.e. languages such that some machine **TxtEx**-identifies each language in $\mathcal{L}$.

The class **TxtEx** is a set theoretic summary of the capability of machines to **TxtEx**-identify collections of r.e. languages.

We now define team identification of languages.

[1] The general case of team function identification was also studied by Osherson, Stob, and Weinstein [21].

Definition 9 (a) A team of n machines, $\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n\}$, is said to $\mathbf{Team}_n^m\mathbf{TxtEx}$-identify a text T just in case at least m members in the team $\mathbf{TxtEx}$-identify T.

(b) A team of n machines $\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n\}$ is said to $\mathbf{Team}_n^m\mathbf{TxtEx}$-identify a language L just in case $\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n\}$ $\mathbf{Team}_n^m\mathbf{TxtEx}$-identify each text for L. In this case we write $L \in \mathbf{Team}_n^m\mathbf{TxtEx}(\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n\})$.

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

Probabilistic language identification is the subject of next definition.

Definition 10 [23, 25] Let $0 \leq p \leq 1$.

(a) $\mathbf{P}$ $\mathbf{Prob}^p\mathbf{TxtEx}$-*identifies* L just in case for each text T for L, $\mathbf{P}$ $\mathbf{TxtEx}$-identifies T with probability at least p. In this case we write $L \in \mathbf{Prob}^p\mathbf{TxtEx}(\mathbf{P})$.

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

As already mentioned, the study of team language identification not only turns out to be more difficult than team function identification, but it also has many surprises. Below, we discuss some of these unexpected results.

In the context of function identification, we have the following result immediately following from the results of Pitt and Smith [27].

$$\mathbf{Team}_4^2\mathbf{Ex} = \mathbf{Team}_2^1\mathbf{Ex}$$

The above result says that the collections of functions that can be identified by teams employing 4 machines and requiring at least 2 to be successful are exactly the same as those collections which can be identified by teams employing 2 machines and requiring at least 1 to be successful.

However, in the context of language identification, we are able to show the following result which says that there are collections of languages that can be identified by teams 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. $\supset$ denotes proper superset.

$$\mathbf{Team}_4^2\mathbf{TxtEx} \supset \mathbf{Team}_2^1\mathbf{TxtEx}$$

As a consequence of the above result, which follows from our Theorem 10, an analog of Pitt's connection does not hold for language identification. This fact turns out to be somewhat surprising because many results about function identification were found to have analogous counterparts in the context of language identification. Even more surprising is the following result which follows from our Theorem 11.

$$\mathbf{Team}_6^3\mathbf{TxtEx} = \mathbf{Team}_2^1\mathbf{TxtEx}$$

We actually complete the picture for team language identification for success ratio 1/2 and as a consequence of our results, we have the following result which says that probabilistic language identification with probability at least 1/2 is strictly more powerful than team identification with success ratio 1/2.

$$\mathbf{Prob}^{\frac{1}{2}}\mathbf{TxtEx} - \bigcup_{j \geq 1} \mathbf{Team}_{2j}^{j}\mathbf{TxtEx} \neq \emptyset$$

The above findings are the subject of Section 5.3. Some of our proofs of the above results use two diagonalization tools described in Section 5.2. These tools, presented in the form of very general theorems, allow us to deduce new diagonalization results from simple arithmetic manipulation of the parameters of known diagonalization arguments. For example, Theorem 7 allows us to employ results of the form $\mathbf{Team}_j^i\mathbf{TxtEx} - \mathbf{Team}_l^k\mathbf{TxtEx} \neq \emptyset$ to prove results of the form $\mathbf{Team}_{j'}^{i'}\mathbf{TxtEx} - \mathbf{Team}_{l'}^{k'}\mathbf{TxtEx} \neq \emptyset$ for 'suitable' values of i', j', k', l' obtainable under 'certain conditions' from i, j, k, l.

In Section 5.4, we again employ the tools of Section 5.2 to give partial picture for success ratios of the form $1/k$, $k > 2$. For example, the following result sheds light on when introducing redundancy in the team yields extra language learning ability.

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

As a consequence of the above result, we have the following relationship between probabilistic language identification with probabilities of the form $1/k$ and team language identification.

$$(\forall k \geq 2)[\mathbf{Prob}^{\frac{1}{k}}\mathbf{TxtEx} - \bigcup_{j \geq 1} \mathbf{Team}^{j}_{j \cdot k}\mathbf{TxtEx} \neq \emptyset]$$

Thus, we are able to establish that for probabilities of the form $1/k$, probabilistic language identification is strictly more powerful than team identification where at least $1/k$ of the members in the team are required to be successful.

In Section 5.5, we present results for some other success ratios and shed light on why general results are difficult to obtain.

Finally, in Section 5.6, we address the problem of why Pitt's connection fails for language identification from positive data, and conjecture that a plausible reason for probabilistic and team identification behaving differently for language identification is the unavailability of negative data. In support of this conjecture, we consider a hypothetical learning criteria called **InfEx**-identification. This criteria is like **TxtEx**-identification except that the learning machine is fed an *informant* of the language instead of a text for the language being learned. An informant, unlike a text which only contains information about what is in the language, contains information about both elements and non-elements of the language.[2] We show that an analog of the Pitt's connection holds for probabilistic **InfEx**-identification and team **InfEx**-identification, as they turn out to be essentially the same notions.

Before we undertake a formal presentation of our study, it is worth noting an aspect of team identification that cannot be overlooked, namely, it may not always be possible to determine which members in the team are successful. This property seems to rob team identification of any possible utility. However, we present below scenarios in which the knowledge of which machines are successful is of no consequence, all that matters is some are.

First, consider a hypothetical situation in which an intelligent species, somewhere in outer space, is attempting to contact other intelligent species (such as humans on earth) by transmitting radio signals in some language (most likely alien to humans). Being a curious species ourselves, we would like to establish a communication link with such a species that is trying to reach out. For this purpose, we could employ a team of, not necessarily cooperating, language learners each of which perform the following three tasks in a loop:

(a) receive and examine strings of a language (eg., from a radio telescope);

(b) guess a grammar for the language whose strings are being received;

(c) transmit messages back to outer space based on the grammar guessed in step 2.

If one or more of the learners in the team is actually, but, possibly unknowingly, successful in learning a grammar for the alien language, a correct communication link would be established between the two species.

Consider another scenario in which two countries, A and B, are at war with each other. Country B uses a secret language to transmit movement orders to its troops. Country A, with an intention to confuse the troops of country B, wants to learn a grammar for country B's secret language so that it can transmit conflicting troop movement instructions in that secret language. To accomplish this task, country A employs a "team" of language learners, each of which perform the following three tasks in a loop:

[2] It is worth noting that the notion of informants is merely theoretical, as for any non-recursive r.e. language, the only informants available are non-recursive. We consider informants purely for gaining a theoretical insight about language learning.

(a) receive and examine strings of country B's secret language;

(b) guess a grammar for the language whose strings are being received;

(c) transmit conflicting messages based on the grammar guessed in step 2 (so that B's troops think that these messages are from B's Generals).

If one or more of the learners in the team is actually, but possibly unknowingly, successful in correctly learning a grammar for country B's secret language, then country A achieves its purpose of confusing the troops of country B.

In both the scenarios described above, we have a team of learners trying to infer a grammar for a language from positive data. The team is successful, just in case, some of the learners in the team are successful. It should be noted that the notion of team language identification models only part of the above scenario, as we ignore in our mathematical model the aspect of learners transmitting messages back. We also mathematically ignore possible detrimental effects of a learner guessing an incorrect grammar and transmitting messages that could interfere with messages from a learner that infers a correct grammar (for example, the string 'baby milk powder factory' in one language could mean the string 'ammunition storage' in another!). In no way are these issues trivial; we simply don't have a formal handle on them at this stage.

3 Notation

Recursion-theoretic concepts not explained below are treated in [29]. N denotes the set of natural numbers, $\{0, 1, 2, \ldots\}$. N^+ denotes the set of positive integers, $\{1, 2, 3, \ldots\}$. $\in$, $\subseteq$, and $\subset$ denote, respectively, membership, containment, and proper containment for sets.

$*$ denotes *unbounded* but *finite*; we let $(\forall n \in N)[n < * < \infty]$. Unless otherwise specified, e, i, j, k, l, m, n, r, s, t, u, v, w, x, y, z, with or without decorations[3], range over N. a, b, c, with or without decorations, range over $N \cup \{*\}$. $[m \,..\, n]$ denotes the set $\{i \mid m \leq i \leq n\}$. We say that a pair (i, j) is less than (k, l) iff $[i < k \ \vee \ [i = k \ \wedge \ j < l]]$.

$\emptyset$ denotes the empty set. A, B, C, S, X, Y, Z, with or without decorations, range over subsets of N. We usually denote finite sets by D. Cardinality of a set D is denoted by $\mathrm{card}(D)$. Maximum and minimum of a set S are denoted by $\max(S)$ and $\min(S)$ respectively. By convention, $\min(\emptyset) = \infty$ and $\max(\emptyset) = 0$.

Let η, with or without decorations, range over partial functions. For $a \in (N \cup \{*\})$, we say that η_1 is an a-variant of η_2 (written $\eta_1 =^a \eta_2$) just in case $\mathrm{card}(\{x \mid \eta_1(x) \neq \eta_2(x)\}) \leq a$. For example, $\eta_1 =^* \eta_2$ means that η_1 and η_2 are finite variants. If $\mathrm{card}(\{x \mid \eta_1(x) \neq \eta_2(x)\}) \not\leq a$, then we say that η_1 is not an a-variant of η_2 (written $\eta_1 \neq^a \eta_2$).

$\langle i, j \rangle$ stands for an arbitrary computable one to one encoding of all pairs of natural numbers onto N [29]. Corresponding projection functions are π_1 and π_2. $(\forall i, j \in N)$ $[\pi_1(\langle i, j \rangle) = i$ and $\pi_2(\langle i, j \rangle) = j$ and $\langle \pi_1(x), \pi_2(x) \rangle = x$]. Similarly, $\langle i_1, i_2, \ldots, i_n \rangle$ denotes a computable one to one encoding of all n-tuples onto N.

The set of all total recursive functions of one variable is denoted by $\mathcal{R}$. f ranges over $\mathcal{R}$. In some situations q, g range over $\mathcal{R}$; in other situations q, g range over N. In some situations p ranges over $\mathcal{R}$; in other situations p is a real number (construed as a probability). For a partial recursive function η, $\mathrm{domain}(\eta)$ denotes the domain of η and $\mathrm{range}(\eta)$ denotes the range of η. $\eta(x)\downarrow$ iff $x \in \mathrm{domain}(\eta)$; $\eta(x)\uparrow$ otherwise.

$\mathcal{E}$ denotes the class of all recursively enumerable languages. L, with or without decorations, ranges over $\mathcal{E}$. $\mathcal{L}$, with or without decorations, ranges over subsets of $\mathcal{E}$. We call the set $\{\langle x, y \rangle \mid \langle x, y \rangle \in L\}$, the x-th cylinder of L. $L_1 \mathbf{\Delta} L_2$ denotes $(L_1 - L_2) \cup (L_2 - L_1)$. For $a \in N \cup \{*\}$, we say that L_1 is an a variant of L_2 (written: $L_1 =^a L_2$) iff $\mathrm{card}(L_1 \mathbf{\Delta} L_2) \leq a$.

φ denotes a standard acceptable programming system (also referred to as standard acceptable numbering) [28, 29]. φ_i denotes the partial recursive function computed by the i^{th} program in the standard acceptable programming system φ. W_i denotes the domain of φ_i. W_i is, then, the

[3]Decorations are subscripts, superscripts, primes and the like.

r.e. set/language ($\subseteq N$) accepted by φ-program i. We can (and do) also think of i as (coding) a (type 0 [16]) grammar for generating W_i. Φ denotes an arbitrary Blum complexity measure [4] for φ. $W_{i,n}$ denotes the set $\{x \leq n \mid \Phi_i(x) \leq n\}$.

The quantifiers '$\stackrel{\infty}{\forall}$' and '$\stackrel{\infty}{\exists}$' mean 'for all but finitely many' and 'there exists infinitely many', respectively.

We let σ, τ, and γ, with or without decorations, range over finite sequences. $\sigma \diamond \tau$ denotes concatenation of σ and τ. We sometimes abuse notation slightly, and use $\sigma_1 \diamond k$ to denote the *concatenation* of k at the end of sequence σ_1; thus $\sigma = \sigma_1 \diamond k$ is defined as follows:

$$\sigma(x) = \begin{cases} \sigma_1(x), & \text{if } x < |\sigma_1|; \\ k, & \text{if } x = |\sigma_1|; \\ \uparrow, & \text{otherwise.} \end{cases}$$

4 Definitions

4.1 Language Identification

Definition 11 [15, 6, 22] Let $a \in N \cup \{*\}$.
(i) $\mathbf{M}$ $\mathbf{TxtEx}^a$-*identifies* $T \Leftrightarrow [\mathbf{M}(T)\downarrow$ and $W_{\mathbf{M}(T)} =^a \text{content}(T)]$.
(ii) $\mathbf{M}$ $\mathbf{TxtEx}^a$-*identifies* L (written: $L \in \mathbf{TxtEx}^a(\mathbf{M})$) $\iff$ $\mathbf{M}$ $\mathbf{TxtEx}^a$-identifies each text for L.
(iii) $\mathbf{TxtEx}^a = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{TxtEx}^a(\mathbf{M})]\}$.

Definition 12 Let a learning machine $\mathbf{M}$ and language L be given. σ is said to be a *stabilizing sequence* for $\mathbf{M}$ on L just in case the following hold:

(a) content$(\sigma) \subseteq L$, and

(b) $(\forall \tau \mid \text{content}(\tau) \subseteq L)$ $[\mathbf{M}(\sigma) = \mathbf{M}(\sigma \diamond \tau)]$.

Definition 13 [3] σ is called a $\mathbf{TxtEx}^a$-*locking sequence* for $\mathbf{M}$ on L just in case $W_{\mathbf{M}(\sigma)} =^a L$ and σ is a stabilizing sequence for $\mathbf{M}$ on L.

Lemma 1 *[3] If* $\mathbf{M}$ $\mathbf{TxtEx}^a$*-identifies L, then there exists a* $\mathbf{TxtEx}^a$*-locking sequence for* $\mathbf{M}$ *on L.*

4.2 Team Identification

A team of learning machines is any multiset of learning machines. We let $\mathcal{M}$, with or without decorations, range over teams of machines. In describing teams of machines, we use the notation for sets with the understanding that these sets are to be treated as multisets. Also, set operations, $\cup$, $\cap$, $\subset$, set difference, etc., on teams result in multiset of machines.

Definition 14 introduces team identification of languages.

Definition 14 Let $m, n \in N^+$ and $a \in N \cup \{*\}$.

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

(b) Let $L \in \mathcal{E}$. A team of n machines $\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n\}$ is said to $\mathbf{Team}_n^m\mathbf{TxtEx}^a$-*identify* L (written: $L \in \mathbf{Team}_n^m\mathbf{TxtEx}^a(\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n\})$) just in case $\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n\}$ $\mathbf{Team}_n^m\mathbf{TxtEx}^a$-identify each text for L.

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

For $\mathbf{Team}_n^m\mathbf{TxtEx}^a$-identification criteria, we refer to the fraction m/n as the *success ratio* of the criteria. In the following, for $i > j$, we take $\mathbf{Team}_j^i\mathbf{TxtEx}^a = \{\emptyset\}$.

Note that in the above definition we have allowed the possibility that for a given language L, different machines in the team may be successful on different texts for L. The following definition describes an alternative formulation in which successful machines in the team are required to be successful on all texts for L.

Definition 15 (a) A team of n machines, $\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n\}$, $\mathbf{Lteam}_n^m\mathbf{TxtEx}^a$-identifies a language L just in case at least m members in the team $\mathbf{TxtEx}^a$-identify L.

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

The next proposition shows that the above two formulations of team language are equivalent.

Proposition 1 *Let m, n such that $m \leq n$ be given. Then,* $\mathbf{Team}_n^m\mathbf{TxtEx}^a = \mathbf{Lteam}_n^m\mathbf{TxtEx}^a$.

PROOF. (Proposition 1) Clearly, $\mathbf{Lteam}_n^m\mathbf{TxtEx}^a \subseteq \mathbf{Team}_n^m\mathbf{TxtEx}^a$. A proof of the other direction requires Fulk's [12, 13] adaptation of the technical machinery first introduced by Blum and Blum [3].

We now show that $\mathbf{Team}_n^m\mathbf{TxtEx}^a \subseteq \mathbf{Lteam}_n^m\mathbf{TxtEx}^a$. Suppose $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n$ are given. We construct $\mathbf{M}'_1, \mathbf{M}'_2, \ldots$ such that

$(\forall L)[(\forall$ texts T for $L)[\mathrm{card}(\{i \mid 1 \leq i \leq n \ \wedge\ \mathbf{M}_i(T)\!\downarrow\ \wedge\ W_{\mathbf{M}_i(T)} = L\}) \geq m] \Rightarrow \mathrm{card}(\{i \mid 1 \leq i \leq n \ \wedge\ L \in \mathbf{TxtEx}^a(\mathbf{M}'_i)\}) \geq m]$

Given, L, let σ_1^L, σ_2^L, …, σ_n^L, be as follows (some of them may be undefined). If σ_i^L does not get defined then, σ_{i+1}^L also does not get defined. Along with σ_i^L we also define m_i^L, for $1 \leq i \leq n$.

Let σ be the lexicographically least sequence, if any, such that

(a) $\mathrm{content}(\sigma) \subseteq L$, and

(b) there exists a $j \in \{1, \ldots, n\}$, such that σ is a stabilizing sequence for $\mathbf{M}_j$ on L.

If such a (lexicographically least) σ exists, then let $\sigma_1^L = \sigma$, and $m_1^L = j$, as in (b).

Now suppose σ_i^L has been defined (note that if σ_i^L does not get defined then σ_{i+1}^l does not get defined). Then, let σ be the lexicographically least extension of σ_i^L, if any, such that

(a') $\mathrm{content}(\sigma) \subseteq L$, and

(b') there exists a $j \in (\{1, \ldots, n\} - \{m_1^L, m_2^L, \ldots, m_i^L\})$, such that σ is a stabilizing sequence for $\mathbf{M}_j$ on L.

If such a (lexicographically least) σ exists, then let $\sigma_{i+1}^L = \sigma$, and $m_{i+1}^L = j$, as in (b').

It is easy to see that if σ_i^L is defined, then σ_i^L and m_i^L can be determined in the limit from a text T for L. Note that this is possible since, if σ is not a locking sequence for $\mathbf{M}$ on L, then one can determine so in the limit (from a text for L). This allows one to determine σ_1^L, then σ_2^L, …, in the limit.

Now we describe the behaviour of $\mathbf{M}'_i$ on text T for L

$\mathbf{M}'_i(T) = \mathbf{M}_{m_i^L}(\sigma_i^L)$, if σ_i^L is defined. $\mathbf{M}'_i(T)$ is undefined otherwise.

Now, consider any L.

Let $i \leq n$, be the largest number such that σ_i^L, is defined. Let T be a text for L such that $\sigma_i^L \subseteq T$, and, for all $j \in (\{1, \ldots, n\} - \{m_1, m_2, \ldots, m_i\})$, $\mathbf{M}_j(T)\!\uparrow$. Note that there exists such a T, since, for all $j \in (\{1, \ldots, n\} - \{m_1, m_2, \ldots, m_i\})$, there does not exist a stabilizing sequence extending σ_i^L, for $\mathbf{M}_j$ on L. Now, on all texts T' for L, for $1 \leq l \leq i$, $\mathbf{M}'_l(T') = \mathbf{M}_{m_l^L}(T)$. Also, note that for $1 \leq l \leq n$, such that $l \notin \{m_1, m_2, \ldots, m_i\}$, $\mathbf{M}_l(T)\!\uparrow$.

It immediately follows that, $[\mathrm{card}(\{i \mid 1 \leq i \leq n \ \wedge\ \mathbf{M}_i(T)\!\downarrow\ \wedge\ W_{\mathbf{M}_i(T)} = L\}) \geq m] \Rightarrow [\mathrm{card}(\{i \mid 1 \leq i \leq n \ \wedge\ L \in \mathbf{TxtEx}^a(\mathbf{M}'_i)\}) \geq m]$. ∎

In the sequel, we only consider $\mathbf{Team}_n^m\mathbf{TxtEx}^a$-identification.

4.3 Probabilistic Identification

A probabilistic learning machine may be thought of as an algorithmic device which has the added ability of basing its actions on the outcome of a random event like a coin flip. More precisely, let t be a positive integer greater that 1. Then, a probabilistic machine $\mathbf{P}$ may be construed as an algorithmic machine that is equipped with a t-sided coin. The response of $\mathbf{P}$ to input σ not only depends upon σ but also on the outcomes of coin flips performed by $\mathbf{P}$ while processing σ. We refer the reader to Pitt [24, 25] for details of probabilistic learning machines.

Let N_m denote the set $\{0, 1, 2, \ldots, m-1\}$. An oracle for a t-sided coin, $t > 1$, also referred to as a t-ary oracle, is an infinite sequence of integers $i_0, i_1, i_2, \ldots$ such that for each $j \in N$, $i_j \in N_t$. (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. Let O be a t-ary oracle and let $\mathbf{P}$ be a probabilistic learning machine. Then $\mathbf{P}^O$ denotes a learner that 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.

We now describe a probability measure on a single coin flip. For a t-sided coin, let $(N_t, \mathcal{B}_t, \mathrm{pr}_t)$ be a probability space on the sample space N_t, where $\mathcal{B}_t$ is the Borel field $\{S \mid S \subseteq N_t\}$ and $\mathrm{pr}_t = \mathrm{card}(S)/t$. We employ this measure to describe a probability measure on t-ary oracles next.

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 $\mathcal{B}_t^\infty$ 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 $(N_t^\infty, \mathcal{B}_t^\infty, \mathrm{pr}_t^\infty)$ be a probability space where pr_t^∞ is defined as follows.

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

Clearly, sets $A_{i_1, i_2, i_3, \ldots, i_n}$ are measurable (i.e. are members of $\mathcal{B}_t^\infty$) [2].

4.3.1 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 $\mathbf{P}$ $\mathbf{TxtEx}^a$-identifying T is taken to be $\mathrm{pr}_t^\infty(\{O \mid \mathbf{P}^O \mathbf{TxtEx}^a\text{-identifies } T\})$. The next lemma establishes that the set $\{O \mid \mathbf{P}^O \mathbf{TxtEx}^a\text{-identifies } T\}$ is measurable.

Lemma 2 [24] *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 16 [24] Let T be a text and $\mathbf{P}$ be a probabilistic machine equipped with a t-sided coin ($t \geq 2$). Then, $\mathrm{pr}_t^\infty(\mathbf{P}\ \mathbf{TxtEx}^a\text{-identifies } T) = \mathrm{pr}_t^\infty(\{O \mid \mathbf{P}^O\ \mathbf{TxtEx}^a\text{-identifies } T\})$.

There is no loss of generality in assuming a two sided coin.

Lemma 3 (Adopted from [24, 25]) *Let $t, t' \geq 2$. 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, $\mathrm{pr}_{t'}^\infty(\mathbf{P}'\ \mathbf{TxtEx}^a\text{-identifies } T) = \mathrm{pr}_t^\infty(\mathbf{P}\ \mathbf{TxtEx}^a\text{-identifies } T)$.*

The next definition describes language identification by probabilistic machines. 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 17 [24] Let $0 \leq p \leq 1$.

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

(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 Results

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

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

In the context of function identification, Osherson, Stob, and Weinstein [21] and Pitt and Smith [27] 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 1 [21, 27] $(\forall j, k \mid \frac{j}{k} > \frac{1}{2})[\mathbf{Team}_k^j\mathbf{Ex} = \mathbf{Ex}]$.

An analog of Theorem 1 for language identification holds for success ratio 2/3 as opposed to success ratio 1/2 for function identification. Corollary 1 to Theorem 2 below says that the collections of languages that can be identified by teams with success ratio greater than 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.[4] Corollary 2 is a similar result about $\mathbf{TxtEx}^*$-identification.

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

Corollary 1 $(\forall j, k \mid \frac{j}{k} > \frac{2}{3})[\mathbf{Team}_k^j\mathbf{TxtEx} = \mathbf{TxtEx}]$.

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

To facilitate the proof of Theorem 2 and other simulation results, we define the following technical notion:

We define grammar majority$(g_1, g_2, \ldots, g_k)$ as follows:

$$W_{\text{majority}(g_1, g_2, \ldots, g_k)} = \{x \mid \text{card}(\{i \mid 1 \leq i \leq k \ \wedge \ x \in W_{g_i}\}) > k/2\}$$

Clearly, majority$(g_1, g_2, \ldots, g_k)$ can be defined using the *s-m-n* theorem [30]. Intuitively, majority$(g_1, g_2, \ldots, g_k)$ is a grammar for a language that consists of all such elements that are enumerated by a majority of grammars in $g_1, g_2, \ldots, g_k$.

Proof of Theorem 2. 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 $\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_k\}$. We define a machine $\mathbf{M}$ that $\mathbf{TxtEx}^{\lceil (j+1)/2 \rceil \cdot a}$-identifies $\mathcal{L}$.

For a finite sequence σ, a text T and a machine $\mathbf{M}'$, let

$$\text{conv}(\mathbf{M}', \sigma) = \max(\{|\tau| \mid \tau \subseteq \sigma \ \wedge \ \mathbf{M}'(\tau) \neq \mathbf{M}'(\sigma)\})$$

$$\text{conv}(\mathbf{M}', T) = \max(\{n \mid \mathbf{M}'(T[n]) \neq \mathbf{M}'(T[n+1])\})$$

Let $m_1^\sigma, m_2^\sigma, \ldots, m_k^\sigma$ be a permutation of $1, 2, \ldots, k$, such that, for $1 \leq r < k$, $[(\text{conv}(\mathbf{M}_{m_r^\sigma}, \sigma), m_r^\sigma) < (\text{conv}(\mathbf{M}_{m_{r+1}^\sigma}, \sigma), m_{r+1}^\sigma)]$. The reader should note that the "$<$" in the previous expression refers to ordering on pairs.

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

We first show that if $\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_k\}$ $\mathbf{Team}_k^j\mathbf{TxtEx}^a$-identifies $L \in \mathcal{L}$, then $\mathbf{M}$ $\mathbf{TxtEx}^*$-identifies L. We will then prove the bound on errors. So suppose $\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_k\}$ $\mathbf{Team}_k^j\mathbf{TxtEx}^a$-identifies $L \in \mathcal{L}$. Suppose T is a text for L. Now at least j of the machines in $\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_k\}$ $\mathbf{TxtEx}^a$-identify L. In particular we have at least j of the machines

[4] Corollary 1 also appears in Osherson, Stob, and Weinstein [21], and may also be shown using an argument from Pitt [24] about probabilistic language learning. The expression "aggregation" is used to denote the simulation of a team by a single machine; see [20] for aggregation results for other identification criteria.

in $\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_k\}$ converge on T. Thus, for $1 \leq r \leq j$, $\lim_{n\to\infty} m_r^{T[n]}\downarrow$ (to say m_r) and $\mathbf{M}_{m_r}(T)\downarrow$. Moreover, $\mathrm{card}(\{r \mid 1 \leq r \leq j \ \wedge\ W_{\mathbf{M}_{m_r}(T)} =^a L\}) \geq j-(k-j) = 2j-k > \frac{j}{2}$ (since at most $k-j$ of the machines in $\{\mathbf{M}_1, \ldots, \mathbf{M}_k\}$ do not $\mathbf{TxtEx}^a$-identify L). It thus follows that $\mathbf{M}(T)\downarrow = \mathrm{majority}(\mathbf{M}_{m_1}(T), \mathbf{M}_{m_2}(T), \ldots, \mathbf{M}_{m_j}(T))$, is a grammar for $*$-variant of L.

To see the bound on errors, consider a text T for a language $L \in \mathcal{L}$. Note that each error committed by the final grammar output by $\mathbf{M}$ on T is also committed by the final grammars of at least $\lceil (j+1)/2 \rceil$ of the j earliest converging machines (on T) in the team $\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_k\}$. Note that at least $\lceil (j+1)/2 \rceil$ of the j earliest converging machines $\mathbf{TxtEx}^a$-identify L. Thus the errors committed by the final grammar of $\mathbf{M}$ is bounded by $\lceil (j+1)/2 \rceil \cdot a$. Thus, if $\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_k\}$ $\mathbf{Team}_k^j\mathbf{TxtEx}^a$-identify $L \in \mathcal{L}$, then $\mathbf{M}$ $\mathbf{TxtEx}^{\lceil (j+1)/2 \rceil \cdot a}$-identifies L. ∎

A slightly better analysis of the errors committed by the simulation given in the above proof shows that

Theorem 3 $(\forall j, k \mid j > 2k/3)(\forall a \in (N \cup \{*\}))[\mathbf{Team}_k^j\mathbf{TxtEx}^a \subseteq \mathbf{TxtEx}^{\lfloor \frac{2j-k}{\lceil (3j-2k-1)/2 \rceil} \cdot a \rfloor}]$.

Corollary 3 to Theorem 4 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 4 is a similar result about $\mathbf{TxtEx}^*$-identification with success ratio exactly 2/3.

Theorem 4 $(\forall j > 0)(\forall a \in N \cup \{*\})[\mathbf{Team}_{3j}^{2j}\mathbf{TxtEx}^a \subseteq \mathbf{Team}_3^2\mathbf{TxtEx}^{(j+1)\cdot a}]$.

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

Corollary 4 $(\forall j > 0)[\mathbf{Team}_{3j}^{2j}\mathbf{TxtEx}^* = \mathbf{Team}_3^2\mathbf{TxtEx}^*]$.

Proof of Theorem 4. Let j and a be as given in the hypothesis of the theorem. Suppose $\{\mathbf{M}_1, \ldots, \mathbf{M}_{3j}\}$ $\mathbf{Team}_{3j}^{2j}\mathbf{TxtEx}^k$-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 2. Let $m_1^\sigma, m_2^\sigma, \ldots, m_{3j}^\sigma$ be a permutation of $1, 2, \ldots, 3j$, such that, for $1 \leq r < 3j$, $[(\mathrm{conv}(\mathbf{M}_{m_r^\sigma}, \sigma), m_r^\sigma) < (\mathrm{conv}(\mathbf{M}_{m_{r+1}^\sigma}, \sigma), m_{r+1}^\sigma)]$.

$$\mathbf{M}_1'(\sigma) = \mathbf{M}_{m_1^\sigma}(\sigma).$$
$$\mathbf{M}_2'(\sigma) = \mathrm{majority}(\mathbf{M}_{m_2^\sigma}(\sigma), \mathbf{M}_{m_3^\sigma}(\sigma), \ldots, \mathbf{M}_{m_{2j}^\sigma}(\sigma)).$$
$$\mathbf{M}_3'(\sigma) = \mathrm{majority}(\mathbf{M}_{m_1^\sigma}(\sigma), \mathbf{M}_{m_2^\sigma}(\sigma), \ldots, \mathbf{M}_{m_{2j+1}^\sigma}(\sigma)).$$

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 $\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_{3j}\}$ converge on T.

In this case, in a way similar to that used in the proof of Theorem 2, we can show that $\mathbf{M}_3'$ $\mathbf{TxtEx}^{(j+1)\cdot a}$-identifies T. Moreover, $\mathbf{M}_1'$ (respectively, $\mathbf{M}_2'$) $\mathbf{TxtEx}^{(j+1)\cdot a}$-identifies T if $\mathbf{M}_{\lim_{s\to\infty} m_1^{T[s]}}$ $\mathbf{TxtEx}^a$-identifies T (respectively, 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 5 $\mathbf{Prob}^{2/3}\mathbf{TxtEx} = \mathbf{Team}_3^2\mathbf{TxtEx}$.

Theorem 6 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 6 $\mathbf{Team}_3^2\mathbf{TxtEx} - \mathbf{TxtEx}^* \neq \emptyset$.

Proof of Theorem 6.

Let $\mathcal{L} = \{L \mid \mathrm{card}(\{i \leq 2 \mid \max(\{y \mid \langle i, y\rangle \in L\}) = y_0 < \infty \text{ and } W_{y_0} = L\}) \geq 2\}$.

We now show that $\mathcal{L} \in \mathbf{Team}_3^2\mathbf{TxtEx}$. Consider a team consisting of three machines $\mathbf{M}_0$, $\mathbf{M}_1$, and $\mathbf{M}_2$. For $0 \leq i \leq 2$, machine $\mathbf{M}_i$ behaves as follows: On $T[n]$, $\mathbf{M}_i$, outputs the maximum y, if any, such that $\langle i, y\rangle \in \mathrm{content}(T[n])$. It is easy to verify that if T is a text for some language in $\mathcal{L}$, then at least two of the machines will converge in the limit to a grammar for content(T). Thus $\mathcal{L} \in \mathbf{Team}_3^2\mathbf{TxtEx}$.

We now show that $\mathcal{L} \notin \mathbf{TxtEx}^*$. Suppose by way of contradiction that some machine $\mathbf{M}$ $\mathbf{TxtEx}^*$-identifies $\mathcal{L}$. Without loss of generality, we assume that $\mathbf{M}$ is order independent [3]. We then show that there exists a language in $\mathcal{L}$ that $\mathbf{M}$ fails to $\mathbf{TxtEx}^*$-identify. The description of this witness proceeds in stages and uses the operator recursion theorem [5]. We first give an informal description of the staging construction, as more complicated versions of this idea are used in some later proofs.

The construction uses a sequence of grammars $p(0), p(1), p(2), \ldots$ defined using the operator recursion theorem. Two grammars $p(0)$ and $p(1)$ play a special role in the construction. Initially $p(0)$ and $p(1)$ are coded into $W_{p(0)}$ and $W_{p(1)}$. This is to ensure that if infinitely many stages are executed, then the language enumerated by $p(0)$ (which would be the same as language enumerated by $p(1)$) is in $\mathcal{L}$.

In stage s, $p(0)$ (in cooperation with $p(2s)$) and $p(1)$ (in cooperation with $p(2s+1)$) try to enumerate two potentially infinitely distinct languages in $\mathcal{L}$. This is achieved by "growing" distinct cylinders infinitely often. Simultaneously an attempt is made to find if $\mathbf{M}$ changes its mind either on $W_{p(0)}$ defined so far or on $W_{p(1)}$ defined so far. If a mind change is found, then both $W_{p(0)}$ and $W_{p(1)}$ defined until the end of stage s are made equal and the next stage is executed.

Now if an attempt to find a mind change is successful at every stage, then both $W_{p(0)}$ and $W_{p(1)}$ are equal and belong to $\mathcal{L}$. But then $\mathbf{M}$ makes infinitely many mind changes on a text for this language and hence fails to $\mathbf{TxtEx}^*$-identify it.

On the other hand, if some stage s starts but does not finish, then $W_{p(0)} (= W_{p(2s)})$ and $W_{p(1)} (= W_{p(2s+1)})$ are two infinitely distinct languages in $\mathcal{L}$. But, $\mathbf{M}$ converges to the same grammar (on some text) for each of these languages, and hence it fails to $\mathbf{TxtEx}^*$-identify at least one of them.

We now proceed formally.

By the operator recursion theorem [5], there exists a 1-1 increasing, nowhere 0, recursive function p such that the $W_{p(i)}$'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 $\mathrm{content}(\sigma_0) = \{\langle 0, p(0)\rangle, \langle 1, p(1)\rangle\}$. Let $W_{p(i)}^s$ denote $W_{p(i)}$ enumerated before stage s. Go to stage 1.

Begin {stage s}

Invariant: $W_{p(0)}^s = W_{p(1)}^s = \mathrm{content}(\sigma_s)$.

1. Enumerate $W_{p(0)}^s$ into $W_{p(2s)}$ 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 $\mathrm{content}(\tau_0) = [W_{p(0)}$ enumerated until now].
   Let τ_1 be an extension of σ_s such that $\mathrm{content}(\tau_1) = [W_{p(1)}$ enumerated until now].
2. Set $x = 0$. 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 $W_{p(1)}, W_{p(2s+1)}$.
   Set $x = x + 1$.
   Go to substage $s' + 1$.
   End {substage s'}

2b. Search for $i \in \{0,1\}$ and $n \in N$ such that $\mathbf{M}(\tau_i \diamond \langle 4+i,0\rangle \diamond \langle 4+i,1\rangle \diamond \ldots \diamond \langle 4+i,n\rangle) \neq \mathbf{M}(\sigma_s)$.

3. If and when 2b succeeds, let i, n be as found in step 2b.
Let $S =$
$[W_{p(0)}$ enumerated until now$]$
$\bigcup$ $[W_{p(1)}$ enumerated until now$]$
$\bigcup\{\langle 4+i,0\rangle, \langle 4+i,1\rangle, \ldots, \langle 4+i,n\rangle\}$.

4. Let σ_{s+1} be an extension of $\tau_i \diamond \langle 4+i,0\rangle \diamond \langle 4+i,1\rangle \diamond \ldots \diamond \langle 4+i,n\rangle$ such that content$(\sigma_{s+1}) = S$.
Enumerate S into $W_{p(0)}$ and $W_{p(1)}$.
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 $\mathbf{TxtEx}^*$-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. For $i \in \{0,1\}$, let $T_i = \tau_i \diamond \langle 4+i,0\rangle \diamond \langle 4+i,1\rangle \diamond \ldots$, where τ_i is as defined in stage s. Clearly, T_i is a text for L_i. 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 $\mathbf{TxtEx}^*$-identify at least one of L_1 and L_2.

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

5.2 Diagonalization Tools

Our proof of Theorem 6 above turns out to be the basic phase of most of the diagonalization results presented in this paper. In fact most diagonalization results can be thought of as dovetailing of this basic diagonalization step. We illustrate the idea in the context of another diagonalization result (a more general version of which will be presented later in this section).

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

A collection of languages that witnesses the above diagonalization is as follows.

Let $\mathcal{L} = \{L \mid \text{card}(\{i \leq 3 \mid \max(\{y \mid \langle i,y\rangle \in L\}) = y_0 < \infty \text{ and } W_{y_0} = L\}) \geq 2\}$.

The reader should note the similarity of the above class with the class witnessing the diagonalization in the proof of Theorem 6. Again, it not too difficult to establish that this class belongs to $\mathbf{Team}_4^2\mathbf{TxtEx}$. The proof that the above class does not belong to $\mathbf{Team}_2^1\mathbf{TxtEx}$ is, however, more complex. This complexity arises from the fact that now the diagonalization has to be carried out against any team consisting of two machines instead of just one machine. This is achieved by nesting the diagonalization in the proof of Theorem 6 twice along different "cylinders." We omit the details (see [18]); a more general result for team ratio $\frac{1}{2}$ will be presented later in this section.

The above discussion points to the desirability of some general tools for diagonalization. All the diagonalization proofs in this paper for $\mathbf{Team}_j^i\mathbf{TxtEx}$ vs $\mathbf{Team}_l^k\mathbf{TxtEx}$ can be seen to display the following properties:

- They can be made to work even if we force certain elements to be in each language of the diagonalizing class (this is the purpose of set S_1 in the following construction).
- They can be made to work even if we place restrictions on which cylinders are to be infinite and which cylinders are to be empty (sets S_4 and S_3, respectively, serve this role in the following construction).

- They can be made to work even if we specify which cylinders are to contain the coded grammar (below, the set S_2 specifies the cylinders into which a grammar for the language can be coded).

Additionally, the changes in the diagonalization proof to ensure the above restrictions can be carried out algorithmically (the predicate PROP below addresses this algorithmic nature of the modification required).

Taking the above discussion into account, we now show how to generalize diagonalization arguments of the form $\mathbf{Team}^i_j\mathbf{TxtEx}-\mathbf{Team}^k_l\mathbf{TxtEx} \neq \emptyset$. In particular we show how, given a theorem of the above form, for parameters i, j, k, l satisfying certain conditions and for new parameters i', j', k', l' satisfying certain conditions, we get a proof of $\mathbf{Team}^{i'}_{j'}\mathbf{TxtEx}-\mathbf{Team}^{k'}_{l'}\mathbf{TxtEx} \neq \emptyset$.

We first define these conditions and then present a general result (Theorem 7 below) which yields new diagonalization results from known ones. We note that these conditions are satisfied by all the diagonalization proofs in the present paper.

We first define a predicate (additional intuitive feel for the predicate is given after the definition). For a recursive function q, and $i, j, k, l \in N^+$, we define predicate PROP(q, i, j, k, l) to be true just in case the following holds: Suppose,

(a) finite sets $S_1, S_2, S_3, S_4, S_2' \subseteq N$,
(b) a team of $\leq l$ machines $\mathcal{M}$,

are given, such that S_2, S_3, S_4 are pairwise disjoint, $S_2' \subseteq S_2$, $\mathrm{card}(S_2) = j$, and $\mathrm{card}(S_2') \leq i$;

then $\mathcal{L}_{q,i,j,k,l,S_1,S_2,S_3,S_4,S_2',\mathcal{M}} \not\subseteq \mathbf{Team}^k_{\mathrm{card}(\mathcal{M})}\mathbf{TxtEx}(\mathcal{M})$, where

$\mathcal{L}_{q,i,j,k,l,S_1,S_2,S_3,S_4,S_2',\mathcal{M}} = \{L \mid$ the following conditions are satisfied
(a) $S_1 \subseteq L$,
(b) $(\forall x \in S_4)[\{y \mid \langle x, y\rangle \in L\}$ is infinite $]$,
(c) $\mathrm{card}(\{x \in S_2 \mid \max(\{y \mid \langle x, y\rangle \in L\}) = y_0 < \infty \ \wedge \ W_{y_0} = L\}) \geq i$,
(d) $(L - S_1) \cap \{\langle x, y\rangle \mid x \in S_3 \ \wedge \ y \in N\} = \emptyset$,
(e) $(\forall x \in S_2')[\max(\{y \mid \langle x, y\rangle \in L\}) = q(S_1, S_2, S_3, S_4, S_2', \mathcal{M}, x)]$,
(f) $(\forall x \in S_2')[S_1 \subseteq W_{q(S_1,S_2,S_3,S_4,S_2',\mathcal{M},x)} \subseteq S_1 \cup \{\langle z, y\rangle \mid z \notin S_3 \ \wedge \ y \in N\}]$.
$\}$

So, S_1 is simply a finite subset of the language (this is to ensure that the diagonalization can still be performed when one requires some finite sets, such as S_1, to be contained in the languages). For each element $x \in S_4$, an infinite subset of the x-th cylinder of N is present in the language. For each element $x \in S_3$, no element from the x-th cylinder of N, except perhaps members of S_1, appears in the language. (S_1, S_3, S_4 thus place certain constraints on what elements are allowed in the language). For at least i elements $x \in S_2$, a finite subset of the x-th cylinder of N is present in the language and the maximum element of this subset codes a grammar for the language (this ensures $\mathbf{Team}^i_j\mathbf{TxtEx}$-identification as the cardinality of S_2 is j). In addition condition (e) requires that, for all $x \in S_2' \subseteq S_2$, the grammar coded in the x-th cylinder can be effectively found, and these grammars behave in a "nice" manner (condition (f)).

We employ the above predicate to prove a theorem which given any known diagonalization of the form $\mathbf{Team}^i_j\mathbf{TxtEx} - \mathbf{Team}^k_l\mathbf{TxtEx} \neq \emptyset$, yields several related diagonalization results.

Theorem 7 *Let $1 \leq i \leq j$ and $0 \leq i_1 \leq i$. If PROP(q, i, j, k, l), then, for i', j', k', l' satisfying the following conditions,*

(a) $i' \leq i$,
(b) $k \leq k'$,
(c) $l' \leq l + \lceil k' - \frac{k'}{\lfloor i/i_1 \rfloor} \rceil$,

(d) $j' \geq j + i - i_1$,
(e) $1 \leq i' \leq j'$ *and* $1 \leq k' \leq l'$,

there exists a recursive q' *such that,* $PROP(q', i', j', k', l')$.

Note that it is sufficient to prove the theorem for equality in conditions (a) $i' = i$, (b) $k = k'$, (c) $l' = l + \lceil k' - \frac{k'}{\lfloor i/i_1 \rfloor} \rceil$, (d) $j' = j + i - i_1$. This is sufficient, since requiring more machines to be successful can only hurt and allowing extra machines in the team (without corresponding increase in the machines required to be successful) can only help. Different values of the parameter i_1 in the above theorem yield different diagonalization results. The use of i_1 in the diagonalization will become clear as we proceed.

Note that if PROP(q, i, j, k, l), then $\mathbf{Team}_j^i\mathbf{TxtEx} - \mathbf{Team}_l^k\mathbf{TxtEx} \neq \emptyset$. This is so because $\mathcal{L} = \bigcup_{\{\mathcal{M} | \mathrm{card}(\mathcal{M}) = l\}} \mathcal{L}_{q,i,j,k,l,\{\langle 0, code(\mathcal{M})\rangle\},\{1,\ldots,j\},\{0\},\emptyset,\emptyset,\mathcal{M}} \in \mathbf{Team}_j^i\mathbf{TxtEx} - \mathbf{Team}_l^k\mathbf{TxtEx}$ (where *code*$(\cdot)$ denotes some fixed coding of sets of machines). To see this, first note that one can construct a set of j machines (effectively in $q, i, j, k, l, S_1, S_2, S_3, S_4, S_2', \mathcal{M}$) witnessing $\mathcal{L}_{q,i,j,k,l,S_1,S_2,S_3,S_4,S_2',\mathcal{M}} \in \mathbf{Team}_j^i\mathbf{TxtEx}$. Thus $\mathcal{L} \in \mathbf{Team}_j^i\mathbf{TxtEx}$ (value of $\mathcal{M}$ can be obtained from the input text). Also, since $\mathcal{L}_{q,i,j,k,l,\{\langle 0,code(\mathcal{M})\rangle\},\{1,\ldots,j\},\{0\},\emptyset,\emptyset,\mathcal{M}} \not\subseteq \mathbf{Team}_l^k\mathbf{TxtEx}(\mathcal{M})$, it follows that $\mathcal{L} \notin \mathbf{Team}_l^k\mathbf{TxtEx}$.

As an application of the above theorem, suppose $\mathbf{Team}_j^i\mathbf{TxtEx} - \mathbf{Team}_l^k\mathbf{TxtEx} \neq \emptyset$ can be shown using a *suitable* proof. Then the above theorem allows us to conclude that $\mathbf{Team}_{j+i}^i\mathbf{TxtEx} - \mathbf{Team}_{l+k}^k\mathbf{TxtEx} \neq \emptyset$ can be shown using a *suitable* proof. By *suitable* proof we mean a proof such that for some q, PROP can be satisfied.

Since all our diagonalization proofs can be easily modified to satisfy PROP, we will use Theorem 7 implicitly to obtain general theorems. Note that in the usage of the above theorem to obtain $\mathbf{Team}_{j'}^{i'}\mathbf{TxtEx} - \mathbf{Team}_{l'}^{k'}\mathbf{TxtEx} \neq \emptyset$ from $\mathbf{Team}_j^i\mathbf{TxtEx} - \mathbf{Team}_l^k\mathbf{TxtEx} \neq \emptyset$, we will usually only specify the value of i_1 and leave the details of verifying that the properties hold to the reader.

Proof of Theorem 7. Suppose $i, j, k, l, q, i', k', j', l', i_1$ are given as above. Suppose a team of $\leq l'$ machines $\mathcal{M}$ is given. Suppose S_1, S_2, S_3, S_4, S_2' be any finite sets such that S_2, S_3, S_4 are pairwise disjoint, $S_2' \subseteq S_2$, $\mathrm{card}(S_2) = j'$, and $\mathrm{card}(S_2') \leq i'$.

Without loss of generality we assume that $i' = i$ (since $\mathbf{Team}_{j'}^{i}\mathbf{TxtEx} \subseteq \mathbf{Team}_{j'}^{i'}\mathbf{TxtEx}$, for $i' \leq i$). Also, without loss of generality, we assume that $\mathrm{card}(\mathcal{M}) = l'$ (since $\mathbf{Team}_{l''}^{k'}\mathbf{TxtEx} \subseteq \mathbf{Team}_{l'}^{k'}\mathbf{TxtEx}$, for $l'' \leq l'$).

We now have to show that there exists a recursive function q' such that

$$\mathcal{L}_{q',i',j',k',l',S_1,S_2,S_3,S_4,S_2',\mathcal{M}} \not\subseteq \mathbf{Team}_{\mathrm{card}(\mathcal{M})}^{k'}\mathbf{TxtEx}(\mathcal{M}).$$

We construct q' using the operator recursion theorem. The proof is based on the following idea. In this proof we work with i special grammars. The argument proceeds in stages. At each stage the following two processes are executed in parallel until a search for the mind change is successful in the first process:

- An attempt is made to find if any one of the k' seemingly most stable machines (in $\mathcal{M}$) makes a mind change.
- The i special grammars are divided into $\lfloor i/i_1 \rfloor$ groups of cardinality i_1 each. Then using these groups we perform distinct diagonalization of the kind done for $\mathbf{Team}_j^i\mathbf{TxtEx}$ versus $\mathbf{Team}_l^k\mathbf{TxtEx}$ (because $j' - i' \geq j - i_1$, we would be able to use this diagonalization: in case i_1 is zero we do not need any of the earlier special grammars in this diagonalization).

If the search for a mind change is successful at each stage, then each of the i special grammars yield the same language and less than k' members of the team, $\mathcal{M}$, converge on this language. If on the other hand, some stage s starts but does not finish, then one of the $\lfloor i/i_1 \rfloor$ groups will yield the desired language witnessing the failure of team $\mathcal{M}$ to $\mathbf{Team}_{l'}^{k'}\mathbf{TxtEx}$-identify $\mathcal{L}_{q',i',j',k',l',S_1,S_2,S_3,S_4,S_2',\mathcal{M}}$.

We now proceed formally.

By a suitably padded version of the operator recursion theorem [5] there exists a recursive, 1–1, q' such that the sets $W_{q'(S_1,S_2,S_3,S_4,S_2',\mathcal{M},x)}$, may be defined as follows in stages. We assume that the padding (to obtain q') is such that, for all $S_1,S_2,S_3,S_4,S_2',\mathcal{M}$, and x, $q'(S_1,S_2,S_3,S_4,S_2',\mathcal{M},x) > \max(\{y \mid \langle x,y\rangle \in S_1\})$. Below, taking $S_1,S_2,S_3,S_4,S_2',\mathcal{M}$ to be fixed we define, for all x, $p(x) = q'(S_1,S_2,S_3,S_4,S_2',\mathcal{M},x)$. Let S_2'' be a set of cardinality i such that $S_2' \subseteq S_2'' \subseteq S_2$. Let conv be as defined in the proof of Theorem 2. For σ, let Z_σ be the (lexicographic least) subset of $\mathcal{M}$ of cardinality k' such that, for each $\mathbf{M} \in Z_\sigma$, for each $\mathbf{M'} \in \mathcal{M} - Z_\sigma$, $\mathrm{conv}(\mathbf{M},\sigma) \leq \mathrm{conv}(\mathbf{M'},\sigma)$. Intuitively Z_σ denotes the k' seemingly most stable machines in $\mathcal{M}$ on σ.

For each $y \in S_2''$, enumerate $S_1 \cup \{\langle x,p(x)\rangle \mid x \in S_2''\}$ into $W_{p(y)}$. Let σ_0 be a sequence such that $\mathrm{content}(\sigma_0) = S_1 \cup \{\langle x,p(x)\rangle \mid x \in S_2''\}$. Let S_5 be a set disjoint from $\{x \mid (\exists y)[\langle x,y\rangle \in S_1]\}, S_2, S_3, S_4$ such that $\mathrm{card}(S_5) = i_1$. Let S_6 be such that $S_5 \subseteq S_6 \subseteq S_5 \cup (S_2 - S_2'')$, and $\mathrm{card}(S_6) = j$. Let $W^s_{p(x)}$ denote $W_{p(x)}$ enumerated before stage s. Go to stage 0.

Stage s

Dovetail steps 1 and 2 until step 1 succeeds. If and when step 1 succeeds go to step 3.

1. Search for an extension τ of σ_s such that $Z_{\sigma_s} \neq Z_\tau$ and $\mathrm{content}(\tau) - \mathrm{content}(\sigma_s) \subseteq \{\langle x,y\rangle \mid x \notin S_3 \cup S_2''\}$.
2. *(* We now set up $\lfloor \frac{i}{i_1} \rfloor$ distinct diagonalizations of the form* $\mathbf{Team}^i_j\mathbf{TxtEx} - \mathbf{Team}^k_l\mathbf{TxtEx}$. *Refer to these diagonalization by diagonalization number w, $0 \leq w < \lfloor \frac{i}{i_1} \rfloor$. *)*

   (We also set up the parameters for these $\lfloor \frac{i}{i_1} \rfloor$ diagonalizations. Intuitively, in diagonalization number w, $X_{1,w}, X_2, X_{3,w}, X_{4,w}, X_2', (\mathcal{M} - Z_{\sigma_s}) \cup \mathcal{M}_w$, correspond to the parameters $S_1,S_2,S_3,S_4,S_2',\mathcal{M}$ in $\mathcal{L}_{q,i,j,k,l,S_1,S_2,S_3,S_4,S_2',\mathcal{M}}$. *)*

   Let $X_2 = S_6$.

   Let $X_2' = S_5$.

   Let $Y_0, Y_1, \ldots, Y_{\lfloor i/i_1 \rfloor -1}$ be pairwise disjoint subsets of S_2'' each of cardinality i_1.

   Let $u_0, u_1, \ldots, u_{\lfloor i/i_1 \rfloor -1}$ be pairwise distinct numbers such that each is greater than $\max(S_2 \cup S_3 \cup S_4 \cup S_5 \cup S_6 \cup \{x \mid (\exists y)[\langle x,y\rangle \in W^s_{p(z)}$ for some $z \in S_2'']\})$. *(* Intuitively, u_w's are new cylinders. They are used to make the languages considered in different diagonalizations different. *)*

   For each $w < \lfloor i/i_1 \rfloor$, let $X_{3,w} = \{u_r \mid r < \lfloor i/i_1 \rfloor \ \wedge\ r \neq w\} \cup S_3 \cup S_2''$.

   For each $w < \lfloor i/i_1 \rfloor$, let $X_{4,w} = \{u_w\} \cup S_4$.

   Let *map* be a mapping from S_2'' to S_5 such that for each $w < \lfloor i/i_1 \rfloor$, $map(Y_w) = S_5$.

   Go to substage 0.

   Substage s'

   For each $w < \lfloor i/i_1 \rfloor$, let $\mathcal{M}_w = \{\mathbf{M} \in Z_{\sigma_s} \mid (\exists y)[\langle u_w,y\rangle \in W_{\mathbf{M}(\sigma_s),s'}] \ \wedge\ (\forall w' < \lfloor i/i_1 \rfloor \mid w' \neq w)(\forall y)[\langle u_{w'},y\rangle \notin W_{\mathbf{M}(\sigma_s),s'}]\}$.

   (Intuitively, machines that converge to grammars which output elements of the form $\langle u_w,y\rangle$ cannot identify languages constructed in the w'-th diagonalization, for $w \neq w'$. This is the motivation for the above definition. Machines in $\mathcal{M}_w$ can participate in diagonalization number w and no other diagonalization (we say that these machines are committed to diagonalization number w). Machines in $Z_{\sigma_s} - \bigcup_{w < \lfloor \frac{i}{i_1} \rfloor} \mathcal{M}_w$ currently seem uncommitted to any particular diagonalization. We will change substage in case these uncommitted machines, are later on found to commit to some diagonalization (see step 2.1). Note that size of $(\mathcal{M} - Z_{\sigma_s}) \cup \mathcal{M}_w$ is $\leq l$, for some diagonalization number w *).*

   For each $w < \lfloor i/i_1 \rfloor$, let $X_{1,w} = \bigcup_{x \in Y_w}[W_{p(x)}$ enumerated until now].

   Dovetail steps 2.1 and 2.2 until step 2.1 succeeds. If and when step 2.1 succeeds, go to substage $s''+1$, where s'' is as found in step 2.1.

   2.1 Search for an $s'' > s'$ and an $\mathbf{M} \in Z_{\sigma_s} - \bigcup_w \mathcal{M}_w$, such that $(\exists w < \lfloor i/i_1 \rfloor)(\exists y)[\langle u_w,y\rangle \in W_{\mathbf{M}(\sigma_s),s''}] \ \wedge\ (\forall w' < \lfloor i/i_1 \rfloor \mid w' \neq w)(\forall y)[\langle u_{w'},y\rangle \notin W_{\mathbf{M}(\sigma_s),s''}]$.

   2.2 Set $t = 0$.

repeat

For each $w < \lfloor i/i_1 \rfloor$, for each $x \in Y_w$ such that $\mathrm{card}(\mathcal{M}_w) \leq l - (l' - k')$, enumerate

$W_{q(X_{1,w}, X_2, X_{3,w}, X_{4,w}, X_2', (\mathcal{M} - Z_{\sigma_s}) \cup \mathcal{M}_w, map(x)), t} - \{\langle x, y \rangle \mid x \in X_{3,w}\}$ into $W_{p(x)}$.

Set $t = t + 1$.

forever

End substage s'

3. Let $X = \bigcup_{x \in S_2''} [W_{p(x)}$ enumerated until now$]$.

   Let σ_{s+1} be an extension of τ such that $\mathrm{content}(\sigma_{s+1}) = \mathrm{content}(\tau) \cup X \cup \{\langle x, s \rangle \mid x \in S_4\}$.

   Enumerate $\mathrm{content}(\sigma_{s+1})$ into $W_{p(x)}$, for $x \in S_2''$.

   Go to stage $s + 1$.

End stage s

Let $\mathcal{L} = \mathcal{L}_{q', i', j', k', l', S_1, S_2, S_3, S_4, S_2', \mathcal{M}}$. We show that $\mathcal{L} \not\subseteq \mathbf{Team}_{l'}^{k'}\mathbf{TxtEx}(\mathcal{M})$. We consider the following cases.

Case 1: All stages terminate.

In this case, let $T = \bigcup_s \mathrm{content}(\sigma_s)$. Clearly, for all $x \in S_2''$, $W_{p(x)} = \mathrm{content}(T) \in \mathcal{L}$. Moreover at most $k' - 1$ of the machines in $\mathcal{M}$ converge on T. Thus $\mathcal{L} \not\subseteq \mathbf{Team}_{l'}^{k'}\mathbf{TxtEx}(\mathcal{M})$.

Case 2: Stage s starts but never terminates.

Note that, in each stage, there can be at most finitely many substages which terminate (i.e. have a successful step 2.1). This is so since step 2.1 can succeed at most once due to each machine in Z_{σ_s}. Let s' be the substage in stage s which starts but never terminates. Let $\mathcal{M}_w$ be as defined in stage s, substage s'. For each $w < \lfloor i/i_1 \rfloor$, let $\mathcal{L}_w = \mathcal{L}_{q, i, j, k, l, X_{1,w}, X_2, X_{3,w}, X_{4,w}, X_2', (\mathcal{M} - Z_{\sigma_s}) \cup \mathcal{M}_w}$. Now for each $w < \lfloor i/i_1 \rfloor$, $\mathcal{L}_w \subseteq \mathcal{L}$ (to see this, first note that languages in $\mathcal{L}_w$ do not contain any element in $\{\langle x, y \rangle \mid x \in X_{3,w}\}$; thus step 2.2 in stage s, substage s', makes, for each $x \in Y_w$, $W_{p(x)} = W_{q(X_{1,w}, X_2, X_{3,w}, X_{4,w}, X_2', (\mathcal{M} - Z_{\sigma_s}) \cup \mathcal{M}_w, map(x))}$. Now the clauses (a)–(f) in the definition of $\mathcal{L}$ can be easily verified using the corresponding clauses in the definition of $\mathcal{L}_w$ and the definition of the parameters). Now, for each $w < \lfloor i/i_1 \rfloor$, $L_w \in \mathcal{L}_w$, $\mathbf{M} \in Z_{\sigma_s} - \mathcal{M}_w$, $[W_{\mathbf{M}(\sigma_s)} \neq L_w]$. Also, for some $w < \lfloor i/i_1 \rfloor$, $\mathrm{card}(\mathcal{M}_w) \leq \lfloor \frac{k'}{\lfloor i/i_1 \rfloor} \rfloor$. Hence, using the fact that $\mathcal{L}_w \not\subseteq \mathbf{Team}_{\mathrm{card}((\mathcal{M} - Z_{\sigma_s}) \cup \mathcal{M}_w)}^{k}\mathbf{TxtEx}((\mathcal{M} - Z_{\sigma_s}) \cup \mathcal{M}_w)$, it follows that $\mathcal{L} \not\subseteq \mathbf{Team}_{\mathrm{card}(\mathcal{M})}^{k'}\mathbf{TxtEx}(\mathcal{M})$. ∎

As an immediate application of the above theorem, we have the following corollary that will be referred to later.

Corollary 5 $(\forall n \in N^+)[\mathbf{Team}_{4n-1}^{2n}\mathbf{TxtEx} - \mathbf{Team}_{2n-1}^{n}\mathbf{TxtEx} \neq \emptyset]$.

Proof. Note that $\mathbf{Team}_{2n}^{2n}\mathbf{TxtEx} - \mathbf{Team}_{n-1}^{n}\mathbf{TxtEx} \neq \emptyset$ (since $\mathbf{Team}_{n-1}^{n}\mathbf{TxtEx} = \emptyset$). Now the Corollary follows, by using Theorem 7, with $i = i' = 2n$, $j = 2n$, $j' = 4n - 1$, $k = k' = n$, $l = n - 1$, $l' = 2n - 1$ and $i_1 = 1$. ∎

We now squeeze some more advantage out of this technique by showing a variant of Theorem 7 which allows us to extend diagonalization results of the form

$$\mathbf{Team}_j^i\mathbf{TxtEx} - \mathbf{Team}_l^k\mathbf{TxtEx}^* \neq \emptyset$$

to related results of the form $\mathbf{Team}_{j'}^{i'}\mathbf{TxtEx} - \mathbf{Team}_{l'}^{k'}\mathbf{TxtEx}^* \neq \emptyset$ for suitable values of i', j', k', and l'. To this end we define a predicate analogous to PROP.

For a recursive function q, and $i,j,k,l \in N^+$, we define the predicate PROPS(q,i,j,k,l) identically to PROP(q,i,j,k,l) except that we have $\mathcal{L}_{q,i,j,k,l,S_1,S_2,S_3,S_4,S_2',\mathcal{M}} \not\subseteq \mathbf{Team}^{k}_{\mathrm{card}(\mathcal{M})}\mathbf{TxtEx}^*(\mathcal{M})$, instead of $\mathcal{L}_{q,i,j,k,l,S_1,S_2,S_3,S_4,S_2',\mathcal{M}} \not\subseteq \mathbf{Team}^{k}_{\mathrm{card}(\mathcal{M})}\mathbf{TxtEx}(\mathcal{M})$. Our treatment below is brief.

We now employ the predicate PROPS to prove the following theorem which is analogous to Theorem 7. The proof of the following theorem is similar to that of Theorem 7 (except that we do not have anything similar to step 2.1, since we cannot use it for $*$-errors).

Theorem 8 *Suppose $1 \leq i \leq j$ and $0 \leq i_1 \leq i$. If PROPS(q,i,j,k,l), then, for i',j',k',l' satisfying the following conditions,*

(a) $i' \leq i$,
(b) $k \leq \lceil k' - \frac{k'}{\lfloor i/i_1 \rfloor} \rceil$,
(c) $l' \leq l + k'$,
(d) $j' \geq j + i - i_1$,
(e) $1 \leq i' \leq j'$ and $1 \leq k' \leq l'$,

there exists a recursive q' such that, PROPS(q',i',j',k',l').

Proof. Suppose $i,j,k,l,q,i',k',j',l',i_1$ are given as above. Without loss of generality we assume $i' = i$.

By a suitably padded version of the operator recursion theorem [5], there exists a recursive, 1–1, q' such that the sets $W_{q'(S_1,S_2,S_3,S_4,S_2',\mathcal{M},x)}$ may be defined as follows. We assume that the padding (to obtain q') is such that, for all $S_1,S_2,S_3,S_4,S_2',\mathcal{M}$, and x, $q'(S_1,S_2,S_3,S_4,S_2',\mathcal{M},x) > \max(\{y \mid \langle x,y\rangle \in S_1\})$. Below, taking $S_1,S_2,S_3,S_4,S_2',\mathcal{M}$ to be fixed we refer to $q'(S_1,S_2,S_3,S_4,S_2',\mathcal{M},x)$ by $p(x)$. Let S_2'' be a set of cardinality i such that $S_2' \subseteq S_2'' \subseteq S_2$. Let conv be as defined in the proof of Theorem 2. For σ, let Z_σ be the (lexicographic least) subset of $\mathcal{M}$ of cardinality k' such that, for each $\mathbf{M} \in Z_\sigma$, for each $\mathbf{M}' \in \mathcal{M} - Z_\sigma$, $\mathrm{conv}(\mathbf{M},\sigma) \leq \mathrm{conv}(\mathbf{M}',\sigma)$.

For each $y \in S_2''$, enumerate $S_1 \cup \{\langle x,p(x)\rangle \mid x \in S_2''\}$ into $W_{p(y)}$. Let σ_0 be a sequence such that $\mathrm{content}(\sigma_0) = S_1 \cup \{\langle x,p(x)\rangle \mid x \in S_2''\}$. Let S_5 be a set disjoint from $\{x \mid (\exists y)[\langle x,y\rangle \in S_1]\}, S_2, S_3, S_4$ such that $\mathrm{card}(S_5) = i_1$. Let S_6 be such that $S_5 \subseteq S_6 \subseteq S_5 \cup (S_2 - S_2'')$, and $\mathrm{card}(S_6) = j$. Let $W^s_{p(x)}$ denote $W_{p(x)}$ enumerated before stage s. Go to stage 0.

Stage s

Dovetail steps 1 and 2 until step 1 succeeds. If and when step 1 succeeds go to step 3.

1. Search for an extension τ of σ_s such that $Z_{\sigma_s} \neq Z_\tau$ and $\mathrm{content}(\tau) - \mathrm{content}(\sigma_s) \subseteq \{\langle x,y\rangle \mid x \notin S_3 \cup S_2''\}$.
2. Let $X_1 = W^s_{p(x)}$, where x is an element of S_2''.
   Let $X_2 = S_6$.
   Let $X_2' = S_5$.
   Let $\mathcal{M}_1 = \mathcal{M} - Z_{\sigma_s}$.
   Let $Y_0, Y_1, \ldots, Y_{\lfloor i/i_1 \rfloor - 1}$ be pairwise disjoint subsets of S_2'' of cardinality i_1 each.
   Let $u_0, u_1, \ldots, u_{\lfloor i/i_1 \rfloor - 1}$ be pairwise distinct numbers such that each is greater than $\max(S_2 \cup S_3 \cup S_4 \cup S_5 \cup S_6)$.
   For each $w < \lfloor i/i_1 \rfloor$, let $X_{3,w} = \{u_r \mid r < \lfloor i/i_1 \rfloor \ \wedge\ r \neq w\} \cup S_3 \cup S_2''$.
   For each $w < \lfloor i/i_1 \rfloor$, let $X_{4,w} = \{u_w\} \cup S_4$.
   Let *map* be a mapping from S_2'' to S_5 such that for each $w < \lfloor i/i_1 \rfloor$, $map(Y_w) = S_5$.
   Set $t = 0$.
   repeat
   For each $w < \lfloor i/i_1 \rfloor$, for each $x \in Y_w$, enumerate $W_{q(X_1,X_2,X_{3,w},X_{4,w},X_2',\mathcal{M}_1,map(x)),t} - \{\langle x,y\rangle \mid x \in X_{3,w}\}$ into $W_{p(x)}$.
   Set $t = t + 1$.
   forever
3. Let $X = \bigcup_{x \in S_2''}[W_{p(x)}$ enumerated until now$]$.
   Let σ_{s+1} be an extension of τ such that $\mathrm{content}(\sigma_{s+1}) = \mathrm{content}(\tau) \cup X \cup \{\langle x,s\rangle \mid x \in S_4\}$.
   Enumerate $\mathrm{content}(\sigma_{s+1})$ into $W_{p(x)}$, for $x \in S_2''$.

Go to stage $s+1$.

End stage s

Let $\mathcal{L} = \mathcal{L}_{q',i',j',k',l',S_1,S_2,S_3,S_4,S_2',\mathcal{M}}$. We show that $\mathcal{L} \not\subseteq \mathbf{Team}_{l'}^{k'}\mathbf{TxtEx}^*(\mathcal{M})$. We consider the following cases.

Case 1: All stages terminate.

In this case, let $T = \bigcup_s \mathrm{content}(\sigma_s)$. Clearly, for all $x \in S_2''$, $W_{p(x)} = \mathrm{content}(T) \in \mathcal{L}$. Moreover, at most $k'-1$ of the machines in $\mathcal{M}$ converge on T. Thus, $\mathcal{L} \not\subseteq \mathbf{Team}_{l'}^{k'}\mathbf{TxtEx}^*(\mathcal{M})$.

Case 2: Stage s starts but never terminates.

Let $\mathcal{M}_1$ be as defined in stage s. For each $w < \lfloor i/i_1 \rfloor$, let $\mathcal{L}_w = \mathcal{L}_{q,i,j,k,l,X_1,X_2,X_{3,w},X_{4,w},X_2',\mathcal{M}_1}$. Now, for each $w < \lfloor i/i_1 \rfloor$, $\mathcal{L}_w \subseteq \mathcal{L}$ (since step 2 in stage s, makes for each $x \in Y_w$, $W_{p(x)} = W_{q(X_1,X_2,X_{3,w},X_{4,w},X_2',\mathcal{M}_1,map(x))}$). Also, for each $w < w' < \lfloor i/i_1 \rfloor$, $L_w \in \mathcal{L}_w$, $L_{w'} \in \mathcal{L}_{w'}$, L_w and $L_{w'}$ are infinitely different. Thus, for some $w < \lfloor i/i_1 \rfloor$, at most $\lfloor \frac{k'}{\lfloor i/i_1 \rfloor} \rfloor$ of the machines in Z_{σ_s}, $\mathbf{TxtEx}^*$-identify a non empty subset of $\mathcal{L}_w$. Thus, since $\mathcal{L}_w \not\subseteq \mathbf{Team}_{\mathrm{card}(\mathcal{M}_1)}^{k}\mathbf{TxtEx}^*(\mathcal{M}_1)$, we have $\mathcal{L} \not\subseteq \mathbf{Team}_{\mathrm{card}(\mathcal{M})}^{k'}\mathbf{TxtEx}^*(\mathcal{M})$. ∎

Note that for all $i \leq j$ and $k > l$, there exists a q such that $\mathrm{PROP}(q,i,j,k,l)$ (respectively, $PROPS(q,i,j,k,l)$).

5.3 Team Language Identification with Success Ratio $\frac{1}{2}$

In the context of functions, the following result immediately follows from Pitt's connection [25] between team function identification and probabilistic function identification.

Theorem 9 [24, 27] $(\forall j > 0)[\mathbf{Team}_{2j}^{j}\mathbf{Ex} = \mathbf{Team}_{2}^{1}\mathbf{Ex}]$.

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, $\mathbf{Team}_{2}^{1}\mathbf{Ex} = \mathbf{Team}_{4}^{2}\mathbf{Ex} = \mathbf{Team}_{6}^{3}\mathbf{Ex} = \cdots$, etc.

Surprisingly, in the context of language identification, we are able to show the following Theorem 10 below which 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 [24] for function inference does *not* lift to language learning!

Theorem 10 $\mathbf{Team}_{4}^{2}\mathbf{TxtEx} - \mathbf{Team}_{2}^{1}\mathbf{TxtEx}^* \neq \emptyset$.

Corollary 6 $(\forall j \in N^+)[\mathbf{Team}_{2j+1}^{j}\mathbf{TxtEx} - \mathbf{Team}_{2}^{1}\mathbf{TxtEx}^* \neq \emptyset]$.

Proof of Theorem 10. By Theorem 6 $\mathbf{Team}_{3}^{2}\mathbf{TxtEx} - \mathbf{Team}_{1}^{1}\mathbf{TxtEx}^* \neq \emptyset$. Theorem now follows by using Theorem 8, with $i = i' = 2$, $j = 3$, $j' = 4$, $i_1 = 1$, $k = k' = 1$, $l = 1$, $l' = 2$. ∎

Even more surprising is Corollary 7 to Theorem 11 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 11 $(\forall j)(\forall i)[\mathbf{Team}_{4j+2}^{2j+1}\mathbf{TxtEx}^{i} \subseteq \mathbf{Team}_{2}^{1}\mathbf{TxtEx}^{i\cdot(j+1)}]$.

Corollary 7 $(\forall j)[\mathbf{Team}_{4j+2}^{2j+1}\mathbf{TxtEx} = \mathbf{Team}_{2}^{1}\mathbf{TxtEx}]$.

Corollary 8 $(\forall j)(\forall i)[\mathbf{Team}^{j+1}_{2j+1}\mathbf{TxtEx}^i \subseteq \mathbf{Team}^1_2\mathbf{TxtEx}^{i\cdot\lceil (j+1)/2\rceil}]$.

Corollary 9 $(\forall j)(\forall i)[\mathbf{Team}^{j+1}_{2j+1}\mathbf{TxtEx}^i \subseteq \mathbf{Team}^{j+2}_{2j+3}\mathbf{TxtEx}^{i\cdot\lceil (j+1)/2\rceil}]$

Proof of Theorem 11. Suppose $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_{4j+2}$ $\mathbf{Team}^{2j+1}_{4j+2}\mathbf{TxtEx}^i$-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 2. Let $m^\sigma_1, m^\sigma_2, \ldots, m^\sigma_{4j+2}$ be a permutation of $1, 2, \ldots, 4j+2$, such that, for $1 \leq r < 4j+2$, $[(\text{conv}(\mathbf{M}_{m^\sigma_r}, \sigma), m^\sigma_r) < (\text{conv}(\mathbf{M}_{m^\sigma_{r+1}}, \sigma), m^\sigma_{r+1})]$.

Let $\text{match}(r, \sigma) = \max(\{n \leq |\sigma| \mid \text{card}((\text{content}(\sigma[n]) - W_{r,|\sigma|}) \cup (W_{r,n} - \text{content}(\sigma))) \leq i\})$. Intuitively, $\text{match}(r, \sigma)$ tells us how much $W_{r,|\sigma|}$ and σ are similar (modulo i errors).

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

$$\mathbf{M}'_1(\sigma) = \text{majority}(\mathbf{M}_{m^\sigma_1}(\sigma), \mathbf{M}_{m^\sigma_2}(\sigma), \ldots, \mathbf{M}_{m^\sigma_{2j+1}}(\sigma)).$$

Let $n^\sigma_1, n^\sigma_2, \ldots, n^\sigma_j$ be the distinct elements of $\{m_r \mid r \in S_\sigma\}$.

$$\mathbf{M}'_2(\sigma) = \text{majority}(\mathbf{M}_{m^\sigma_{2j+2}}(\sigma), \mathbf{M}_{m^\sigma_{2j+3}}(\sigma), \ldots, \mathbf{M}_{m^\sigma_{3j+2}}(\sigma), \mathbf{M}_{n^\sigma_1}(\sigma), \ldots, \mathbf{M}_{n^\sigma_j}(\sigma)).$$

Now suppose T is a text for $L \in \mathcal{L}$. Then at least $2j+1$ of the machines $\mathbf{M}_1, \ldots, \mathbf{M}_{4j+2}$ converge on T to a grammar for i variant of L. Thus, for $1 \leq r \leq 2j+1$, $\lim_{n\to\infty} m^{T[n]}_r\!\downarrow$, to say m_r, and $\mathbf{M}_{m_r}(T)\!\downarrow$. Now if $\text{card}(\{r \mid 1 \leq r \leq 2j+1 \wedge W_{\mathbf{M}_{m_r}(T)} =^i L\}) \geq j+1$, then $\mathbf{M}'_1(T)\!\downarrow = \text{majority}(\mathbf{M}_{m_1}(T), \mathbf{M}_{m_2}(T), \ldots, \mathbf{M}_{m_{2j+1}}(T))$ is a grammar for an $i \cdot (j+1)$ variant of L. On the other hand if $\text{card}(\{r \mid 1 \leq r \leq 2j+1 \wedge W_{\mathbf{M}_{m_r}(T)} =^i L\}) < j+1$, then, for $2j+1 < r \leq 3j+2$, $\lim_{n\to\infty} m^{T[n]}_r\!\downarrow$, to say m_r, and $\mathbf{M}_{m_r}(T)\!\downarrow$. Moreover $\text{card}(\{r \mid 1 \leq r \leq 3j+2 \wedge W_{\mathbf{M}_{m_r}(T)} =^i L\}) \geq j+1$. Moreover, since $\text{card}(\{r \mid 1 \leq r \leq 2j+1 \wedge W_{\mathbf{M}_{m_r}(T)} =^i L\}) < j+1$, we have that $\lim_{n\to\infty} S_{T[n]}$ converges, to say S, and $(\forall r \in \{1, 2\ldots, 2j+1\} - S)[W_{\mathbf{M}_{m_r}(T)} \neq^i L]$. It immediately follows that $\text{card}(\{r \mid [2j+1 < r \leq 3j+2 \vee r \in S] \wedge W_{\mathbf{M}_{m_r}(T)} =^i L\}) \geq j+1$.

Now, $\mathbf{M}'_2(T)\!\downarrow = \text{majority}(\mathbf{M}_{m_{2j+2}}(T), \mathbf{M}_{m_{2j+3}}(T), \ldots, \mathbf{M}_{m_{3j+2}}(T), \mathbf{M}_{n_1}(T), \ldots, \mathbf{M}_{n_j}(T))$, where $n_1, \ldots, n_j$ are the different members of $\{m_r \mid r \in S\}$. Thus $\mathbf{M}'_2(T)$ is a grammar for an $i \cdot (j+1)$ variant of L.

From the above analysis we have: $\{\mathbf{M}'_1, \mathbf{M}'_2\}$ witness that $\mathcal{L} \in \mathbf{Team}^1_2\mathbf{TxtEx}^{i\cdot(j+1)}$. ∎

Finally, we settle the question for team success ratio 1/2 by establishing Theorem 12 and 13 below. We note that the proofs of these two theorems are the most complicated in this paper.

Theorem 12 $(\forall n \in N^+)[\mathbf{Team}^{2n}_{4n}\mathbf{TxtEx} - \mathbf{Team}^n_{2n}\mathbf{TxtEx} \neq \emptyset]$.

Proof of Theorem 12. Consider the following class of languages.

$\mathcal{L} = \{L \mid \max(\{i < 4n \mid \text{card}(\{x \mid \langle i, x\rangle \in L\}) = x_0 < \infty \wedge W_{x_0} = L\}) \geq 2n\}$.

For $i \in N$, we call the set $\{\langle i, x\rangle \mid \langle i, x\rangle \in L\}$ as the i-th cylinder of L. $\max(\{x \mid \langle i, x\rangle \in L\})$, if it exists, is called the grammar coded into the i-th cylinder of L.

It is easy to see that $\mathcal{L} \in \mathbf{Team}^{2n}_{4n}\mathbf{TxtEx}$. Suppose by way of contradiction that the team $\{\mathbf{M}_0, \mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_{2n-1}\}$ is such that $\mathcal{L} \subseteq \mathbf{Team}^n_{2n}\mathbf{TxtEx}(\{\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_{2n-1}\})$. Then by the implicit use of the operator recursion theorem [5], there exists a 1-1, recursive, increasing p such that $W_{p(\cdot)}$ may be described as follows.

Recall that $[x_1 \,..\, x_2]$ denotes the set $\{x \mid x_1 \leq x \leq x_2\}$. In the following argument, the bulk of the work for diagonalization is done in steps 4 and 5. On the completion of step 5, step 6 easily achieves diagonalization using essentially the technique developed in the proof of Theorem 6. We give an informal overview of the proof.

Step 3 in the construction attempts to find if one of the most seemingly stable n machines in the team $\{\mathbf{M}_0, \mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_{2n-1}\}$ makes a mind change. If the search for such a mind change succeeds infinitely often, then less than n out of the $2n$ machines in the team are successful, achieving the diagonalization. So suppose that step 3 succeeds only finitely often. Let stage s

be the stage in which the search for mind change does not succeed. Step 6 of this stage s, if reached, essentially carries out a diagonalization of $\mathbf{Team}^{2n}_{4n-1}\mathbf{TxtEx}$ versus $\mathbf{Team}^{n}_{2n-1}\mathbf{TxtEx}$. This is essentially the diagonalization argument used in Corollary 5. For this we need to "spoil" the grammar output by at least one of the n seemingly most stable machines, while having spoiled at most one of the grammars coded into the first $2n$ cylinders of the language being considered for diagonalization. Steps 4 and 5 essentially try to achieve this aim. This is done as follows. First, step 4 temporarily "spoils" a set X (of size n) of grammars coded into the first $2n$ cylinders forcing grammars output by at least $\lceil \mathrm{card}(X)/2 \rceil$ of the n seemingly most stable machines to be temporarily spoiled. (This is essentially done using a $\mathbf{Team}^{2r}_{3r}\mathbf{TxtEx}$ versus $\mathbf{Team}^{u}_{v}\mathbf{TxtEx}$ diagonalization, where $u > 2v/3$). Step 5 essentially iterates the process of step 4 reducing the cardinality of X by nearly half in each iteration until X becomes a singleton set. In case steps 4,5 do not succeed in their aim, the diagonalization performed in these steps, gives us the required diagonalizing language. While going through the construction, we suggest that the reader simultaneously go through the case analysis after the construction (Case 2.1 corresponding to step 4; Case 2.2 corresponding to step 5 and Case 2.3 corresponding to step 6).

We now proceed formally. Let $\mathbf{lmc}$ be a function such that $\mathbf{lmc}(\mathbf{M}, \sigma) = \max(\{|\tau| \mid \tau \subseteq \sigma \wedge \mathbf{M}(\tau) \neq \mathbf{M}(\sigma)\})$. Enumerate $\langle 0, p(0)\rangle, \langle 1, p(1)\rangle, \ldots, \langle 2n-1, p(2n-1)\rangle$ in $W_{p(0)}, W_{p(1)}, \ldots, W_{p(2n-1)}$. Let σ_0 be such that $\mathrm{content}(\sigma_0) = \{\langle 0, p(0)\rangle, \langle 1, p(1)\rangle, \ldots, \langle 2n-1, p(2n-1)\rangle\}$. Set avail $= 2n-1$ (intuitively, avail denotes the least number such that, for all $i >$ avail, $p(i)$ is available for diagonalization). $W^s_{p(\cdot)}$ denotes $W_{p(\cdot)}$ enumerated before stage s. Go to stage 0.

Begin stage s

Invariant $W^s_{p(0)} = W^s_{p(i)} = \mathrm{content}(\sigma_s)$, for all $i < 2n$.

1. Let $Z \subseteq [0 \,..\, 2n-1]$ be such that, $\mathrm{card}(Z) = n$ and for all $i \in Z$ and for all $j \in ([0 \,..\, 2n-1] - Z)$, $\mathbf{lmc}(\mathbf{M}_i, \sigma_s) \leq \mathbf{lmc}(\mathbf{M}_j, \sigma_s)$.
   (∗ Intuitively, Z denotes the set of n machines which have not changed their conjectures recently on σ_s, i.e., they are the seemingly n most "stable" machines on σ_s. ∗)
2. Dovetail steps 3 and 4–6 until step 3 succeeds. If and when step 3 succeeds, go to step 7.
3. Search for an extension τ of σ_s such that, for some $i \in Z$, $\mathbf{M}_i(\sigma_s) \neq \mathbf{M}_i(\tau)$ and $\mathrm{content}(\tau) - \mathrm{content}(\sigma_s) \subseteq \{\langle x, y\rangle \mid x \geq 2n\}$.
   (∗ Note that if this step succeeds infinitely often, then less than n members of the team $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_{2n}$ converge on a suitable text for some language in $\mathcal{L}$. ∗)
4. *(∗ Intuitively, the aim of this step is to temporarily spoil at least $n/2$ machines in Z, while temporarily spoiling at most n of the grammars coded into the first $2n$ cylinders of the diagonalizing language. Simultaneously a diagonalization similar to $\mathbf{Team}^{2n}_{3n}\mathbf{TxtEx}$, versus $\mathbf{Team}^{u}_{v}\mathbf{TxtEx}$, for $u > 2v/3$, is carried out. This diagonalization uses cylinders $[0 \,..\, n-1]$ and $[2n \,..\, 4n-1]$ in the diagonalizing language for coding of grammars. It uses the fact that if step 4a does not succeed then $< 3n/2$ of the machines in $\mathbf{M}_0, \ldots, \mathbf{M}_{2n-1}$ can potentially identify any language considered in the diagonalization in step 4b. Note that if the aim of temporarily spoiling at least $n/2$ machines in Z is not successful then the diagonalization will succeed. ∗)*
   For each $i < n$, let $q_i = p(\text{avail} + 1 + i)$.
   Set avail $=$ avail $+ n$.
   For each $i < n$, enumerate $\langle 2n + i, q_i\rangle$ in $W_{p(0)}$.
   For each $i < n$, enumerate $[W_{p(0)}$ enumerated until now$]$ into $W_{p(i)}$ and W_{q_i}.
   Set $m = 1 + \max(\{x \mid \{\langle 4n, x\rangle, \langle 4n+1, x\rangle\} \cap [W_{p(0)}$ enumerated until now$] \neq \emptyset\})$.
   Dovetail steps 4a and 4b until, if ever, step 4a succeeds. If and when step 4a succeeds, go to step 5.

4a. Search for $Y \subseteq Z$ such that $\mathrm{card}(Y) \geq n/2$ and for each $i \in Y$, there exists an $l \in \{4n, 4n+1\}$ and an $x \geq m$ such that $W_{\mathbf{M}_i(\sigma_s)}$ enumerates $\langle l, x\rangle$.

4b. Let τ_0 be an extension of σ_s such that $\mathrm{content}(\tau_0) = [W_{p(0)}$ enumerated until now$]$. Go to substage 4b:0.

   Begin substage 4b:t

(* *Invariant: for all $i < n$, $j < n$, $[W_{p(i)}$ enumerated until now$] = [W_{q_j}$ enumerated until now$] = [W_{p(0)}$ enumerated until now$]$.* *)

4b.1. For each $i < n$, let $q^1_{n+i} = p(\text{avail} + 1 + i)$.
For each $i < n$, let $q^2_{n+i} = p(\text{avail} + n + 1 + i)$.
Set avail $=$ avail $+ 2n$.
Let $Z' \subseteq ([0 \ .. \ 2n-1] - Z)$ be such that $\text{card}(Z') = \lceil n/2 + 1/2 \rceil$ and, for all $i \in Z'$ and $j \in ([0 \ .. \ 2n-1] - (Z \cup Z'))$, $\mathbf{lmc}(\mathbf{M}_i, \tau_t) \leq \mathbf{lmc}(\mathbf{M}_j, \tau_t)$.

4b.2. Set $m_1 = 1 + \max(\{x \mid \{\langle 4n, x\rangle, \langle 4n+1, x\rangle\} \cap [W_{p(0)}$ enumerated until now$] \neq \emptyset\})$.
For each $i < n$, enumerate $[W_{p(0)}$ enumerated until now $]$ into $W_{q^1_{n+i}}$ and $W_{q^2_{n+i}}$.
For each $i < n$ and $j < n$, enumerate $\langle 3n+i, q^1_{n+i}\rangle$ in $W_{p(j)}$ and $W_{q^1_{n+j}}$.
For each $j < n$, enumerate $\langle 4n, m_1\rangle$ in $W_{p(j)}$ and $W_{q^1_{n+j}}$.
For each $i < n$ and $j < n$, enumerate $\langle 3n+i, q^2_{n+i}\rangle$ in W_{q_j} and $W_{q^2_{n+j}}$.
For each $j < n$, enumerate $\langle 4n+1, m_1\rangle$ in W_{q_j} and $W_{q^2_{n+j}}$.

4b.3. Search for a γ extending τ_t and $i \in Z'$ such that $\mathbf{M}_i(\gamma) \neq \mathbf{M}_i(\tau_t)$ and $\text{content}(\gamma) - \text{content}(\tau_t) \subseteq \{\langle 3n+i, q^1_{n+i}\rangle, \langle 3n+i, q^2_{n+i}\rangle \mid i < n\} \cup \{\langle 4n, m_1\rangle, \langle 4n+1, m_1\rangle\}$.

4b.4. If and when such a γ is found in step 4b.3.
Let $S = \text{content}(\gamma) \cup [W_{p(0)}$ enumerated until now$] \cup [W_{q_0}$ enumerated until now$]$.
For each $i < n$, enumerate S into $W_{p(i)}$ and W_{q_i}.
Let τ_{t+1} be an extension of γ such that $\text{content}(\tau_{t+1}) = S$.
Go to substage 4b:$(t+1)$.

End substage 4b:t

5. (* *Note that $\text{card}(Y) \geq n/2$. Thus at least $n/2$ grammars in Z are temporarily spoiled, whereas at most n grammars coded into the first $2n$ cylinders (denoted by X below) of the diagonalizing language are temporarily spoiled. The aim of this step is to inductively reduce the size of X in each iteration of the while loop, while ensuring that $\text{card}(Y) \geq \text{card}(X)/2$ at the end of each iteration. If successful, this would eventually give us $\text{card}(X) \geq 1$ and $\text{card}(Y) \geq 1$. Simultaneously, in each iteration, a diagonalization of the form: $\mathbf{Team}^{2n}_{3n}\mathbf{TxtEx}$ versus $\mathbf{Team}^{u}_{v}\mathbf{TxtEx}$, for $u > 2v/3$, is carried out. If the aim of step 5 is not successful, then this diagonalization will succeed, and give us a diagonalizing language.* *)

Let Y be as found in step 4a.
Set $v = 4n+2$. Set $X = [0 \ .. \ (n-1)]$.
while $\text{card}(X) > 1$ **do**

Let $S = \bigcup_{i \in ([0 \ .. \ 2n-1] - X)} [W_{p(i)}$ enumerated until now$]$.
For each $i \in ([0 \ .. \ 2n-1] - X)$, enumerate S in $W_{p(i)}$.

(* *Invariants maintained by the while loop at this point are:*

(i) $(\forall j, j' \in ([0 \ .. \ 2n-1] - X))[[W_{p(j)}$ enumerated until now$] = [W_{p(j')}$ enumerated until now$]]$.

(ii) $(\forall j \in Y)(\exists x \mid 4n \leq \pi_1(x) < v)[x \in W_{\mathbf{M}_j(\sigma_s)} \wedge (\forall j' \in [0 \ .. \ 2n-1] - X)[x \notin [W_{p(j')}$ enumerated until now $]]]$

(Note that invariant (ii) will also be satisfied when the loop is exited. Intuitively, this invariant means that currently machines in Y and the grammars coded into cylinders in X are temporarily "spoiled". And thus they cannot take part in the diagonalization in this iteration. These would remain spoiled till the end of current iteration, when X and Y are redefined.)

(iii) $\text{card}(Y) \geq \text{card}(X)/2$.

(iv) $\text{card}(X) \leq n$. *)

(* *Moreover, after each iteration of the while loop, $\text{card}(X)$ decreases (actually $\text{card}(X)$ nearly halves after each iteration)* *).

(* *Intuitively, the aim of this iteration is to temporarily spoil at least $\text{card}(X)/4$ machines in Z, while temporarily spoiling at most $\lceil \text{card}(X)/2 \rceil$ of the first $2n$ cylinders of the diagonalizing language. Simultaneously a diagonalization similar* *to* $\mathbf{Team}_{3n}^{2n}\mathbf{TxtEx}$ *versus* $\mathbf{Team}_{v}^{u}\mathbf{TxtEx}$*, for* $u > 2v/3$*, is carried out. This diagonalization uses cylinders* $([0 \,..\, 2n-1] \cup [2n \,..\, 3n + \mathrm{card}(X) - 1]) - X$ *in the diagonalizing languages for coding of grammars. Note that if the aim of temporarily spoiling at least* $\mathrm{card}(X)/4$ *machines in* Z *is not successful then the diagonalization will succeed.* $*$)

For each $i < \mathrm{card}(X)$, let $q_i = p(\mathrm{avail} + 1 + i)$.

Set $\mathrm{avail} = \mathrm{avail} + \mathrm{card}(X)$.

Let $X_1, X_2 \subseteq ([0 \,..\, 2n-1] - X)$ be such that, $\mathrm{card}(X_1) = \lfloor \mathrm{card}(X)/2 \rfloor$, $\mathrm{card}(X_2) = \lceil \mathrm{card}(X)/2 \rceil$ and $X_1 \cap X_2 = \emptyset$.

For $i \in X_1$ and $j < \mathrm{card}(X)$, enumerate $[W_{p(i)}$ enumerated until now$]$ into W_{q_j}.

For each $i < \mathrm{card}(X)$ and $j \in ([0 \,..\, 2n-1] - X)$ and $k < \mathrm{card}(X)$, enumerate $\langle 2n+i, q_i \rangle$ in $W_{p(j)}$ and W_{q_k}.

5a. Let τ_0 be an extension of σ_s such that $\mathrm{content}(\tau_0) = [W_{q_0}$ enumerated until now$]$.

Go to substage 5:0.

Begin substage 5:t

(∗ *Invariant: for all* $i \in [0 \,..\, 2n-1] - X$*, for all* $j < \mathrm{card}(X)$*,* $[W_{p(i)}$ *enumerated until now* $] = [W_{q_j}$ *enumerated until now*$]$. ∗)

For each $i < 2n - \mathrm{card}(X)$, let $q^1_{\mathrm{card}(X)+i} = p(\mathrm{avail} + 1 + i)$.

For each $i < 2n - \mathrm{card}(X)$, let $q^2_{\mathrm{card}(X)+i} = p(\mathrm{avail} + 2n - \mathrm{card}(X) + 1 + i)$.

Set $\mathrm{avail} = \mathrm{avail} + 4n - (2 \cdot \mathrm{card}(X))$.

Let $Z' \subseteq ([0 \,..\, 2n-1] - Z)$ be such that $\mathrm{card}(Z') = \mathrm{card}(Y)$ and, for all $i \in Z'$ and $j \in ([0 \,..\, 2n-1] - (Z \cup Z'))$, $\mathbf{lmc}(\mathbf{M}_i, \tau_t) \leq \mathbf{lmc}(\mathbf{M}_j, \tau_t)$.

Set $m_1 = 1 + \max(\{x \mid \{\langle v, x\rangle, \langle v+1, x\rangle\} \cap [W_{q_0}$ enumerated until now$] \neq \emptyset\})$.

For each $i < 2n - \mathrm{card}(X)$, enumerate $[W_{q_0}$ enumerated until now$]$ into $W_{q^1_{\mathrm{card}(X)+i}}$ and $W_{q^2_{\mathrm{card}(X)+i}}$.

For each $i < 2n - \mathrm{card}(X)$, $j \in X_1$ and $k < \mathrm{card}(X_2)$, enumerate $\langle 2n + \mathrm{card}(X) + i, q^1_{\mathrm{card}(X)+i} \rangle$ in $W_{p(j)}, W_{q_k}, W_{q^1_{\mathrm{card}(X)+i}}$.

For each $i < 2n - \mathrm{card}(X)$, $j \in X_1$ and $k < \mathrm{card}(X_2)$, enumerate $\langle v, m_1 \rangle$ in $W_{p(j)}, W_{q_k}, W_{q^1_{\mathrm{card}(X)+i}}$.

For each $i < 2n - \mathrm{card}(X)$, $j \in X_2$ and $k < \mathrm{card}(X_1)$, enumerate $\langle 2n + \mathrm{card}(X) + i, q^2_{\mathrm{card}(X)+i} \rangle$ in $W_{p(j)}, W_{q_{\mathrm{card}(X_2)+k}}, W_{q^2_{\mathrm{card}(X)+i}}$.

For each $i < 2n - \mathrm{card}(X)$, $j \in X_2$ and $k < \mathrm{card}(X_1)$, enumerate $\langle v+1, m_1 \rangle$ in $W_{p(j)}, W_{q_{\mathrm{card}(X_2)+k}}, W_{q^2_{\mathrm{card}(X)+i}}$.

Dovetail steps 5a.1 and 5a.2 until, if ever, one of them succeeds. If step 5a.1 succeeds before step 5a.2 does, if ever, then go to step 5b. If step 5a.2 succeeds before step 5a.1 does, if ever, then go to step 5a.3.

5a.1. Search for a $Y' \subseteq (Z - Y)$, such that $\mathrm{card}(Y') = \mathrm{card}(Y)$ and, for each $i \in Y'$, there exists an $l \in \{v, v+1\}$ and an $x \geq m_1$ such that $W_{\mathbf{M}_i(\sigma_s)}$ enumerates $\langle l, x \rangle$.

5a.2. Search for an extension γ of τ_t and an $i \in Z'$ such that $\mathbf{M}_i(\tau_t) \neq \mathbf{M}_i(\gamma)$ and $\mathrm{content}(\gamma) - \mathrm{content}(\tau_t) \subseteq \{\langle 2n + \mathrm{card}(X) + i, q^1_{\mathrm{card}(X)+i}\rangle, \langle 2n + \mathrm{card}(X) + i, q^2_{\mathrm{card}(X)+i}\rangle \mid i < 2n - \mathrm{card}(X)\} \cup \{\langle v, m_1\rangle, \langle v+1, m_1\rangle\}$.

5a.3. Let γ be as found in step 5a.2.

Let $S = \mathrm{content}(\gamma) \cup [W_{q_0}$ enumerated until now$] \cup [W_{q_{\mathrm{card}(X)-1}}$ enumerated until now$]$.

For each $j \in [0 \,..\, 2n-1] - X$, enumerate S into $W_{p(j)}$.

For each $i < \mathrm{card}(X)$, enumerate S into W_{q_i}.

Let τ_{t+1} be an extension of γ such that $\mathrm{content}(\tau_{t+1}) = S$.

Go to substage 5:$t+1$.

End substage 5:t

5b. Let Y' be as found in step 5a.1.

Set $Y_1 = \{i \in Y' \mid W_{\mathbf{M}_i(\sigma_s)}$ enumerates $\langle v, x\rangle$ for some $x \geq m_1$ as observed in step

5a.1}.

Set $Y_2 = \{i \in Y' - Y_1 \mid W_{\mathbf{M}_i(\sigma_s)}$ enumerates $\langle v+1, x\rangle$ for some $x \geq m_1$ as observed in step 5a.1}.

if $\mathrm{card}(Y_1)/\mathrm{card}(X_1) \geq 1/2$, **then** set $X = X_1, Y = Y_1$.

else set $X = X_2, Y = Y_2$.

endif

Set $v = v + 2$.

endwhile

6. (* *Note that* $\mathrm{card}(X) = 1$ *and* $\mathrm{card}(Y) \geq 1$. *Thus at least one of the machines in* Z *is spoiled, whereas only one of the grammars coded into the first* $2n$ *cylinders of the diagonalizing language is spoiled. Now it is possible to do a diagonalization of the form* $\mathbf{Team}^{2n}_{4n-1}\mathbf{TxtEx}$ *versus* $\mathbf{Team}^{n}_{2n-1}\mathbf{TxtEx}$. *)

Set $v = v + 2$.

Set $q_0 = p(\mathrm{avail} + 1)$.

Set $\mathrm{avail} = \mathrm{avail} + 1$.

Let $i_0, i_1, \ldots, i_{2n-1}$ be such that $\{p(i_j) \mid j < 2n\} = \{p(j) \mid j \in ([0 \,..\, 2n-1] - X)\} \cup \{q_0\}$.

Let $S = \{\langle 2n, q_0\rangle\} \cup \bigcup_{i \in [0 \,..\, 2n-1]-X}[W_{p(i)}$ enumerated until now].

For each $i \in ([0 \,..\, 2n-1] - X)$, enumerate S into $W_{p(i)}$ and W_{q_0}.

Let τ_0 be an extension of σ_s such that $\mathrm{content}(\tau_0) = W_{p(0)}$ enumerated until now.

Go to substage 6:0.

Begin substage 6:t

For each $i < 2n-1$ and $j < 2n$, let $q^j_{1+i} = p(\mathrm{avail} + 1 + j \cdot (2n-1) + i)$.

Set $\mathrm{avail} = \mathrm{avail} + 2n \cdot (2n-1)$.

Let $Y' \subseteq ([0 \,..\, 2n-1] - Z)$ be such that $\mathrm{card}(Y') = \mathrm{card}(Y)$ and, for $i \in Y'$ and $j \in ([0 \,..\, 2n-1] - (Z \cup Y'))$, $\mathbf{lmc}(\mathbf{M}_i, \tau_t) \leq \mathbf{lmc}(\mathbf{M}_j, \tau_t)$.

For each $i < 2n-1$, $j < 2n$, enumerate $\langle 2n+1+i, q^j_{1+i}\rangle$ in $W_{p(i_j)}$.

Set $m_1 = 1 + \max(\{x \mid (\exists w, j \mid j < 2n)[\langle w, x\rangle \in [W_{p(i_j)}$ enumerated until now]]}).

For each $j < 2n$, enumerate $\langle v + j, m_1\rangle$ in $W_{p(i_j)}$.

For each $j < 2n$ and $i < 2n-1$, enumerate [$W_{p(i_j)}$ enumerated until now] into $W_{q^j_{1+i}}$.

6a. Search for an extension γ of τ_t and $i \in ((Z \cup Y') - Y)$, such that $\mathbf{M}_i(\tau_t) \neq \mathbf{M}_i(\gamma)$ and $\mathrm{content}(\gamma) - \mathrm{content}(\tau_t) \subseteq \{\langle 2n+1+i, q^j_{1+i}\rangle \mid j < 2n \wedge i < 2n-1\} \cup \{\langle v+j, m_1\rangle \mid j < 2n\}$.

6b. Let γ be as found in step 6a.

Let $S = \mathrm{content}(\gamma) \cup \bigcup_{j<2n}[W_{p(i_j)}$ enumerated until now].

For each $j < 2n$, enumerate S into $W_{p(i_j)}$.

Let τ_{t+1} be an extension of τ_t such that $\mathrm{content}(\tau_{t+1}) = S$.

Go to substage 6:$t+1$.

End substage 6:t

7. If and when step 3 succeeds, let τ be as found in step 3.

Let $S = \mathrm{content}(\tau) \cup \bigcup_{i<2n}[W_{p(i)}$ enumerated until now].

For each $i < 2n$, enumerate S into $W_{p(i)}$.

Let σ_{s+1} be as extension of τ such that $\mathrm{content}(\sigma_{s+1}) = S$.

Set $\mathrm{avail} = \max(\{\mathrm{avail}\} \cup \{x \mid (\exists i < 4n)[\langle i, p(x)\rangle \in S]\})$.

Go to stage $s+1$.

End stage s

Now we consider the following cases.

Case 1: All stages terminate.

In this case, clearly $W_{p(0)} = W_{p(1)} = W_{p(2)} = \ldots = W_{p(2n-1)}$. Let $L = W_{p(0)}$. Clearly, for $i < 2n$, $\max(\{x \mid \langle i, x\rangle \in L\}) = p(i)$. Thus $L \in \mathcal{L}$. Also $T = \bigcup_s \sigma_s$ is a text for L. However at most $n-1$ of the machines $\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_{2n-1}$ converge on T.

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

Let Z be as defined in stage s. Now for $i \in Z$ and any text T such that $\sigma_s \subseteq T$, and $\mathrm{content}(T) \subseteq \mathrm{content}(\sigma_s) \cup \{\langle x, y\rangle \mid x \geq 2n, y \in N\}$, $\mathbf{M}_i(T) = \mathbf{M}_i(\sigma_s)$. We now consider following subcases. All step numbers and substages referred to below stand for the corresponding steps and substages in stage s.

Case 2.1: In stage s the procedure enters but does not leave step 4.

For each $i < n$, let q_i be as defined in step 4. Let m be as defined in step 4. Note that the number of i's in Z, such that $(\exists x \geq m)(\exists l \in \{4n, 4n+1\})[\langle l, x\rangle \in W_{\mathbf{M}_i(\sigma_s)}]$ is less than $n/2$. Let τ_t be as defined in step 4b.

Case 2.1.1: All substages at step 4b terminate.

In this case, clearly for $i < n$ and $j < n$, $W_{p(i)} = W_{q_j}$. Let $L = W_{p(0)}$. Clearly, $L \in \mathcal{L}$. Moreover $\{\langle 4n, x\rangle \mid \langle 4n, x\rangle \in L\}$ is infinite. Also because step 4a does not succeed and step 4b.3 succeeds infinitely often, $\mathrm{card}(\{i < 2n \mid \mathbf{M}_i\ \mathbf{TxtEx}$ identifies $L\}) < (\lceil n/2 + 1/2\rceil - 1) + n/2$. Thus $L \notin \mathbf{Team}^n_{2n}\mathbf{TxtEx}(\{\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_{2n-1}\})$.

Case 2.1.2: Some substage 4b:t at step 4b starts but does not terminate.

In this case, for $i < n$, let q^1_{n+i}, q^2_{n+i}, be as defined in step 4b.1 of substage 4b:t. Clearly, $W_{p(0)} = W_{p(1)} = \cdots = W_{p(n-1)} = W_{q^1_n} = W_{q^1_{n+1}} = \ldots = W_{q^1_{2n-1}}$ and $W_{q_0} = W_{q_1} = \cdots = W_{q_{n-1}} = W_{q^2_n} = W_{q^2_{n+1}} = \ldots = W_{q^2_{2n-1}}$. Let $L_1 = W_{p(0)}$ and $L_2 = W_{q_0}$. It is easy to see that $L_1, L_2 \in \mathcal{L}$ and $L_1 \neq L_2$. Moreover, for all $i \in Z \cup Z'$, for any text T for L_1 or L_2 such that $\tau_t \subseteq T$, $\mathbf{M}_i(T) = \mathbf{M}_i(\tau_t)$. This, along with the fact that step 4a does not succeed, implies that at least one of L_1 or L_2 is $\mathbf{TxtEx}$-identified by less than $n - \lceil n/2 + 1/2\rceil + \frac{n/2 + \lceil n/2 + 1/2\rceil}{2}$ of the machines in $\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_{2n-1}$.

Case 2.2: In stage s the procedure reaches step 5 but does not reach step 6.

Let X, Y be as in the last iteration of the while loop which is (partly) executed in step 5. Also for at least $\mathrm{card}(Y)$ many i's in Z, $W_{\mathbf{M}_i(\sigma_s)}$ enumerates some element (since step 4a/5a.1 (in the previous while loop) succeeded) which is neither in the language L defined in Case 2.2.1 below, nor in L_1 or L_2 defined in Case 2.2.2 below; thus, $\mathbf{M}_i$ does not $\mathbf{TxtEx}$-identify either of the languages L, L_1 and L_2. For each $i < \mathrm{card}(X)$, let q_i be as defined in the last iteration of the while loop in step 5. Let τ_t be as defined in the last iteration of the while loop in step 5.

Case 2.2.1: All substages in the last iteration of the while loop in step 5 terminate.

In this case, clearly for $i \in ([0\,..\,2n-1] - X)$ and $j < \mathrm{card}(X)$, $W_{p(i)} = W_{q_j}$. Let $L = W_{q_0}$. Clearly, $L \in \mathcal{L}$. Let $T = \bigcup_t \tau_t$. It is easy to verify that T is a text for L. Moreover, for less than $\mathrm{card}(Y)$ many i's in $([0\,..\,2n-1] - Z)$, $\mathbf{M}_i$ converges on T. Thus, since there are at least $\mathrm{card}(Y)$ many i's in Z such that $W_{\mathbf{M}_i(\sigma_s)}$ enumerates some element which is not in L (since step 4a/5a.1 (in the previous while loop) succeeded and the invariants of the while loop are satisfied), we have $\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_{2n-1}$ do not $\mathbf{Team}^n_{2n}\mathbf{TxtEx}$-identify L.

Case 2.2.2: Some substage 5:t in step 5 starts but does not terminate.

In this case, for $i < (2n - \mathrm{card}(X))$, let $q^1_{\mathrm{card}(X)+i}$ and $q^2_{\mathrm{card}(X)+i}$ be as defined in substage 5:t of the last iteration of the while loop in step 5. Clearly, for $i \in X_1$, $j < \mathrm{card}(X_2)$ and $k < 2n - \mathrm{card}(X)$, $W_{p(i)} = W_{q_j} = W_{q^1_{\mathrm{card}(X)+k}}$. Also, for $i \in X_2$, $j < \mathrm{card}(X_1)$ and $k < 2n - \mathrm{card}(X)$, $W_{p(i)} = W_{q_{\mathrm{card}(X_2)+j}} = W_{q^2_{\mathrm{card}(X)+k}}$. Let $L_1 =$

W_{q_0} and $L_2 = W_{q_{\mathrm{card}(X)-1}}$. Clearly, both L_1 and L_2 are members of $\mathcal{L}$. Also, $L_1 \neq L_2$.

Also since steps 5a.1, 5a.2 do not succeed in substage 5:t, arguing in a way similar to that in case 2.1.2 we have that, at least one of L_1, L_2 is **TxtEx**-identified by less than n many machines in $\{\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_{2n-1}\}$.

Case 2.3: In stage s the procedure reaches step 6.

In this case, for each $i \in Y$, $W_{\mathbf{M}_i(\sigma_s)}$ enumerates an element (due to completion of all iterations of the while loop in step 5) which neither is in the language, L, defined in Case 2.3.1 below nor belongs to any language in $\{L_j \mid j < 2n-1\}$ defined in Case 2.3.2 below; thus, $\mathbf{M}_i$ does not **TxtEx** identify either L or any language in $\{L_j \mid j < 2n-1\}$. Let τ_t be as defined in step 6.

Case 2.3.1: All substages in step 6 terminate.

In this case clearly, for $i \in ([0 \, .. \, 2n-1] - X)$, $W_{p(i)} = W_{q_0}$. Let $L = W_{q_0}$. Clearly, $L \in \mathcal{L}$. Let $T = \bigcup_t \tau_t$. Now, the number of i's in $([0 \, .. \, 2n-1] - Z)$ such that $\mathbf{M}_i$ converges on T is $<$ card(Y). Moreover, at least card(Y) of the machines in Z converge to incorrect grammars (note that the invariant (ii) at the beginning of the while loop in step 5 is also satisfied when the loop is exited). Thus, $L \notin \mathbf{Team}^n_{2n}(\{\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_{2n-1}\})$.

Case 2.3.2: Some substage 6:t at step 6 starts but does not terminate.

In this case for $j < 2n$ and $i < 2n-1$, let q^j_{1+i} be as defined in substage 6:t. Also, let $i_0, \ldots, i_{2n-1}$ be as defined in step 6. Clearly, for $j < 2n$ and $i < 2n-1$, $W_{p(i_j)} = W_{q^j_{1+i}}$. Let $L_j = W_{p(i_j)}$. Clearly, each of the languages in $\{L_i \mid i < 2n\}$ belong to $\mathcal{L}$ and are pairwise distinct. Now for $i < 2n$, let T_i be a text for L_i such that $\tau_t \subseteq T_i$. Now it is easy to verify that, for each $j \in Z \cup Y'$ and $i < 2n$, $\mathbf{M}_j(T_i) = \mathbf{M}_j(\tau_t)$. Since, for each $j \in ((Z \cup Y') - Y)$, $\mathbf{M}_j(\tau_t)$, can each be grammars for at most one of $L_0, L_1, \ldots, L_{2n-1}$, we have that $\{L_0, L_1, \ldots, L_{2n-1}\} \not\subseteq \mathbf{Team}^n_{2n}(\{\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_{2n-1}\})$.

From the above cases it follows that $\mathcal{L} \notin \mathbf{Team}^n_{2n}\mathbf{TxtEx}$. ∎

The above diagonalization can be generalized to show the following.

Theorem 13 $(\forall n, m \in N^+ \mid 2n$ *does not divide* $m)[\mathbf{Team}^{2n}_{4n}\mathbf{TxtEx} - \mathbf{Team}^m_{2m}\mathbf{TxtEx} \neq \emptyset]$.

We omit a proof of the theorem because a simple modification of our proof of Theorem 12 suffices. The only changes required are that in the diagonalization procedure instead of searching for $\geq r$ machines to converge to a grammar (or, for $\geq r$ converged grammars to output a particular value), we search for $\geq r \cdot m/n$ machines (or, grammars) in this case. Thus, at the end of step 5, we will have at least $\lceil \frac{m}{2n} \rceil$ of the m converged machines converge to a grammar which enumerates something 'extra.' Step 6 then utilizes the fact that $\mathbf{Team}^{2n}_{4n-1}\mathbf{TxtEx}$ can diagonalize against $\mathbf{Team}^r_w\mathbf{TxtEx}$, if $r/w > 2n/(4n-1)$. We leave the details to the reader.

Corollary 10 $(\forall m, n \in N^+)[\mathbf{Team}^m_{2m}\mathbf{TxtEx} \subseteq \mathbf{Team}^n_{2n}\mathbf{TxtEx} \Leftrightarrow [m$ *divides* $n \bigvee m$ *is odd*$]]$.

Corollary 11 $\mathbf{Prob}^{1/2}\mathbf{TxtEx} - \bigcup_m \mathbf{Team}^m_{2m}\mathbf{TxtEx} \neq \emptyset$.

PROOF. Let $\mathcal{L}$ be defined as follows. Let $\mathcal{L}_n = \{L \mid \mathrm{card}(\{i < 4n \mid \max(\{x \mid \langle i, x\rangle \in L\}) = x_0 < \infty \wedge W_{x_0} = L\}) \geq 2n\}$.

Let $\mathcal{L} = \{L' \mid (\exists n, L \in \mathcal{L}_n)[L' = \{\langle 0, n\rangle\} \cup \{\langle 1, x\rangle \mid x \in L\}]\}$. It is easy to observe that $\mathcal{L} \in \mathbf{Prob}^{1/2}\mathbf{TxtEx}$. By a simple modification of our proof of Theorem 12 it can be shown that $\mathcal{L} \notin \bigcup_m \mathbf{Team}^m_{2m}\mathbf{TxtEx}$. ∎

The above corollary 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. In Corollary 13, we establish a similar result for the ratio $1/k, k > 2$.

5.4 Team Language Identification for Success Ratio $\frac{1}{k}$, $k > 2$.

We now employ Theorem 7 to deduce the following using Theorem 13.

Theorem 14 $(\forall k \geq 2)(\forall$ *even* $j > 1)(\forall i \mid j$ *does not divide* $i)[\mathbf{Team}^{j}_{j\cdot k}\mathbf{TxtEx} - \mathbf{Team}^{i}_{i\cdot k}\mathbf{TxtEx} \neq \emptyset]$.

Proof. By Induction on k. Note that base case ($k = 2$) follows by Theorem 13. Now suppose $\mathbf{Team}^{j}_{jk}\mathbf{TxtEx} - \mathbf{Team}^{i}_{ik}\mathbf{TxtEx} \neq \emptyset$. Using Theorem 7 with $i_1 = 0$, we have $\mathbf{Team}^{j}_{(k+1)j}\mathbf{TxtEx} - \mathbf{Team}^{i}_{(k+1)i}\mathbf{TxtEx} \neq \emptyset$. ∎

We do not know if the above theorem can be extended to show that, $(\forall k \geq 2)(\forall$ even $j > 1)(\forall i \mid j$ does not divide $i)[\mathbf{Team}^{j}_{j\cdot k}\mathbf{TxtEx} - \mathbf{Team}^{i}_{i\cdot k}\mathbf{TxtEx}^{*} \neq \emptyset]$.

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

Corollary 13 $(\forall k \geq 2)[\mathbf{Prob}^{1/k}\mathbf{TxtEx} - \bigcup_j \mathbf{Team}^{j}_{j\cdot k}\mathbf{TxtEx} \neq \emptyset]$.

The above Corollary can be proved using a trick similar to that used to prove Corollary 11. We omit the details.

We next present some more applications of Theorems 7 and 8.

Theorem 15 *For each* $m > n \in N^{+}$, $r \geq 3$
$\mathbf{Team}^{m}_{r\cdot m}\mathbf{TxtEx} - \mathbf{Team}^{n}_{r\cdot n}\mathbf{TxtEx} \neq \emptyset$.

Proof. If m is even then the theorem follows from Theorem 14. Suppose m is odd. Then by Theorem 14, $\mathbf{Team}^{m+1}_{2m+2}\mathbf{TxtEx} - \mathbf{Team}^{n}_{2n}\mathbf{TxtEx} \neq \emptyset$. Thus, we have $\mathbf{Team}^{m}_{2m+1}\mathbf{TxtEx} - \mathbf{Team}^{n}_{2n}\mathbf{TxtEx} \neq \emptyset$. Using Theorem 7 with $i_1 = 1$, we get $\mathbf{Team}^{m}_{3m}\mathbf{TxtEx} - \mathbf{Team}^{n}_{3n}\mathbf{TxtEx} \neq \emptyset$. Now using Theorem 7 repeatedly with $i_1 = 0$ we get the result. ∎

Theorem 16 *For each* $r \in N$, $\mathbf{Team}^{3}_{3+2r}\mathbf{TxtEx} - \mathbf{Team}^{2}_{2r}\mathbf{TxtEx}^{*} \neq \emptyset$.

Proof. The theorem is trivially true for $r = 0$. Since $\mathbf{Team}^{2}_{3}\mathbf{TxtEx} - \mathbf{TxtEx}^{*} \neq \emptyset$ and $\mathbf{Team}^{2}_{3}\mathbf{TxtEx} \subseteq \mathbf{Team}^{1}_{2}\mathbf{TxtEx}$, we have $\mathbf{Team}^{3}_{5}\mathbf{TxtEx} - \mathbf{Team}^{2}_{2}\mathbf{TxtEx}^{*} \neq \emptyset$. Using Theorem 8 repeatedly with $i_1 = 1$, we get $\mathbf{Team}^{3}_{3+2r}\mathbf{TxtEx} - \mathbf{Team}^{2}_{2r}\mathbf{TxtEx}^{*} \neq \emptyset$, for $r \geq 1$. ∎

Theorem 17 *For each* $r \geq 3$, $\mathbf{Team}^{3}_{3r}\mathbf{TxtEx} - \mathbf{Team}^{j}_{jr}\mathbf{TxtEx} \neq \emptyset$, *if* j *is not divisible by 3.*

Proof. As a Corollary to Theorem 19 below we have $\mathbf{Team}^{3}_{5}\mathbf{TxtEx} - \mathbf{Team}^{j}_{\lfloor \frac{5j}{3} \rfloor}\mathbf{TxtEx} \neq \emptyset$. Using Theorem 7 with $i_1 = 1$, we get $\mathbf{Team}^{3}_{7}\mathbf{TxtEx} - \mathbf{Team}^{j}_{\lfloor \frac{5j}{3} \rfloor + \lceil 2j/3 \rceil}\mathbf{TxtEx} \neq \emptyset$, and then $\mathbf{Team}^{3}_{9}\mathbf{TxtEx} - \mathbf{Team}^{j}_{3j}\mathbf{TxtEx} \neq \emptyset$. Now again using Theorem 7 repeatedly with $i_1 = 0$, we get $\mathbf{Team}^{3}_{3r}\mathbf{TxtEx} - \mathbf{Team}^{j}_{jr}\mathbf{TxtEx} \neq \emptyset$, for $r \geq 3$. ∎

A generalization of the above theorem shows that

Theorem 18 *For all* i, *for each* $r \geq i$, $\mathbf{Team}^{i}_{i\cdot r}\mathbf{TxtEx} - \mathbf{Team}^{j}_{j\cdot r}\mathbf{TxtEx} \neq \emptyset$, *if* j *is not divisible by* i.

5.5 On the Difficulty of Obtaining General Results

Despite the useful tools of Section 5.2, general results are difficult to come by for success ratios $< 1/2$ and for between success ratios $1/2$ and $2/3$. In this section, we present two results: the first (Theorem 19) illustrates the kind of results that we can obtain (using the methods of section 5.2), the second (Theorem 21) sheds light on why general results are difficult to obtain.

Corollary 14 below gives a hierarchy when more than half of the team members are required to be successful.

Theorem 19 *Suppose* $n < \lceil m \cdot \frac{2r+1}{r+1} \rceil$. $\mathbf{Team}^{r+1}_{2r+1}\mathbf{TxtEx} - \mathbf{Team}^{m}_{n}\mathbf{TxtEx}^* \neq \emptyset$.

PROOF. Clearly, $\mathbf{Team}^{r+1}_{r+1}\mathbf{TxtEx} - \mathbf{Team}^{\lceil \frac{mr}{r+1} \rceil}_{n-m}\mathbf{TxtEx}^* \neq \emptyset$ (since $\lceil \frac{mr}{r+1} \rceil > n-m$). The theorem now follows by using Theorem 8 with $i_1 = 1$. ∎

Corollary 14 $(\forall r)[\mathbf{Team}^{r+2}_{2r+3}\mathbf{TxtEx} - \mathbf{Team}^{r+1}_{2r+1}\mathbf{TxtEx}^* \neq \emptyset]$.

A generalization of a detailed proof of Theorem 19 can be used to show the following Theorem 20. We omit the details.

Theorem 20 $(\forall p, r \mid p > \frac{r+1}{2r+1})[\mathbf{Team}^{r+1}_{2r+1}\mathbf{TxtEx} - \mathbf{Prob}^{p}\mathbf{TxtEx} \neq \emptyset]$.

Theorem 21 below shows that there exist i, j, k, l such that

$$\mathbf{Team}^{i}_{j}\mathbf{TxtEx} = \mathbf{Team}^{k}_{l}\mathbf{TxtEx} \text{ for } \frac{i}{j} \neq \frac{k}{l}, \text{ and both } \frac{i}{j} \text{ and } \frac{k}{l} \text{ are } \leq \frac{2}{3}.$$

Thus, we *cannot* hope to prove a general theorem which separates $\mathbf{Team}^{i}_{j}\mathbf{TxtEx}$ and $\mathbf{Team}^{k}_{l}\mathbf{TxtEx}$ whenever $\frac{i}{j} \neq \frac{k}{l}$.

Theorem 21 $(\forall i, j \mid i/j > 5/8)[\mathbf{Team}^{i}_{j}\mathbf{TxtEx} \subseteq \mathbf{Team}^{2}_{3}\mathbf{TxtEx}]$.

Corollary 15 $(\forall i, j \mid 5/8 < i/j \leq 2/3)[\mathbf{Team}^{i}_{j}\mathbf{TxtEx} = \mathbf{Team}^{2}_{3}\mathbf{TxtEx}]$.

In [20, Lemma 4], the following lemma was established.

Lemma 4 *Suppose* $r, w \in N$ *are given such that* $r \geq w > 2r/5$. *There exist recursive functions* G_1 *and* G_2 *such that,* $(\forall p_1, p_2, \ldots, p_r)(\forall L)[\text{card}(\{i \mid 1 \leq i \leq r \ \wedge \ W_{p_i} = L\}) \geq w \Rightarrow W_{G_1(p_1,\ldots,p_r)} = L \ \vee \ W_{G_2(p_1,\ldots,p_r)} = L]$.

The proof of the above lemma actually established the following stronger result.

Lemma 5 *Suppose* $r, w \in N$ *are given such that* $r \geq w > 2r/5$. *There exist recursive functions* G_1 *and* G_2 *such that,* $(\forall p_1, p_2, \ldots, p_r)(\forall L)[\text{card}(\{i \mid 1 \leq i \leq r \ \wedge \ W_{p_i} = L\}) \geq w \Rightarrow W_{G_1(p_1,\ldots,p_r)} = L \vee W_{G_2(p_1,\ldots,p_r)} = L]$. *In addition,* $(\forall x \in W_{G_1(p_1,\ldots,p_r)} \cup W_{G_2(p_1,\ldots,p_r)})[\text{card}(\{i \mid 1 \leq i \leq r \ \wedge \ x \in W_{p_i}\}) \geq w]$.

The above lemma can be extended to obtain the following.

Lemma 6 *Suppose* $r, w \in N$ *are given such that* $r \geq w > 2r/5$. *There exist recursive functions* G'_1 *and* G'_2 *such that,* $(\forall p_1, p_2, \ldots, p_r)(\forall L)[\text{card}(\{i \mid 1 \leq i \leq r \ \wedge \ W_{p_i} = L\}) \geq w \Rightarrow W_{G'_1(p_1,\ldots,p_r)} = L \ \vee \ W_{G'_2(p_1,\ldots,p_r)} = L]$. *Moreover if* $\text{card}(\{i \mid 1 \leq i \leq r \ \wedge \ W_{p_i} = L\}) > 3r/5$, *then* $W_{G'_1(p_1,\ldots,p_r)} = W_{G'_2(p_1,\ldots,p_r)} = L$.

PROOF. Let G_1, G_2 be as given by Lemma 5. By s-m-n, there exist recursive G'_1, G'_2 such that the following holds.

$W_{G'_1(p_1,p_2,\ldots,p_r)} = W_{G_1(p_1,p_2,\ldots,p_r)} \cup \{x \mid \text{card}(\{s \mid 1 \leq s \leq r \ \wedge \ x \in W_{p_s}\}) > 3r/5\}$.
$W_{G'_2(p_1,p_2,\ldots,p_r)} = W_{G_2(p_1,p_2,\ldots,p_r)} \cup \{x \mid \text{card}(\{s \mid 1 \leq s \leq r \ \wedge \ x \in W_{p_s}\}) > 3r/5\}$.
Using Lemma 5, it is easy to see that G'_1, G'_2 satisfy the properties claimed. ∎

Now using the above lemma, we give a proof of Theorem 21.

Proof of Theorem 21. Suppose machines $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_j$ are given. We define $\mathbf{M}_1', \mathbf{M}_2', \mathbf{M}_3'$ as follows. Let G_1', G_2' be as given by Lemma 6, for $r = j, w = i$. Let conv be as defined in the proof of Theorem 2. Let $m_1^\sigma, m_2^\sigma, \ldots, m_k^\sigma$ be a permutation of $1, 2, \ldots, k$, such that, for $1 \leq r < k$, $[(\mathrm{conv}(\mathbf{M}_{m_r^\sigma}, \sigma), m_r^\sigma) < (\mathrm{conv}(\mathbf{M}_{m_{r+1}^\sigma}, \sigma), m_{r+1}^\sigma)]$. Note that, according to our notation, the "$<$" in the previous expression refers to ordering on pairs.

Let $\mathbf{M}_1'(\sigma) = G_1'(\mathbf{M}_{m_1^\sigma}(\sigma), \mathbf{M}_{m_2^\sigma}(\sigma), \ldots, \mathbf{M}_{m_i^\sigma}(\sigma))$.
Let $\mathbf{M}_2'(\sigma) = G_2'(\mathbf{M}_{m_1^\sigma}(\sigma), \mathbf{M}_{m_2^\sigma}(\sigma), \ldots, \mathbf{M}_{m_i^\sigma}(\sigma))$.
Let $\mathbf{M}_3'(\sigma) = \mathrm{majority}(\mathbf{M}_{m_1^\sigma}(\sigma), \mathbf{M}_{m_2^\sigma}(\sigma), \ldots, \mathbf{M}_{m^\sigma_{\lceil (7j)/8 \rceil}}(\sigma))$.

Now suppose $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_j$, $\mathbf{Team}_j^i\mathbf{TxtEx}$-identify L and T is a text for L. Clearly, if at least i of the j machines, $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_j$, $\mathbf{TxtEx}$-identify T, then at least $2i - j > 2j/5$ of the first i converging machines in $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_j$, $\mathbf{TxtEx}$-identify T. Thus by Lemma 6 it follows that at least one of $\mathbf{M}_1', \mathbf{M}_2'$ $\mathbf{TxtEx}$-identifies T. Moreover, if at least $3i/5$ of the first i converging machines in $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_j$, $\mathbf{TxtEx}$-identify L, then both $\mathbf{M}_1', \mathbf{M}_2'$ $\mathbf{TxtEx}$-identify T. On the other hand if fewer than $3i/5$ of the first i converging machines in $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_j$, $\mathbf{TxtEx}$-identify T, then at least $7j/8$ of the machines in $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_j$, converge on T, and thus $\mathbf{M}_3'$ $\mathbf{TxtEx}$-identifies T. The theorem follows. ∎

A generalization of the above method can be used to show that,

Theorem 22 $(\forall p > 5/8)[\mathbf{Prob}^p\mathbf{TxtEx} \subseteq \mathbf{Team}_3^2\mathbf{TxtEx}]$.

Theorem 23 $(\forall l_1, l_2, k_1, k_2 \geq 1 \mid l_2 \geq 5l_1/2 - 1, k_2 < 3k_1/2 + \lceil \frac{k_1(l_1-1)}{l_1} \rceil)[\mathbf{Team}_{l_2}^{l_1}\mathbf{TxtEx} - \mathbf{Team}_{k_2}^{k_1}\mathbf{TxtEx} \neq \emptyset]$.

Proof. Since $l_1/(l_2 - l_1 + 1) \leq 2/3$ and $k_1/(k_2 - \lceil \frac{k_1 \cdot (l_1-1)}{l_1} \rceil) > 2/3$, we have, $\mathbf{Team}_{l_2-l_1+1}^{l_1}\mathbf{TxtEx} - \mathbf{Team}^{k_1}_{k_2 - \lceil \frac{k_1 \cdot (l_1-1)}{l_1} \rceil}\mathbf{TxtEx} \neq \emptyset$. Now using Theorem 7 with $i_1 = 1$, we get $\mathbf{Team}_{l_2}^{l_1}\mathbf{TxtEx} - \mathbf{Team}_{k_2}^{k_1}\mathbf{TxtEx} \neq \emptyset$. ∎

Iterating the above method we get,

Theorem 24 $(\forall w)(\forall l_1, l_2, k_1, k_2 \geq 1 \mid l_2 \geq \frac{3l_1}{2} + w(l_1 - 1) \wedge k_2 < \frac{3k_1}{2} + w \cdot \lceil \frac{k_1(l_1-1)}{l_1} \rceil)[\mathbf{Team}_{l_2}^{l_1}\mathbf{TxtEx} - \mathbf{Team}_{k_2}^{k_1}\mathbf{TxtEx} \neq \emptyset]$.

Theorem 25 $(\forall l_1, l_2, k_1, k_2 \geq 1 \mid l_2 \geq 5l_1/2 - 1, k_2 < k_1 + \frac{3}{2} \cdot \lceil \frac{k_1(l_1-1)}{l_1} \rceil)[\mathbf{Team}_{l_2}^{l_1}\mathbf{TxtEx} - \mathbf{Team}_{k_2}^{k_1}\mathbf{TxtEx}^* \neq \emptyset]$.

Proof. Since $l_1/(l_2 - l_1 + 1) \leq 2/3$ and $\lceil k_1(l_1-1)/l_1 \rceil/(k_2 - k_1) > 2/3$, we have, $\mathbf{Team}_{l_2-l_1+1}^{l_1}\mathbf{TxtEx} - \mathbf{Team}_{k_2-k_1}^{\lceil \frac{k_1 \cdot (l_1-1)}{l_1} \rceil}\mathbf{TxtEx}^* \neq \emptyset$. Now using Theorem 8 with $i_1 = 1$, we get $\mathbf{Team}_{l_2}^{l_1}\mathbf{TxtEx} - \mathbf{Team}_{k_2}^{k_1}\mathbf{TxtEx} \neq \emptyset$. ∎

Theorem 26 $(\forall k, l \mid k > 2l/5)[\mathbf{Team}_l^k\mathbf{TxtEx} \subseteq \mathbf{Team}_3^1\mathbf{TxtEx}]$.

Proof of Theorem 26. By Corollary 8 we know that for any m and n, such that $m > n/2$, $\mathbf{Team}_n^m\mathbf{TxtEx} \subseteq \mathbf{Team}_2^1\mathbf{TxtEx}$. Suppose machines $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_l$ are given. For $\emptyset \neq S \subseteq \{1, 2, \ldots, l\}$, let $\mathbf{M}_S^1$, $\mathbf{M}_S^2$ denote the two machines which $\mathbf{Team}_2^1\mathbf{TxtEx}$-identify any language which is $\mathbf{Team}_{\mathrm{card}(S)}^{\lfloor \mathrm{card}(S)/2 \rfloor + 1}$-identified by machines $\{\mathbf{M}_i\}_{i \in S}$.

We now define $\mathbf{M}_a, \mathbf{M}_b$, and $\mathbf{M}_c$ which $\mathbf{Team}_3^1\mathbf{TxtEx}$-identify any language which is $\mathbf{Team}_l^k\mathbf{TxtEx}$-identified by $\{\mathbf{M}_i\}_{1 \leq i \leq l}$. Let conv be as defined in the proof of Theorem 2. Suppose σ is given. Let $S_\sigma \subseteq \{1, 2, \ldots, l\}$ be the lexicographically least set of cardinality k such that, for each $i \in S_\sigma$ and each $i' \in \{1, 2, \ldots, l\} - S_\sigma$, $\mathrm{conv}(\mathbf{M}_i, \sigma) \leq \mathrm{conv}(\mathbf{M}_{i'}, \sigma)$. Let the members of S_σ be $i_1, i_2, ..., i_k$. Let $\mathbf{M}_a(\sigma) = \mathrm{majority}(\mathbf{M}_{i_1}(\sigma), \mathbf{M}_{i_2}(\sigma), \ldots, \mathbf{M}_{i_k}(\sigma))$.

Let $\text{match}(i,\sigma) = \max(\{x \leq |\sigma| \mid (\text{content}(\sigma[x]) \subseteq W_{i,|\sigma|}) \ \wedge \ (W_{i,x} \subseteq \text{content}(\sigma))\})$. Let $X_\sigma \subseteq S_\sigma$ be a (lexicographically least) set of cardinality $\lceil k/2 \rceil$ such that for each $i \in X_\sigma$ and each $i' \in S_\sigma - X_\sigma$, $\text{match}(\mathbf{M}_i(\sigma),\sigma) \leq \text{match}(\mathbf{M}_{i'}(\sigma),\sigma)$.

Let $\mathbf{M}_b(\sigma) = \mathbf{M}^1_{\{1,2,\ldots,l\}-X_\sigma}(\sigma)$ and $\mathbf{M}_c(\sigma) = \mathbf{M}^2_{\{1,2,\ldots,l\}-X_\sigma}(\sigma)$.

Now, suppose $\{\mathbf{M}_i\}_{1\leq i\leq l}$ $\mathbf{Team}^k_l\mathbf{TxtEx}$-identify content$(T)$. Then, $S = \lim_{n\to\infty} S_{T[n]}$ consists of a subset (of $\{1,2,\ldots,l\}$) of cardinality k such that, for each i in S, $\mathbf{M}_i$ converges on T.

Now, if majority of machines in S, $\mathbf{TxtEx}$-identify T then so does $\mathbf{M}_a$. If majority of machines in S do not $\mathbf{TxtEx}$-identify T, then $X = \lim_{n\to\infty} X_{T[n]}$ exists and the elements of X do not $\mathbf{TxtEx}$-identify T; this implies that at least k of $\{\mathbf{M}_1,\mathbf{M}_2,\ldots,\mathbf{M}_l\} - \{\mathbf{M}_i \mid i \in X\}$ do. Thus, at least one of $\mathbf{M}_b$, $\mathbf{M}_c$ $\mathbf{TxtEx}$-identifies T. ∎

An extension of the above proof yields the following result.

Theorem 27 $(\forall k,l,i \mid k > 2l/5)[\mathbf{Team}^k_l\mathbf{TxtEx}^i \subseteq \mathbf{Team}^1_3\mathbf{TxtEx}^{i\cdot\lceil\frac{k}{2}\rceil}]$.

We end this section by stating results that provide more evidence of the complexity of team identification of languages. The first collection of results (Corollary 16 just below to Theorem 27 above together with Theorems 28 and 29 below) show that there exist identification classes $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ such that $\mathbf{A} \subset \mathbf{B}$, but both $\mathbf{A}$, $\mathbf{C}$ and $\mathbf{B}$, $\mathbf{C}$ are incomparable to each other.

Corollary 16 $\mathbf{Team}^3_7\mathbf{TxtEx} \subseteq \mathbf{Team}^1_3\mathbf{TxtEx}$.

Theorem 28 $\mathbf{Team}^1_3\mathbf{TxtEx} - \mathbf{Team}^3_7\mathbf{TxtEx} \neq \emptyset$.

PROOF. Follows from team function hierarchy of Smith [32], $(\forall n \in N^+)[\mathbf{Team}^1_n\mathbf{Ex} \subset \mathbf{Team}^1_{n+1}\mathbf{Ex}]$, and Pitt's connection for functions [25], $(\forall p \mid 0 < p \leq 1)(\forall n)[1/(n+1) < p \leq 1/n \Rightarrow \mathbf{Team}^1_n\mathbf{Ex} = \mathbf{Prob}^p\mathbf{Ex}]$. ∎

Theorem 29 $\mathbf{Team}^2_5\mathbf{TxtEx} - \mathbf{Team}^1_3\mathbf{TxtEx} \neq \emptyset$.

PROOF. By Theorem 10 $\mathbf{Team}^2_4\mathbf{TxtEx} - \mathbf{Team}^1_2\mathbf{TxtEx} \neq \emptyset$. The theorem now follows using Theorem 7 with $i_1 = 1$. ∎

Theorem 30 $\mathbf{Team}^3_7\mathbf{TxtEx} - \mathbf{Team}^2_5\mathbf{TxtEx} \neq \emptyset$.

PROOF. $\mathbf{Team}^3_5\mathbf{TxtEx} - \mathbf{Team}^2_3\mathbf{TxtEx} \neq \emptyset$ by Corollary 14. Theorem now follows using Theorem 7 with $i_1 = 1$. ∎ Our second collection of results (Theorem 31 and 32 below) shows that sometimes allowing successful members in the team to make a finite, but unbounded, number of mistakes compensates for weaker teams. More specifically, Theorem 31 below shows that all such collections of languages that can be identified by teams of 8 machines requiring at least 5 to be successful can be identified by some team of 3 machines requiring at least 2 to be successful if successful members of this latter team are allowed to converge to grammars which make a finite, but unbounded, number of mistakes. On the other hand, Theorem 32 shows that there are collections of languages that can be identified by teams of 8 machines requiring at least 5 to be successful, but which collections cannot be identified by any team of 3 machine requiring at least 2 to be successful if the number of mistakes allowed in the final grammars of the successful members of the latter team is bounded in advance.

Theorem 31 $\mathbf{Team}^5_8\mathbf{TxtEx} \subseteq \mathbf{Team}^2_3\mathbf{TxtEx}^*$.

PROOF. We omit the proof. The idea is similar to that used in Theorem 21. ∎

Theorem 32 $(\forall j \in N)[\mathbf{Team}^5_8\mathbf{TxtEx} - \mathbf{Team}^2_3\mathbf{TxtEx}^j \neq \emptyset]$.

We omit the proof of the above theorem. The idea is similar to that used in proving Theorem 12.

We finally note that many additional results can be shown to hold for team language identification. We do not present them here because they are of partial nature only.

5.6 Team and Probabilistic Identification of Languages from Informants

Finally, we consider identification from both positive and negative data. Identification from texts is an abstraction of learning from positive data. Similarly, learning from both positive and negative data can be abstracted as identification from informants. The notion of informants, defined below, was first considered by Gold [15].

Definition 18 A text I is called an *informant* for a language L just in case $\text{content}(I) = \{\langle x, 1\rangle \mid x \in L\} \cup \{\langle x, 0\rangle \mid x \notin L\}$.

The next definition formalizes identification from informants.

Definition 19 (a) $\mathbf{M}$ $\mathbf{InfEx}$-identifies L (written: $L \in \mathbf{InfEx}(\mathbf{M})$) just in case $\mathbf{M}$, fed any informant for L converges to a grammar for L.

(b) $\mathbf{InfEx} = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{InfEx}(\mathbf{M})]\}$.

We can similarly define $\mathbf{Prob}^p\mathbf{InfEx}$-identification and $\mathbf{Team}_n^m\mathbf{InfEx}$-identification. The following result says that Pitt's connection holds for language identification if the machines are also presented with information about what is not in the language. This result strongly suggests that the complications arising in the study of team $\mathbf{TxtEx}$-identification may be due to the lack of negative data.

Theorem 33 $(\forall p \mid 1/(n+1) < p \leq 1/n)$ $[\mathbf{Team}_n^1\mathbf{InfEx} = \mathbf{Prob}^p\mathbf{InfEx}]$.

A close inspection of Pitt's proof for function identification yields a proof for the above theorem; we omit details.

6 Conclusions

The present paper studied the computational limits on team identification of r.e. languages from positive data. It was shown that the notions of probabilistic language identification and team function identification turn out to be different. In fact, it was established that for probabilities of the form $1/k$, probabilistic identification of languages is strictly more powerful than team identification of languages where at least $1/k$ of the members in the team are required to be successful.

We also presented two very general tools that allowed us to easily prove new diagonalization results from known ones. Some results were also presented which shed light on the difficulty of obtaining general results. An attempt was made to pinpoint the reason behind why probabilistic identification is different from team identification for languages by showing that an analog of Pitt's connection holds for language identification if the learning agent is also presented with negative information.

Finally we note that results from [24] could be used to show that for $\mathbf{TxtBc}$-identification (see [6] for definition), if $i > j/2$, then $\mathbf{Team}_j^i\mathbf{TxtBc} = \mathbf{TxtBc}$ (also see [20]). Thus, team inference with respect to $\mathbf{TxtBc}$-identification behaves differently from team inference with respect to $\mathbf{TxtEx}$-identification. A study of probabilistic and team identification for $\mathbf{TxtBc}$-identification on the lines of the present paper is open. We also note that the structure of team language identification is similar to the structure of finite identification (identification without any mind changes) of functions by a team for success ratios $\geq 2/3$ (see [17]). For other success ratios, the structure of team language identification is different from finite identification of functions by a team [9, 11, 10, 33, 17, 8, 7].

Acknowledgements

We would like to thank the referees for several helpful comments and suggestions. We also thank John Case for suggesting the topic. We thank John Case, Mark Fulk, Lata Narayanan Dan Osherson, and Rajeev Raman for providing several helpful discussions and encouragement.

During the early stages of this work, Sanjay Jain was affiliated with the Department of Computer Science, University of Rochester and the Department of Computer and Information Sciences, University of Delaware. He was supported in part by NSF grant CCR 832-0136 at the University of Rochester

During the early stages of this work, Arun Sharma was affiliated with the Department of Computer Science, SUNY at Buffalo, Department of Computer and Information Sciences, University of Delaware, and the Department of Brain and Cognitive Sciences, MIT. He was supported by NSF grant CCR 871-3846 at SUNY at Buffalo and University of Delaware, and by a Siemens Corporation grant at MIT. At UNSW, this work has been supported by a grant from the Australian Research Council.

References

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

[2] P. Billingsley. *Probability and Measure*. Willey and sons, New York, 1995. 3rd edition.

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

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

[5] J. Case. Periodicity in generations of automata. *Mathematical Systems Theory*, 8:15–32, 1974.

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

[7] 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.

[8] 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.

[9] R. Freivalds. Functions computable in the limit by probabilistic machines. In *Mathematical Foundations of Computer Science*, pages 77–87, 1974.

[10] R. Freivalds. Finite identification of general recursive functions by probabilistic strategies. In *Proceedings of the Conference on Fundamentals of Computation Theory*, pages 138–145. Akademie-Verlag, Berlin, 1979.

[11] R. Freivalds. On the principle capabilities of probabilistic algorithms in inductive inference. *Semiotika Inform*, 12:137–140, 1979.

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

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

[14] Gill. Computational complexity of probabilistic turing machines. *SIAM Journal of Computing*, 1977.

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

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

[17] 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.

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

[19] 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.

[20] S. Jain and A. Sharma. On aggregating teams of learning machines. *Theoretical Computer Science A*, 137(1):85–108, January 1995.

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

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

[23] L. Pitt. A characterization of probabilistic inference. In *Proceedings of the 25th Symposium on the Foundations of Computer Science*, 1984.

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

[25] L. Pitt. Probabilistic inductive inference. *Journal of the ACM*, 36:383–433, 1989.

[26] L. Pitt and C. Smith. Probability and plurality for aggregations of learning machines. In *Proceedings of the 14th International Colloquium on Automata, Languages and Programming*, 1987.

[27] L. Pitt and C. Smith. Probability and plurality for aggregations of learning machines. *Information and Computation*, 77:77–92, 1988.

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

[29] H. Rogers. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill, New York, 1967. Reprinted by MIT Press, Cambridge, Massachusetts in 1987.

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

[31] C. Smith. The power of parallelism for automatic program synthesis. In *Proceedings of the 22nd Symposium on the Foundations of Computer Science*, 1981.

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

[33] M. Velauthapillai. Inductive inference with bounded number of mind changes. In *Proceedings of the Second Annual Workshop on Computational Learning Theory, Santa Cruz, California*, pages 200–213. Morgan Kaufmann Publishers, Inc., August 1989.