What is the search space of the regular inference ?*

P. Dupont [(1)], L. Miclet [(1,2)] and E. Vidal [(3)]

(1) France Télécom - CNET/LAA/TSS/RCP

2, route de Trégastel 22301 Lannion Cedex (France)

(2) Ecole Nationale Supérieure de Sciences Appliquées et Technologie - IRISA

6, rue de Kérampont 22305 Lannion Cedex (France)

(3) DSIC - Universidad Politécnica de Valencia

Camino de Vera s/n, 46071 Valencia (Spain)

E-mail: dupont@lannion.cnet.fr

Abstract

This paper revisits the theory of regular inference, in particular by extending the definition of *structural completeness* of a positive sample and by demonstrating two basic theorems. This framework enables to state the regular inference problem as a search through a boolean lattice built from the positive sample. Several properties of the search space are studied and generalization criteria are discussed. In this framework, the concept of *border set* is introduced, that is the set of the most general solutions excluding a negative sample. Finally, the complexity of regular language identification from both a theoritical and a practical point of view is discussed.

1 Introduction

Regular inference is the process of learning a regular language from a set of examples, consisting of a positive sample, i.e. a finite subset of a regular language. A negative sample, i.e. a finite set of strings not belonging to this language, may also be available.

This problem has been studied as early as the growth of the theory of formal grammars. It has an obvious theoritical interest and also an important range of applications, in particular in the fields of *Identification of Sequential Processes*, *Pattern Recognition*, *Speech* and *Natural Language Processing*. The theoretical complexity of this problem is now well established [17] and many empirical algorithms have been also devised, most of them using only a positive sample [6]. However, regular inference as a "generalization as search" problem has not yet been stated in a comprehensive manner. This is the point of view that we develop throughout this paper.

*Published in ICGI'94, Lectures Notes in Computer Science, No. 862, Grammatical Inference and Applications, Springer-Verlag, pp. 25-37, 1994.

We recall in section 2 the basic algebraic notions which will be necessary to define the space of possible solutions to the inference problem: regular languages, finite automata, partitions of a finite set, derivation of an automaton with respect to a partition of its state set, lattices of automata. We revisit the notion of *structural completeness* of a sample with respect to an automaton by extending its definition.

Section 3 is devoted to the detailed description of the search space of the inference problem. We review here the basic theorems which characterize this space and demonstrate them consistently with our definition of structural completeness. It follows that the search space is a boolean lattice of automata built from a positive sample. We state some properties of this lattice. In particular, we show that building the lattice from the *maximal canonical automaton* of a positive sample is more general than from the *prefix tree acceptor*.

We present in section 4 some generalization criteria which may be used to guide the search. In particular, if a negative sample is available, the inference problem may be viewed as the *minimal DFA consistency problem* (section 4.1). Along the same lines, we define in section 4.2 the concept of *border set*, that is the set of the most general solutions consistent with some given positive and negative samples, and we study some properties of this set.

Finally, we recall in section 5 the theoritical complexity results of regular language identification and point out that, while the exact identification of a regular language from a positive and a negative sample is a probably intractable problem, an approximate identification, in some sense, can be achieved in polynomial time.

2 Definitions and notations

The reader familiar with the classical definitions of automata theory may omit section 2.1. We introduce these notions here for the sake of completeness.

2.1 Languages, automata and partitions

2.1.1 Basic definitions

Let Σ denote a finite alphabet, let u, v, w denote elements of Σ^*, i.e. strings over Σ, and let λ denote the empty string. Let $|u|$ denote the length of the string u. We say that u is a *prefix* of v if there exists w such that $uw = v$. We say that u is a *suffix* of v if there exists w such that $wu = v$. A language L is any subset of Σ^*. Let $Pr(L) = \{u | \exists v, uv \in L\}$ denote the set of prefixes of L. Let $L/u = \{v | uv \in L\}$ denote the *left-quotient* of L by u. We have $L/u \neq \phi$ iff $u \in Pr(L)$.

2.1.2 Finite automata

A *finite automaton* is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ where Q is a finite set of *states*, Σ is an *alphabet*, δ is a *transition function*, i.e. a mapping from QxΣ to 2^Q, q_0 is the *initial state* and F is a subset of Q identifying the *final* or *accepting states*. If for any q in Q and any a in Σ, $\delta(q, a)$ has at most one member, respectively exactly one member, the automaton

A is said to be *deterministic*, respectively *complete*. In the sequel, we shall denote DFA, respectively NFA, a deterministic, respectively a non-deterministic, finite automaton.

2.1.3 Language accepted by a finite automaton

An *acceptance* of a string $u = a_1 \dots a_l$ by a (possibly non-deterministic) automaton A defines a (possibly non-unique) sequence of $l + 1$ states $(q^0, \dots, q^l)$ such that $q^0 = q_0$, $q^l \in F$ and $q^{i+1} \in \delta(q^i, a_{i+1})$, for $0 \leq i \leq l - 1$. These $l + 1$ states are said to be *reached* for this acceptance and the state q^l is said to be *used as accepting state*. Similarly, the l transitions (i.e. elements of δ) are said to be *exercized* by this acceptance.

An automaton A is said to be *ambiguous* if there exists at least one string u for which there are several acceptances. In that case, the automaton A is necessarily non-deterministic. A set of transitions being exercized by a set of strings S_s is the union of the sets of transitions exercized by some acceptance of each string in S_s. The corresponding states are said to be *reached* for some acceptance of S_s.

The language $L(A)$ accepted by a finite automaton A is the set of strings accepted by A. A language L is accepted by a finite automaton A if and only if it is a regular set, that is it may be defined by a regular expression [1]. In the sequel, an automaton will denote a *finite* automaton and a language will denote a *regular* language.

A state q of an automaton A is said to be *useful* if there exists a string u in $L(A)$ for which the state q may be reached for some acceptance of u. Otherwise, the state is said to be *useless*. An automaton that contains no useless state is said to be *stripped*.

Let $A(L) = (Q, \Sigma, \delta, q_0, F)$ denote a DFA which has the minimal number of states for a given language L. $A(L)$ is also called the *canonical automaton* of L and may be defined as follows:

$Q = \{L/u \mid u \in Pr(L)\}$,
$q_0 = L/\lambda$,
$F = \{L/u \mid u \in L\}$,
$\delta(L/u, a) = L/ua$ where $u, ua \in Pr(L)$.

$A(L)$ is unique up to a renumbering of its states, that is, every deterministic automaton which exactly accepts L and contains the minimal number of states for this language, is *isomorphic* to $A(L)$ [1].

2.1.4 Derived automata

For any set S, a partition π is a set of pairwise disjoint nonempty subsets of S whose union is S. Let s denote an element of S and let $B(s, \pi)$ denote the unique element, or *block*, of π containing s. We say that π_i *refines*, or *is finer than*, π_j if and only if every block of π_j is a union of one or several blocks of π_i.

If $A = (Q, \Sigma, \delta, q_0, F)$ is an automaton, the automaton $A/\pi = (Q', \Sigma, \delta', B(q_0, \pi), F')$ *derived from A with respect to the partition π* of Q, also called the *quotient automaton* A/π, is defined as follows:

$Q' = Q/\pi = \{B(q, \pi) | q \in Q\}$, $F' = \{B \in Q' | B \cap F \neq \phi\}$,
$\delta' : Q' \text{x} \Sigma \rightarrow 2^{Q'} : \forall B, B' \in Q', \forall a \in \Sigma, B' \in \delta'(B, a) \text{iff} \exists q, q' \in Q, q \in B, q' \in B'$ and $q' \in \delta(q, a)$.

The states of Q belonging to the same block B of the partition π are said to be *merged* together.

2.2 Language samples and associated automata

Let I_+ denote a finite subset, called *positive sample*, of any language L. Let I_- denote a finite subset, called *negative sample*, of the complementary language $\Sigma^* - L$. Consequently, I_+ and I_- are two disjoint finite subsets of Σ^*.

2.2.1 Structural completeness

A sample I_+ is said to be *structurally complete* with respect to an automaton A accepting L, if there exists an acceptance $\mathcal{AC}(I_+, A)$ of I_+ such that:
(1) every transition of A is exercized.
(2) every element of F (the final state set of A) is used as accepting state.
Remark that the classical definitions [4, 12, 13, 3] of structural completeness did not include the second condition. We shall see in section 3.1 that this second condition is also necessary.

2.2.2 Maximal Canonical Automaton, Prefix Tree Acceptor and Universal Automaton

Let $I_+ = \{u_1, \ldots, u_M\}$ be a positive sample, where $u_i = a_{i,1} \ldots a_{i,|u_i|}$, $1 \leq i \leq M = |I_+|$.

Let $MCA(I_+) = (Q, \Sigma, \delta, q_0, F)$ denote the *maximal canonical automaton with respect to* I_+ [12]. It is constructed as follows:
Σ is the alphabet on which I_+ is defined.
$Q = \{v_{i,j} | 1 \leq i \leq M,\ 1 \leq j \leq |u_i|,\ v_{i,j} = a_{i,1} \ldots a_{i,j}\} \cup \{\lambda\}$, $q_0 = \lambda$, $F = I_+$,
$\delta(\lambda, a) = \{v | v = a_{i,1}, a = a_{i,1}, 1 \leq i \leq M\}$,
$\delta(v_{i,j}, a) = \{v_{i,j+1} | v_{i,j+1} = v_{i,j} a_{i,j+1} \text{ and } a = a_{i,j+1}\}$, $1 \leq i \leq M$, $1 \leq j \leq |u_i| - 1$.
Consequently, $L(MCA(I_+)) = I_+$ and $MCA(I_+)$ is the largest stripped automaton, i.e. the automaton having the largest number of useful states, with respect to which I_+ is structurally complete. Note that $MCA(I_+)$ is generally non-deterministic.

Let $PTA(I_+)$ denote the *prefix tree acceptor* of I_+ [3]. It is the quotient automaton $MCA(I_+)/\pi_{I_+}$ where the partition π_{I_+} is defined as follows:
$B(q, \pi_{I_+}) = B(q', \pi_{I_+})$ iff $Pr(q) = Pr(q')$.
That is, the $PTA(I_+)$ is obtained from the $MCA(I_+)$ by merging states sharing the same prefixes. Note that $PTA(I_+)$ is deterministic.

Let UA denote the *universal automaton.* It accepts all the strings defined over the alphabet Σ, i.e. $L(UA) = \Sigma^*$, and it is the smallest automaton with respect to which every sample of Σ^* is structurally complete.

2.3 Lattices of automata

Let $P(A)$ denote the set of partitions of the state set of an automaton A. Let $r(\pi_i)$, or simply r_i, denote the number of blocks of the partition π_i. Let $\pi_1 = \{B_{11}, \ldots, B_{1r_1}\}$ and π_2 be two partitions of $P(A)$. We say that π_2 *directly derives from* π_1 if the partition π_2 is constructed from π_1 as follows: $\pi_2 = \{B_{1j} \cup B_{1k}\} \cup (\pi_1 \setminus \{B_{1j}, B_{1k}\})$, for some j, k between 1 and r_1, $j \neq k$. Consequently, $r_2 = r_1 - 1$.

This derivation operation defines a partial order relation on $P(A)$, which we shall denote $\preceq$. In particular, we have $\pi_1 \preceq \pi_2$. Let $\ll$ denote its transitive closure. In other words, $\pi_i \ll \pi_j$ if and only if π_i is finer than π_j. By extension, we say that A/π_i is *finer than* A/π_j and that A/π_j *derives from* A/π_i. By construction of a quotient automaton, we have the language inclusion property [4] which may be reformulated as follows:

$$A/\pi_i \ll A/\pi_j \text{ if } L(A/\pi_i) \subseteq L(A/\pi_j).$$

The set of automata partially ordered by the relation $\preceq$ is a boolean lattice, that we shall denote $Lat(A)$, of which A and UA (i.e. the universal automaton) are respectively the null and universal elements.

The *depth* of an automaton A/π in $Lat(A)$ is given by $N - r(\pi)$, where N is the number of states of A. Consequently, the depth of the automaton A in $Lat(A)$ is equal to 0 while the depth of the universal automaton UA is equal to $N - 1$.

3 The search space of the regular inference problem

Regular inference may be defined as the discovery of an unknown automaton A from which an observed positive sample I_+ is supposed to have been generated. Given the additional hypothesis of structural completeness of I_+, this problem may be considered as a search through a boolean lattice built from the positive information. This property follows from two (partially demonstrated) theorems [12, 3], that we revisit in section 3.1.

3.1 Basic theorems

Theorem 1 *Let I_+ be a positive sample of any regular language L and let A be any automaton accepting exactly L. If I_+ is structurally complete with respect to A then A belongs to $Lat(MCA(I_+))$.*

Proof.
We shall construct the $MCA(I_+)$ from an acceptance of I_+ by A. This construction defines a partition π such that A is isomorphic to $MCA(I_+)/\pi$.
Let $I_+ = \{u_1, \ldots, u_M\}$, where $u_i = a_{i,1} \ldots a_{i,|u_i|}$, $1 \leq i \leq M$, be a positive sample which is structurally complete with respect to A.
The acceptance $\mathcal{AC}(I_+, A)$ defines for each string u_i a sequence $(q_A^{i,0}, \ldots, q_A^{i,|u_i|})$ of $|u_i| + 1$ states, where $q_A^{i,0} = q_{0_A}$, $q_A^{i,|u_i|} \in F_A$ and $q_A^{i,j+1} \in \delta_A(q_A^{i,j}, a_{i,j+1})$, $1 \leq i \leq M, 0 \leq j \leq |u_i| - 1$.
We start the construction of the MCA by considering its initial state, $q_{0_{MCA}}$ and let define $q_{MCA}^{i,0} = q_{0_{MCA}}$, $1 \leq i \leq M$. Each time a transition of A is exercized, we add a new state in Q_{MCA} and we adapt the transition function δ_{MCA} as follows:

$$q_A^{i,j+1} \in \delta_A(q_A^{i,j}, a_{i,j+1}) \Leftrightarrow q_{MCA}^{i,j+1} \in \delta_{MCA}(q_{MCA}^{i,j}, a_{i,j+1}), 1 \leq i \leq M, 0 \leq j \leq |u_i| - 1.$$

Moreover, we construct the final state set F_{MCA} as follows:

$$F_{MCA} = \{q_{MCA}^{i,|u_i|}, 1 \leq i \leq M\}.$$

We also define a function φ which maps the states of MCA over those of A:

$$\varphi : \; Q_{MCA} \rightarrow Q_A \; : \varphi(q_{MCA}^{i,j}) = q_A, \text{ whenever } q_A = q_A^{i,j}, \; 1 \leq i \leq M, 0 \leq j \leq |u_i|.$$

Finally, let the partition π be defined as follows:

$$B(q^l_{MCA}, \pi) = B'(q^k_{MCA}, \pi) \text{ iff } \varphi(q^l_{MCA}) = \varphi(q^k_{MCA}).$$

The very definition of the partition π implies that A is isomorphic to MCA/π, since the structural completeness of I_+ implies that $\delta_{MCA/\pi}$ exactly corresponds to δ_A. The second condition on the structural completeness of I_+ imposes that

$$\forall q_A \in F_A, \exists i, 1 \leq i \leq M \text{ such that } q_A^{i,|u_i|} = q_A.$$

Consequently, $F_{MCA} = \{q_{MCA} | \varphi(q_{MCA}) \cap F_A \neq \phi\}$ and $F_{MCA/\pi}$ exactly corresponds to F_A. □

Theorem 2 *Let I_+ be a positive sample of any regular language L and let $A(L)$ denote the canonical automaton accepting L. If I_+ is structurally complete with respect to $A(L)$ then $A(L)$ belongs to $Lat(PTA(I_+))$.*

Proof.
We may follow the same proof than in the theorem 1, except that the acceptance of I_+ by $A(L)$ is now unique since $A(L)$ is deterministic. Similarly, each entry of the transition function of the $PTA(I_+)$ contains at most one member, since the $PTA(I_+)$ is deterministic. □

To illustrate the last part of the demonstration of the theorem 1, we consider the following example. According to our definition, the sample $I_+ = \{ab\}$ is not structurally complete with respect to the automaton A of figure 1, where q_1 and q_2 are both final states. For this reason, the automaton A may not be derived from the $MCA(I_+)$, and obviously neither from the $PTA(I_+)$. Indeed, it is not possible to define a partition π of the state set Q_{MCA} such that A corresponds to $MCA(I_+)/\pi$ with $q_1 \in F_{MCA/\pi}$. On the contrary, the sample $\{ab, a\}$ is structurally complete with respect to the automaton A which may thus be derived from the MCA, or from the PTA, associated with this last sample.

Figure 1: Automaton A and the $MCA(I_+)$.

Theorem 3 *Let I_+ be a positive sample. Let $\mathcal{A}$ be the set of automata such that I_+ is structurally complete with respect to any automaton belonging to $\mathcal{A}$. The set $\mathcal{A}$ is equal to $Lat(MCA(I_+))$.*

Proof.
(1) *If I_+ is structurally complete with respect to an automaton A, then A belongs to $Lat(MCA(I_+))$.* This is exactly the theorem 1.
(2) *If an automaton A belongs to $Lat(MCA(I_+))$ then I_+ is structurally complete with respect to A* [4]. By construction, I_+ is structurally complete with respect to $MCA(I_+)$. Since every automaton in $Lat(MCA(I_+))$ may be derived from the $MCA(I_+)$ for some partition π, the result follows directly from the definition of a quotient automaton. □

3.2 Some properties of the search space

From the section 3.1 we know that if we have a positive sample I_+ of an unknown language L and if we suppose that I_+ is structurally complete with respect to an unknown automaton A accepting exactly L, we may derive A for some partition π of the state set of $MCA(I_+)$. The inference problem may thus be seen as a search through a boolean lattice for the partition π.

Let us first mention that the theorem 1 of section 3.1 is more general than the theorem 2 in the sense that it only supposes the structural completeness of I_+ with respect to any automaton accepting a language L, not necessarily with respect to the canonical automaton[1] for L. Moreover, we have the property 1.

Property 1 $Lat(PTA(I_+)) \subseteq Lat(MCA(I_+))$.

This is clear since $PTA(I_+)$ may be defined as a quotient automaton of $MCA(I_+)$. Besides as the $Lat(PTA(I_+))$ is generally *properly* included in $Lat(MCA(I_+))$, searching in $Lat(PTA(I_+))$ instead of $Lat(MCA(I_+))$ allows for reducing the search space. We shall now detail other properties of these search spaces.

The first important feature is that each element in the lattices corresponds to a particular automaton, but many different automata represent the same language. Therefore, we might think of restricting the search to deterministic automata. However, we have the property 2.

Property 2 *There exist positive samples I_+ for which some languages are only represented by NFA's in $Lat(MCA(I_+))$.*

Note that the same property holds for $Lat(PTA(I_+))$ as a consequence of the property 1. Consider the NFA of figure 2. It accepts the language L_1 defined by the regular expression $(ba^*a)^*$. The sample $I_+ = \{baa\}$ is structurally complete with respect to this automaton. However, this sample is not structurally complete with respect to the minimal DFA for the same language, and thus the same holds for any DFA accepting the language L_1. Consequently, we may not identify the language L_1 if we restrict the search to DFA's.

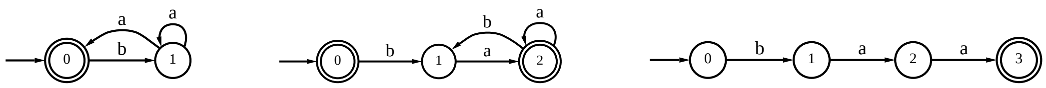

Figure 2: $NFA(L_1)$, $A(L_1)$ and $MCA(I_+)$, with $L_1 = (ba^*a)^*$.

Property 3 *There exist positive samples I_+ for which the set of languages which may be identified from $Lat(PTA(I_+))$ is properly included in the set of languages which may be identified from $Lat(MCA(I_+))$.*

By the property 1 we know that the set of automata which may be derived from the $MCA(I_+)$ includes the set of those which may be derived from the $PTA(I_+)$. The property 3 follows from the fact that sometimes this inclusion is proper and that the automata

[1] A structurally complete sample with respect to $A(L)$ is sometimes called *representative for L* [15].

which belongs to $Lat(MCA(I_+))$ and not to $Lat(PTA(I_+))$, have no equivalent automaton, i.e. any automaton accepting the same language, in $Lat(PTA(I_+))$.

Let us consider the language $L_2 = (ba + b(aa)^*)$, which may be represented by the NFA of figure 3 and its corresponding canonical automaton $A(L_2)$. Let the positive sample I_+ be equal to $\{ba, baa\}$. The associated $MCA(I_+)$ and $PTA(I_+)$ are represented in the figure 4. We may clearly derive $NFA(L_2)$ from the $MCA(I_+)$ while not from the $PTA(I_+)$. This is due to the fact that the states q_1 and q_2 of the MCA, respectively q_3 and q_4, are merged together in the PTA. Moreover, since I_+ is structurally complete with respect to $NFA(L_2)$ but not with respect to $A(L_2)$, this last automaton may not be derived from the PTA and neither any deterministic automaton accepting the same language.

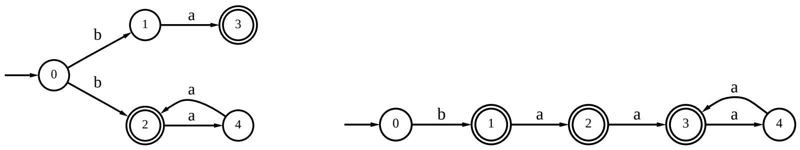

Figure 3: $NFA(L_2)$ and $A(L_2)$, with $L_2 = (ba + b(aa)^*)$.

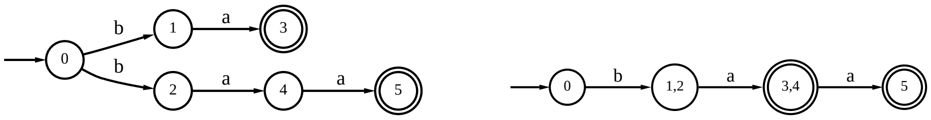

Figure 4: $MCA(I_+)$ and $PTA(I_+)$, with $I_+ = \{ba, baa\}$.

4 Guiding the search for the unknown automaton

4.1 Generalization criteria

Given a positive sample I_+ of a regular language, the criterion which will guide the search for some unknown automaton A accepting I_+ has to be defined. Indeed, we want to generalize in some sense the information contained in this sample.

We know, by construction, that any automaton derived from the $MCA(I_+)$ accepts a language including I_+. In that sense, any automaton belonging to $Lat(MCA(I_+))$ constitutes a possible generalization of the positive sample. Therefore, one may be interested in finding an automaton identifying a language belonging to a particular subclass of regular languages, for example, the class of *k-reversible* languages [3] or the class of *k-testable* languages [5].

Suppose now that a negative sample of the unknown language L is available. Then, the inference problem may be stated as the discovery of an automaton which is *compatible*, or *consistent*, with the positive and negative sample. That is, we look for an automaton A such that $I_+ \in L(A)$ and $I_- \cap L(A) = \phi$. There are many such automata in $Lat(MCA(I_+))$ and, in particular, the MCA satisfies these requirements. However, it does not generalize the positive learning data in the sense that the language accepted strictly corresponds to the positive sample. For this reason, we may look for the DFA, consistent with the positive and negative sample, which moreover contains the minimal number of states. This is the so-called *minimal DFA consistency problem.* The simplicity of the inferred automaton is here chosen as generalization criterion. Remark, however, that this problem has been proved to be NP-Hard (see section 5).

4.2 Border Set

Relating the minimal DFA consistency problem with the characterization of the search space given in section 3, gives rise to the concept of *border set.*

Definition 4.2.1 *An antistring $\overline{as}$ in a lattice of automata is a set of automata such that any element of $\overline{as}$ is not related by $\preceq$ with any other element of $\overline{as}$.*

Definition 4.2.2 *An automaton is said to be at a maximal depth in a lattice of automata, if there is no automaton A' which may be derived from A such that $L(A') \cap I_- = \phi$.*

Definition 4.2.3 *The border set $BS_{MCA}(I_+, I_-)$ is the antistring in $Lat(MCA(I_+))$, of which each element is at a maximal depth.*

Definition 4.2.4 *The border set $BS_{PTA}(I_+, I_-)$ is the antistring in $Lat(PTA(I_+))$, of which each element is at a maximal depth.*

Consequently, the border set of a lattice is the set of automata which correspond to the limit of the generalization under the control of the negative sample. The minimal DFA consistency problem may now be viewed as the discovery of the smallest DFA in a border set.

Property 4 $BS_{PTA}(I_+, I_-) \subseteq BS_{MCA}(I_+, I_-)$.

This is a direct consequence of the property 1.

Property 5 *There may be several distinct languages represented by the automata belonging to $BS_{MCA}(I_+, I_-)$.*

This property follows from the fact that the partial order relation introduced in section 2.3 applies to automata but not to languages.

Property 6 *There may exist NFA's belonging to $BS_{MCA}(I_+, I_-)$ which contain less states than the minimal consistent DFA.*

In particular, this is the case when the minimal consistent DFA corresponds to a language which may be represented by an NFA having less states.

Property 7 *All deterministic automata belonging to $BS_{MCA}(I_+, I_-)$ are necessarily minimal for the language they accept.*

Note that the same property holds for $BS_{PTA}(I_+, I_-)$ as a consequence of the property 4.
Proof.
Let A be a DFA accepting the language L and consistent with I_+ and I_-. Let A be not isomorphic to the canonical automaton $A(L)$. We may construct the partition π of the state set of A in such a way that A/π is obtained from A by merging states sharing the same suffixes (or tails). By construction, $A/\pi = A(L)$. Hence, $A \ll A(L)$ and the depth of $A(L)$ is higher than the depth of A. Consequently, whathever the belonging of $A(L)$ to $BS_{MCA}(I_+, I_-)$, A may not belong to it. $\square$
This property states that any deterministic automaton which is not minimal for the language it accepts, may not belong to $BS_{MCA}(I_+, I_-)$. On the other hand, there exist canonical automata which do not belong to $BS_{MCA}(I_+, I_-)$ since, firstly, all such automata are not necessarily compatible with I_- and, secondly, it may be possible to derive from some canonical automaton another automaton which belongs to $BS_{MCA}(I_+, I_-)$.

An algorithm to construct the border set by enumeration under the control of I_- has been given in [14]. Its computational cost is obviously not tractable and several approximations have been proposed [14].

5 The complexity of regular language identification

In section 4.1, we have essentially limited the discussion to the minimal DFA consistency problem. The relation of this problem with the most general problem of learning a (regular) language needs some clarification. For this purpose, we recall the most classical paradigm of language learning that is *the identification in the limit* proposed by Gold [7]. Next, we mention the computational complexity results of automaton identification and, finally, we discuss the feasibility of approximate identification from sparse data.

5.1 Automaton identification in the limit

There are two closely related variations of identification in the limit [8]. The first one corresponds to the case of *requested data* in which as much data as needed is supposed to be available, that is, may be requested. Here, the learning algorithm is supplied with a growing sequence of data $D_1, D_2, \ldots, D_i, \ldots$ compatible with an arbitrary target machine A. At each discrete time step i the learning algorithm must propose an hypothesis (a DFA, e.g.) $H(D_i)$ representing the guessed solution at step i. The algorithm is said to have the *identification in the limit* property if, after some finite time index t, all the guessed solutions $H(D_i)$, with $i \geq t$, are the same and $L(H(D_i)) = L(A)$.

In the case of *given data*, i.e. the set D of available learning data is fixed, the learning algorithm must propose an hypothesis $H(D)$. This algorithm is said to have the identification in the limit property if, for any target machine A, it is possible to define a set D_A^r as follows [8]:

$$D_A^r \text{ is a subset of } \Sigma^* \text{ such that } \forall D \supseteq D_A^r \;,\; L(H(D)) = L(A).$$

We call the set D_A^r a *representative sample*[2] for $L(A)$ (see section 4.2). Note that the representative sample not only depends on the language to be identified but also on the

[2]This term has not to be confused with the definition of Muggleton [15].

learning algorithm.

Gold showed that, using both positive and negative examples, all the *recursively enumerable* classes of languages, the regular one in particular, can be identified in the limit. In contrast, no *superfinite* class of languages, i.e. one that contains all finite languages and at least one that is infinite, can be identified in the limit using only positive examples [7].

5.2 Theoritical and practical complexity of automaton identification

An excellent overview on the *computational complexity of DFA learning* is given by Pitt [17]. We will not discuss in deep these topics here but rather we will briefly present some facts that may perhaps help seeding new light into the long controversial theme of the *practical* tractability of learning DFA's from positive and negative examples.

The key negative theoretical evidence can be summarized as follows :

- Finding the smallest DFA consistent with I_+ and I_- is NP-Hard [2, 8].
- The minimal DFA consistency problem can not be approximated within any polynomial of the size of the optimal solution [18].
- Approximate inference of finite automata from sparse labeled examples is NP-Hard if an adversary chooses both the target machine and the training set [10].

On the other hand, Trakhtenbrot and Barzdin proposed an $\mathcal{O}(mn^2)$ state-merging algorithm [19] for building the smallest DFA consistent with a *complete* labeled learning sample, where m is the size of the PTA built from the positive information and n is the size of the target DFA. This complete learning sample is made of all the strings up to the length l, each of them being labeled either as positive or negative. The length l depends on some features[3] of the target automaton and has been shown to be equal to $2n-1$, in the *worst case* [19]. Whenever n is large, the size of the learning sample becomes prohibitive. Furthermore, Angluin has shown that, in the worst case, the exact automaton identification is not feasible if a vanishingly small fraction of the complete learning sample is lacking [2].

Fortunately enough, the computational complexity seems to be better in the *average case*. For DFA's randomly drawn from a well defined probability distribution, the expected value of the complete sample size is given by [19]:

$$|S|_{complete} = (|\Sigma|^2 n^C \log_2 n - 1)/(|\Sigma| - 1).$$

where C only depends on $|\Sigma|$.

The expected average size of a complete learning sample, at least for small alphabets, might suggest that exact identification of average DFA's is feasible. However, there is no practical guarantee to have a complete sample available, whatever its size. One should only assume the availability of *sparse* data. In this framework, Lang has empirically shown [11] that highly accurate *approximate* identification may be achieved by using a

[3]Those features are the *degree of distinguibility* of the automaton and its *depth* in the sense defined in [19].

(randomly drawn) vanishingly small fraction of the complete learning sample; that is, a fraction which decreases as the size of the target automaton increases.

The polynomial algorithm proposed by Lang [11], and independently by Oncina and Garcia [16], is reminiscent of the Trakhtenbrot and Barzdin's algorithm [19]. In this technique, a greedy search in $Lat(PTA(I_+))$ yields a "locally optimal" solution to the consistency problem. Indeed, the solution produced by this algorithm is a DFA belonging to $BS_{PTA}(I_+, I_-)$. By the property 7, we know that this DFA is a canonical automaton. However, it is not necessarily the smallest DFA consistent with the data except if the data contain a representative sample. In other words, when the learning data are sufficiently representative, this algorithm is proved to produce the canonical automaton of the language to be identified. Moreover, this automaton is also the solution to the minimal DFA consistency problem in that particular case. Note, finally, that Oncina and Garcia have shown that the size of a representative sample proper to this algorithm is $\mathcal{O}(n^2)$ [16].

References

[1] A. Aho and J. Ullman, *The Theory of Parsing, Translation and Compiling, Vol. 1: Parsing*, Series in Automatic Computation, Prentice-Hall, Englewood, Cliffs, 1972.

[2] D. Angluin, *On the Complexity of Minimum Inference of Regular Sets*, Information and Control, Vol. 39, pp. 337-350, 1978.

[3] D. Angluin, *Inference of Reversible Languages*, Journal of the ACM, Vol. 29, No. 3, pp. 741-765, 1982.

[4] K.S. Fu and T.L. Booth, *Grammatical Inference: Introduction and Survey*, IEEE Transactions on SMC, Part 1: Vol. 5, pp. 85-111, Part 2: Vol. 5, pp. 409-423, 1975.

[5] P. Garcia and E. Vidal, *Inference of K-testable languages in the strict sense and applications to syntactic pattern recognition*, IEEE Transactions on PAMI, Vol. 12, No. 9, pp. 920-925, 1990.

[6] J. Gregor, *Data-driven Inductive Inference of Finite-state Automata*, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 8, No. 1, pp. 305-322, 1994.

[7] E.M. Gold, *Language Identification in the Limit*, Information and Control, Vol. 10, No. 5, pp. 447-474, 1967.

[8] E.M. Gold, *Complexity of Automaton Identification from Given Data*, Information and Control, Vol. 37, pp. 302-320, 1978.

[9] M.A. Harrison, *Introduction to Formal Language Theory*, Addison-Wesley, Reading, Massachusetts, 1978.

[10] M. Kearns and L. Valiant, *Cryptographic Limitations on Learning Boolean Formulae and Finite Automata*, Proc. of the 21st ACM Symposium on Theory of Computing, pp. 433-444, 1989.

[11] K.J. Lang, *Random DFA's can be Approximately Learned from Sparse Uniform Examples*, Proc. of the 5th ACM workshop on Computational Learning Theory, pp. 45-52, 1992.

[12] L. Miclet, *Inférence de Grammaires Régulières*, Thèse de Docteur-Ingénieur, E.N.S.T., Paris, France, 1979.

[13] L. Miclet, *Regular Inference with a Tail-Clustering Method*, IEEE Trans. on SMC, Vol. 10, pp. 737-743, 1980.

[14] L. Miclet and C. de Gentile, *Inférence Grammaticale à partir d'Exemples et de Contre-Exemples : deux Algorithmes Optimaux (BIG et RIG) et une Version Heuristique (BRIG)*, Actes des JFA-94, Strasbourg, France, pp. F1-F13, 1994.

[15] S. Muggleton, *Induction of Regular Languages from Positive Examples*, Turing Institute Report, TIRM-84-009, 1984.

[16] J. Oncina and P. Garcia, *Inferring Regular Languages in Polynomial Update Time*, Pattern Recognition and Image Analysis, N. Perrez de la Blanca, A. Sanfeliu and E. Vidal (editors), Series in Machine Perception and Artificial Intelligence, Vol. 1, pp. 49-61, World Scientific, 1992.

[17] L. Pitt, *Inductive Inference, DFA's, and Computational Complexity*, Lecture Notes in Artificial Intelligence, K.P. Jantke (editor), No. 397, Springer-Verlag, Berlin, pp. 18-44, 1989.

[18] L. Pitt and M. Warmuth, *The Minimum Consistent DFA Problem Cannot be Approximated Within any Polynomial*, Tech. Report UIUCDCS-R-89-1499, University of Illinois, 1989.

[19] B. Trakhtenbrot and Ya. Barzdin, *Finite Automata: Behavior and Synthesis*, North Holland Pub. Comp., Amsterdam, 1973.