

Learning of Bi- ω Languages from Factors

M. Jayasrirani

Arignar Anna Government Arts College, Walajapet

MJAYASRIRANI@YAHOO.CO.IN

D.G. Thomas

Madras Christian College, Chennai - 600 059

DGTHOMASMCC@YAHOO.COM

M.H. Begam

Arignar Anna Government Arts College, Walajapet

HUMROSIA_BEGAM@YAHOO.CO.IN

J.D. Emerald

Arignar Anna Government Arts College, Walajapet

EMERALD_SHEELA@YAHOO.COM

Editors: Jeffrey Heinz, Colin de la Higuera and Tim Oates

Abstract

De la Higuera and Janodet (2001) gave a polynomial algorithm that identifies the class of safe ω -languages which is a subclass of deterministic ω -languages from positive and negative prefixes. As an extension of this work we study the learning of the family of bi- ω languages.

Keywords: DB-machine, bi- ω language, learning.

1. Introduction

Nivat (1979) has made a study of infinite words and infinite successful computations in an attempt to define the semantics of recursive programs. He has considered a method of generating infinite words by algebraic or context-free grammars. Bi-infinite words or two sided infinite words are natural extensions of infinite words and have also been objects of interest and study. The theory of finite automata has been extended to bi-infinite words by Nivat and Perrin (1986) and the study has been continued (Beauquier, 1986; Devolder and Litovsky, 1991).

Maler and Pnueli (1995) have exhibited a learning algorithm for a subclass of ω -regular languages which is recognized by a finite automaton with Büchi condition and is also recognized by a finite automaton with Muller condition from membership queries and counter examples based on the framework suggested by Angluin (1987). A linear time learning algorithm in Gold's framework of identification in the limit from positive data (Gold, 1967) is given for a subclass of ω -regular languages with restricted superset queries by Saoudi and Yokomori (1994).

S. Gnanasekaran and Thomas (2001) have introduced a new class called Büchi local ω -languages and proved that the class of ω -regular languages is an alphabetic morphic image of the class of Büchi local ω -languages. They gave a polynomial algorithm that identifies the class of ω -regular languages. Thomas et al. Thomas et al. (2002) extended this result to bi- ω regular languages.

de la Higuera and Janodet (2001) investigated the question of inferring ω -languages through the prefixes of accepted or rejected infinite words. They gave a polynomial al-

gorithm that identifies the class of safe ω -languages which is a subclass of deterministic ω -languages from positive and negative prefixes.

As an extension of this work, in this paper, we exhibit that the classes of bi- ω regular languages and deterministic bi- ω languages are not identifiable in the limit using factors of accepted and rejected bi- ω words and the subclass of bi- ω regular languages called bi- ω safe languages are learnable from positive and negative factors of bi- ω words.

2. Bi- ω Languages and Results

In this section we recall the notions of bi- ω languages (Beauquier, 1986; Nivat and Perrin, 1986) and prove certain results.

An alphabet Σ is a finite set whose members are called symbols. A finite string or word on an alphabet Σ is a finite sequence of zero or more symbols of Σ . The set of all words on Σ is denoted by Σ^* and ϵ denotes the empty word which has length zero.

If $x, y \in \Sigma^*$ then x is a prefix of xy and y is a suffix of xy . If $x, y, z \in \Sigma^*$ then y is a subword or factor of xyz . For each $x \in \Sigma^*$, $Fact(x)$ is the collection of all factors of x and $Fact(L) = \cup_{x \in L} Fact(x)$.

An infinite word or a right-infinite word or an ω -word on an alphabet Σ is a mapping from the set N of natural numbers to Σ . An infinite word x on Σ is written as $x = x_1x_2x_3 \dots$ ($x_i \in \Sigma, i \geq 1$). Let left-infinite word x on an alphabet Σ is mapping from the set Z^- of negative integers to Σ and is written as $x = \dots x_{-3}x_{-2}x_{-1}$ ($x_{-i} \in \Sigma, i \geq 1$).

Let Σ^Z denote the set of all mappings from the set of integers into Σ . A bi-infinite word is a class under the equivalence relation ρ over Σ^Z , defined by $u\rho v$, where $u = (u_i)_{i \in Z}$ and $v = (v_j)_{j \in Z}$, if and only if there exists some $p \in Z$ such that for every $n \in Z$, $u_{n+p} = v_n$. A bi-infinite word is written as $x = \dots x_{-3}x_{-2}x_{-1}x_0x_1x_2x_3 \dots$ ($x_i \in \Sigma, i \in Z$). The set of infinite words on Σ is denoted by Σ^ω , the set of all left-infinite words on Σ is denoted by ${}^\omega\Sigma$ and set of all bi-infinite words on Σ is denoted by ${}^\omega\Sigma^\omega$.

Definition 1 A Büchi (finite) automaton M over an alphabet Σ is

$M = \langle Q, \Sigma, E_M, I_{inf}, T_{inf} \rangle$ where Q is a finite set of states; Σ is a finite alphabet; E_M is a subset of $Q \times \Sigma \times Q$, called the set of arrows; $I_{inf} \subseteq Q$ is a set of left-infinite repetitive states; $T_{inf} \subseteq Q$ is a set of right-infinite repetitive states.

An arrow $c = (p, a, q)$ is also denoted by $c : p \xrightarrow{a} q$. A path in M is a bi-infinite sequence of arrows $c_i, i \in Z$, namely $c = \dots c_{-2}c_{-1}c_0c_1c_2 \dots$. The arrows in the paths are consecutive in the sense that $c_{i+1} : q_i \xrightarrow{a_{i+1}} q_{i+1}$ where $i \in Z$. The bi-infinite word $a = \dots a_{-2}a_{-1}a_0a_1a_2 \dots$ is called the label of the path c . A notation $c : I_{inf} \xrightarrow{w} T_{inf}$ denotes a bi-infinite path c of label w which passes through states of I_{inf} infinitely often on the left and through states of T_{inf} infinitely often on the right. Let

$$L^{\omega\omega}(M) = \{w \in {}^\omega\Sigma^\omega / \exists (c : I_{inf} \xrightarrow{w} T_{inf})\}$$

be a bi- ω language accepted by M . Let $Rec({}^\omega\Sigma^\omega) = \{L \subseteq {}^\omega\Sigma^\omega / \exists \text{ a Büchi automaton } M \text{ such that } L^{\omega\omega}(M) = L\}$ and $Rec({}^\omega\Sigma^\omega)$ is known as the family of all bi- ω regular languages each of which is a finite union of sets of the form

$$\begin{aligned} {}^\omega XY Z^\omega = \{u \in {}^\omega\Sigma^\omega / u = \dots x_n \dots x_2 x_1 y z_1 z_2 \dots z_{n-1} z_n \dots, \text{ with } x_i \in X, \\ y \in Y \text{ and } z_i \in Z, i \geq 1\} \end{aligned}$$

where X, Y and Z are regular languages in Σ^* . The Büchi automaton M over Σ is said to be deterministic if for every three states $q, q_1, q_2 \in Q$ and every symbol $a \in \Sigma$, $(q, a, q_1) \in E_M$ and $(q, a, q_2) \in E_M \Rightarrow q_1 = q_2$.

Definition 2 A bi- ω language L is safe if

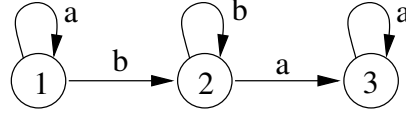
$\forall x \in {}^\omega\Sigma^\omega, \forall v \in \text{Fact}(x), \exists u \in {}^\omega\Sigma, w \in \Sigma^\omega$ such that $uvw \in L \Rightarrow x \in L$

i.e., $\forall x \in {}^\omega\Sigma^\omega, \text{Fact}(x) \subseteq \text{Fact}(L) \Rightarrow x \in L$

i.e., $\forall x \in {}^\omega\Sigma^\omega, x \notin L \Rightarrow \exists v \in \text{Fact}(x)$ such that $\forall u \in {}^\omega\Sigma^\omega, w \in \Sigma^\omega, uvw \notin L$

Let $\text{Safe}_{\omega\omega}(\Sigma)$ denote the class of all safe bi- ω regular languages.

Example 1 ${}^\omega ab^\omega + {}^\omega ba^\omega + {}^\omega ab^*a^\omega + {}^\omega b^\omega$ is a safe language accepted by the automaton $M = \langle Q, \Sigma, \delta, I_{\text{linf}}, T_{\text{rinf}} \rangle$, $Q = \{1, 2, 3\}$, $\Sigma = \{a, b\}$, $I_{\text{linf}} = \{1, 2\}$, $T_{\text{rinf}} = \{2, 3\}$ and δ is the transition function.



But ${}^\omega ab^\omega + {}^\omega ba^\omega + {}^\omega ab^*a^\omega$ is not a safe language because every factor b^k of ${}^\omega b^\omega$ is a factor of ${}^\omega ab^\omega + {}^\omega ba^\omega + {}^\omega ab^*a^\omega + {}^\omega b^\omega$. It follows that the class of all safe bi- ω regular languages is a subclass of all deterministic bi- ω languages. Hence $\text{Safe}_{\omega\omega}(\Sigma) \neq \text{Det}_{\omega\omega}(\Sigma)$.

Definition 3 A DB-machine is a deterministic Büchi automaton where $I_{\text{linf}} = Q = T_{\text{rinf}}$.

Definition 4 (Beauquier, 1986) A language L is bi- ω regular if there exist three sequences of regular languages $(A_i)_{i \in \{1, \dots, n\}}$, $(B_i)_{i \in \{1, \dots, n\}}$ and $(C_i)_{i \in \{1, \dots, n\}}$ such that

$$L = \bigcup_{i=1}^n {}^\omega A_i B_i C_i^\omega.$$

We note that

$$\text{Fact}(L) = \text{Fact} \left(\bigcup_{i=1}^n {}^\omega A_i B_i C_i^\omega \right) = \bigcup_{i=1}^n \text{Fact}({}^\omega A_i B_i C_i^\omega)$$

$$\begin{aligned} \text{Fact}({}^\omega A_i B_i C_i^\omega) = & \text{Suff}(A_i) A_i^* B_i C_i^* \text{Pref}(C_i) \cup \text{Suff}(A_i) A_i^* \text{Pref}(B_i) \cup \\ & \text{Suff}(B_i) C_i^* \text{Pref}(C_i) \cup \text{Suff}(C_i) C_i^* \text{Pref}(C_i) \cup \\ & \text{Suff}(A_i) A_i^* \text{Pref}(A_i) \cup \text{Fact}(B_i) \end{aligned}$$

where $\text{Fact}(L)$ is the set of all factors (finite words) of members of $L \subseteq {}^\omega\Sigma^\omega$, $\text{Pref}(C)$ is the set of all prefixes (finite words) of members of $C \subseteq \Sigma^*$ and $\text{Suff}(A)$ is the set of all suffixes (finite words) of members of $A \subseteq \Sigma^*$.

Definition 5 (de la Higuera and Janodet, 2001) A language $P \subseteq \Sigma^*$ is a regular factor language if (1) P is regular (2) Every factor of a word of P is a word of P i.e., $\forall u \in \Sigma^*, \forall a, b \in \Sigma, aub \in P \Rightarrow u \in P$ (3) Every word of P is a proper factor of another word of P i.e., $\forall u \in P, \exists a, b \in \Sigma$ such that $aub \in P$.

A deterministic finite state automaton (dfa) is a factor automaton (factor dfa) if and only if (1) Every state is initial and final (2) Every state is alive i.e., $\forall q \in Q, \exists a, b \in \Sigma$ and $q' \in Q, \delta(q, b) \in Q$ and $\delta(q', a) = q$.

Proposition 6

1. If L is a bi- ω regular language then $Fact(L)$ is a regular factor language.
2. If P is a regular factor language, then there exists a factor automaton which recognizes P .
3. If $A = \langle Q, \Sigma, \delta, I, T \rangle$ is a factor automaton then the language $\mathcal{L}(M)$ recognized by the DB-machine $M = \langle Q, \Sigma, \delta, I_{inf}, T_{inf} \rangle$ is bi- ω regular and satisfies $\mathcal{L}(A) = Fact\mathcal{L}(M)$.

Theorem 7 L is a safe bi- ω regular language if and only if L is recognized by a DB-machine.

3. Learning of Bi- ω Languages

In this section, we define 'positive factors' and 'negative factors' of a word in a bi- ω language L and exhibit that bi- ω safe languages are learnable from positive and negative factors of bi- ω words.

Definition 8 Let $v \in \Sigma^*$

1. v is an \exists positive factor of L iff $\exists u \in {}^\omega\Sigma, w \in \Sigma^\omega$ such that $uvw \in L$
2. v is an \forall positive factor of L iff $\forall u \in {}^\omega\Sigma, w \in \Sigma^\omega$ such that $uvw \in L$
3. v is an \exists negative factor of L iff $\exists u \in {}^\omega\Sigma, w \in \Sigma^\omega$ such that $uvw \notin L$
4. v is an \forall negative factor of L iff $\forall u \in {}^\omega\Sigma, w \in \Sigma^\omega$ such that $uvw \notin L$

Given a bi- ω language L , let $P_\forall(L)$ denote the set of all \forall -positive factors of L , $P_\exists(L)$ denote the set of all \exists positive factors of L , $N_\forall(L)$ the set of \forall -negative factors of L and $N_\exists(L)$ the set of all \exists -negative factors of L .

Two finite sets $S+$ and $S-$ of finite words form together a set of (p, n) examples for a bi- ω language if and only if $S+ \subseteq P_p(L)$ and $S- \subseteq N_n(L)$.

Without explaining the notions and notations which can be naturally extended for bi- ω languages from the case of ω -languages (de la Higuera and Janodet, 2001), we obtain the following results.

Lemma 9 Let \mathbb{L} be a class of bi- ω languages and \mathbb{R} a class of representations for \mathbb{L} . If there exist L_1 and L_2 in \mathbb{L} such that $L_1 \neq L_2$, $P_p(L_1) = P_p(L_2)$ and $N_n(L_1) = N_n(L_2)$ then the problem $\langle \mathbb{L}_{\mathbb{R}}, idlim, \langle p, n \rangle \rangle$ has a negative status.

Theorem 10 For any class of representations \mathbb{R} and for all $(p, n) \in \{\exists, \forall\} \times \{\exists, \forall\}$ problems $\langle Reg_{\omega\omega}(\Sigma)_R, idlim, (p, n) \rangle$ and $\langle Det_{\omega\omega}(\Sigma)_R, idlim, (p, n) \rangle$ have negative status where $idlim$ stands for identification in the limit.

Theorem 11 $\langle Safe_{\omega\omega}(\Sigma)_{DBM}, polyid, (\exists, \forall) \rangle$ has a positive status where $polyid$ stands for polynomially identifiable in the limit.

The result is an extension of the result given in (de la Higuera and Janodet, 2001).

References

- D. Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75:87–106, 1987.
- D. Beauquier. Thin homogeneous sets of factors. In *LNCS*, volume 241, pages 239–251, 1986.
- C. de la Higuera and J.C Janodet. Inference of ω -languages from prefixes. *LNAI*, 2225: 364–378, 2001.
- J. Devolder and I. Litovsky. Finitely generated bi- ω -languages. *Theoretical Computer Science*, 85(1):33–52, 1991.
- M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.
- O. Maler and A. Pnueli. On the learnability of infinitary regular sets. *Information and Computation*, 118:316–326, 1995.
- M. Nivat. Infinite words, infinite trees and infinite computations. 109:1–52, 1979.
- M. Nivat and D. Perrin. Ensembles reconnaissables de mots bi-infinis. *Canadian Journal of Mathematics*, 38:513–537, 1986.
- K.G. Subramanian S. Gnanasekaran, V.R. Dare and D.G. Thomas. Learning ω -regular languages. In *Proceedings of the International Symposium on Artificial Intelligence*, pages 206–211. Allied Publishers, 2001.
- A. Saoudi and T. Yokomori. Learning local and recognizable ω -languages and monadic logic programs. In *Proceedings of EuroColt 1993*, pages 157–169. Oxford University Press, 1994.
- D.G. Thomas, M.H. Begam, K.G. Subramanian, and S. Gnanasekaran. Learning of regular bi- ω languages. *LNAI*, 2424:283–292, 2002.

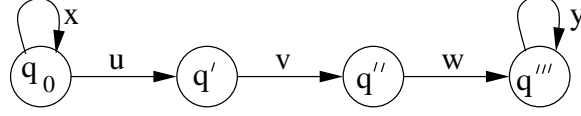
Appendix

Proof of Proposition 1.

1. L is a bi- ω regular language. So $L = \bigcup_{i=1}^n {}^\omega A_i B_i C_i^\omega$ where A_i, B_i, C_i are regular sets. Since regular languages are closed under union, product and star, $Fact(\bigcup_{i=1}^n {}^\omega A_i B_i C_i^\omega)$ is a regular language. Hence $Fact(L)$ is regular.
2. Let P be a regular factor language. P is recognized by a dfa A which is minimal and dead state free. As P is a factor language, every state of this automaton is initial and final. Moreover let q_1 be a state of A and u , a word of P , such that $\delta(q, u) = q_1$. By the definition of a factor language, there exist $a, b \in \Sigma$ such that $aub \in P$. So $\delta(q', a) = q$ and $\delta(q_1, b)$ is necessarily defined. Hence q is alive.
3. Let $A = \langle Q, \Sigma, \delta, I, T \rangle$ be a factor automaton. Consider the corresponding DB-machine $M = \langle Q, \Sigma, \delta, I_{linf}, T_{rinf} \rangle$ (where $I_{linf} = T_{rinf} = Q$). Let us prove that $Fact\mathcal{L}(M) = \mathcal{L}(A)$.
Let $v \in Fact\mathcal{L}(M)$.

Then there exist $u \in {}^\omega\Sigma$ and $w \in \Sigma^\omega$ such that $uvw \in \mathcal{L}(M)$. So there exists a state $q \in Q$ such that $\delta(q, v) \in Q$ and hence $v \in \mathcal{L}(A)$.

Conversely let $v \in \mathcal{L}(A)$ and $q'' = \delta(q', v)$, $q', q'' \in Q$. As q' and q'' are alive, we can build two words x, u, w, y such that $\delta(q_0, x) = q_0$, $\delta(q_0, u) = q'$, $\delta(q'', w) = q'''$ and $\delta(q''', y) = q'''$.



Clearly the bi-infinite path of label ${}^\omega x u v w y^\omega$ passes through the state q_0 infinitely often on the left and the state q''' infinitely often on the right. So ${}^\omega x u v w y^\omega \in \mathcal{L}(M)$. Thus $v \in \text{Fact}\mathcal{L}(M)$.

Proof of Theorem 1. Let L be a language recognized by a DB-machine $M = \langle Q, \Sigma, \delta, Q, Q \rangle$ and $w \in {}^\omega\Sigma^\omega$. Assume that every factor w_i of w can be continued into a word of L recognized by M . The mapping $c : N \rightarrow Q$ such that $c(i+1) = \delta(c(i), w_i)$ is a run of M on w_i and hence on w . Since all the states of M are marked, this run is successful and so $w \in L$. Hence L is a safe bi- ω regular language.

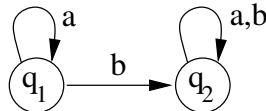
Conversely, let L be a safe bi- ω regular language. By proposition 6, $\text{Fact}(L)$ is a regular factor language which is recognized by some factor automaton $A = \langle Q, \Sigma, \delta, Q, Q \rangle$. We claim that L is recognized by the DB-machine $M = \langle Q, \Sigma, \delta, Q, Q \rangle$. By Proposition 6, the language $\mathcal{L}(M)$ satisfies $\text{Fact}\mathcal{L}(M) = \mathcal{L}(A)$. By the first part of the proof, $\mathcal{L}(M)$ is a safe language since M is a DB-machine. So L and $\mathcal{L}(M)$ are both safe languages such that $\text{Fact}(L) = \text{Fact}\mathcal{L}(M) = \mathcal{L}(A)$. Assume that there exists a word $w \in L$ and not in $\mathcal{L}(M)$ (or vice-versa). As $\text{Fact}(L) = \text{Fact}\mathcal{L}(M)$, every factor of w is in $\text{Fact}\mathcal{L}(M)$. Since $\mathcal{L}(M)$ is a safe language, w itself is in $\mathcal{L}(M)$, a contradiction. So $L = \mathcal{L}(M)$.

Proof of Theorem 2. We prove the theorem by giving counterexample.

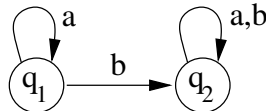
$$L_1 = {}^\omega a b a^* (a + b)^\omega, \quad L_2 = {}^\omega a^\omega + {}^\omega a b a^* (a + b)^\omega$$

These languages are accepted by the following automata, respectively.

- 1) $Q = \{q_1, q_2\}$, $I_{\text{inf}} = \{q_1\}$, $T_{\text{inf}} = \{q_2\}$



- 2) $Q = \{q_1, q_2\}$, $I_{\text{inf}} = \{q_1\}$, $T_{\text{inf}} = \{q_1, q_2\}$



But it is clear that whatever the choice of quantifiers p and n , languages P_p and N_n are identical in both the cases.

Formally, $P_\exists(L_1) = P_\exists(L_2) = \Sigma^*$; $P_\forall(L_1) = P_\forall(L_2) = a^* b a^* (a + b)^*$; $N_\exists(L_1) = N_\exists(L_2) = a^* + a^* b a^*$; $N_\forall(L_1) = N_\forall(L_2) = \phi$.