

Inferring subclasses of regular languages faster using *RPNI* and forbidden configurations.*

A. Cano, J. Ruiz and P. García.

Depto. de Sistemas Informáticos y Computación.

Universidad Politécnica de Valencia.

Valencia (Spain).

email:{acano,jruiz,pgarcia}@dsic.upv.es

Abstract

Many varieties of regular languages have characterizations in terms of forbidden-patterns of their accepting finite automata. The use of patterns while inferring languages belonging to those families through the *RPNI*-Lang algorithm help to avoid overgeneralization in the same way as negative samples do. The aim of this paper is to describe and prove the convergence of a modification of the *RPNI*-Lang algorithm that we call *FCRPNI*. Preliminary experiments done seem to show that the convergence when we use *FCRPNI* for some subfamilies of regular languages is achieved faster than when we use just the *RPNI* algorithm.

Keywords: Variety of languages, grammatical inference, forbidden configurations.

1 Introduction

Finite automata identification from positive and negative samples is a solved problem in automata theory. If we exclude the general enumeration algorithm there are two constructive algorithms that identify any deterministic finite automaton in deterministic polynomial time.

The first one (*TBG*) is due to Trakhtenbrot and Barzdin [21] by one hand and Gold [10] by the other and the second is due to Oncina and García

*Work partially supported by the Spanish CICYT under contract TIC2000-1153

(*RPNI*) [16] and to Lang [14]. This later algorithm behaves better than *TBG* does [7] as it makes better use of the information contained in the data and thus the convergence is achieved more rapidly.

Even so, the convergence is still slow, which restricts the use of this algorithm in real learning tasks. Several approaches have been proposed aiming to overcome this difficulty. If we exclude the probabilistic approaches [20], [3] and we remain in the classical model of identification in the limit, most of the proposals consist in the modification of *RPNI*-Lang algorithm by means of introducing heuristics that modify the order of merging states from the prefix tree acceptor [11], [15], [5].

The theoretical interest of those approaches is somehow relative as they usually do not guarantee the identification in the limit and then the inferred languages can not be characterized.

The situation today with respect to the inference from positive and negative samples is similar to the situation existing with the inference from only positive samples in the 80's. The distinction between heuristic and characterizable methods [1] opened new possibilities of research [2], [8], [18] etc. Angluin's proposal can be summarized as follows: as the family of regular languages is not identifiable with the use of only positive samples, let us find subclasses which are identifiable in this way and let us propose algorithms for them.

Coming back to inference from complete presentation we can postulate the following question: if we restrict ourselves to the inference of subclasses of regular languages, can we speed up the convergence without dropping the requisite of identification? The starting point of our proposal is to answer positively to this question, it seems that if we restrict the searching space, the identification process will need less information to converge.

Research in language theory has been fruitful in the discovery of subclasses of the regular languages family. The concept of variety of regular languages (also known as pseudovariety), that is, a class of regular languages closed under boolean operations, inverse morphisms and quotients, and its one to one correspondence with the algebraic concept of pseudovariety of semigroups (monoids) [6] can be useful in grammatical inference.

There exist many varieties of languages which are decidable, meaning that given a *DFA* that recognizes a language, its membership to a certain variety can be decided. Among the decidable varieties we have the well known families of finite and cofinite languages, definite, reverse definite and generalized definite languages, locally testable and piecewise testable languages, star free languages, (see [6] or [17]) etc. Note that neither of those families are identifiable from only positive samples. Without considering now facts about the complexity of the algorithms for each particular situation, the decidability

of those families permit a simple modification of the *RPNI*-Lang algorithm: each time that this algorithm tries a merge, besides the usual considerations we have to make sure that the resulting automaton (the language) can still belong to the variety to which we have restricted the inference.

The work we present here wants to be the beginning of a research in order to study the influence that restrictions in the learning domain may have in regular language inference from positive and negative data. We have paid no attention to the time complexity of the proposed method. It is obvious that this complexity will depend on the variety under study and, in fact, there will be varieties for which this method could not be applied for complexity reasons. We describe the method and prove it converges. The experimental results are very limited and they are restricted to two examples of varieties, the families of star-free and piecewise testable languages.

2 Definitions and notation.

In this section we will describe some facts about semigroups and formal languages in order to make the notation understandable to the reader. For further details about the definitions, the reader is referred to [12] (formal languages) and to [6] and [17] (varieties of finite semigroups).

2.1 Automata, languages and semigroups

Throughout this paper Σ will denote a finite alphabet and Σ^* will be the free monoid generated by Σ with concatenation as the internal law and λ as neutral element. A *language* L over Σ is a subset of Σ^* . The elements of L are called *words*. Given $x \in \Sigma^*$, if $x = uv$ with $u, v \in \Sigma^*$, then u (resp. v) is called *prefix* (resp. *suffix*) of x .

A deterministic finite automaton (*DFA*) is a quintuple $A = (Q, \Sigma, \cdot, q_0, F)$ where Q is a finite set of states, Σ is a finite alphabet, $q_0 \in Q$ is the initial state, $F \subseteq Q$ is the set of final states and \cdot is a partial function that maps $Q \times \Sigma$ in Q , which can be easily extended to words. A word x is accepted by an automaton A if $q_0 \cdot x \in F$. The set of words accepted by A is denoted by $L(A)$.

Given an automaton A , $\forall a \in \Sigma$, we can define the function $a^A : Q \rightarrow Q$ as $qa^A = q \cdot a$, $\forall q \in Q$. For $x \in \Sigma^*$, the function $x^A : Q \rightarrow Q$ is defined inductively: λ^A is the identity on Q and $(xa)^A = x^A a^A$, $\forall a \in \Sigma$. Clearly, $\forall x, y \in \Sigma^*$, $(xy)^A = (x^A)(y^A)$. The set $\{a^A : a \in \Sigma\}$ is denoted by M_A . The set of functions $\{x^A : x \in \Sigma^+\}$ is a finite semigroup under the operation of composition of functions, and is denoted as S_A and called *semigroup of* A .

A Moore machine is a 6-tuple $M = (Q, \Sigma, \Gamma, \cdot, q_0, \Phi)$, where Σ (resp. Γ) is the input (resp. output) alphabet, \cdot is a partial function that maps $Q \times \Sigma$ in Q and Φ is a function that maps Q in Γ called *output function*. The behavior of M is given by the partial function $t_M : \Sigma^* \rightarrow \Gamma$ defined as $t_M(x) = \Phi(q_0 \cdot x)$ for every $x \in \Sigma^*$ such that $q_0 \cdot x$ is defined.

Given two finite sets of words D_+ and D_- , we define the (D_+, D_-) -*prefix Moore machine* ($PTM(D_+, D_-)$) as the Moore machine having $\Gamma = \{0, 1, \uparrow\}$, $Q = \text{Pr}(D_+ \cup D_-)$, $q_0 = \lambda$ and $u \cdot a = ua$ if $u, ua \in Q$ and $a \in \Sigma$. For every state u , the value of the output function associated to u is 1, 0 or \uparrow (undefined) depending whether u belongs to D_+ , to D_- or to the complementary set of $D_+ \cup D_-$.

A Moore machine $M = (Q, \Sigma, \{0, 1, \uparrow\}, \delta, q_0, \Phi)$ is *consistent* with (D_+, D_-) if $\forall x \in D_+$ we have $\Phi(q_0 \cdot x) = 1$ and $\forall x \in D_-$ we have $\Phi(q_0 \cdot x) = 0$.

2.2 Varieties of finite semigroups and languages

A *finite semigroup* (resp. *monoid*) is a couple formed from a finite set and an internal associative operation (resp. that has a neutral element).

For every $L \subseteq \Sigma^*$, the congruence \sim_L defined as $x \sim_L y \Leftrightarrow (\forall u, v \in \Sigma^*, uxv \in L \Leftrightarrow uyv \in L)$, is called the *syntactic congruence* of L and it is the coarsest congruence that saturates L . Σ^* / \sim_L is called the *syntactic monoid of L* and is denoted as $S(L)$. The morphism $\varphi : \Sigma^* \rightarrow S(L)$, that maps each word to its equivalence class modulo \sim_L is called the *syntactic morphism* of L .

A *variety of finite monoids* (also denoted as *pseudovariety*) is a class of finite monoids closed under morphic images, submonoids and finite direct products.

A *variety of recognizable languages* is a class of languages closed under finite union and intersection, complement, inverse morphisms and right and left quotients. Eilenberg [6] proved that varieties of finite monoids and varieties of languages are in one-to-one correspondence. If \mathbf{V} is a variety of semigroups, we denote as $\mathcal{L}_{\mathbf{V}}(\Sigma^*)$ the variety of languages over Σ whose syntactic semigroups lie in \mathbf{V} .

Some instances of this correspondence that will be used throughout this paper are the relations between:

- The variety of locally testable languages and the variety of locally idempotent and commutative semigroups.
- The variety of piecewise testable languages and the variety of J-trivial semigroups.

- The variety of star-free languages and the variety of aperiodic semi-groups.

2.3 Forbidden configurations

Given an automaton $A = (Q, \Sigma, \cdot, q_0, F)$, the set of all paths in A defines an infinite labelled graph $G(A)$ where the set of vertices is Q and the set of edges is $\{(q, w, q \cdot w) : q \in Q, w \in \Sigma^+\}$. A labelled subgraph \mathbb{P} of $G(A)$ is said to be a configuration, or a pattern, present in A .

The forbidden-pattern characterizations have been developed in the study of the relations between logic and formal languages. They are results of the following type: "A language L belongs to a class \mathcal{C} if and only if the accepting finite automaton does not have subgraph \mathbb{P} in its transition graph". Usually, forbidden-pattern characterizations imply the decidability of the characterized class, since we only have to test whether the forbidden-pattern occurs in an automaton.

For many varieties of regular languages the forbidden-pattern characterization is well known, but for others the question remains open, for example it can be shown that the semigroup of a (minimal) deterministic automaton A is idempotent if and only if there exist no configuration of A of the form depicted in figure 1, where $x \in \Sigma^+$ and $p \neq q$.

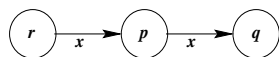


Figure 1: Forbidden configuration for an automaton whose semigroup is idempotent

Forbidden-pattern characterization for several varieties of languages can be found in [4], [9] and [19].

3 A description of the *FCRPNI* algorithm

We suppose that the reader is familiar with *RPNI*-Lang algorithm [16], [14]. The version we use presents *DFA*s as Moore machines with output belonging to the set $\{0, 1, \uparrow\}$. Any word x such that $\Phi(q_0 \cdot x) = 0$ (resp. 1) (resp. \uparrow) is considered negative (resp. positive) (resp. not defined). The only changes that our algorithm makes with respect to *RPNI*-Lang algorithm is that before we definitively merge two states, we test if the resulting automaton can

still belong to the considered variety. This is done by looking for possible forbidden configurations in the stable part of the automaton. In the sequel, the modification we propose will be denoted as *FCRPNI* (Forbidden Patterns *RPNI*).

Merging two states in *RPNI*-Lang algorithm possibly makes some other states to be merged in order to guarantee the determinism of the automaton. Then this automaton is tested for consistency with the data (a state can not be positive and negative) and if it is not consistent we have to undo the merging.

Besides that test, *FCRPNI* has to establish that the current automaton can still belong to the variety to be learned. This is done by testing whether the forbidden-patterns occur in the consolidated part of the automaton.

The following example illustrates the differences between *RPNI*-Lang and *FCRPNI* algorithms, we restrict ourselves to the variety of star-free languages. We recall that a language is star-free if and only if the minimal automaton that recognizes it is permutation-free¹.

Example 1 Let $L = aa^*$ (which is star-free) and let $D_+ = \{a, aaa\}$ and $D_- = \{\lambda\}$. The prefix tree acceptor is represented in Figure 2

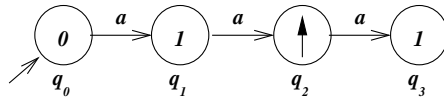


Figure 2: Prefix Moore machine for the sample $D_+ = \{a, aaa\}$ and $D_- = \{\lambda\}$.

RPNI-Lang tries to merge q_0 and q_1 but the consistency test fails. In the following step it tries to merge q_0 and q_2 which implies the merging of q_1 and q_3 . The result is depicted in Figure 3.

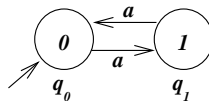


Figure 3: *DFA* output by *RPNI* algorithm on input of $D_+ = \{a, aaa\}$ and $D_- = \{\lambda\}$.

As every state in the prefix Moore machine has been considered, the algorithm finishes. We see that the input was not enough to learn aa^* .

¹An automaton has a permutation if there exists a subset $P \subseteq Q$ and a word x such that $P \cdot x = P$, where $P \cdot x = \cup_{p \in P} p \cdot x$

FCRPNI with the restriction to star-free languages proceeds in the same way but as the automaton depicted in Figure 3 fails in the test of forbidden patterns for star-free languages, it can not merge states q_0 and q_2 . The following step is to try to merge q_1 and q_2 which implies the merging of q_1 and q_3 . The final result is the automaton represented in figure 4.

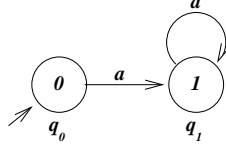


Figure 4: *DFA output by FCRPNI algorithm on input of $D_+ = \{a, aaa\}$ and $D_- = \{\lambda\}$.*

3.1 Convergence of FCRPNI algorithm.

Fact. Let $L \in \mathcal{L}_V(\Sigma^*)$ be the target language and let $D_+ \cup D_-$ a complete sample for the inference of language L using *RPNI-Lang* algorithm. At every step this algorithm outputs a finite automaton with no forbidden configurations as those forbidden patterns may only appear in consolidated part of the automaton and that subautomaton remains stable during the rest of the process.

Proposition 2 *FCRPNI algorithm identifies any variety $\mathcal{L}_V(\Sigma^*)$ from a complete presentation in the limit.*

Proof. Let $L \in \mathcal{L}_V(\Sigma^*)$ be the target language. If the samples used for inference contain a characteristic sample of L for the *RPNI-Lang* algorithm, no forbidden configuration can appear in the stable part of the automaton during the process of inference, and *FCRPNI* algorithm behaves in the same way as *RPNI-Lang* algorithm does and thus, it identifies the language L . ■

Proposition 3 *Let $L \in \mathcal{L}_V(\Sigma^*)$ be the target language and let $D_+ \cup D_-$ an arbitrary sample such that when it is used as input for the *RPNI-Lang* algorithm, it outputs the canonical acceptor for L . Then *FCRPNI* algorithm outputs the same automaton.*

Proof. It is obvious that if *RPNI-Lang* algorithm enters in a forbidden configuration it will not identify language L . ■

Propositions 2 and 3 show in theory that if we restrict the identification to varieties of regular languages, *FCRPNI* algorithm works somehow better

than *RPNI*-Lang does. It is clear that there are situations in which the former converges whereas the later does not (see example 1). The cost is that we have to check for forbidden configurations, so only a complete set of experiments will allow us to quantify the differences.

4 Experimental results.

We have done some small experiments in order to compare the behavior of *RPNI* and *FCRPNI* for two varieties of formal languages, the variety of piecewise testable languages and the variety of star-free languages.

Description of the experiments:

- We work with minimal automata having 5 states, the alphabet is $\Sigma = \{a, b\}$. Each of them recognizes either a piecewise testable language or a star-free one. We obtain them beginning with larger automata, we then minimize them and discard the automata which do not have the required size. Afterwards we calculate the transformation semigroup of each automaton and discard the ones that does not belong to the required class.
- For the learning process we use randomly generated strings of length less than or equal to 10 over Σ . The number of them is shown in the tables that describe the results of the experiments.
- The comparison of the obtained automata is done using all the words of length less than or equal to 15 not used in the learning process.
- We have done 200 experiments for each different types of languages.

Table 1 (resp. Table 2) show the mean of the error rate (percentage of words not correctly classified) in terms of the number of words used in the learning process when *RPNI*-Lang and *FCRPNI* are used for inference of piecewise testable (resp. star-free) languages.

| Number of samples used for inference | 20 | 40 | 60 | 80 | 100 |
|--------------------------------------|-------|------|------|------|------|
| error rate of <i>RPNI</i> -Lang | 12,12 | 3,42 | 2,00 | 0,93 | 0,77 |
| error rate of <i>FCRPNI</i> | 9,73 | 2,57 | 0,62 | 0,26 | 0,11 |

Table 1: Mean of the error rate when of *RPNI* and *FCRPNI* are used for the inference of piecewise testable languages in terms of the number of words used in the learning process

| | | | | | |
|--------------------------------------|------|------|------|------|------|
| Number of samples used for inference | 20 | 40 | 60 | 80 | 100 |
| error rate of <i>RPNI</i> -Lang | 8,10 | 1,63 | 0,75 | 0,28 | 0,24 |
| error rate of <i>FCRPNI</i> | 7,11 | 1,23 | 0,38 | 0,13 | 0,12 |

Table 2: Mean of the error rate when of *RPNI* and *FCRPNI* are used for the inference of star-free languages in terms of the number of words used in the learning process

5 Conclusions and future work.

We have described *FCRPNI*, a modification of *RPNI*-Lang algorithm aiming to show how the restriction of the learning domain to certain well characterized subclasses of the family regular languages affects to the convergence.

We have proved that if *RPNI*-Lang algorithm has been given enough samples to converge, so does *FCRPNI*. Then, this later algorithm converges faster than the former one at the cost of:

- Restrict the domain of the inference process.
- The additional cost of having to do the forbidden-pattern tests.

We have done some preliminary experiments for two varieties of languages, the aperiodic and the piecewise testable. Although the error rates for *FCRPNI* are better in both cases, as the size of the automata we have used is very small, we can not be conclusive about how much better they are.

As future work we should make efficient algorithms for some of the varieties for which it is possible and we should make a complete experimentation to measure how both algorithms behave.

References

- [1] Angluin, D. *Inductive Inference of Formal Languages from Positive Data*. Inform and Control, pp. 117-135 (1980).
- [2] Angluin, D. *Inference of Reversible Languages*. Journal of the ACM, Vol 29-3. pp. 741-765 (1982).
- [3] Carrasco, R. and Oncina, J. *Learning Stochastic Regular Grammars by means of a State Merging Method*. In Grammatical Inference and Applications. R. Carrasco and J. Oncina (Eds.). LNAI 862. Springer-Verlag, pp. 139-152 (1994).
- [4] Cohen, J. Perrin D. and Pin J-E. *On the expressive power of temporal logic*. Journal of computer and System Sciences 46, pp 271-294 (1993).

- [5] Coste, F. and Nicolas J. *How considering Incompatible State Mergings May Reduce the DFA induction Search Tree*. In Grammatical Inference. V. Honavar and G. Slutzki (Eds.) LNAI 1433. Springer-Verlag, pp 199-210 (1998).
- [6] Eilenberg, S. *Automata, Languages and Machines*, Vol A and B (Academic Press, 1976)
- [7] García, P. Cano, A. and Ruiz, J. *A comparative study of two algorithms for automata identification*. In Grammatical Inference: Algorithms and Applications. A.L. Oliveira (Ed.) LNAI 189. Springer-Verlag, pp. 115-126 (2000).
- [8] García P. and Vidal E. *Inference of k -Testable languages in the Stric Sense and Applications to Syntactic Pattern Recognition*. IEEE Transactions on Pattern Analysis and Machine Intelligence 12/9, pp 920-925 (1990).
- [9] Glaßer, C. *Forbidden-Patterns and Word Extensions for Concatenation Hierarchies*. Ph dissertation, Würzburg University, Germany, 2001.
- [10] Gold , M. *Complexity of Automaton Identification from Given Data*. Information and Control 37, pp 302-320 (1978).
- [11] de la Higuera, C. Oncina, J. and Vidal, E. *Data dependant vs data independant algorithms*. In Grammatical Inference: Learning Syntax from Sentences. L. Miclet and C. de la Higuera (Eds.). LNAI 1147. Springer-Verlag, pp. 313-325 (1996).
- [12] Hopcroft, J. and Ullman, J. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley (1979).
- [13] Juillé , H. and Pollack J. *A Stochastic Search Approach to Grammar Induction*. In Grammatical Inference. V. Honavar and G. Slutzki (Eds.) LNAI 1433. Springer-Verlag, pp 126-137 (1998).
- [14] Lang , K.J. *Random DFA's can be Approximately Learned from Sparse Uniform Examples*. In Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory, pp 45-52. (1992).
- [15] Lang , K.J., Pearlmutter B.A. and Price R.A. *Results on the Abbadingo One DFA Learning Competition and a New Evidence-Driven State Merging Algorithm*. In Grammatical Inference. V. Honavar and G. Slutzki (Eds.) LNAI 1433. Springer-Verlag, pp 1-12 (1998).
- [16] Oncina, J. and García, P. *Inferring Regular Languages in Polynomial Updated Time*. In Pattern Recognition and Image Analysys. Pérez de la Blanca, Sanfeliú and Vidal (Eds.) World Scientific. (1992).

- [17] Pin, J. *Varieties of formal languages*. Plenum. (1986).
- [18] Ruiz, J. and García, P. *Learning k -piecewise testable languages from positive data*. In Grammatical Inference: Learning Syntax from Sentences. L. Miclet and C. de la Higuera (Eds.). LNAI 1147. Springer-Verlag, pp. 203-210 (1996).
- [19] Schmitz, H. *The Forbidden-Pattern approach to Concatenation Hierarchies*. Ph dissertation, Würzburg University, Germany, 2001.
- [20] Stolcke, A. and Omohundro, S. *Inducing Probabilistic Grammars by Bayesian Model Merging*. In Grammatical Inference and Applications. R. Carrasco and J. Oncina (Eds.). LNAI 862. Springer-Verlag, pp. 106-118 (1994).
- [21] Trakhtenbrot B. and Barzdin Ya. *Finite Automata: Behavior and Synthesis*. North Holland Publishing Company. (1973).
- [22] Vidal, E. and Llorens, S. *Using Knowledge to improve N -Gram Language Modelling through the MGGI Methodology*. In Grammatical Inference: Learning Syntax from Sentences. L. Miclet and C. de la Higuera (Eds.). LNAI 1147. Springer-Verlag, pp. 179-190 (1996).