SCS&E Report 9301
February, 1993

Computational Limits on Team Identification of Languages

Sanjay Jain and Arun Sharma

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
THE UNIVERSITY OF NEW SOUTH WALES

Abstract

A team of learning machines is essentially a multiset of learning machines. A team is said to successfully identify a concept just in case each member of some nonempty subset 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.

Some preliminary results were first reported at the *17th International Colloquium on Automata, Languages and Programming*, Warwick University, July 1990.

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

At the same time, 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 Buffalo and University of Delaware, and by a Siemens Corporation grant at MIT.

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 essentially a multiset of machines. A team is said to identify a language if each member of some nonempty subset 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 [21, 23] 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 one of our results (Theorem 10). 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. Section 3 introduces the notation and Section 4 describes the definitions 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 $\mathbf{M}$ till 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 computable 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 computable functions, $\{f[n] \mid f \text{ is a computable function and } n \in N\}$, 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 a program index 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 [2] 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 J. Case and extensively studied by Smith [29, 30]. The next definition describes team identification of functions. Recall that a team of machines is essentially 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 [29, 30] and $\mathbf{Team}_n^m\mathbf{Ex}$-identification was studied by Osherson, Stob, and Weinstein [18]. Pitt [21] 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 delay the formal details of the probability of identification till Section 4.5. Below, $\mathbf{P}$ ranges over probabilistic machines.

Definition 3 [21, 23] 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 [21, 23]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 [24, 25] 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.

A *text* for a language L is a mapping T from N into $N \cup \{\#\}$ such that L is the set of natural numbers in the range of T. Intuitively, a text T for a language L is a presentation of elements of L (possibly repeated) and no non-elements of L; #'s in the presentation may be thought of as modeling pauses in data input. *content*(T) denotes the set of natural numbers in the range of T. (Thus, the content of a text never includes #.) The initial sequence of text T of length n is denoted $T[n]$. The set of all finite initial sequences of N and #'s is denoted SEQ. It is easy to see that there exists a computable bijection between SEQ and N. Members of SEQ are inputs to machines that learn grammars (acceptors) for r.e. languages. We also fix an acceptable programming system and interpret the output of a language learning machine as the index of a program in this system. Then, a program conjectured by a machine in response to a finite initial sequence may be viewed as a candidate accepting grammar for the language being learned. We say that $\mathbf{M}$ converges on text T to i just in case for all but finitely many n, $\mathbf{M}(T[n]) = i$. The following definition introduces Gold's criteria for successful identification of languages.

Definition 4 [15]

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

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

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

The class $\mathbf{TxtEx}$ is a set theoretic summary of the capability of machines to $\mathbf{TxtEx}$-identify collections of r.e. languages. We now define team identification of languages.

Definition 5 (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.

[1] The general case of team function identification was also studied by Osherson, Stob, and Weinstein [18].

(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 $\mathcal{L}$.

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. It can be shown that an alternative formulation in which successful machines in the team are required to be successful on all texts for L is equivalent to our definition in the sense that both formulations yield the same collections of identifiable languages (the reader is directed to Fulk [12, 13] for arguments of such equivalences).

Probabilistic language identification is the subject of next definition. Again, as was the case with probabilistic function identification, we delay the formal details of probability of identification in the following definition to Section 4.5.

Definition 6 [21, 23] 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 [25].

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

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.

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

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 prove 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 $\mathbf{Team}^{i}_{j}\mathbf{TxtEx} - \mathbf{Team}^{k}_{l}\mathbf{TxtEx} \neq \emptyset$ to prove results of the form $\mathbf{Team}^{i'}_{j'}\mathbf{TxtEx} - \mathbf{Team}^{k'}_{l'}\mathbf{TxtEx} \neq \emptyset$ 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.

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

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:

[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 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:

(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 [27]. 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, 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 $[i < j \ \vee \ [i = j \ \wedge \ k < l]]$.

$\emptyset$ denotes the empty set. A, B, C, S, X, Y, Z, with or without decorations, range over subsets of N. We reserve A^m to range over multisets with elements from N. We usually denote finite sets by D. $\mathrm{card}(D)$ denotes the cardinality of the finite set D. $\mathrm{card}(A^m)$ denotes the number of (not necessarily distinct) elements of the multiset A^m. Similarly, set operations, $\cap$, $\cup$, $\subset$, set difference, on multisets producing multisets can be defined (for example $\{1,1,2\} \cup \{1\} = \{1,1,1,2\}$ and $\{1,1,2\} - \{1\} = \{1,2\}$). $\max(\ ), \min(\ )$ denote the maximum and minimum of a set respectively. We take $\min(\emptyset)$ to be ∞ and $\max(\emptyset)$ to be 0.

Let η, with or without decoration, 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 $\mathrm{card}(\{x \mid \eta_1(x) \neq \eta_2(x)\}) \leq a$. Otherwise we say that η_1 is not an a-variant of η_2 (written $\eta_1 \neq^a \eta_2$).

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 natnum. 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 $\mathcal{E}$. φ denotes a standard acceptable programming system (also referred to as standard acceptable numbering) [26, 27]. 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 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 [3] for φ. $W_{i,n}$ denotes the set $\{x \leq n \mid \Phi_i(x) \leq n\}$.

$\langle i, j \rangle$ stands for an arbitrary computable one to one encoding of all pairs of natural numbers onto N [27]. 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 quantifiers '$\overset{\infty}{\forall}$' and '$\overset{\infty}{\exists}$' mean 'for all but finitely many' and 'there exists infinitely many', respectively.

4 Definitions

4.1 Learning Machines

In Definition 7 below, we formally introduce what we mean by a machine that learns a function, and in Definition 9, we do the same for a machine that learns a language.

We assume, without loss of generality, that the graph of a function is fed to a machine in canonical order. For $f \in \mathcal{R}$ and $n \in N$, we let $f[n]$ denote the finite initial segment $\{(x, f(x)) \mid x < n\}$. Clearly, $f[0]$ denotes the empty segment. SEG denotes the set of all finite initial segments, $\{f[n] \mid f \in \mathcal{R} \wedge n \in N\}$. Note that $f[n] \cup \{(n, x)\}$ is a new finite initial segment of length $n + 1$ formed by extending $f[n]$ suitably.

Definition 7 [15] A *function learning machine* is an algorithmic device which computes a mapping from SEG into N.

The output of a function learning machine $\mathbf{M}$ on initial segment $f[n]$, denoted $\mathbf{M}(f[n])$, is interpreted as the index of a program in our fixed acceptable programming system φ.

We now consider language learning machines. Definition 8 below introduces a notion that facilitates discussion about elements of a language being fed to a learning machine.

Definition 8 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. We let σ, τ, and γ, with or without decorations, range over finite sequences. SEQ denotes the set of all finite sequences. $\sigma_1 \diamond k$ denotes the *concatenation* of k at the end of sequence σ_1, where $\sigma = \sigma_1 \diamond k$ is defined as follows:

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

Definition 9 A *language learning machine* is an algorithmic device which 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 initial segments, SEG, can be coded onto N. Also, the set of all finite sequences of natural numbers and #'s, SEQ, can be coded onto N. Thus, in both Definitions 7 and 9, we can view these machines as taking natural numbers as input and emitting natural numbers as output. Henceforth, we will refer to both function-learning machines and language-learning machines as just learning machines, or simply as machines. We let $\mathbf{M}$, with or without decorations, range over learning machines.

4.2 Function Identification

In Definition 10 below we spell out what it means for a learning machine on a function to converge in the limit.

Definition 10 Suppose $\mathbf{M}$ is a learning machine and f is a computable function. $\mathbf{M}(f)\downarrow$ (read: $\mathbf{M}(f)$ *converges*) $\iff$ $(\exists i)(\overset{\infty}{\forall} n)\ [\mathbf{M}(f[n]) = i]$. If $\mathbf{M}(f)\downarrow$, then $\mathbf{M}(f)$ is defined = the unique i such that $(\overset{\infty}{\forall} n)[\mathbf{M}(f[n]) = i]$, otherwise we say that $\mathbf{M}(f)$ diverges (written: $\mathbf{M}(f)\uparrow$).

The next definition introduces Gold's criteria for successful identification of a function.

Definition 11 [15, 2, 6] Let $a \in N \cup \{*\}$.
(i) $\mathbf{M}$ $\mathbf{Ex}^a$-*identifies* f (written: $f \in \mathbf{Ex}^a(\mathbf{M})$) $\iff$ $(\exists i \mid \varphi_i =^a f)[\mathbf{M}(f)\downarrow = i]$.
(ii) $\mathbf{Ex}^a = \{\mathcal{S} \mid (\exists \mathbf{M})[\mathcal{S} \subseteq \mathbf{Ex}^a(\mathbf{M})]\}$.

Case and Smith [6] motivate anomalies (or, mistakes) in the final programs in Definition 11 from the fact that physicists sometimes do employ explanations with anomalies. The $a = *$ case was introduced by L. Blum and M. Blum [2] and the other $a > 0$ cases were first considered by Case and Smith [6].

4.3 Language Identification

Definition 12 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 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 superscripts, range over texts. $T[n]$ denotes the finite initial sequence of T with length n. Hence, $\mathrm{domain}(T[n]) = \{x \mid x < n\}$.

In Definition 13 below we spell out what it means for a learning machine on a text to converge in the limit.

Definition 13 Suppose $\mathbf{M}$ is a learning machine and T is a text. $\mathbf{M}(T)\downarrow$ (read: $\mathbf{M}(T)$ *converges*) $\iff$ $(\exists i)(\stackrel{\infty}{\forall} n)\ [\mathbf{M}(T[n]) = i]$. If $\mathbf{M}(T)\downarrow$, then $\mathbf{M}(T)$ is defined = the unique i such that $(\stackrel{\infty}{\forall} n)[\mathbf{M}(T[n]) = i]$, otherwise we say that $\mathbf{M}(T)$ diverges (written: $\mathbf{M}(T)\uparrow$).

Definition 14 [15, 5, 20] Let $a \in N \cup \{*\}$.
(i) $\mathbf{M}$ $\mathbf{TxtEx}^a$-*identifies* $T \Leftrightarrow [\mathbf{M}(T)\downarrow$ and $W_{\mathbf{M}(T)} =^a \mathrm{content}(T)]$.
(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) $\mathbf{TxtEx}^a = \{\mathcal{L} \mid (\exists \mathbf{M})[\mathcal{L} \subseteq \mathbf{TxtEx}^a(\mathbf{M})]\}$.

4.4 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 15 introduces team identification of functions and Definition 16 introduces team identification of languages.

Definition 15 [30, 19] Let $a \in N \cup \{*\}$ and let $m, n \in N^+$.

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

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

Definition 16 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 $i_1, i_2, \ldots, i_m$, $1 \le i_1 < i_2 < \cdots < i_m \le n$, 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 both $\mathbf{Team}_n^m\mathbf{Ex}^a$-identification criteria and $\mathbf{Team}_n^m\mathbf{TxtEx}^a$-identification criteria, we refer to the fraction m/n as the *success ratio* of the criteria. In the following, for $i > j$, we take $\mathbf{Team}_j^i\mathbf{TxtEx}^a = \{\emptyset\}$.

4.5 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 make these notions precise below; we closely follow Pitt [22, 23].

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. Observe that a t-ary oracle is somewhat like a text for the finite language N_t, and notations for texts carry over to oracles, that is, the n^{th} member of O is denoted O_n and the initial finite sequence of O of length n is denoted $O[n]$. The set $\{O[n] \mid O \text{ is a } t\text{-ary oracle and } n \in N\}$ is the collection of all finite t-ary sequences. (A typical variable for finite t-ary sequences is ρ). Similarly, the length of a finite t-ary sequence ρ is denoted $|\rho|$; for $n < |\rho|$, the n^{th} member of ρ is denoted by ρ_n and the initial sequence of length n in ρ is denoted by $\rho[n]$.

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

Similarly, we can describe the behavior of $\mathbf{P}$ for a given t-ary oracle O. $\mathbf{P}^O$ behaves like $\mathbf{P}$ except whenever $\mathbf{P}$ flips its coin, $\mathbf{P}^O$ reads the result of the coin flip from the oracle O, that is, the result of the first coin flip is O_0, the result of the second coin flip is O_1, and so on and so forth.

We now describe a probability measure on a single coin flip. For a t-sided coin, let $(N_t, \mathcal{B}_t, \text{pr}_t)$ 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 $\text{pr}_t = \text{card}(S)/t$. Intuitively, this measure simply says that the probability of the outcome of flipping a t-sided coin belonging to a set $S \subseteq N_t$ is $\text{card}(S)/t$. We 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 $(N_t^\infty, \mathcal{B}_t^\infty, \text{pr}_t^\infty)$ 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 $\text{pr}_t^\infty(A_{i_1, i_2, \ldots, i_n}) = \prod_{j=1}^n \text{pr}_t(A_{i_j})$, for each choice of n integers $i_1, i_2, \ldots, i_n$.

Clearly, sets $A_{i_1, i_2, i_3, \ldots, i_n}$ are measurable.

4.5.1 Probabilistic Function Identification

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

Lemma 1 [22, 23] *Let* $\mathbf{P}$ *be a probabilistic machine and let* $f \in \mathcal{R}$*. Then* $\{O \mid \mathbf{P}^O\ \mathbf{Ex}^a\text{-identifies } f\}$ *is measurable.*

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

Definition 17 [22, 23] Let $f \in \mathcal{R}$ and $\mathbf{P}$ be a probabilistic machine equipped with a t-sided coin ($t \geq 2$). Then, $\mathrm{pr}_t^\infty(\mathbf{P}\ \mathbf{Ex}^a\text{-identifies } f) = \mathrm{pr}_t^\infty(\{O \mid \mathbf{P}^O\ \mathbf{Ex}^a\text{-identifies } f\})$.

The next lemma says that we do not sacrifice any learning power by restricting our attention to the investigation of identification by probabilistic machine equipped with only a two-sided coin.

Lemma 2 (Adopted from [22, 23]) *Let* $t, t' > 2$*. Let* $\mathbf{P}$ *be a probabilistic machine with a* t*-sided coin. Then, there exists a probabilistic machine* $\mathbf{P}'$ *with a* t'*-sided coin such that for each* $f \in \mathcal{R}$, $\mathrm{pr}_{t'}^\infty(\mathbf{P}'\ \mathbf{Ex}^a\text{-identifies } f) = \mathrm{pr}_t^\infty(\mathbf{P}\ \mathbf{Ex}^a\text{-identifies } f)$.

The next definition describes function 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. Also, we are at liberty to use whatever value of the number of sides of a coin that is convenient for the presentation at hand.

Definition 18 [22, 23] Let $0 \leq p \leq 1$.

(a) $\mathbf{P}$ $\mathbf{Prob}^p\mathbf{Ex}^a$*-identifies* f (written: $f \in \mathbf{Prob}^p\mathbf{Ex}^a(\mathbf{P})$) just in case $\mathrm{pr}(\mathbf{P}\ \mathbf{Ex}^a\text{-identifies } f) \geq p$.

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

4.5.2 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 3 [22] *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 19 [22] Let T be a text and $\mathbf{P}$ be a probabilistic machine equipped with a t-sided coin ($t \geq 2$). Then, $\mathrm{pr}_t^\infty(\mathbf{P}\ \mathbf{TxtEx}^a\text{-identifies } T) = \mathrm{pr}_t^\infty(\{O \mid \mathbf{P}^O\ \mathbf{TxtEx}^a\text{-identifies } T\})$.

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

Lemma 4 (Adopted from [22, 23]) *Let* $t, t' > 2$*. Let* $\mathbf{P}$ *be a probabilistic machine with a* t*-sided coin. Then, there exists a probabilistic machine* $\mathbf{P}'$ *with a* t'*-sided coin such that for each text* T, $\mathrm{pr}_{t'}^\infty(\mathbf{P}'\ \mathbf{TxtEx}^a\text{-identifies } T) = \mathrm{pr}_t^\infty(\mathbf{P}\ \mathbf{TxtEx}^a\text{-identifies } T)$.

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

Definition 20 [22] Let $0 \le p \le 1$.

(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 $\mathrm{pr}(\mathbf{P}\ \mathbf{TxtEx}^a\text{-identifies } T) \ge p$.

(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 $\ge \frac{2}{3}$

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 [18] and Pitt and Smith [25] 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 [18, 25] $(\forall j,k \mid \frac{j}{k} > \frac{1}{2})(\forall a)[\mathbf{Team}_k^j\mathbf{Ex}^a = \mathbf{Ex}^a]$.

Surprisingly, an analog of Theorem 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 identifies by a single machine.[3] Corollary 2 is a similar result about $\mathbf{TxtEx}^*$-identification.

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

Corollary 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:

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

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

Clearly, majority(A^m) can be defined using the *s-m-n* theorem [28]. Intuitively, majority(A^m) is a grammar for a language that consists of all such elements that are enumerated by a majority of grammars in A^m. Below, whenever we use a set as an argument to majority we assume the argument to be a multiset.

[3] Corollary 1 also appears in Osherson, Stob, and Weinstein [18], and may also be shown using an argument from Pitt [22] about probabilistic language learning.

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 $\mathcal{L}$.

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

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

It is easy to verify that 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 $(\forall j, k \mid j > 2k/3)(\forall a \in (N \cup \{*\}))[\mathbf{Team}_k^j\mathbf{TxtEx}^a \subseteq \mathbf{TxtEx}^{\lfloor \frac{2j-k}{\lfloor (3j-2k+1)/2 \rfloor} \cdot a \rfloor}]$.

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

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 clearly, $\mathbf{M}_3'$ $\mathbf{TxtEx}^{(j+1)\cdot a}$-identifies T. Moreover, $\mathbf{M}_1'$ ($\mathbf{M}_2'$) $\mathbf{TxtEx}^{(j+1)\cdot a}$-identifies T if $\mathbf{M}_{\lim_{s\to\infty} m_1^{T[s]}}$ $\mathbf{TxtEx}^a$-identifies T (does not $\mathbf{TxtEx}^a$-identifies 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 $\mathcal{L} = \{L \mid (\exists \text{ distinct } x_1, x_2 \in \{0,1,2\})(\text{for } i = 1,2)[\{y \mid \langle x_i, y\rangle \in L\}$ is non-empty and finite and $W_{\max(\{y \mid \langle x_i, y\rangle \in L\})} = L]\}$.

Clearly, $\mathcal{L} \in \mathbf{Team}_3^2\mathbf{TxtEx}$. Suppose by way of contradiction some machine $\mathbf{M}$ $\mathbf{TxtEx}^*$-identifies $\mathcal{L}$. Without loss of generality, assume that $\mathbf{M}$ is order independent [2]. Then, by the operator recursion theorem [4], there exists a 1-1 increasing, nowhere 0, recursive function p such that $W_{p(i)}$'s can be described as follows.

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

Begin {stage s}

1. Enumerate $W_{p(0)}^s \bigcup W_{p(1)}^s$ in $W_{p(0)}, W_{p(1)}, W_{p(2s)}$, and $W_{p(2s+1)}$.
   Enumerate $\langle 2, p(2s)\rangle$ in $W_{p(0)}, W_{p(2s)}$.
   Enumerate $\langle 2, p(2s+1)\rangle$ in $W_{p(1)}, W_{p(2s+1)}$.
   Let τ_0 be an extension of σ_s such that content$(\tau_0) = W_{p(0)}$ enumerated till now.
   Let τ_1 be an extension of σ_s such that content$(\tau_1) = W_{p(1)}$ enumerated till now.
2. Let $x = 0$. Dovetail steps 2a and 2b until, if ever, step 2b succeeds. If and when step 2b succeeds, go to step 3.
   - 2a. Go to substage 0.
     Begin {substage s'}
     - Enumerate $\langle 4, x\rangle$ in $W_{p(0)}, W_{p(2s)}$.
     - Enumerate $\langle 5, x\rangle$ in $W_{p(1)}, W_{p(2s+1)}$.
     - Let $x = x + 1$.
     - Go to substage $s' + 1$.

     End {substage s'}
   - 2b. Search for $i \in \{0,1\}$ and $n \in N$ such that $\mathbf{M}(\tau_i \diamond \langle 4+i, 0\rangle \diamond \langle 4+i, 1\rangle, \ldots, \langle 4+i, n\rangle) \neq \mathbf{M}(\sigma_s)$.
3. If and when 2b succeeds, let i, n be as found in step 2b.
   Let $S =$
   $W_{p(0)}$ enumerated till now
   $\bigcup W_{p(1)}$ enumerated till now
   $\bigcup\{\langle 4+i, 0\rangle, \langle 4+i, 1\rangle, \ldots, \langle 4+i, n\rangle\}$.
4. Let σ_{s+1} = 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 in $W_{p(0)}$.
   Go to stage $s + 1$.

End {stage s}

Consider the following cases:

Case 1: All stages terminate.

In this case, let $L = W_{p(0)} = W_{p(1)} \in \mathcal{L}$. Let $T = \bigcup_s \sigma_s$. Clearly, T is a text for L. But, $\mathbf{M}$ on T makes infinitely many mind changes (since the only way in which infinitely many stages can be completed is by the success of step 2b infinitely often). Thus, $\mathbf{M}$ does not $\mathbf{TxtEx}^*$-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. Let $T_i = \tau_i \diamond \langle 4+i, 0\rangle \diamond \langle 4+i, 1\rangle \diamond \ldots \diamond \langle 4+i, n\rangle$, where $i \in \{0, 1\}$ and τ_i is as defined in stage s. Now, $\mathbf{M}$ converges to $\mathbf{M}(\sigma_s)$ for both T_1 and T_2. Since L_1, L_2 are infinitely different from each other, $W_{\mathbf{M}(\sigma_s)}$ is infinitely different from at least one of L_1 and L_2. Hence, $\mathbf{M}$ does not $\mathbf{TxtEx}^*$-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

In order to avoid details and to simplify many diagonalization proofs in the sequel, 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 would like to note that these conditions are satisfied by all the diagonalization proofs in the present paper.

For a recursive function q, and $i, j, k, l \in N^+$, we define the predicate PROP(q, i, j, k, l) to be true just in case given

(a) finite sets S_1, S_2, S_3, S_4, S'_2,

(b) a team of $\leq l$ machines $\mathcal{M}$,

such that S_2, S_3, S_4 are pairwise disjoint, $S'_2 \subseteq S_2$, card$(S_2) = j$, and card$(S'_2) \leq i$, then $\mathcal{L}_{q,i,j,k,l,S_1,S_2,S_3,S_4,S'_2,\mathcal{M}} \not\subseteq \mathbf{Team}^k_{\text{card}(\mathcal{M})}\mathbf{TxtEx}(\mathcal{M})$, 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) $(\forall x \in S_4)[\text{card}(\{y \mid \langle x, y\rangle \in L\}) = \infty]$,
(c) $\text{card}(\{x \in S_2 \mid \max(\{y \mid \langle x, y\rangle \in L\}) \text{ exists } \wedge W_{\max(\{y \mid \langle x,y\rangle \in L\})} = L\}) \geq i$,
(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) $(\forall x \in S'_2)[S_1 \subseteq W_{q(S_1,S_2,S_3,S_4,S'_2,\mathcal{M},x)} \subseteq S_1 \cup \{\langle z, y\rangle \mid z \notin S_3 \ \wedge \ y \in N\}]$.
$\}$

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').

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 [4] 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 refer to $q'(S_1, S_2, S_3, S_4, S_2', \mathcal{M}, x)$ by $p(x)$. Without loss of generality we assume $\text{card}(\mathcal{M}) = l'$. 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 $\mathbf{M}' \in \mathcal{M} - Z_\sigma$, $\text{conv}(\mathbf{M}, \sigma) \leq \text{conv}(\mathbf{M}', \sigma)$.

For $y \in S_2''$, enumerate $S_1 \cup \{\langle x, p(x)\rangle \mid x \in S_2''\}$ in $W_{p(y)}$. Let σ_0 be a sequence such that $\text{content}(\sigma_0) = S_1 \cup \{\langle x, p(x)\rangle \mid x \in S_2''\}$. Let S_5 be a set disjoint from S_1, S_2, S_3, S_4 such that $\text{card}(S_5) = i_1$. Let S_6 be such that $S_5 \subseteq S_6 \subseteq S_5 \cup (S_2 - S_2'')$, and $\text{card}(S_6) = j$. 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 $\text{content}(\tau) - \text{content}(\sigma_s) \subseteq \{\langle x, y\rangle \mid x \notin S_3 \cup S_2''\}$.
2. Let $X_2 = S_6$.

   Let $X_2' = S_5$.

   For $w < \lfloor i/i_1 \rfloor$, let Y_w be pairwise disjoint subsets of S_2'' of cardinality i_1 each.

   For $w < \lfloor i/i_1 \rfloor$, let u_w 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 $z \in S_2''\})$.

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

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

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

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

   2.1 Search for an $s'' > s'$, $\mathbf{M} \in Z_{\sigma_s} - \bigcup_w \mathcal{M}_w$, 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 Let $t = 0$.

   repeat

   For each $w < \lfloor i/i_1 \rfloor$, for each $x \in Y_w$ such that $\text{card}(\mathcal{M}_w) \leq l - (l' - k')$, enumerate $W_{q(X_{1,w},X_2,X_{3,w},X_{4,w},X_2',(\mathcal{M}-Z_{\sigma_s})\cup\mathcal{M}_w,map(x)),t} - \{\langle x, y\rangle \mid x \in S_3 \cup S_2''\}$ in $W_{p(x)}$.

   Let $t = t + 1$.

   forever

   End substage s'

3. Let $X = \bigcup_{x \in S_2''} W_{p(x)}$ enumerated till now.
   Let σ_{s+1} be an extension of τ such that $\text{content}(\sigma_{s+1}) = \text{content}(\tau) \cup X \cup \{\langle x, s\rangle \mid x \in S_4\}$.
   Enumerate $\text{content}(\sigma_{s+1})$ 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 $T = \bigcup_s \text{content}(\sigma_s)$. Clearly, for all $x \in S_2''$, $W_{p(x)} = \text{content}(T) \in \mathcal{L}$. 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.

It is easy to see that there can be at most finitely many substages in each stage which terminate. 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 $w < \lfloor i/i_1 \rfloor$, $\mathcal{L}_w \subseteq \mathcal{L}$ (since step 2.2 in stage s, substage s', makes, for each $x \in Y_w$, $W_{p(x)} = W_{q(X_{1,w},X_2,X_{3,w},X_{4,w},X_2',(\mathcal{M}-S_{\sigma_s})\cup\mathcal{M}_w,map(x))}$). Also, for each $w, w' < \lfloor i/i_1 \rfloor$, $w \neq w'$, $L_w \in \mathcal{L}_w$, $\mathbf{M} \in \mathcal{M}_{w'}$, $(\exists y)[\langle u_{w'}, y\rangle \in W_{\mathbf{M}(\sigma_s)} - L_w]$. Also, for some $w < \lfloor i/i_1 \rfloor$, $\text{card}(\mathcal{M}_w) \leq \lfloor \frac{k'}{\lfloor i/i_1 \rfloor} \rfloor$. Thus, since $\mathcal{L}_w \not\subseteq \mathbf{Team}_l^k\mathbf{TxtEx}((\mathcal{M} - Z_{\sigma_s}) \cup \mathcal{M}_w)$, we have $\mathcal{L} \not\subseteq \mathbf{Team}_{l'}^{k'}\mathbf{TxtEx}(\mathcal{M})$. ∎

Note that if $\text{PROP}(q,i,j,k,l)$, then $\mathbf{Team}_j^i\mathbf{TxtEx} - \mathbf{Team}_l^k\mathbf{TxtEx} \neq \emptyset$. This is so because $\mathcal{L} = \bigcup_{\{\mathcal{M} \mid \text{card}(\mathcal{M})=l\}} \mathcal{L}_{q,i,j,k,l,\{\langle 0,code(\mathcal{M})\rangle\},\{1,\ldots,j\},\{0\},\emptyset,\emptyset,\mathcal{M}} \in \mathbf{Team}_j^i\mathbf{TxtEx} - \mathbf{Team}_l^k\mathbf{TxtEx}$. 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.

Theorem 7 allowed us to extend 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'.

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 $\text{PROPS}(q,i,j,k,l)$ to be true just in case given

(a) finite sets S_1, S_2, S_3, S_4, S_2',

(b) a team of l machines $\mathcal{M}$,

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 $\mathcal{L}_{q,i,j,k,l,S_1,S_2,S_3,S_4,S'_2,\mathcal{M}} \not\subseteq \mathbf{Team}_l^k\mathbf{TxtEx}^*(\mathcal{M})$, 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) $(\forall x \in S_4)[\mathrm{card}(\{y \mid \langle x,y\rangle \in L\}) = \infty]$,
(c) $\mathrm{card}(\{x \in S_2 \mid \max(\{y \mid \langle x,y\rangle \in L\})$ exists $\wedge\ W_{\max(\{y \mid \langle x,y\rangle \in L\})} = L\}) \geq i$,
(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) $(\forall x \in S'_2)[S_1 \subseteq W_{q(S_1,S_2,S_3,S_4,S'_2,\mathcal{M},x)} \subseteq S_1 \cup \{\langle x,y\rangle \mid x \notin S_3 \ \wedge\ y \in N\}$.
$\}$

We now employ the predicate PROPS to prove the following theorem which is analogous to Theorem 7.

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 [4], 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 $y \in S''_2$, enumerate $S_1 \cup \{\langle x,p(x)\rangle \mid x \in S''_2\}$ in $W_{p(y)}$. 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 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 $\mathcal{M}_1 = \mathcal{M} - Z_{\sigma_s}$.
For $w < \lfloor i/i_1 \rfloor$, let Y_w be pairwise disjoint subsets of S''_2 of cardinality i_1 each.
For $w < \lfloor i/i_1 \rfloor$, let u_w 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 $w < \lfloor i/i_1 \rfloor$, let $X_{3,w} = \{u_r \mid r < \lfloor i/i_1 \rfloor \ \wedge \ r \neq w\} \cup S_3 \cup S''_2$.
For $w < \lfloor i/i_1 \rfloor$, let $X_{4,w} = \{u_w\} \cup S_4$.
Let *map* be a mapping from S''_2 to S_5 such that for each $w < \lfloor i/i_1 \rfloor$, $map(Y_w) = S_5$.
Let $t = 0$.
repeat
For each $w < \lfloor i/i_1 \rfloor$, for each $x \in Y_w$, enumerate $W_{q(X_1, X_2, X_{3,w}, X_{4,w}, X'_2, \mathcal{M}_1, map(x)), t}$ in $W_{p(x)}$.
Let $t = t + 1$.
forever

3. Let $X = \bigcup_{x \in S''_2} W_{p(x)}$ enumerated till now.
Let σ_{s+1} be an extension of τ such that $\text{content}(\sigma_{s+1}) = \text{content}(\tau) \cup X \cup \{\langle x, s \rangle \mid x \in S_4\}$.
Enumerate $\text{content}(\sigma_{s+1})$ 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 $\mathcal{L} \not\subseteq \mathbf{Team}^{k'}_{l'}\mathbf{TxtEx}^*(\mathcal{M})$. We consider the following cases.

Case 1: All stages terminate.

In this case, let $T = \bigcup_s \text{content}(\sigma_s)$. Clearly, for all $x \in S''_2$, $W_{p(x)} = \text{content}(T) \in \mathcal{L}$. Moreover, at most $k' - 1$ of the machines in $\mathcal{M}$ converge on T. Thus, $\mathcal{L} \not\subseteq \mathbf{Team}^{k'}_{l'}\mathbf{TxtEx}^*(\mathcal{M})$.

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 $\mathcal{L}_w \not\subseteq \mathbf{Team}^{k}_{l}\mathbf{TxtEx}^*(\mathcal{M}_1)$, we have $\mathcal{L} \not\subseteq \mathbf{Team}^{k'}_{l'}\mathbf{TxtEx}^*(\mathcal{M})$. ∎

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

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 [23] between team function identification and probabilistic function identification.

Theorem 9 [22, 25] $(\forall j > 0)[\mathbf{Team}^{j}_{2j}\mathbf{Ex} = \mathbf{Team}^{1}_{2}\mathbf{Ex}]$.

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 [22] 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 5 $\mathbf{Team}_{2j+1}^{j}\mathbf{TxtEx} - \mathbf{Team}_2^1\mathbf{TxtEx}^* \neq \emptyset$.

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 6 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 6 $(\forall j)[\mathbf{Team}_{4j+2}^{2j+1}\mathbf{TxtEx} = \mathbf{Team}_2^1\mathbf{TxtEx}]$.

Corollary 7 $(\forall j)(\forall i)[\mathbf{Team}_{2j+1}^{j+1}\mathbf{TxtEx}^i \subseteq \mathbf{Team}_2^1\mathbf{TxtEx}^{i\cdot\lceil (j+1)/2\rceil}]$.

Corollary 8 $(\forall j)(\forall i)[\mathbf{Team}_{2j+1}^{j+1}\mathbf{TxtEx}^i \subseteq \mathbf{Team}_{2j+3}^{j+2}\mathbf{TxtEx}^{i\cdot\lceil (j+1)/2\rceil}]$

Proof of Theorem 11. Suppose $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_{4j+2}$ $\mathbf{Team}_{4j+2}^{2j+1}\mathbf{TxtEx}^i$-identify $\mathcal{L}$. Let $\mathbf{M}_1'$ and $\mathbf{M}_2'$ be defined as follows.

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

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

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

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

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

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

Finally, we settle the question for team success ratio 1/2 by establishing Theorem 13 below. We would like to note that our proof of the following theorem turns out to be the most complicated in the present paper.

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

PROOF OF THEOREM 12. Consider the following class of languages.

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

It is easy to see that $\mathcal{L} \in \mathbf{Team}_{4n}^{2n}\mathbf{TxtEx}$. Suppose by way of contradiction that the team $\{\mathbf{M}_0, \mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_{2n-1}\}$ are such that $\mathcal{L} \subseteq \mathbf{Team}_{2n}^{n}\mathbf{TxtEx}(\{\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_{2n-1}\})$. Then by the implicit use of the operator recursion theorem [4], 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 step 5. Step 4 sets up the conditions for step 5 to act. On the completion of step 5, step 6 easily achieves diagonalization using essentially the technique developed in the proof of Theorem 6.

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 $\text{content}(\sigma_0) = \{\langle 0, p(0)\rangle, \langle 1, p(1)\rangle, \ldots, \langle 2n-1, p(2n-1)\rangle\}$. Let avail $= 2n-1$ (intuitively, avail denotes the least number such that, for all $i >$ avail, $p(i)$ is available for diagonalization). Go to stage 0.

Begin stage s

1. Let $Z \subseteq [0 \,..\, 2n-1]$ be such that, $\text{card}(Z) = n$ and for $i \in Z$ and for $j \in ([0 \,..\, 2n-1] - Z)$, $\mathbf{lmc}(\mathbf{M}_i, \sigma_s) \leq \mathbf{lmc}(\mathbf{M}_j, \sigma_s)$.
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 $\text{content}(\tau) - \text{content}(\sigma_s) \subseteq \{\langle x, y\rangle \mid x \geq 2n\}$.
4. For $i < n$, let $q_i = p(\text{avail} + 1 + i)$.
   Let avail = avail + n.
   For $i < n$, enumerate $\langle 2n + i, q_i\rangle$ into $W_{p(0)}$.
   For $i < n$, enumerate $W_{p(0)}$ enumerated till now into $W_{p(i)}$ and W_{q_i}.
   Let $m = 1 + \max(\{x \mid \{\langle 4n, x\rangle, \langle 4n+1, x\rangle\} \cap (W_{p(0)} \text{ enumerated till now}) \neq \emptyset\})$.
   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 $\text{card}(Y) \geq n/2$ 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 $\text{content}(\tau_0) = W_{p(0)}$ enumerated till now. Go to substage 4b—0.

Begin substage 4b—t

4b.1. For $i < n$, let $q^1_{n+i} = p(\text{avail} + 1 + i)$.
For $i < n$, let $q^2_{n+i} = p(\text{avail} + n + 1 + i)$.
Let 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. Let $m_1 = 1 + \max(\{x \mid \{\langle 4n, x\rangle, \langle 4n+1, x\rangle\} \cap (W_{p(0)} \text{ enumerated till now}) \neq \emptyset\})$.
For $i < n$, enumerate $W_{p(0)}$ enumerated till now into $W_{q^1_{n+i}}$ and $W_{q^2_{n+i}}$.
For $i < n$ and $j < n$, enumerate $\langle 3n + i, q^1_{n+i}\rangle$ into $W_{p(j)}$ and $W_{q^1_{n+j}}$.
For $j < n$, enumerate $\langle 4n, m_1\rangle$ into $W_{p(j)}$ and $W_{q^1_{n+j}}$.
For $i < n$ and $j < n$, enumerate $\langle 3n + i, q^2_{n+i}\rangle$ into W_{q_j} and $W_{q^2_{n+j}}$.
For $j < n$, enumerate $\langle 4n+1, m_1\rangle$ into 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 content(γ) − 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 $S =$ content(γ) $\cup W_{p(0)}$ enumerated till now $\cup W_{q_0}$ enumerated till now.

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

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

Go to substage 4b—$t+1$.

End substage 4b—t

5. Let Y be as found in step 4a.

Let $v = 4n+2$. $X = \{x \mid x < n\}$.

while card$(X) > 1$ **do**

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

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

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

(iv) card$(X) \leq n$. ∗)

(∗ *Moreover, after each iteration of the while loop,* card(X) *decreases (actually* card(X) *nearly halves after each iteration)* ∗).

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

Let avail = avail + card(X).

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

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

For $i <$ card(X) and $j \in X_1 \cup X_2$ and $k <$ card(X), enumerate $\langle 2n+i, q_i\rangle$ into $W_{p(j)}$ and W_{q_k}.

Let τ_0 be an extension of σ_s such that content$(\tau_0) = W_{q_0}$ enumerated till now.

Go to substage 5—0.

Begin substage 5—t

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

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

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

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

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

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

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

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

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

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

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

5a. Search for a $Y' \subseteq (Z - Y)$, such that $\text{card}(Y') = \text{card}(Y)$ 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 $\langle l, x\rangle$.

5b. 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 $\text{content}(\gamma) - \text{content}(\tau_t) \subseteq \{\langle 2n + \text{card}(X) + i, q^1_{\text{card}(X)+i}\rangle, \langle 2n + \text{card}(X) + i, q^2_{\text{card}(X)+i}\rangle \mid i < 2n - \text{card}(X)\} \cup \{\langle v, m_1\rangle, \langle v+1, m_1\rangle\}$.

5c. Let γ be as found in step 5c.

Let $S = \text{content}(\gamma) \cup W_{q_0}$ enumerated till now $\cup W_{q_{\text{card}(X)-1}}$ enumerated till now.

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

For each $q \in \{q_i \mid i < \text{card}(X)\}$, enumerate S into W_q.

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

Go to substage 5—$t+1$.

End substage 5—t

5d. Let Y' be as found in step 5a.

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

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

if $\text{card}(Y_1)/\text{card}(X_1) > 1/2$, **then** let $X = X_1, Y = Y_1$.

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

endif

$v = v + 2$.

endwhile

6. (* *Note that* $\text{card}(X) = 1$ *and* $\text{card}(Y) \geq 1$. *)

Let $v = v + 2$.

Let $q_0 = p(\text{avail} + 1)$.

Let $\text{avail} = \text{avail} + 1$.

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

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

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

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

Go to substage 6—0.

Begin substage 6—t

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

Let $\text{avail} = \text{avail} + 2n \cdot (2n-1)$.

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

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

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

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

For $j < 2n$ and $i < 2n-1$, enumerate $W_{p(i_j)}$ enumerated till 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 $\text{content}(\gamma) - \text{content}(\tau_t) \subseteq \{\langle 2n+1+i, q^j_{1+i}\rangle \mid j < 2n \wedge i < 2n-1\} \cup \{\langle v+j, m_1\rangle \mid j < 2n\}$.

6c. Let γ be as found in step 6a.
Let $S = \text{content}(\gamma) \cup \bigcup_{j<2n} W_{p(i_j)}$ enumerated till now.
For $j < 2n$, enumerate S into $W_{p(i_j)}$.
Let τ_{t+1} be an extension of τ_t such that $\text{content}(\tau_{t+1}) = S$.
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 $S = \text{content}(\tau) \cup \bigcup_{i<2n} W_{p(i)}$ enumerated till now.
For $i < 2n$, enumerate S into $W_{p(i)}$.
Let σ_{s+1} be as extension of τ such that $\text{content}(\sigma_{s+1}) = S$.
Let $\text{avail} = \max(\{\text{avail}\} \cup \{x \mid (\exists i < 4n)[\langle i, p(x)\rangle \in S]\})$.
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 $\text{content}(T) \subseteq \text{content}(\sigma_s) \cup \{\langle x, y\rangle \mid x \geq 2n, y \in N\}$, $\mathbf{M}_i(T) = \mathbf{M}_i(\sigma_s)$. 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 does not reach step 5.

For $i < n$, 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, $\text{card}(\{i \mid \mathbf{M}_i \ \mathbf{TxtEx} \text{ identifies } L\}) < (\lceil n/2 + 1/2\rceil - 1) + n/2$. 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 (in the previous while loop) succeeded) which is neither in the language L defined in Case 2.2.1 below nor in L_1, 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 $i < \mathrm{card}(X)$, 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 $T = \bigcup_t \tau_t$. Moreover, for less than $\mathrm{card}(Y)$ many i's in $([0 \,..\, 2n-1] - Z)$, $\mathbf{M}_i$ converges on T.

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, 5b do not succeed in substage 5—t, at least one of L_1, L_2 is $\mathbf{TxtEx}$-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 Cases 2.3.2 below; thus, $\mathbf{M}_i$ does not $\mathbf{TxtEx}$ 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 $< \mathrm{card}(Y)$. 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 step substage 6—t. Also, let $i_0, \ldots, i_{2n-1}$ be as defined in substage 6—t. 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 $\tau_t \subseteq T_k$. 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 9 $(\forall m, n \in N^+)[\mathbf{Team}^m_{2m}\mathbf{TxtEx} \subseteq \mathbf{Team}^n_{2n}\mathbf{TxtEx} \Leftrightarrow [m$ *divides* $n \bigvee m$ *is odd*$]]$.

Corollary 10 $\mathbf{Prob}^{1/2}\mathbf{TxtEx} - \bigcup_m \mathbf{Team}^m_{2m}\mathbf{TxtEx} \neq \emptyset$.

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 ration 1/2. In the next section, 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 show 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 11 $(\forall a \in N)(\forall k \geq 2)(\forall$ *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 12 $(\forall k \geq 2)[\mathbf{Prob}^{1/k}\mathbf{TxtEx} - \bigcup_j \mathbf{Team}^j_{j\cdot k}\mathbf{TxtEx} \neq \emptyset]$.

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

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

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 $\mathbf{Team}^m_{3m}\mathbf{TxtEx} - \mathbf{Team}^n_{3n}\mathbf{TxtEx} \neq \emptyset$. Using Theorem 7 repeatedly with $i_1 = 0$ we get the result. ∎

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

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 ratio $< 1/2$ and for between success ratio $1/2$ 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 13 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 $\lceil\frac{mr}{r+1}\rceil > n - m$). Theorem now follows by using Theorem 8 with $i_1 = 1$. ∎

Corollary 13 $(\forall r)[\mathbf{Team}^{r+2}_{2r+3}\mathbf{TxtEx} - \mathbf{Team}^{r+1}_{2r+1}\mathbf{TxtEx}^* \neq \emptyset]$.

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 $\mathbf{Team}^7_{11}\mathbf{TxtEx} \subseteq \mathbf{Team}^2_3\mathbf{TxtEx}$.

Corollary 14 $\mathbf{Team}^7_{11}\mathbf{TxtEx} = \mathbf{Team}^2_3\mathbf{TxtEx}$.

PROOF OF THEOREM 21. Given a team $\{\mathbf{M}_1, \ldots, \mathbf{M}_{11}\}$, we will construct three learning machines $\mathbf{M}'_1, \mathbf{M}'_2$, and $\mathbf{M}'_3$ such that the team $\{\mathbf{M}'_1, \mathbf{M}'_2, \mathbf{M}'_3\}$ $\mathbf{Team}_3^2\mathbf{TxtEx}$-identifies any language $\mathbf{Team}_{11}^7\mathbf{TxtEx}$-identified by the team $\{\mathbf{M}_1, \ldots, \mathbf{M}_{11}\}$. Let conv be as defined in the proof of Theorem 2. Let $m_1^\sigma, m_2^\sigma, \ldots, m_{11}^\sigma$ be a permutation of $1, 2, \ldots, 11$, such that, for $1 \leq r < 11$, $[(\text{conv}(\mathbf{M}_{m_r^\sigma}, \sigma), m_r^\sigma) < (\text{conv}(\mathbf{M}_{m_{r+1}^\sigma}, \sigma), m_{r+1}^\sigma)]$. Let match be as defined in the proof of Theorem 11 (with $i = 0$). Let $\text{similar}(i, j, n) = \max(\{n_1 \leq n \mid W_{i,n_1} \subseteq W_{j,n} \wedge W_{j,n_1} \subseteq W_{i,n}\})$. Intuitively, similar computes the closeness between two grammars. It denotes the point where it appears that the languages accepted by the two grammars differ.

Let $r_1^\sigma, \ldots, r_7^\sigma$ be a permutation of $m_1^\sigma, \ldots, m_7^\sigma$, be such that for $1 \leq l \leq 6$. $(\text{match}(\mathbf{M}_{r_l^\sigma}(\sigma), \sigma), r_l^\sigma) > (\text{match}(\mathbf{M}_{r_{l+1}^\sigma}(\sigma), \sigma), r_{l+1}^\sigma)$.

$\mathbf{M}'_3$ on σ outputs

$$\text{majority}(\mathbf{M}_{r_1^\sigma}(\sigma), \mathbf{M}_{r_2^\sigma}(\sigma), \mathbf{M}_{r_3^\sigma}(\sigma), \mathbf{M}_{r_4^\sigma}(\sigma), \mathbf{M}_{r_5^\sigma}(\sigma), \mathbf{M}_{m_8^\sigma}(\sigma), \mathbf{M}_{m_9^\sigma}(\sigma)).$$

Suppose a text T is given for $L \in \mathbf{Team}_{11}^7\mathbf{TxtEx}(\{\mathbf{M}_1, \ldots, \mathbf{M}_{11}\})$. Clearly, for $1 \leq j \leq 7$, $\lim_{n\to\infty} m_j^{T[n]}$ exists. Let $\mathcal{M} = \{\mathbf{M}_{\lim_{s\to\infty} m_j^{T[s]}} \mid 1 \leq j \leq 7\}$. Now, $\mathbf{M}'_3$ $\mathbf{TxtEx}$-identifies T if at least 2 of the machines in $\mathcal{M}$ converge to a wrong grammar on T. $\mathbf{M}'_1, \mathbf{M}'_2$ will be constructed so that if at least 6 of the machines in $\mathcal{M}$ converge, on T, to a correct grammar, then $\mathbf{M}'_1, \mathbf{M}'_2$ $\mathbf{TxtEx}$-identify T. Otherwise, at least one of $\mathbf{M}'_1, \mathbf{M}'_2$ $\mathbf{TxtEx}$-identifies T. Note that at least 3 of the machines in $\mathcal{M}$ $\mathbf{TxtEx}$-identify T.

$\mathbf{M}'_1$ on σ outputs $G_1(\mathbf{M}_{m_1^\sigma}(\sigma), \ldots, \mathbf{M}_{m_7^\sigma}(\sigma))$ and $\mathbf{M}'_2$ on σ outputs $G_2(\mathbf{M}_{m_1^\sigma}(\sigma), \ldots, \mathbf{M}_{m_7^\sigma}(\sigma))$, where G_1, G_2 are as defined below.

Given $g_1, g_2, \ldots, g_7$, $W_{G_1(g_1,\ldots,g_7)}, W_{G_2(g_1,\ldots,g_7)}$ is defined as follows.

Let $n_0 = 0$, $m1_0 = m2_0 = 0$. For $1 \leq i \leq 7$, let $g'_{i,0} = g_i$. Let $G'_{1,0} = G_1(g_1, \ldots, g_7)$ and $G'_{2,0} = G_2(g_1, \ldots, g_7)$. We will enumerate elements in $W_{G_1(g_1,\ldots,g_7)}$, $W_{G_2(g_1,\ldots,g_7)}$ in stages. $G'_{1,s}$, $G'_{2,s}$ will be a permutation of $G_1(g_1, \ldots, g_7), G_2(g_1, \ldots, g_7)$ and $g'_{1,s}, \ldots, g'_{7,s}$ will be a permutation of $g_1, \ldots, g_7$. This is just for the ease of presentation.

Begin $\{\text{stage } s\}$

Search for $n > n_s$ such that there exist distinct $p_1, p_2, p_3 \in [1 \,..\, 7]$, such that $\text{similar}(g'_{r,s}, g'_{l,s}, n) > n_s$, for $r, l \in \{p_1, p_2, p_3\}$.

If and when such an n is found, let $n_{s+1} = n$.

Let $p_1, p_2, p_3 \in [1 \,..\, 7]$ be such that p_1, p_2, p_3 are distinct and $\min(\text{similar}(g'_{p_1,s}, g'_{p_2,s}, n), \text{similar}(g'_{p_1,s}, g'_{p_3,s}, n), \text{similar}(g'_{p_2,s}, g'_{p_3,s}, n))$ is maximized.

Let $p_4, p_5, p_6 \in [1 \,..\, 7] - \{p_1, p_2, p_3\}$ be such that p_4, p_5, p_6 are distinct and $\min(\text{similar}(g'_{p_4,s}, g'_{p_5,s}, n), \text{similar}(g'_{p_4,s}, g'_{p_6,s}, n), \text{similar}(g'_{p_5,s}, g'_{p_6,s}, n))$ is maximized.

Let $m1_{s+1} = \min(\text{similar}(g'_{p_1,s}, g'_{p_2,s}, n), \text{similar}(g'_{p_1,s}, g'_{p_3,s}, n), \text{similar}(g'_{p_2,s}, g'_{p_3,s}, n))$.

Let $m2_{s+1} = \min(\text{similar}(g'_{p_4,s}, g'_{p_5,s}, n), \text{similar}(g'_{p_4,s}, g'_{p_6,s}, n), \text{similar}(g'_{p_5,s}, g'_{p_6,s}, n))$.

If $\text{card}(\{p_1, p_2, p_3\} \bigcap \{1, 2, 3\}) \leq 1$, then let $G'_{1,s+1} = G'_{2,s}$ and $G'_{2,s+1} = G'_{1,s}$.

Enumerate $W_{g'_{p_1,s}, m1_{s+1}} \cup W_{g'_{p_2,s}, m1_{s+1}} \cup W_{g'_{p_3,s}, m1_{s+1}}$ in $W_{G'_{1,s+1}}$.

Enumerate $W_{g'_{p_4,s}, m2_{s+1}} \cup W_{g'_{p_5,s}, m2_{s+1}} \cup W_{g'_{p_6,s}, m2_{s+1}}$ in $W_{G'_{2,s+1}}$.

Let $g'_{i,s+1} = g'_{p_i,s}$ for $1 \leq i < 7$.

Let $g'_{7,s+1} = g'_{\text{pleft},s}$, where $\text{pleft} \in ([1 \,..\, 7] - \{p_1, \ldots, p_6\})$.

Go to stage $s + 1$.

End $\{\text{stage } s\}$

It is easy to see that if at least 6 of the first 7 converging machines **TxtEx**-identify T, then both $\mathbf{M}'_1$ and $\mathbf{M}'_2$ do. We prove below that at least one of $\mathbf{M}'_1, \mathbf{M}'_2$ **TxtEx**-identify T, if it is **TxtEx**-identified by at least 3 of the first seven converging machines. It is sufficient to show that if at least 3 of $g_1, \ldots, g_7$ are grammars for a language L, then at least one of $G_1(g_1, \ldots, g_7)$, $G_2(g_1, \ldots, g_7)$, accepts L. For $r \leq 1$, let $W^s_{G_r}$ denote W_{G_r} enumerated before stage s. It is easy to show by induction that, before stage s following hold.

1. $W^s_{G'_{1,s}} = W_{g'_{1,s}, m1_s} \cup W_{g'_{2,s}, m1_s} \cup W_{g'_{3,s}, m1_s}$.
2. $W^s_{G'_{2,s}} \supseteq W_{g'_{4,s}, m2_s} \cup W_{g'_{5,s}, m2_s} \cup W_{g'_{6,s}, m2_s}$.
3. $W^s_{G'_{1,s}} \subseteq W_{g'_{1,s}, n_s} \cap W_{g'_{2,s}, n_s} \cap W_{g'_{3,s}, n_s}$.
4. $W^s_{G'_{2,s}} - [W_{g'_{4,s}, m2_s} \cup W_{g'_{5,s}, m2_s} \cup W_{g'_{6,s}, m2_s}] \subseteq W^s_{G'_{1,s}}$.
5. $(\forall x \in W^s_{G'_{2,s}})(\exists \text{ distinct } j, k \in [4:7])[x \in W_{g'_{j,s}, n_s} \wedge x \in W_{g'_{k,s}, n_s}]$.
6. $m1_s \geq m2_s$.

Thus, if at least 3 of $g_1, \ldots, g_7$ are grammars for L, then at least one of $G_1(g_1, \ldots, g_7)$, $G_2(g_1, \ldots, g_7)$, enumerates L. ∎

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

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

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

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

Iterating the above method we get,

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

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

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

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

Proof of Theorem 26. By Corollary 7 we know that for any m and n, such that $m > n/2$, $\mathbf{Team}^m_n\mathbf{TxtEx} \subseteq \mathbf{Team}^1_2\mathbf{TxtEx}$. Suppose machines $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_l$ are given. For $\emptyset \neq S \subseteq \{1, 2, \ldots, l\}$, let $\mathbf{M}^1_S$, $\mathbf{M}^2_S$ denote the two machines which $\mathbf{Team}^1_2\mathbf{TxtEx}$-identify any language which is $\mathbf{Team}^{\lfloor \mathrm{card}(S)/2 \rfloor + 1}_{\mathrm{card}(S)}$-identified by machines $\{\mathbf{M}_i\}_{i \in S}$.

We now define $\mathbf{M}_a, \mathbf{M}_b$, and $\mathbf{M}_c$ which $\mathbf{Team}^1_3\mathbf{TxtEx}$-identify any language which is $\mathbf{Team}^k_l\mathbf{TxtEx}$-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 $i' \in \{1,2,\ldots,l\} - S_\sigma$, $\text{conv}(\mathbf{M}_i,\sigma) \leq \text{conv}(\mathbf{M}_{i'},\sigma)$. Then, let $\mathbf{M}_a(\sigma) = \text{majority}(\{\mathbf{M}_r(\sigma) \mid r \in S_\sigma\})$.

Let $\text{match}(i,\sigma) = \max(\{x \leq |\sigma| \mid (\text{content}(\sigma[x]) \subseteq W_{r,|\sigma|}) \ \wedge \ (W_{r,x} \subseteq \text{content}(\sigma))\})$. 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 15 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 15 $\mathbf{Team}^3_7\mathbf{TxtEx} \subseteq \mathbf{Team}^1_3\mathbf{TxtEx}$.

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

PROOF. Follows from team function hierarchy of Smith [30], $(\forall n \in N^+)[\mathbf{Team}^1_n\mathbf{Ex} \subset \mathbf{Team}^1_{n+1}\mathbf{Ex}]$, and Pitt's connection for functions [23], $(\forall p \mid 0 < p \leq 1)(\forall n)[1/(n+1) < p \leq 1/n \Rightarrow \mathbf{Team}^1_n\mathbf{Ex} = \mathbf{Prob}^p\mathbf{Ex}]$. ▮

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

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

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

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

The next definition formalizes identification from informants.

Definition 22 (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 [22] could be used to show that for **TxtBc**-identification (see [5] for definition), if $i > j/2$, then $\mathbf{Team}^i_j\mathbf{TxtBc} = \mathbf{TxtBc}$. Thus, team inference with respect to **TxtBc**-identification behaves differently from team inference with respect to **TxtEx**-identification. A study of probabilistic and team identification for **TxtBc**-identification on the lines of the present paper is open. We would also like to 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, 31, 17, 8, 7].

Acknowledgements

We would like thank John Case for suggesting this investigation, providing helpful critical comments, and discussing various aspects of this work. We would also like to express our gratitude to Mark Fulk and Dan Osherson for supporting us during this research at various stages. Lata Narayanan and Rajeev Raman provided helpful discussion.

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

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

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

[5] J. Case and C. Lynes. Machine inductive inference and language identification. *Lecture Notes in Computer Science*, 140:107–115, 1982.

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

[7] R. P. Daley, B. Kalyanasundaram, and M. Velauthapillai. Breaking the probability 1/2 barrier in fin-type learning. In *Proceedings of the Workshop on Computational Learning Theory*, 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. *Mathematical Foundations of Computer Science*, 1975.

[10] R. Freivalds. Finite identification of general recursive functions by probabilistic strategies. In *Proceedings of the Conference on Algebraic, Arithmetic and Categorical Methods in Computation Theory*, pages 138–145. Akedemie-Verlag, Berlin, 1979.

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

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

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

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

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

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

[24] 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.

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

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

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

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

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

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

[31] M. Velauthapillai. Inductive inference with bounded number of mind changes. In *Proceedings of the Workshop on Computational Learning Theory*, pages 200–213, 1989.