On the Minimality of Definite Tell-tale Sets in Finite Identification of Languages

Nina Gierasimczuk **Dick de Jongh**
Institute for Logic, Language and Computation
Universiteit van Amsterdam
{n.gierasimczuk,d.h.j.dejongh}@uva.nl

Abstract

This work is concerned with notions of minimality of definite tell-tale sets used in finite identification of languages. We base our considerations on the observation that a class of languages is finitely identifiable from positive data if every element of this class includes a definite tell-tale set, i.e. a finite subset that separates the language from other possibilities (Mukouchi, 1992). We discuss two notions of minimality of definite tell-tales in finite identification. We show that the problem of finding a minimal definite tell-tale for one language from a finitely identifiable finite class of finite sets is PTIME computable, while the problem of finding a minimal-size definite tell-tale set is NP-complete. In the last section, we restrict out attention to preset learners, finite learners who use in their identification task one fixed DFTT for each language. We define the non-effective fastest learning function that performs finite identification on the basis of minimal definite finite tell-tale sets. In connection to the fastest learner we show that the problem of finding the set of all minimal definite tell-tale sets for a finite language from a finite class is NP-hard. We conclude with results about recursion-theoretic complexity, showing that finding the minimal size DFTTs is not always possible in a recursive manner.

1 Introduction

To finitely identify a language means to be able to recognize it with certainty after receiving some (specific) finite sample of this language. Such a finite sample that suffices for finite identification is called *definite finite tell-tale* (DFTT, for short, see: (Mukouchi, 1992)). One can see such a DFTT as the collection of the most characteristic elements of the set, but it has also a different connotation that is based on the *eliminative power* of its elements. We can think of the information that is carried by a particular sample of the language in a negative way, as showing which of the hypotheses are inconsistent with the information that has arrived, and thereby eliminating them. A set is finitely identifiable if it has comprised in a finite subset the power of eliminating all languages under consideration different from itself. These finite subsets are the definite finite tell-tales.

From the characterization of finite identifiability (Mukouchi, 1992), we know that if a class of languages is finitely identifiable, then the identification can be done on the basis of corresponding DFTTs, i.e. finite subsets of the original languages that contain a sample that is essential for finite identifiability. A number of issues emerge when analyzing computational properties of these definite finite tell-tales. Since finite tell-tales are by no means unique it is obviously useful to obtain small tell-tales. In this context we are particularly interested in the possibility of distinguishing two notions of minimality for DFTTs. A *minimal DFTT* is a DFTT that cannot be further reduced without losing its eliminative power with respect to a class of languages. A *minimal-size DFTT* of a set L, is a DFTT that is one of those which are smallest among all possible DFTTs of L.

In order to investigate the computational complexity of finding such minimal DFTTs, we will have to restrict ourselves to finite classes of languages. This may seem to be a very heavy restriction, but it creates the possibility of grasping important aspects of the complexity of finite identification. It also allows discussing the complexity of finite identifiability from the perspective of a teacher. In particular, it allows estimating how complex it is to find an optimal sample of a language allowing finite identifiability with respect to a certain class, and expose it to the pupil. We also investigate the analogous problems concerning recursion-theoretic complexity.

The plan of the paper is as follows. We introduce some basic notions, and provide the definition and characterization of finite identifiability. Then we present the refined notions of minimal DFTT, and minimal-size DFTT. We show that the problem of finding a minimal-size DFTT is NP-complete (using the well-known Minimum Set Cover Problem (Garey & Johnson, 1979)), while the problem of finding any minimal DFTT is PTIME computable. Therefore, it can be argued that it is harder for a teacher to provide a minimal-size optimal sample, than just any minimal information. With this context in mind we restrict the attention to preset learners, learning functions which use for the identification of each language one fixed preset DFTT. In the end we define the non-effective fastest learning function that performs finite identification on the basis of minimal DFTTs. Its computational complexity in the finite case turns out to be NP-hard. Similar results are obtained in the analogous recursion-theoretic context. As so often the best is the enemy of the better.

2 Notation and Basic Definitions

Let U be an infinite recursive set, we call any $L \subseteq U$ a language. A class of languages $\mathcal{L} = \{L_1, L_2, \ldots\}$ is an indexed family of recursive languages if there is a computable function $f : \mathbb{N} \times U \to \{0, 1\}$, such that

$$f(i, w) = \begin{cases} 1 & \text{if } w \in L_i, \\ 0 & \text{if } w \notin L_i. \end{cases}$$

In the rest of this paper we will consider indexed families of recursive languages.

Later on, in order to consider computational complexity, we will devote our attention to finite classes of sets. Then we take the class $\mathcal{L}$ to be $\{L_1, L_2, \ldots, L_n\}$. We use $\mathcal{I}_{\mathcal{L}} = \{i \mid L_i \in \mathcal{L}\}$ to denote the set of indices of sets in $\mathcal{L}$.

Definition 1 (Environment) *By an environment ε of L we mean an infinite sequence of elements from L enumerating all and only the elements from L (allowing repetitions).*

Definition 2 (Notation) *We will use the following notation:*

- *ε_n is the n-th element of ε;*
- *$\varepsilon | n$ is the sequence $(\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{n-1})$;*
- *$set(\varepsilon)$ is the set of elements that occur in ε;*
- *φ is a learning function — a recursive map from finite data sequences to indexes of hypotheses, $\varphi : U^* \to \mathbb{N} \cup \{\uparrow\}$. The function is allowed to refrain from giving a natural number answer, in that case the output is marked by $\uparrow$. In particular, as we will see below, in finite identification the function is defined to give a natural number as an answer from $I_{\mathcal{L}}$ only once.*

Finite identifiability of a class of languages from positive data is defined by the following chain of conditions.

Definition 3 (Finite identification, FIN) *A learning function φ:*

1. *finitely identifies $L_i \in \mathcal{L}$ on ε iff, when inductively given ε, at some point φ outputs i, and stops;*
2. *finitely identifies $L_i \in \mathcal{L}$ iff it finitely identifies L_i on every ε for L_i;*
3. *finitely identifies $\mathcal{L}$ iff it finitely identifies every $L_i \in \mathcal{L}$;*
4. *a class $\mathcal{L}$ is finitely identifiable iff there is a learning function φ that finitely identifies $\mathcal{L}$.*

In the last section we will relax the condition of recursivity of φ to discuss some case of non-effective finite identifiability.

3 Definite tell-tale sets and finite identification

A necessary and sufficient condition for finite identifiability has already been formulated in the literature (Mukouchi, 1992). It involves a modified, stronger notion of finite tell-tale (Angluin, 1980), namely the *definite finite tell-tale*.

Definition 4 (Mukouchi, 1992) *A set S_i is a definite finite tell-tale for $L_i \in \mathcal{L}$ if*

1. *$S_i \subseteq L_i$,*

2. S_i *is finite, and*
3. *for any index* j*, if* $S_i \subseteq L_j$ *then* $L_i = L_j$*.*

Finite identifiability can be then characterized in the following way.

Theorem 5 (Mukouchi, 1992) *A class* $\mathcal{L}$ *is finitely identifiable from positive data iff there is an effective procedure that on input* i *produces all elements of a definite finite tell-tale of* L_i*.*

In other words, each set in a finitely identifiable class contains a finite subset that distinguishes it from all other sets in the class.

4 Eliminative Power and Finite Identifiability

Identifiability in the limit (Gold, 1967) of a class of languages guarantees the existence of a reliable strategy that allows for convergence to a correct hypothesis for every language from the class. The exact moment at which a correct hypothesis has been reached is not known and in general can be uncomputable. Things are different in finite identifiability. Here, the learning function is allowed to answer only once. Hence, the conjecture is based on certainty. In other words, the learner must know that the answer she gives is true, because there is no place for a change of mind later.

Knowing that one hypothesis is true means to be able to exclude all other possibilities. In this section we define the notion of *eliminative power* of a piece of information, which reflects the informative strength of data with respect to a certain class of sets.

Definition 6 *Let us take* $\mathcal{L}$ *— an indexed class of recursive languages, and* $x \in \bigcup \mathcal{L}$*.* The eliminative power of x with respect to $\mathcal{L}$ *is determined by a function* $El_{\mathcal{L}} : \bigcup \mathcal{L} \to \wp(\mathbb{N})$*, such that:*

$$El_{\mathcal{L}}(x) = \{i \mid x \notin L_i \ \& \ L_i \in \mathcal{L}\}.$$

Additionally, we will use $El_{\mathcal{L}}(X)$ *for* $\bigcup_{x \in X} El_{\mathcal{L}}(x)$*.*

In other words, function $El_{\mathcal{L}}$ takes x and outputs the set of indexes of all the sets in $\mathcal{L}$ that are inconsistent with x, and therefore in the light of the information x they can be "eliminated".

We can now characterize finite identifiability in terms of the eliminative power. The following is easy to observe.

Proposition 1 *A set* S_i *is a definite tell-tale set of* $L_i \in \mathcal{L}$ *iff*

1. $S_i \subseteq L_i$*,*
2. S_i *is finite, and*
3. $El_{\mathcal{L}}(S_i) = \mathcal{I}_{\mathcal{L}} - \{i\}$*.*

Moreover, from Theorem 5 we know that finite identifiability of an indexed family of recursive languages requires that every set in a class has a DFTT. Formally:

Theorem 7 *A class* $\mathcal{L}$ *is finitely identifiable from positive data iff there is an effective procedure that for any* i *supplies a finite set* $S_i \subseteq L_i$*, such that*

$$El_{\mathcal{L}}(S_i) = \mathcal{I}_{\mathcal{L}} - \{i\}.$$

4.1 The Computational Complexity of Finite Identifiability Check

As has already been mentioned in the introduction, we aim at analyzing the computational complexity of finding DFTTs. In order to do that we restrict to finite classes of finite sets. One may ask about the purpose of further reduction of sets that are already finite. In fact, if a finite class of finite sets is finitely identifiable, then each element of the class is already its own DFTT. However, finite sets can be much larger than their DFTTs. For example, we can take a class of the following form:

$$\mathcal{L} = \{L_i = \{2i, 2^i \text{ first odd integers}\} \mid i = 1, \ldots, n\}.$$

In the case of $\mathcal{L}$ a reduction to the minimal information that suffices for finite identification makes a significant difference in the length of the process of learning.

Theorem 8 *Checking whether a finite class of finite sets is finitely identifiable is polynomial with respect to the size of the class, i.e. the number of sets in the class and the maximal cardinality of sets in the class.*

Proof: The procedure consists of computing $El_{\mathcal{L}}(x)$ and checking whether for each $L_i \in \mathcal{L}$, $El(L_i) = \mathcal{I}_{\mathcal{L}} - \{i\}$.

Let us first focus on computing $El_{\mathcal{L}}(x)$ for $\bigcup \mathcal{L}$. We take $\mathcal{L}$ and assume that $|\mathcal{L}| = m$, and that the largest set in $\mathcal{L}$ has n elements.

In the first steps we have to obtain $\bigcup \mathcal{L}$. After this,we list for each element of $\bigcup \mathcal{L}$ the indices of the sets to which the element does not belong. In this step we have computed $El_{\mathcal{L}}(x)$ for each $x \in \bigcup \mathcal{L}$. All components of this procedure can clearly be performed in polynomial time with respect to m and n. It remains to check whether for all $L_i \in \mathcal{L}$, $\bigcup_{x \in L_i} El_{\mathcal{L}}(x) = \mathcal{I}_{\mathcal{L}} - \{i\}$. This involves essentially only the operation of sum. ■

From this analysis we conclude that checking whether a finite class of finite sets is finitely identifiable is quite easy, polynomial task. Nevertheless, as we saw in the example in the beginning of this section, it can be time consuming if n is a large number.

5 Finding a Minimal Definite Finite Tell-tale

We are now ready to introduce one of the two nonequivalent notions of minimality of the DFTTs. We will call S_i a minimal DFTT of L_i in $\mathcal{L}$ if and only if all the elements of the sets in S_i are essential for finite identification of L_i in $\mathcal{L}$, i.e. taking an element out of the set S_i will decrease its eliminative power with respect to $\mathcal{L}$, and hence it will no longer be a DFTT. We will show that a language can have many minimal DFTTs of different cardinalities. This will give us cause to introduce another notion of minimality — minimal-size DFTT.

Definition 9 *Let us take a finitely identifiable indexed family of recursive languages $\mathcal{L}$, and $L_i \in \mathcal{L}$. A* minimal DFTT *of L_i in $\mathcal{L}$ is an $S_i \subseteq L_i$, such that*

1. *S_i is a DFTT for L_i in $\mathcal{L}$, and*
2. *$\forall X \subset S_i \; El_{\mathcal{L}}(X) \neq \mathcal{I}_{\mathcal{L}} - \{i\}$.*

Theorem 10 *Let $\mathcal{L}$ be a finitely identifiable finite class of finite sets. Finding a minimal DFTT of $L_i \in \mathcal{L}$ can be done in polynomial time.*

Proof: Assume that the class $\mathcal{L}$: is finitely identifiable, finite and consists only of finite sets. From the assumptions, we know that for each $L_i \in \mathcal{L}$ a DFTT exists, in fact L_i is its own DFTT.

The following procedure yields a minimal DFTT for each $L_i \in \mathcal{L}$.

We want to find a set $X \subseteq L_i$ such that

$$El(X) = \mathcal{I}_{\mathcal{L}} - \{i\}, \text{ but } \forall Y \subset X \; El(Y) \neq \mathcal{I}_{\mathcal{L}} - \{i\}.$$

First we set $X := L_i$.

We look for $x \in X$ such that $El(X - \{x\}) = \mathcal{I}_{\mathcal{L}} - \{i\}$. If there is no such element, X is the desired DFTT. If there is such an x, we set $X := X - \{x\}$, and repeat the procedure.

Let $|L_i| = n$, where $|\cdot|$ stands for cardinality. The number of comparisons needed for finding a minimal DFTT of L_i in $\mathcal{L}$ is bounded by n^2. ■

5.1 Example

Let us consider the class

$$\mathcal{L} = \{L_1 = \{1, 3, 4\}, L_2 = \{2, 4, 5\}, L_3 = \{1, 3, 5\}, L_4 = \{4, 6\}\}.$$

The procedure of finding minimal DFTTs for sets in $\mathcal{L}$ is as follows.

1. We construct a list of the elements from $\bigcup \mathcal{L}$.
2. With each element x from $\bigcup \mathcal{L}$ we associate $El_{\mathcal{L}}(x) = \{i \,|\, x \notin L_i\}$, i.e. the set of indices of sets to which x does not belong (names of sets that are inconsistent with x). Table 1 shows the result of the two first steps for $\mathcal{L}$.
3. The next step is to find minimal DFTTs for every set in the class $\mathcal{L}$. As an example, let us take the first set $L_1 = \{1, 2, 3\}$. We order elements of L_1, and take the first element of the ordering. Let it be 1. We compute $El_{\mathcal{L}}(L_1 - \{1\})$, it turns out to be $\{2, 3, 4\}$. We therefore accept the set $\{3, 4\}$ as a smaller DFTT for L_1. Then we try to further reduce the obtained DFTT, by checking the next element in the ordering, let it be 3. $El_{\mathcal{L}}(\{3, 4\} - \{3\}) = \{4\} \neq \{2, 3, 4\}$, so 3 cannot be subtracted without loss of eliminative power. We perform the same check for the last

Table 1: Eliminative power of the elements in $\bigcup \mathcal{L}$ with respect to $\mathcal{L}$

x	$El_{\mathcal{L}}(x)$
1	$\{2,4\}$
2	$\{1,3,4\}$
3	$\{2,4\}$
4	$\{3\}$
5	$\{1,4\}$
6	$\{1,2,3\}$

Table 2: DFTTs of $\mathcal{L}$

set	a minimal DFTT
$\{1,3,4\}$	$\{3,4\}$
$\{2,4,5\}$	$\{4,5\}$
$\{1,3,5\}$	$\{3,5\}$
$\{4,6\}$	$\{6\}$

singleton $\{4\}$. It turns out that $\{3,4\}$ cannot further be reduced. We give $\{3,4\}$ as a minimal DFTT of L_1.[1]

4. We perform the same procedure for all the sets in $\mathcal{L}$. As a result we get minimal DFTTs for each $L_i \in \mathcal{L}$ presented in table 2.

6 Finding a Minimal-Size Definite Finite Tell-tale

We can use the notion of eliminative power to construct a procedure for finding minimal-size DFTTs of a finitely identifiable class $\mathcal{L}$.

Let us again take $\mathcal{L}$ — a class of finite sets. We assume that $|\mathcal{L}| = m$.

To find a DFTT of minimal size for the set $L_i \in \mathcal{L}$, one has to perform a search through all the subsets of L_i starting from singletons, looking for the first X_i, such that $El(X_i) = \mathcal{I}_{\mathcal{L}} - \{i\}$.

DFTTs of minimal size need not be unique. Which one is encountered first depends on the manner of performing the search. Below we describe the example discussed before.

6.1 Example

Let us consider again the class from Example 5.1, namely

$$\mathcal{L} = \{L_1 = \{1,3,4\}, L_2 = \{2,4,5\}, L_3 = \{1,3,5\}, L_4 = \{4,6\}\}.$$

1. We construct a list of the elements from $\bigcup \mathcal{L}$.

2. With each element x from $\bigcup \mathcal{L}$ we associate $El_{\mathcal{L}}(x) = \{i | x \notin L_i\}$, i.e. the set of hypotheses for sets to which x does not belong (names of sets that are inconsistent with x). Table 1 presents the result of the two first steps for $\mathcal{L}$.

3. The next step is to find minimal-size DFTTs for every set in the class $\mathcal{L}$. As an example, let us take the first set $L_1 = \{1,3,4\}$. We are looking for $X \subseteq L_1$ of minimal size, such that $El_{\mathcal{L}}(X) = \mathcal{I}_{\mathcal{L}} - \{L_1\}$.
   (a) We look for $\{x\}$ such that $x \in L_1$ and $El_{\mathcal{L}}(\{x\}) = \{2,3,4\}$. There is no such singleton.
   (b) We look for $\{x,y\}$ such that $x, y \in L_1$ and $El_{\mathcal{L}}(\{x,y\}) = \{2,3,4\}$. There are two such sets: $\{1,4\}$ and $\{3,4\}$.

4. We perform the same procedure for all $L_i \in \mathcal{L}$. As a result we get minimal-size DFTTs for each of $\mathcal{L}$, the result is presented in table 3.

Table 3: Minimal-size DFTTs of $\mathcal{L}$

set	minimal-size DFTTs
$\{1,3,4\}$	$\{1,4\}$ or $\{3,4\}$
$\{2,4,5\}$	$\{2\}$
$\{1,3,5\}$	$\{1,5\}$ or $\{3,5\}$
$\{4,6\}$	$\{6\}$

Table 4: A comparison of minimal and minimal-size DFTTs of $\mathcal{L}$

set	a minimal DFTT	minimal-size DFTTs
$\{1,3,4\}$	$\{3,4\}$	$\{1,4\}$ or $\{3,4\}$
$\{2,4,5\}$	$\{4,5\}$	$\{2\}$
$\{1,3,5\}$	$\{3,5\}$	$\{1,5\}$ or $\{3,5\}$
$\{4,6\}$	$\{6\}$	$\{6\}$

Let us now compare the two resulting reductions of sets from $\mathcal{L}$ (see Table 4). The case of L_2 shows that the two procedures give different outcomes. The actual difference in this case is not huge, but in bigger sets it can be significant.

6.2 Running time

Let us now analyze and discuss the running time of such a procedure.

First we need to compute $El_{\mathcal{L}}(x)$ for $\bigcup \mathcal{L}$. From the Theorem 8 we know that it can be done in polynomial time.

Now, let us approximate the number of steps needed to find a minimal-size DFTT of a chosen set $L_i \in \mathcal{L}$. We again assume that $|\mathcal{L}| = m$, and L_i has n elements.

In the procedure described above we performed a search through, in the worst case, all combinations from 1 to $|L_i|$, to find the right set $X \subseteq L_i$, such that $El_{\mathcal{L}}(X)$ satisfies the condition of eliminative all hypothesis but h_i. So, for each set L_i, the number of comparisons that have to be performed is:

$$n + \frac{n!}{2!(n-2)!} + \frac{n!}{3!(n-3)!} + \ldots + 1 = 2^{n-1}$$

6.3 Computational Complexity

It is costly to find minimal-size DFTTs. As we have seen above, our procedure leads to a complete search through the large space of all subsets of a language. We call this computational problem the Minimal-size DFTT Problem, and define it formally below. The problem is checking whether $L_i \in \mathcal{L}$ has a DFTT of size k or smaller.

Definition 11 (Minimal-size DFTT **Problem)**

Instance *A finite class of finite sets $\mathcal{L}$, a set $L_i \in \mathcal{L}$, and a positive integer $k \leq |L_i|$.*

Question *Is there a minimal DFTT $X_i \subseteq L_i$ of size $\leq k$?*

We are going to show that the Minimal-size DFTT Problem is NP-complete by pointing out that it is equivalent to the Minimum Set Cover Problem, which is know to be NP-complete (Karp, 1972). Let us recall it below.

Definition 12 (Minimum Set Cover **Problem)**

Instance: *Collection P of subsets of a finite set F, positive integer $k \leq |P|$.*

Question: *Does P contain a cover for X of size k or less, i.e. a subset $P' \subseteq P$ with $|P'| \leq k$ such that every element of X belongs to at least one member of P'?*

[1]Checking only singletons is enough because the eliminative power of sets is defined as the sum of the eliminative power of its elements.

Theorem 13 *The* MINIMAL-SIZE DFTT *Problem is NP-complete.*

Proof: First let us observe that by Theorem 7, MINIMAL-SIZE DFTT Problem is equivalent to the following Problem:

Definition 14 (MINIMAL-SIZE DFTT' **Problem)**

Instance: *Collection* $\{El(x)|x \in L_i\}$*, positive integer* $k \leq |L_i|$.

Question: *Does* $\{El(x)|x \in L_i\}$ *contain a cover for* $\mathcal{I}_{\mathcal{L}} - \{i\}$ *of size* k *or less, i.e. a subset* $Y_i \subseteq \{El(x)|x \in L_i\}$ *with* $|Y_i| \leq k$ *such that every element of* $\{El(x)|x \in L_i\}$ *belongs to at least one member of* Y_i*?*

It is easy to observe that MINIMAL-SIZE DFTT' Problem is a notational variant of MINIMUM SET COVER Problem, i.e. we take $F = \mathcal{I}_{\mathcal{L}}$, $P = \{El(x)|x \in L_i\}$ (and therefore $|P| = |L_i|$), and $X = \mathcal{L}_{\mathcal{L}} - \{i\}$. Therefore MINIMAL-SIZE DFTT' Problem is NP-complete. Since MINIMAL-SIZE DFTT' Problem is equivalent to MINIMAL-SIZE DFTT Problem, we conclude that MINIMAL-SIZE DFTT Problem is also NP-complete. ■

The MINIMAL-SIZE DFTT Problem may have to be solved by an optimal ("good") teacher, who is expected to give only relevant information, and guarantee fast learning. Of course such a teacher may have insight in the construction of $\mathcal{L}$ and then may have knowledge of minimal-size DFTTs in a different manner.

7 Preset Learning

In this section we will discuss the notion of *preset learner*, i.e. a learner that exclusively makes use of DFTTs for each language of the class. This concept is based on the intuition that this is the most simple-minded way of going about identifying a language finitely. Moreover it is very easy to teach to a learner. The notion is most important in the case of finite classes of languages because in the infinite case there seems to be not much other recourse for finite learners anyway. Again the complexity of finding minimal size DFTTs is the main theme of this section.

We begin with a general discussion without restricting $\mathcal{L}$ to being finite. Let us take a finitely identifiable class $\mathcal{L}$, and $L_i \in \mathcal{L}$ and consider the collection S^i of all minimal DFTTs of $L_i \in \mathcal{L}$.

Proposition 2 ε *is an environment for* $L_i \in \mathcal{L}$ *iff for some* $n \in \mathbb{N}$*,* $set(\varepsilon|n)$ *is a superset of some* S^i_j *in* S^i*.*

Proof:

($\Rightarrow$) Since $\mathcal{L}$ is finitely identifiable, at some stage $\varepsilon|n$ in ε for L_i a DFTT S'_i for L_i has to occur. Therefore there has to exist a minimal $S^i_j \subseteq S'_i \subseteq set(\varepsilon|n)$.

($\Leftarrow$) trivial. ■

This means that for every environment of $L_i \in \mathcal{L}$ there is a finite point at which the elements of some minimal definite tell-tale have been enumerated and the learner can safely and economically conclude L_i. Using Proposition 2 we can then define a finite learner who decides on the right language at the first moment that this is possible. In order to define the fastest preset learner we have to consider a learner that can make use of any minimal DFTT. Therefore, for every set the learner has access to the set of all minimal DFFTs. So, strictly speaking it is not a preset learner.

Definition 15 *The fastest preset learner* φ_{fast} *is defined in the following way.*

$$\varphi_{fast}(\varepsilon|n) = \begin{cases} i & \textit{if } \exists S^i_j \in S^i(S^i_j \subseteq set(\varepsilon|n) \text{ \& } S^i_j \nsubseteq set(\varepsilon|n-1)), \\ \uparrow & \textit{otherwise.} \end{cases}$$

The function φ_{fast} will identify the collection $\mathcal{L}$ because an environment for a language L_i will give at a proper time a (minimal) DFTT for that language. On the other hand, if $\varphi_{fast}(\varepsilon|n) = i$, then $set(\varepsilon|k)$ for $k < n$ does not have enough eliminative power to exclude all other hypotheses in $\mathcal{L}$, and hence no other successful learners could conjecture L_i either.

Clearly φ_{fast} is not defined in an effective manner and therefore does not adhere to Definition 2. If we are in the case of a finite set of finite languages the definition is effective. In the next subsection we discuss its complexity.

7.1 Computational Complexity

Let us again go back to the case of a finite class of finite sets. To compute the set S^i of all minimal DFTTs of $L_i \in \mathcal{L}$ we need to perform the procedure for finding a minimal DFTT for all possible orderings of elements in L_i. Therefore the simple procedure described earlier has to be performed $n!$ times. This indicates that finding the set of all minimal DFTTs is quite costly. In fact we show that the problem is NP-hard.

Proposition 3 *Finding S^i of $L_i \in \mathcal{L}$ is NP-hard.*

Proof: It is easy to observe that once we enumerate S^i of L_i, we can find a minimal-size DFTT of L_i in polynomial time, by simply picking one of the smallest sets in S^i. This means that the MINIMAL-SIZE DFTT Problem for L_i can be polynomially reduced to the problem of finding S^i for L_i. ■

The fastest learner does not have to work this way. Since in the finite case languages are their own DFTTs, the Learner can use these to arrive at a conclusion as quickly as possible. Fast finite identification on the basis of DFTTs of finite classes of finite sets has then the same complexity as regular finite identification.

7.2 Recursion-theoretic Complexity

In this section we return to the case of infinite classes of languages. We will present results concerning minimal DFTTs and minimal-size DFTTs. We show that for recursive (preset) learning functions there are definite restrictions with regard to the minimality of the DFTTs they can use.

In the following we will use the manner of speech where we will say for example that e is a Turing machine if we mean that e is an integer that codes a Turing machine, M_e, and that $f(a) = \{b, c\}$ if $f(a)$ codes the finite set containing just b and c.

Let us now recall the Kleene's T-Predicate.

Definition 16 (T-predicate) *$T(e, x, y)$ holds iff e is a Turing machine and y is a computation of e with input x.*

With the use of the T-predicate the Halting Problem can be formulated in the following way:

$$M_e(e) \downarrow \iff \exists y T(e, e, y).$$

Theorem 17 *There exists a preset learner φ with an effective function f that gives a DFTT $f(i) = X_i$ for each L_i, but for which no recursive function g exists such that for each i, $g(i)$ is the set of all minimal DFTTs for L_i.*

Proof: Let us consider the following class of finite sets $\mathcal{L} = \{L_i | i \in \mathcal{L}\}$, such that:

$$L_i = \{2i, 2(\mu y T(i, i, y)) + 1\}.$$

We can easily observe that $|L_i| = 2 \iff M_i(i) \downarrow$. Note that it is decidable for an arbitrary natural number whether it is a member of L_i. If n is even it should be $2i$. If n is odd $\frac{n-1}{2}$ should code a computation of M_i on i. Minimal DFTTs of L_i are $\{2i\}$, and, in case $M_i(i) \downarrow$, $\{2(\mu y T(i, i, y)) + 1\}$. It is clear that the function f such that

$$f(i) = \{\{2i\}, \{2(\mu y T(i, i, y)) + 1\}\}$$

cannot be given by a total recursive function, because its existence would solve Halting Problem. ■

We can improve on this theorem by showing that a minimal-size DFTT can not always be given effectively.

Theorem 18 *There exists preset learner φ with an effective function f that gives a DFTT X_i for each L_i, but for which no recursive function g exists such that for each i, $g(i)$ is a minimal size DFTT for L_i.*

Proof: Let $j : \mathbb{N}^2 \to \mathbb{N}$ be a recursive pairing function (bijection) with inverses j_1 and j_2. We now consider the class $\mathcal{L} = \{L_i | i \in \mathcal{L}\}$ such that:

$$L_i = \{3j_1(i), 3j_2(i) + 1, 3(\mu y T(i, i, y)) + 2\}.$$

Note that a minimal-size DFTT of L_i is $\{3j_1(i), 3j_2(i) + 1\}$ or $\{3(\mu y T(i, i, y)) + 2\}$ if $M_i(i) \downarrow$. Clearly the minimal size DFTTs of L_i cannot be given by a total recursive function g, because the existence of g would solve Halting Problem. ■

The next question that comes to mind is whether it is always possible to effectively obtain a minimal DFTT. Our further work on the subject involves an attempt to answer this question.

8 Conclusions and Future Work

We used a characterization of finite identification of languages from positive data to discuss the complexity of finding an optimal learning and teaching strategy in finite identification. We introduced two notions: minimal DFTT and minimal-size DFTT. By viewing the informativeness of examples as their power to eliminate certain conjectures, we have checked the computational complexity of 'finite teachability' from minimal DFTT and minimal-size DFTT. In the former case the problem turns out to be PTIME computable, while the latter falls into the NP-complete class of problems. This suggests that it is easy to teach in a way that avoids irrelevant information but it is potentially difficult to teach in the most effective way. In the last section we restrict our attention to preset learners and we use our results on minimality to define a non-effective fastest preset learner who has access to all minimal DFTTs in a class and decides on a language as soon as possible. The essential part of this method is to find a set of all minimal DFTTs for a language. We show that this problem is NP-hard. We show some related results in the analogous recursion-theoretic context.

Some links between finite identification and dynamic epistemic logic (see e.g. (van Ditmarsch et al., 2007) for an introduction) have already been established ((Gierasimczuk, 2009), (Dégremont & Gierasimczuk, 2009)). For dynamic epistemic logic the restriction to finite sets of finite languages is very natural, so the analysis of its complexity that we present in this paper can be applied to strengthen this connection.

In the future we want to investigate more complex languages that consist of strings. We will be able to analyze another notion of minimality of relevant information, based not on the amount of data but on its simplicity.

References

Angluin, D. (1980). Inductive inference of formal languages from positive data. *Information and Control*, *45(2)*, 117–135.

Dégremont, C., & Gierasimczuk, N. (2009). Can doxastic agents learn? On the temporal structure of learning. *Proc. 2nd LORI* (pp. 90–104). Springer.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability.* San Francisco: W. H. Freeman and Co.

Gierasimczuk, N. (2009). Bridging learning theory and dynamic epistemic logic. *Synthese*, *169*, 371–384.

Gold, E. (1967). Language identification in the limit. *Information and Control*, *10*, 447–474.

Karp, R. M. (1972). Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher (Eds.), *Complexity of computer computations*, 85–103. Plenum Press.

Mukouchi, Y. (1992). Characterization of finite identification. *AII '92: Proceedings of the International Workshop on Analogical and Inductive Inference* (pp. 260–267). London, UK: Springer-Verlag.

van Ditmarsch, H., van der Hoek, W., & Kooi, B. (2007). *Dynamic epistemic logic.* Springer Netherlands.