Approximative Learning of Regular Languages

Henning Fernau

WSI-2001-2

Henning Fernau

Wilhelm-Schickard-Institut für Informatik
Universität Tübingen
Sand 13
D-72076 Tübingen
Germany

E-Mail: fernau@informatik.uni-tuebingen.de
Telefon: (07071) 29-77565
Telefax: (07071) 29-5061

ISSN 0946-3852

Approximative Learning of Regular Languages

Henning Fernau
Wilhelm-Schickard-Institut für Informatik
Universität Tübingen
Sand 13, D-72076 Tübingen, Germany
fernau@informatik.uni-tuebingen.de

July 24, 2001

Abstract

We show how appropriately chosen functions f which we call *distinguishing* can be used to make deterministic finite automata backward deterministic. These ideas have been exploited to design regular language classes called f-distinguishable which are identifiable in the limit from positive samples. Special cases of this approach are the k-reversible and terminal distinguishable languages as discussed in [1, 4, 6, 21, 22]. Here, we give new characterizations of these language classes. Moreover, we show that all regular languages can be approximated in the setting introduced by Kobayashi and Yokomori [15]. Finally, we prove that the class of all function-distinguishable languages is equal to the class of regular languages.

1 Introduction

Identification in the limit from positive samples, also known as *exact learning from text* as proposed by Gold [11], is one of the oldest yet most important models of grammatical inference. Since not all regular languages can be learned exactly from text, the characterization of *identifiable* subclasses of

regular languages is a useful line of research, because the regular languages are a very basic language family.

One possible idea to overcome this weakness is to use the idea of approximate learning. In this area, various approaches were considered, e.g., metric space approaches [25], allowing absolute or relative errors [2, 23], or lattice-theoretic approaches [14, 15] (based on [26, 17, 18, 19]). We will focus on the last approach in the following.

In [7], we introduced the so-called function-distinguishable languages as a rich source of examples of identifiable language families. Among the language families which turn out to be special cases of our approach are the k-reversible languages [1] and the terminal-distinguishable languages [21, 22], which belong, according to Gregor [13], to the most popular identifiable regular language classes. Moreover, we have shown [7] how to transfer the ideas underlying the well-known identifiable language classes of k-testable languages, k-piecewise testable languages and threshold testable languages to our setting. In a nutshell, an identification algorithm for f-distinguishable languages assigns to every finite set of samples $I_+ \subseteq T^*$ the smallest[1] f-distinguishable language containing I_+ by subsequently merging states which cause conflicts to the definition of f-distinguishable automata, starting with the simple prefix tree automaton accepting I_+.

In this paper, we firstly give a further useful characterization of function-distinguishable languages which has been also employed in other papers [5, 8, 9]. This also allows us to define a possible merging-state inference strategy in a concise manner. Then, we focus on questions of approximability of regular languages by function-distinguishable languages in the setting introduced by Kobayashi and Yokomori [15].

The paper is organized as follows: In Section 2, we provide the necessary background from formal language theory and in sections 3 and 4, we introduce the central concepts of the paper, namely the so-called distinguishing functions and the function distinguishable automata and languages. In Section 5, we discuss an alternative definition of the function canonical automata which we used as compact presentation in other papers. In Section 6, we show how to approximate arbitrary regular languages by using function-distinguishable languages, based on the notion of upper-best approximation in the limit introduced by Kobayashi and Yokomori in [14, 15]. Section 7

[1]This is well-defined, since each class of f-distinguishable languages is closed under intersection, see Theorem 5.

concludes the paper, indicating practical applications of our method and extensions to non-regular language families. Moreover, there we list some (open) complexity problems related to our work.

An extended abstract of this report will appear in the Proceedings of SOFSEM'01.

2 General definitions

Σ^* is the set of words over the alphabet Σ. Σ^k ($\Sigma^{<k}$) collects the words whose lengths are equal to (less than) k. λ denotes the empty word. Pref(L) is the set of prefixes of L and $u^{-1}L = \{ v \in \Sigma^* | uv \in L \}$ is the quotient of $L \subseteq \Sigma^*$ by u.

We assume that the reader knows that regular languages can be characterized by (deterministic) finite automata $A = (Q, T, \delta, q_0, Q_F)$, where Q is the state set, $\delta \subseteq Q \times T \times Q$ is the transition relation, $q_0 \in Q$ is the initial state and $Q_F \subseteq Q$ is the set of final states. As usual, δ^* denotes the extension of the transition relation to arbitrarily long input words. The language defined by an automaton A is written $L(A)$. An automaton is called *stripped* iff all states are accessible from the initial state and all states lead to some final state. Observe that the transition function of a stripped deterministic finite automaton is not total in general.

We denote the minimal deterministic automaton of the regular language L by $A(L)$. Recall that $A(L) = (Q, T, \delta, q_0, Q_F)$ can be described as follows: $Q = \{u^{-1}L | u \in \text{Pref}(L)\}$, $q_0 = \lambda^{-1}L = L$; $Q_F = \{u^{-1}L | u \in L\}$; and $\delta(u^{-1}L, a) = (ua)^{-1}L$ with $u, ua \in \text{Pref}(L)$, $a \in T$. According to our definition, any minimal deterministic automaton is stripped.

Furthermore, we need two automata constructions in the following:
The *product automaton* $A = A_1 \times A_2$ of two automata $A_i = (Q_i, T, \delta_i, q_{0,i}, Q_{F,i})$ for $i = 1, 2$ is defined as $A = (Q, T, \delta, q_0, Q_F)$ with $Q = Q_1 \times Q_2$, $q_0 = (q_{0,1}, q_{0,2})$, $Q_F = Q_{F,1} \times Q_{F,2}$, $((q_1, q_2), a, (q_1', q_2')) \in \delta$ iff $(q_1, a, q_1') \in \delta_1$ and $(q_2, a, q_2') \in \delta_2$.

A *partition* of a set S is a collection of pairwise disjoint nonempty subsets of S whose union is S. If π is a partition of S, then, for any element $s \in S$, there is a unique element of π containing s, which we denote $B(s, \pi)$ and call the *block* of π containing s. A partition π is said to *refine* another partition π' iff every block of π' is a union of blocks of π. If π is any partition of the state set Q of the automaton $A = (Q, T, \delta, q_0, Q_F)$, then the *quotient automaton*

$\pi^{-1}A = (\pi^{-1}Q, T, \delta', B(q_0, \pi), \pi^{-1}Q_F)$ is given by $\pi^{-1}\hat{Q} = \{ B(q, \pi) \mid q \in \hat{Q} \}$ (for $\hat{Q} \subseteq Q$) and $(B_1, a, B_2) \in \delta'$ iff $\exists q_1 \in B_1 \exists q_2 \in B_2 : (q_1, a, q_2) \in \delta$.

3 Distinguishing functions

In order to avoid cumbersome case discussions, let us fix now T as the input alphabet of the finite automata we are going to discuss.

Definition 1 Let F be some finite set. A mapping $f : T^* \to F$ is called a *distinguishing function* if $f(w) = f(z)$ implies $f(wu) = f(zu)$ for all $u, w, z \in T^*$.

In the literature, we can find the terminal function [22]

$$\mathrm{Ter}(x) = \{ a \in T \mid \exists u, v \in T^* : uav = x \}$$

and, more generally, the k-terminal function [6]

$$\begin{aligned} \mathrm{Ter}_k(x) &= (\pi_k(x), \mu_k(x), \sigma_k(x)), \quad \text{where} \\ \mu_k(x) &= \{ a \in T^{k+1} \mid \exists u, v \in T^* : uav = x \} \end{aligned}$$

and $\pi_k(x)$ [$\sigma_k(x)$] is the prefix [suffix] of length k of x if $x \notin T^{<k}$, and $\pi_k(x) = \sigma_k(x) = x$ if $x \in T^{<k}$. The example $f(x) = \sigma_k(x)$ leads to the k-reversible languages, confer [1, 6]. In particular, the trivial distinguishing function, whose range is a singleton set, characterizes the 0-reversible languages. Other examples of distinguishing functions in the context of even linear languages can be found in [5, 21].

Observe that every regular language R induces, via its Nerode equivalence classes, a distinguishing function f_R, where $f_R(w)$ maps w to the equivalence class containing w. Especially, T^* leads to a trivial distinguishing function $f_{T^*} : T^* \to \{q\}$, and the class of f_{T^*}-distinguishable languages coincides with the class of 0-reversible languages [1] over the alphabet T.

In some sense, these are the only distinguishing functions, since one can associate to every distinguishing function f a finite automaton $A_f = (F, T, \delta_f, f(\lambda), F)$ by setting $\delta_f(q, a) = f(wa)$, where $w \in f^{-1}(q)$ can be chosen arbitrarily, since f is a distinguishing function.

4 Function distinguishable languages

Here, we will formally introduce function distinguishable languages and discuss some formal language properties.

Definition 2 Let $A = (Q, T, \delta, q_0, Q_F)$ be a finite automaton. Let $f : T^* \to F$ be a distinguishing function. A is called *f-distinguishable* if:

1. A is deterministic.
2. For all states $q \in Q$ and all $x, y \in T^*$ with $\delta^*(q_0, x) = \delta^*(q_0, y) = q$, we have $f(x) = f(y)$.

   (In other words, for $q \in Q$, $f(q) := f(x)$ for some x with $\delta^*(q_0, x) = q$ is well-defined.)
3. For all $q_1, q_2 \in Q$, $q_1 \neq q_2$, with either (a) $q_1, q_2 \in Q_F$ or (b) there exist $q_3 \in Q$ and $a \in T$ with $\delta(q_1, a) = \delta(q_2, a) = q_3$, we have $f(q_1) \neq f(q_2)$.

A language is called *f-distinguishable* iff it can be accepted by an f-distinguishable automaton. The family of f-distinguishable languages is denoted by f-DL.

We need a suitable notion of a canonical automaton in the following.

Definition 3 Let $f : T^* \to F$ be a distinguishing function and let $L \subseteq T^*$ be a regular set. Let $A(L, f)$ be the stripped subautomaton of the product automaton $A(L) \times A_f$, i.e., delete all states that are not accessible from the initial state or do not lead into a final state of $A(L) \times A_f$. $A(L, f)$ is called *f-canonical automaton* of L.

Observe that the class f-DL formally fixes the alphabet of the languages by the range of f. As we have already seen by the examples for distinguishing functions listed above, f can oftenly defined for *all* alphabets. Taking this generic point of view, for example, Ter-DL is just the class of (reversals of) terminal distinguishable languages [5, 22], where the alphabet is left unspecified.

For example, for each distinguishing function f, the associated automaton A_f is f-distinguishable. This simple observation leads us to:

Theorem 4 *A language is function-distinguishable iff it is regular.*

Proof. Let L be a regular language. Consider the canonical automaton A_L for L. It is quite easy to see that A_L is f_L-distinguishable. □

In other words, $\{ f\text{-DL} \mid f$ is a distinguishing function $\}$ gives a finer classification of all regular languages. This finer classification is necessary, since it is well known that the class of all regular languages is not identifiable in the limit from positive data [11].

The following theorem generalizes the corresponding assertion for k-reversible languages as stated by Angluin [1].

Theorem 5 *For each distinguishing function f, f-DL is closed under intersection.*

Proof. The standard product automaton construction is applicable. □

To the contrary, f-DL is *not* closed under union nor complement in general, see [1]. According to Pin [20], the union closure of the 0-reversible languages is characterized by another class of regular languages which he calls reversible. He calls a language L *reversible* iff there is a finite automaton A accepting L such that A is deterministic and codeterministic but has possibly several initial and several accepting states. Sometimes, such automata are also called injective automata or permutation automata.

5 An alternative presentation

In [7], we developed a generic merging state algorithm for f-DL which paralleled the approach of Angluin for 0-reversible languages. More precisely, the algorithm, when given an input sample I_+, starts with the prefix tree acceptor $PTA(I_+)$ (as defined below). If $A_f(I_+)$ ($L_f(I_+)$, resp.) denotes the output automaton (output language, resp.) of the merging state inference algorithm when given I_+, then (disregarding automaton ismorphism) $A(L_f(I_+), f) = A_f(I_+)$, see [7]. In their works, Radhakrishnan and Nagaraja [22] do not start with the PTA of the given input data set I_+ but rather with a so-called "skeletal grammar" for the given input data set I_+, which corresponds to the "maximal canonical automaton" $MCA(I_+)$ in the framework of Dupont and Miclet [3]. Here, we describe a related algorithm for learning f-DL-languages. This way, we also yield an alternative characterization of f-DL.

Consider an input sample set $I_+ = \{w_1, \ldots, w_M\} \subseteq T^+$.[2] Let $w_i = a_{i1} \ldots a_{in_i}$, where $a_{ij} \in T$, $1 \leq i \leq M$, $1 \leq j \leq n_i$. The *skeletal automaton* for the sample set is defined as

$$\begin{aligned} A_S(I_+) &= (Q_S, T, \delta_S, Q_0, Q_f), \quad \text{where} \\ Q_S &= \{\, q_{ij} \mid 1 \leq i \leq M, 1 \leq j \leq n_i + 1 \,\}, \\ \delta_S &= \{\, (q_{ij}, a_{i,j}, q_{i,j+1}) \mid 1 \leq i \leq M, 1 \leq j \leq n_i \,\}, \\ Q_0 &= \{\, q_{i1} \mid 1 \leq i \leq M \,\} \quad \text{and} \\ Q_f &= \{\, q_{i,n_i+1} \mid 1 \leq i \leq M \,\}. \end{aligned}$$

Observe that we allow a *set* of initial states. The *frontier string* of q_{ij} is defined by $\mathrm{FS}(q_{ij}) = a_{ij} \ldots a_{in_i}$. The *head string* of q_{ij} is defined by the equation $\mathrm{HS}(q_{ij})\mathrm{FS}(q_{ij}) = w_i$, i.e., $\mathrm{HS}(q_{ij}) = a_{i1} \ldots a_{i,j-1}$. In other words, $\mathrm{HS}(q_{ij})$ is the unique string leading from an initial state into q_{ij}, and $\mathrm{FS}(q_{ij})$ is the unique string leading from q_{ij} into a final state.[3] Therefore, the skeletal automaton of a sample set simply spells all words of the sample set in a trivial fashion. Two things can be easily observed.

1. The state partition π of Q_S induced by $q \equiv q'$ iff $\mathrm{HS}(q) = \mathrm{HS}(q')$ yields the prefix tree acceptor, i.e., $PTA(I_+) = \pi^{-1} A_S(I_+)$.

2. Since there is only one word leading to any q, namely $\mathrm{HS}(q)$, $f(q) = f(\mathrm{HS}(q))$ can be uniquely defined.

Now, for $q_{ij}, q_{k\ell} \in Q_S$, define $q_{ij} \rightleftharpoons_f q_{k\ell}$ iff (1) $\mathrm{HS}(q_{ij}) = \mathrm{HS}(q_{k\ell})$ or (2) $\mathrm{FS}(q_{ij}) = \mathrm{FS}(q_{k\ell})$, as well as $f(q_{ij}) = f(q_{k\ell})$.

The following assertion is easily verified:

Lemma 6 *For each distinguishing function f and each finite language I_+, $\rightleftharpoons_f$ is a reflexive symmetric relation on the set Q_S of states of $A_S(I_+)$.*

In general, $\rightleftharpoons_f$ is not an equivalence relation on the state set of A_S, as the following example shows:

[2] The inclusion of the empty word would introduce some unnecessary technalities.

[3] In order to overcome unnecessary technical complications, we underline here that we are dealing with a sample set, i.e., we do not consider repitions of sample words which are allowed in Gold's model in general.

Example 7 Consider the trivial distinguishing function σ_0 and $I_+ = \{a, aa\}$. The skeletal automaton has state transitions (q_{11}, a, q_{12}), (q_{21}, a, q_{22}) and (q_{22}, a, q_{23}). Since $\mathrm{HS}(q_{11}) = \mathrm{HS}(q_{21}) = \lambda$ and $\mathrm{HS}(q_{12}) = \mathrm{HS}(q_{22}) = a$, as well as $\mathrm{FS}(q_{12}) = \mathrm{FS}(q_{23}) = \lambda$, $\mathrm{FS}(q_{11}) = \mathrm{FS}(q_{22}) = a$ and $\mathrm{FS}(q_{21}) = aa$, all states in Q_S are σ_0-equivalent, but $q_{11} \not\rightleftharpoons_{\sigma_0} q_{12}$.

Therefore, we define $\equiv_f := (\rightleftharpoons_f)^+$, denoting in this way the transitive closure of the original relation. The following lemmma is again an easy exercise left to the reader.

Lemma 8 *For each distinguishing function f and each finite language I_+, $\equiv_f$ is an equivalence relation on the state set of $A_S(I_+)$.*

We consider now the automaton $\pi_f^{-1}A_S(I_+)$, where π_f is the partition induced by the equivalence relation $\equiv_f$. We like to show that $A_f(I_+) = \pi_f^{-1}A_S(I_+)$. As a preparatory stage, we prove:

Lemma 9 *For each distinguishing function f and each finite language I_+, $\pi_f^{-1}A_S(I_+)$ is an f-distinguishable automaton.*

Proof. We have to verify the three conditions posed upon f-distinguishable automata for $\pi_f^{-1}A_S(I_+)$. Let δ denote the transition relation of $\pi_f^{-1}A_S(I_+)$ and $\bar{q}_0$ its initial state. (We use barred state notations for states of $\pi_f^{-1}A_S(I_+)$ and non-barred notations for states of $A_S(I_+)$.)
ad 1.: Consider an input word w with $q_1, q_2 \in \delta^*(\bar{q}_0, w)$. Then, there are some $q_{ij} \in \bar{q}_1$ and $q_{k\ell} \in \bar{q}_2$ (recall that $\bar{q}_1, \bar{q}_2$ are both sets of states of $A_S(I_+)$) with $\mathrm{HS}(q_{ij}) = w$ and $\mathrm{HS}(q_{k\ell}) = w$. Hence, $q_{ij} \rightleftharpoons_f q_{k\ell}$, which means that $\bar{q}_1 = \bar{q}_2$, since $\bar{q}_1$ and $\bar{q}_2$ are equivalence classes of states of $A_S(I_+)$.
ad 2.: Observe that $f(q)$ is well-defined for every state q of $A_S(I_+)$. It is easy to check that if $q \rightleftharpoons_f q'$, then $f(q) = f(q')$. Since $q, q' \in \bar{q}$ iff $q \equiv_f q'$ iff $q \rightleftharpoons_f^+ q'$, $f(q) = f(q')$ immediately follows by the transitivity of equality.
ad 3.: It can be shown similar to point 1 (formally by induction). □

Theorem 10 *For each distinguishing function f and each sample set I_+, $A_f(I_+) = \pi_f^{-1}A_S(I_+)$ (up to isomorphism).*

Proof. According to [3], we can consider $\pi_f^{-1}A_S(I_+)$ as being obtained by a sequence of merging state steps, merging only two states at a time. Without loss of generality, such a sequence of mergings might start with "repairing" violations of the determinism requirement, so that we obtain $PTA(I_+)$ as an

intermediate automaton. Similar to the reasoning in the previous lemma, the reader may verify that each of these merging steps can be justified also by the existence of conflicts in the merged states according to inference algorithm sketched in the introduction. Since we have shown the correctness of that inference algorithm in [7], the assertion of this theorem follows, as well. □

This argument justifies the presentation of certain subcases of function distinguishable languages as done in [5, 8].

6 Approximation

Kobayashi and Yokomori introduced in [14, 15] the notion of upper-best approximation in the limit of a target language with respect to the hypothesis space. They showed that regular languages can be upper-best approximated by k-reversible languages for any fixed k. Here, we shall prove that similar results are true for any class f-DL. In particular, this implies that, given any enumeration of an arbitrary regular language to some identification algorithm for f-DL, this algorithm will converge, yielding some well-defined result. Especially, the terminal distinguishable languages can be used to approximate all regular languages in a precise sense. This is interesting, since already Radhakrishnan and Nagaraja observed in [22] on an empirical basis that their algorithm converges for regular languages, but not for context-free languages. The approximation notion developed by Kobayashi and Yokomori gives a mathematical explanation of this empirical observation.

Firstly, we give the necessary definitions due to Kobayashi and Yokomori.

Let $\mathcal{L}$ be a language class and L be a language possibly outside $\mathcal{L}$. An *upper-best approximation* $\bar{\mathcal{L}}L$ *of* L *with respect to* $\mathcal{L}$ is defined to be a language L_* containing L such that for any $L' \in \mathcal{L}$ with $L \subseteq L'$, $L_* \subseteq L'$ holds. If such an L_* does not exist, $\bar{\mathcal{L}}L$ is undefined.

Remark 1 If $\mathcal{L}$ is closed under intersection, then L_* is uniquely defined.

Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two language classes. We say that $\mathcal{L}_1$ has the *upper-best approximation property (u.b.a.p.) with respect to* $\mathcal{L}_2$ iff, for every $L \in \mathcal{L}_2$, $\bar{\mathcal{L}}_1 L$ is defined.

Consider an inference machine I to which as input an arbitrary language $L \in \mathcal{L}$ may be enumerated (possibly with repetitions) in an arbitrary order, i.e., I receives an infinite input stream of words $E(1)$, $E(2)$, ..., where

$E : \mathbb{N} \to L$ is an enumeration of L. We say that I *identifies an upper-best approximation of* L *in the limit (from positive data) by* $\mathcal{L}$ if I reacts on an enumeration of L with an output device stream $D_i \in \mathcal{D}$ such that there is an $N(E)$ so that, for all $n \geq N(E)$, we have $D_n = D_{N(E)}$ and, moreover, the language defined by $D_{N(E)}$ equals $\bar{\mathcal{L}}L$. A language class $\mathcal{L}_1$ is called *upper-best approximately identifiable in the limit (from positive data) by* $\mathcal{L}_2$ iff there exists an inference machine I which identifies an upper-best approximation of each $L \in \mathcal{L}_1$ in the limit (from positive data) by $\mathcal{L}_2$. Observe that this notion of identifiability coincides with Gold's classical notion of learning in the limit in the case when $\mathcal{L}_1 = \mathcal{L}_2$.

Consider a language class $\mathcal{L}$ and a language L from it. A finite subset $F \subseteq L$ is called a *characteristic sample* of L with respect to $\mathcal{L}$ iff, for any $L' \in \mathcal{L}$, $F \subseteq L'$ implies that $L \subseteq L'$.

Now, fix some distinguishing function f. We call a language $L \subseteq T^*$ *pseudo-f-distinguishable* iff, for all $u_1, u_2, v \in T^*$ with $f(u_1) = f(u_2)$, we have $u_1^{-1}L = u_2^{-1}L$ whenever $\{u_1 v, u_2 v\} \subseteq L$. By the characterization theorem derived in [7], $L \in f$-DL iff L is pseudo-f-distinguishable and regular.

Immediately from the definition, we may conclude:

Proposition 11 *Let $L_1 \subseteq L_2 \subseteq \ldots$ be any ascending sequence of pseudo-f-distinguishable languages. Then, $\bigcup_{i \geq 1} L_i$ is pseudo-f-distinguishable.* □

For brevity, we write $u_1 \equiv_{L,f} u_2$ iff $u_1^{-1}L = u_2^{-1}L$ and $f(u_1) = f(u_2)$.

Remark 2 If $L \subseteq T^*$ is a regular language and if $f : T^* \to F$ is some distinguishing function, then the number of equivalence classes of $\equiv_{L,f}$ equals the number of states of A_L (plus one) times $|F|$, and this is just the number of states of $A(L, f)$ (plus $|F|$).

Let $L \subseteq T^*$ be some language. For any integer i, we will define $R_f(i, L)$ as follows:

1. $R_f(0, L) = L$ and
2. $R_f(i, L) = R_f(i-1, L) \cup \{\, u_2 w \mid u_1 v, u_2 v, u_1 w \in R_f(i-1, L) \wedge f(u_1) = f(u_2) \,\}$ for $i \geq 1$.

Furthermore, set $R_f(L) = \bigcup_{i \geq 0} R_f(i, L)$.

Observe that, by definition, a language is pseudo-k-reversible [15] iff it is pseudo-σ_k-distinguishable. Moreover, the operator R_k introduced in [15] is written as R_{σ_k} in our notation.

Since R_f turns out to be a hull operator, the following statement is obvious.

Proposition 12 *For any language L and any distinguishing function f, $R_f(L)$ is the smallest pseudo-f-distinguishable language containing L.* □

Lemma 13 *Let $L \subseteq T^*$ be any language. If u_1 and u_2 are prefixes of L, then $u_1 \equiv_{L,f} u_2$ implies that $u_1^{-1}R_f(L) = u_2^{-1}R_f(L)$.*

Proof. Let u_1 and u_2 be prefixes of L with $u_1 \equiv_{L,f} u_2$. By definition of $\equiv_{L,f}$, $u_1^{-1}L = u_2^{-1}L \neq \emptyset$. Hence, there is a string v so that $\{u_1v, u_2v\} \subseteq L \subseteq R_f(L)$. Furthermore, by definition of $\equiv_{L,f}$, $f(u_1) = f(u_2)$. Since $R_f(L)$ is pseudo-f-distinguishable due to Proposition 12, $u_1^{-1}R_f(L) = u_2^{-1}R_f(L)$. □

Lemma 14 *Let $L \subseteq T^*$ be any language and let f be any distinguishing function. Then, for any prefix w_1 of $R_f(L)$, there exists a prefix w_2 of L with $w_1^{-1}R_f(L) = w_2^{-1}R_f(L)$.*

Proof. Since w_1 is a prefix of $R_f(L)$ iff w_1 is a prefix of $R_f(i, L)$ for some $i \geq 0$, it suffices to show the following claim by induction:

> Let $i \geq 0$. Then, for any prefix w_1 of $R_f(i, L)$, there exists a prefix w_2 of L with $w_1^{-1}R_f(L) = w_2^{-1}R_f(L)$.

Trivially, the claim is true when $i = 0$, since $R_f(0, L) = L$.

As induction hypothesis, assume that the claim is shown for $i = \ell$. Hence, we have to consider some $w_1 \in \mathrm{Pref}(R_f(\ell + 1, L)) \setminus \mathrm{Pref}(R_f(\ell, L))$ in the induction step. Consider some $w_1z \in R_f(\ell + 1, L) \setminus R_f(\ell, L)$. This means that there are strings $u_1, v, w \in T^*$ with $\{u_1v, u_2v, u_1w\} \subseteq R_f(\ell, L)$, $f(u_1) = f(u_2)$ and $u_2w = w_1z$. If $|u_2| \geq |w_1|$, w_1 is a prefix of $u_2w \in R_f(\ell, L)$ in contrast to our assumption. Therefore, we have $w_1 = u_2v'$ for some $v' \in T^+$. Since $R_f(L)$ is pseudeo-f-distinguishable and $\{u_1v, u_2v\} \subseteq R_f(L)$ as well as $f(u_1) = f(u_2)$, $u_1^{-1}R_f(L) = u_2^{-1}R_f(L)$, which yields $w_1^{-1}R_f(L) = (u_2v')^{-1}R_f(L) = (u_1v')^{-1}R_f(L)$. Since v' is a prefix of w, u_1v' is a prefix of $u_1w \in R_f(\ell, L)$. By induction hypothesis, there is a prefix w_2 of L such that $w_2^{-1}R_f(L) = (u_1v')^{-1}R_f(L) = w_1^{-1}R_f(L)$. □

By a reasoning completely analogous to [15], we may conclude:

Theorem 15 *For any distinguishing function f, the class f-DL has the u.b.a.p. with respect to the class of regular languages.* □

Observe that the number of states of $A_{R_f(L)}$ is closely related to the number of states of $A(L, f)$, see Remark 2.

Theorem 16 *For any distinguishing function f, the class of regular languages is upper-best approximately identifiable in the limit from positive data by f-DL.* □

In the spirit of [16, Cor. 2], it is possible to obtain other, new identifiable classes of regular languages as homomorphic images of an arbitrary class f-DL (for each fixed distinguishing function f).

7 Discussion

We have proposed a large collection of families of languages, each of which is identifiable in the limit from positive samples, hence extending previous works. We feel that deterministic methods yielding characterizable regular subclasses (such as the ones proposed in this paper) are quite important for practical applications, since they could be understood more precisely than mere heuristics, so that one can prove certain properties about the algorithms. Moreover, the approach of this paper allows one to make the bias (which each regular language identification algorithm necessarily has) explicit and transparent to the user: The bias consists in (1) the restriction to regular languages and (2) the choice of a particular distinguishing function f. Detailed comments in this direction can be found in [9].

We will provide a publicly accessible prototype learning algorithm for (each of the families) f-DL in the near future. A user can then firstly look for an appropriate f by making learning experiments with typical languages he expects to be representative for the languages in his particular application. If there are only few "typical languages" $L_1, \ldots, L_r$ in the beginning, one could also start with $f_{L_1} \times \cdots \times f_{L_r}$, where $f \times g$ is defined as $(f \times g)(x) = (f(x), g(x))$, see the proof of Theorem 4. After this "bias training phase", the user may then use the such-chosen learning algorithm (or better, an improved implementation for the specific choice of f) for his actual application.

Even if the particular class f-DL chosen by the user does not completely comprise all languages the identification machine IM will be confronted with,

Theorem 16 suggests that, in the case that a regular language which does not lie in f-DL is enumerated to IM, some reasonable outcome will be produced in a reasonable time.

If the application suggests that the languages which are to be inferred are non-regular, methods such as those suggested in [21] can be transferred. This is most easily done by using the concept of *control languages* as undertaken in [4, 5] or [24, Section 4] or by using the related concept of *permutations*, see [10].

We conclude this report with posing several complexity questions that naturally arise when being faced with the problem of choosing an appropriate bias for the learning algorithm. Let us assume that the user knows several "typical" languages $L_1, \ldots, L_r$. Possibly, the choice of $f_{L_1} \times \cdots \times f_{L_r}$ as distinguishing function has a range which is too large for practical implementation. Recall that the identification algorithm proposed in [7] exponentially depends on the size of the range of the distinguishing function. Therefore, the following problem is of interest:
Problem 1(r): Given $L_1, \ldots, L_r$, find a distinguishing function f with minimal range such that $L_1, \ldots, L_r$ lie all within f-DL. Although we expect this problem to be NP-hard, we have yet no proof. We even suspect the problem is hard in the special case when $r = 1$.

What could we do if a user cannot tell a good representative set of languages $L_1, \ldots, L_r$ in advance, i.e., with complete automaton specification, or what if the representative languages do not come as a whole but one by one in an on-line fashion? Then, the following incremental version of a "training phase" (which can also be incorporated into the actual "learning phase") for finding the suitable distinguishing function might be an alternative way of getting a good distinguishing function:

1. Start with $f = \sigma_0$, i.e., the trivial distinguishing function.

2. LOOP i: Enumerate $L_i = \{w_{i1}, w_{i2}, \ldots\}$; let $I_{ij} = \{w_{i1}, \ldots, w_{ij}\}$.

3. LOOP j: Consider I_{ij}, $j = 1, \ldots$
   Let L_{ij} be the smallest f-distinguishable language containing I_{ij}.
   IF $L_{ij} = L_i$ THEN continue with LOOP i
   IF $L_{ij} \subseteq L_i$ THEN continue with LOOP j
   IF $L_{ij} = L_i$ THEN modify(f) and continue with LOOP i

We still have to specify the function modify(f). What is the current situation

when calling that function? We know that

$$R_f(L_i) \supseteq L_{ij} \supsetneq L_i.$$

Therefore, $L_i \notin f$-DL. A possible modification would be to put $f := f \times f_{L_{ij}}$. In order to get distinguishable functions with small range, an alternative would be to look for the f_{ij} with smallest range such that $L_{ij} \in f \times f_{ij}$-DL. We also suspect that this optimization problem (Problem 2) is NP-hard. Observe that in the case $f = \sigma_0$ Problem 2 coincides with Problem 1(1).

A related problem is the following: Given $L \subsetneq L'$, find a minimal deterministic finite automaton A such that $L = L' \cap L(A)$ (intersection problem). Observe that this question is known to be NP-hard due to Gold [12]. An answer to the latter question could possibly be helpful for solving Problem 2: Take $L = L_i$ and $L' = R_f(L_i)$. Then, a solution A to the intersection problems defines a distinguishing function f_A such that $f := f \times f_A$ hopefully yields a good solution to Problem 2.

Observe that all the above problems basically lead to some sort of modification of the hypothesis space. More precisely, the hypothesis space used up to a certain point may be refuted, i.e., extended in our case, if it has proven to be unsufficient. Therefore, it may be interesting to investigate possible connections to learning models which incorporate refutations, see [18, 19].

Acknowledgments: We gratefully acknowledge discussions with S. Kobayashi and K. Reinhardt.

References

[1] D. Angluin. Inference of reversible languages. *Journal of the Association for Computing Machinery*, 29(3):741–765, 1982.

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

[3] P. Dupont and L. Miclet. Inférence grammaticale régulière: fondements théoriques et principaux algorithmes. Technical Report RR-3449, INRIA, 1998.

[4] H. Fernau. Learning of terminal distinguishable languages. Technical Report WSI–99–23, Universität Tübingen (Germany), Wilhelm-Schickard-Institut für Informatik, 1999. Short

version published in the proceedings of AMAI 2000, see `http://rutcor.rutgers.edu/~amai/AcceptedCont.htm`.

[5] H. Fernau. Identifying terminal distinguishable languages. Submitted revised version of [4].

[6] H. Fernau. k-gram extensions of terminal distinguishable languages. In *Proc. 15th International Conference on Pattern Recognition*. 2nd Volume, pp. 125–128, IEEE Press, 2000.

[7] H. Fernau. Identification of function distinguishable languages. In *Proc. 11th International Conference Algorithmic Learning Theory (ALT)*, volume 1968 of *LNCS/LNAI*, pages 116–130. Springer, 2000.

[8] H. Fernau. Parallel communicating grammar systems with terminal transmission. *Acta Informatica*, 37:511–540, 2001.

[9] H. Fernau. Learning XML Grammars. Technical Report No. WSI–2001–1, Universität Tübingen (Germany), Wilhelm-Schickard-Institut für Informatik, 2001. Revised version to appear in: *Proceedings of Machine Learning and Data Mining MLDM'01*, volume 2123 of *LNCS/LNAI*. Springer, 2001.

[10] H. Fernau and J. M. Sempere. Permutations and control sets for learning non-regular language families. In *Proc. 5th International Colloquium on Grammatical Inference (ICGI): Algorithms and Applications*, volume 1891 of *LNCS/LNAI*, pages 75–88. Springer, 2000.

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

[12] E. M. Gold. Complexity of automaton identification from given data. *Information and Control* 37:302–320, 1978.

[13] J. Gregor. Data-driven inductive inference of finite-state automata. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(1):305–322, 1994.

[14] S. Kobayashi and T. Yokomori. On approximately identifying concept classes in the limit. In *Proc. 6th International Conference Algorithmic Learning Theory (ALT)*, volume 997 of *LNCS/LNAI*, pages 298–312. Springer, 1995.

[15] S. Kobayashi and T. Yokomori. Learning approximately regular languages with reversible languages. *Theoretical Computer Science*, 174:251–257, 1997.

[16] S. Kobayashi and T. Yokomori. Identifiability of subspaces and homomorphic images of zero-reversible languages. In *Proc. 8th International Conference Algorithmic Learning Theory (ALT)*, volume 1316 of *LNCS/LNAI*, pages 48–61. Springer, 1997.

[17] T. Motoki, T. Shinohara and K. Wright. The correct definition of finite elasticity: Corrigendum to identification of unions. In *COLT'91*, page 375. Morgan Kaufmann, 1991.

[18] Y. Mukouchi. Inductive inference of an approximate concept from positive data. In *Proc. Algorithmic Learning Theory ALT'94*, volume 872 of *LNCS/LNAI*, pages 484–499. Springer, 1994.

[19] Y. Mukouchi and S. Arikawa. Inductive inference machines that can refute hypothesis spaces. In *Proc. 4th Workshop on Algorithmic Learning Theory ALT'93*, volume 744 of *LNCS/LNAI*, pages 123–137. Springer, 1993.

[20] J. E. Pin. On the languages accepted by finite reversible automata. In *14th ICALP'87*, volume 267 of *LNCS*, pages 237–249, 1987.

[21] V. Radhakrishnan. *Grammatical Inference from Positive Data: An Effective Integrated Approach.* PhD thesis, Department of Computer Science and Engineering, Indian Institute of Technology, Bombay (India), 1987.

[22] V. Radhakrishnan and G. Nagaraja. Inference of regular grammars via skeletons. *IEEE Transactions on Systems, Man and Cybernetics*, 17(6):982–992, 1987.

[23] J. S. Royer. Inductive inference of approximations. *Information and Control*, 70:156–178, 1986.

[24] Y. Takada. A hierarchy of language families learnable by regular language learning. *Information and Computation*, 123:138–145, 1995.

[25] R. M. Wharton. Approximate language identification. *Information and Control*, 26:236–255, 1974.

[26] K. Wright. Identification of unions and languages drawn from an identifiable class. In *COLT'89*, pages 328–333. Morgan Kaufmann, 1989.