Learning Locally Testable Even Linear Languages from Positive Data[⋆]

José M. Sempere and Pedro García

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Valencia, Spain
{jsempere,pgarcia}@dsic.upv.es

Abstract. Learning from positive data is a center goal in grammatical inference. Some language classes have been characterized in order to allow its learning from *text*. There are two different approaches to this topic: (i) reducing the new classes to well known ones, and (ii) designing new learning algorithms for the new classes. In this work we will use reduction techniques to define new classes of even linear languages which can be inferred from positive data only. We will center our attention to inferable classes based on local testability features. So, the learning processes for such classes of even linear languages can be performed by using algorithms for locally testable regular languages.
Keywords: *Learning from positive data, local testability, even linear languages.*

1 Introduction

Grammatical inference [AS83, Sa97] is the inductive learning approach where target concepts are represented through objects from Formal Language theory (typically finite automata, formal grammars and Turing machines-like acceptors or generators). Learning new language classes from different information protocols is one of the most important goals in grammatical inference. Along the time, there have been characterized many language classes that can be inferred from only positive data. Learning from positive data [An80] is one of the most attractive information protocols for grammatical inference. It excludes negative information and all the problems concerning its characterization in terms of noisy or incomplete data. Nevertheless, the main problem with learning from positive data is the *overgeneralization* effect to formulate hypotheses. Anyway, some subclasses of regular languages have been characterized and proved to be inferred from positive data. First, talking about regular languages, we can mention, among others, reversible languages [An82, Mä00], k-testable languages in the strict sense [GVO90], different subfamilies of testable languages including k-testable languages [Ru97], terminal distinguishable languages [Fe99, RN87a, RN87b] and generalizations of k-testable plus terminal distinguishable languages [Fe00a]. Talking about even linear languages, we can refer

[⋆] Work supported by the Spanish CICYT under contract TIC2000-1153.

to terminal distinguishable even linear languages [RN88, Fe99] and deterministic even linear languages which are an extension of regular reversible ones [KMT97]. A generalization to function distinguishable language learning can be found in [Fe00b]. Finally, for other language classes, there have been different approaches to the same direction [EST96, EST98, EST00].

In this work, we will explore how k-testability can be extended to act over even linear languages and, therefore, can be taken into account to design new learning algorithms based on old ones. Our success criterion will be Gold's *identification in the limit* [Go67] and the information protocol will be only positive strings of the target language. We will propose some front-end transformations in order to carry out the learning methods.

The structure of this work is as follows: First, we will give the definitions and results that we will use in order to define new language classes. We will introduce different classes of even linear languages whose definitions depend on how the reductions to regular languages are performed. For every language class we will propose an integrated protocol to efficiently learn target languages of every class.

2 Basic Concepts and Notation

The basic concepts of formal language theory that we will use can be consulted in [HU79]. First, Σ will denote an alphabet and $card(\Sigma)$ its cardinality. Σ^* will be the set of words over Σ. The empty string will be denoted by λ. The reversal of a string x will be denoted by x^r. The product of the strings x and y will be denoted by xy. The set of strings with length less than or equal to k (resp. less than k, equal to k) will be denoted by $\Sigma^{\leq k}$ (resp. $\Sigma^{<k}$, Σ^k). Given any string $x \in \Sigma^*$, its length will be denoted by $|x|$. A language L defined over Σ is any subset of Σ^*.

A grammar is a construct $G = (N, \Sigma, P, S)$ where, N and Σ are two disjoint alphabets of nonterminal and terminal symbols, P is a finite set of production rules and $S \in N$ is the axiom of the grammar. The relationship between strings of symbols that can eventually appear during a derivation process in the grammar will be denoted by $\overset{*}{\underset{G}{\Rightarrow}}$. So, the language obtained by the grammar G is the set $L(G) = \{x \in \Sigma^* : S \overset{*}{\underset{G}{\Rightarrow}} x\}$. The class of regular languages will be denoted by $\mathcal{REG}$ and the class of context-free languages will be denoted by $\mathcal{CF}$.

Even linear language class ($\mathcal{ELL}$) was first introduced by Amar and Putzolu in [AP64]. An even linear grammar $G = (N, \Sigma, P, S)$ is characterized through the forms in which every production appears in the grammar. There are two possible forms for every production in G:

1. $A \rightarrow xBy$ with $A, B \in N$ and $x, y \in \Sigma^k$ for any $k > 0$
2. $A \rightarrow x$ with $A \in N$ and $x \in \Sigma^*$

The following normal form of the productions can be applied to any even linear grammar to obtain an equivalent one:

1. $A \rightarrow aBb$ with $A, B \in N$ and $a, b \in \Sigma$
2. $A \rightarrow x$ with $A \in N$ and $x \in \Sigma \cup \{\lambda\}$

From now on we will work with grammars in the previously defined normal form. It is well known that $\mathcal{REG} \subset \mathcal{ELL} \subset \mathcal{CF}$. Amar and Putzolu [AP64] proved that any even linear language can be characterized by a finite index *quasi-congruence* over pair of strings (x, y). Later, Takada [Ta88] showed how any even linear language can be obtained from a regular control set together with a predefined universal grammar. He designed an inference protocol for even linear languages based on a reduction of the input sample to a different one which induces a regular language. Sempere and García [SG94] defined a transformation $\sigma : \Sigma^* \rightarrow (\Sigma\Sigma)^*\Sigma \cup \lambda$ in the following way

1. $(\forall a \in \Sigma \cup \{\lambda\})\ \sigma(a) = a$
2. $(\forall a, b \in \Sigma)\ (\forall x \in \Sigma^*)\ \sigma(axb) = [ab]\sigma(x)$

The transformation σ^{-1} was defined in a similar way:

1. $(\forall a \in \Sigma \cup \{\lambda\})\ \sigma^{-1}(a) = a$
2. $(\forall a, b \in \Sigma)\ (\forall x \in \Sigma^*)\ \sigma^{-1}([ab]x) = a\sigma^{-1}(x)b$

Sempere and García [SG94] showed that L is an even linear language iff $\sigma(L)$ is regular. So, in a similar manner as Takada did, they reduced the problem of learning even linear languages to the problem of learning regular ones.

3 Locally Testable Even Linear Languages

Locally testable languages were first introduced by McNaughton and Papert [McP71]. We will give some definitions to establish the characteristics of such languages.

Definition 1. *Let $k > 0$ and for every string $x \in \Sigma^*$ we define the k-testable vector of x as the tuple $v_k(x) = (i_k(x), t_k(x), f_k(x))$ where*

$$i_k(x) = \begin{cases} x & \text{if} \quad |x| < k \\ u : x = uv, |u| = k-1 & \text{if} \quad |x| \geq k \end{cases}$$

$$f_k(x) = \begin{cases} x & \text{if} \quad |x| < k \\ v : x = uv, |v| = k-1 & \text{if} \quad |x| \geq k \end{cases}$$

$$t_k(x) = \{v : x = uvw, u \in \Sigma^*, w \in \Sigma^*, |v| = k\}$$

Definition 2. *For every $k > 0$, we define the equivalence relation $\equiv_k \subseteq \Sigma^* \times \Sigma^*$ as follows:*

$$(\forall x, y \in \Sigma^*) \quad x \equiv_k y \Longleftrightarrow v_k(x) = v_k(y)$$

It has been proved that $\equiv_k$ is a finite index relation and $\equiv_{k+1}$ refines $\equiv_k$.

Definition 3. *We will say that $L \subseteq \Sigma^*$ is k-testable iff it is the union of some equivalence classes of $\equiv_k$. L is locally testable if it is k-testable for any $k > 0$.*

We will denote the family of k-testable languages by k-$\mathcal{TL}$ and the class of locally testable languages by $\mathcal{LT}$. It is proved that $\mathcal{LT} \subset \mathcal{REG}$

Another language class which is intimately related to $\mathcal{LT}$ is the class of locally testable languages in the strict sense [Ga88]. We will define it as follows:

Definition 4. *Let Σ be an alphabet and $Z_k = (\Sigma, I_k, F_k, T_k)$ where $I_k, F_k \subseteq \Sigma^{\leq k-1}$ and $T_k \subseteq \Sigma^k$. Then, L is a k-testable language in the strict sense if it is defined by the following expression*

$$L = (I_k \Sigma^*) \cap (\Sigma^* F_k) - (\Sigma^* T_k \Sigma^*)$$

We will say that L is locally testable in the strict sense if it is k-testable in the strict sense for any $k > 0$.

We will denote the family of k-testable languages in the strict sense by k-$\mathcal{LTSS}$ and the class of locally languages in the Strict Sense by $\mathcal{LTSS}$. It is proved that $\mathcal{LTSS} \subset \mathcal{REG}$. Furthermore, k-$\mathcal{LT}$ is the boolean closure of k-$\mathcal{LTSS}$. Observe that both classes, k-$\mathcal{LT}$ and k-$\mathcal{LTSS}$ are closed under reversal.

In order to work with even linear languages we must use σ transformation together with a morphism $g : (\Sigma\Sigma) \cup \Sigma \rightarrow \Delta$, where Δ is an alphabet such that $card(\Delta) \geq card(\Sigma)^2 + 2 \cdot card(\Sigma)$. So, if $\Sigma = \{a_1, a_2, \cdots, a_n\}$ and $\Delta = \{b_1, b_2, \cdots, b_m\}$ then g will be defined as follows

1. $(\forall a_i \in \Sigma)\ g(a_i) = b_i$
2. $(\forall a_i, a_j \in \Sigma)\ g([a_i a_j]) = b_{n+i \cdot n+j}$

Definition 5. *Let $L \in \mathcal{ELL}$. Then, L is a locally testable even linear language iff $g(\sigma(L)) \in \mathcal{LT}$. We will say that L is a locally testable even linear language in the strict sense iff $g(\sigma(L)) \in \mathcal{LTSS}$.*

We will denote the class of locally testable even linear languages as $\mathcal{LT}_{\mathcal{ELL}}$ and the class of locally testable even linear languages in the strict sense as $\mathcal{LTSS}_{\mathcal{ELL}}$. From the last definition, we can design a reduction technique to learn languages from $\mathcal{LT}_{\mathcal{ELL}}$ or $\mathcal{LTSS}_{\mathcal{ELL}}$ by applying learning algorithms for $\mathcal{LT}$ or $\mathcal{LTSS}$. Let us suppose that the learning algorithm for $\mathcal{LT}$ is $A_{\mathcal{LT}}$ as described in [Ru97], then the script for such a reduction is showed in Figure 1.

```
Input: A finite sample $R$
Method:
    $S = \emptyset$
    $\forall x \in R$
        $S = S \cup g(\sigma(x))$
    $\mathcal{H} = A_{\mathcal{LT}}(S)$
Output: $\sigma^{-1}(g^{-1}(\mathcal{H}))$
```

Figure 1. *A method to infer languages from* $\mathcal{LT}_{\mathcal{ELL}}$

The script to learn languages from $\mathcal{LTSS}_{\mathcal{ELL}}$ is similar to the previous one. In this case, we should use a method to infer languages from $\mathcal{LTSS}$ as in [GVO90]. Let us suppose that such a method is $A_{\mathcal{LTSS}}$, then the inferring method is showed in Figure 2

```
Input: A finite sample $R$
Method:
    $S = \emptyset$
    $\forall x \in R$
        $S = S \cup g(\sigma(x))$
    $\mathcal{H} = A_{\mathcal{LTSS}}(S)$
Output: $\sigma^{-1}(g^{-1}(\mathcal{H}))$
```

Figure 2. *A method to infer languages from* $\mathcal{LTSS}_{\mathcal{ELL}}$

Example 1. Let us suppose that sample R is defined by the set

$$R = \{10, 10001110, 111000111000, 11100001111000\}$$

then $\sigma(R)$ equals to the set

$$\{[10], [10][01][01][01], [10][10][10][01][01][01], [10][10][10][01][01][01][01]\}$$

If we apply the morphism g such that $g([10]) = a$ and $g([01]) = b$ then $g(\sigma(R)) = \{a, abbb, aaabbb, aaabbbb\}$. Now, by applying $A_{\mathcal{LTSS}}(S)$ to $g(\sigma(R))$, with $k = 2$, we obtain an hypothesis $\mathcal{H}$ such that $\sigma^{-1}(g^{-1}(\mathcal{H}))$ is the following even linear grammar: $S \to 1A0$; $A \to 1A0 \mid 0B1 \mid \lambda$; $B \to 1A0 \mid 0B1 \mid \lambda$.

A Characterization of Locally Testable Even Linear Grammars

Now, once we have efficiently solved the problem of learning locally testable even linear languages, we can take profit of the reductions we have applied in order to describe some features that makes an even linear grammar a locally testable one. Let us formalize such characteristics.

Property 1. Let $k > 0$ and let $G = (N, \Sigma, P, S)$ be an even linear grammar such that the following condition holds: $(\forall A \in N)$ if $S \underset{G}{\overset{*}{\Rightarrow}} x_1Ay_1$ and $S \underset{G}{\overset{*}{\Rightarrow}} x_2Ay_2$ then $\forall w \in \Sigma^*$ $g(\sigma(x_1wy_1)) \equiv_k g(\sigma(x_2wy_2))$. Then $L(G) \in k - \mathcal{LT}_{\mathcal{ELL}}$

Proof.

Let us suppose that the grammar G holds the condition that the property states. Then, obviously, the strings of the language $L(G)$ can be obtained by $\mathcal{O}(card(N))$ equivalence classes according to $\equiv_k$.

$\square$

A similar result can be obtained in order to characterize those even linear grammars that generate languages from $\mathcal{LTSS}_{ELL}$.

4 More on Locally Testable Even Linear Languages

We have showed some relationships between even linear and regular languages through the transformations σ and g. Now, we will change the front-end string interface in order to characterize new language classes inside $\mathcal{ELL}$ that benefits from local testability in a different manner as described before.

Half-splitting Strings

Take notice that, in every even linear language, all the strings can be separated by right and left sides with equal length and a (possibly λ) centered string. The sides of every string can be mapped to derivations in even linear grammars

So, we will define a new transformation `HalfSplit` over strings which will separate them in two halves with the same length. For every string $x \in \Sigma^*$, we will define $\texttt{HalfSplit}(x) = \{x_1, a, x_2^r\}$ with $x = x_1 a x_2$ and $|x_1| = |x_2|$ and $a \in \Sigma \cup \{\lambda\}$.

Now, we will extend the operation over any finite sample S.

$$\texttt{HalfSplit}(S) = \{L(S), M(S), R(S)\}$$

where
$L(S) = \{x_1 \in \Sigma^* : \exists x \in S, \texttt{HalfSplit}(x) = \{x_1, a, x_2\}\}$,
$R(S) = \{(x_2)^r \in \Sigma^* : \exists x \in S, \texttt{HalfSplit}(x) = \{x_1, a, x_2\}\}$ and
$M(S) = \{a \in \Sigma \cup \{\lambda\} : \exists x \in S, \texttt{HalfSplit}(x) = \{x_1, a, x_2\}\}$.

Synchronized Vs. Non Synchronized Half-splitting

With the help of `HalfSplit`, we can obtain hypotheses called *constructors* which will guide the learning process to obtain the final hypothesis. In Figure 3 we show the scripting method to obtain constructors from `HalfSplit` operation by using learning algorithms for $\mathcal{LT}$ language class.

```
Input: A finite sample $S$
Method:
        $T_L = L(S) \in \texttt{HalfSplit}(S)$
        $T_R = R(S) \in \texttt{HalfSplit}(S)$
        $T_M = M(S) \in \texttt{HalfSplit}(S)$
        $\mathcal{H}_L = A_{\mathcal{LT}}(T_L)$
        $\mathcal{H}_R = A_{\mathcal{LT}}(T_R)$
        $\mathcal{H}_M = T_M$
Output: $\{\mathcal{H}_L, \mathcal{H}_M, \mathcal{H}_R\}$
```

Figure 3. *A method to infer constructors for synchronized and non synchronized half-splitting*

Example 2. Let $S = \{abbaabab, abbabaababab, abbabbabbaababababab\}$, then $\texttt{HalfSplit}(S) = \{L(S), R, (S), M(S)\}$, where
$L(S) = \{abba, abbaba, abbabbabba\}$,
$R(S) = \{baba, bababa, bababababa\}$ and
$M(S) = \{\lambda\}$.

Now, by applying learning algorithm $A_{\mathcal{LT}}$, with $k = 2$ we obtain constructors $\mathcal{H}_L$ and $\mathcal{H}_R$ with $\mathcal{H}_M = \{\lambda\}$ where the equivalence classes constructed by the learning algorithm are the following
$\mathcal{H}_L = \{[\lambda]_{\equiv_2}, [a]_{\equiv_2}, [ab]_{\equiv_2}, [abb]_{\equiv_2}, [abba]_{\equiv_2}, [abbab]_{\equiv_2}\}$
$\mathcal{H}_R = \{[\lambda]_{\equiv_2}, [b]_{\equiv_2}, [ba]_{\equiv_2}, [bab]_{\equiv_2}, [baba]_{\equiv_2}\}$

After obtaining constructors from a finite sample then we can define different even linear language subclasses. Our motivation is to combine different equivalence classes from the relation $\equiv_k$ that has been obtained during the learning process. It is obvious that the sets that we have obtained from `HalfSplit` form an input sample for a hidden locally testable regular language.

We will define *synchronicity* among the equivalence classes that we have obtained apart from $L(S)$ and $R(S)$. Synchronicity means that, if we take a string from the input sample $x = x_1ax_2 \in S$, then the equivalence class obtained for x_1 must be linked to the equivalence class obtained for $(x_2)^r$. So, we can define the relation $\sim_k$ between strings as follows:$(\forall x, y \in \Sigma^*)$ with $|x_1| = |x_2|, |y_1| = |y_2|$, $x = x_1ax_2$, $y = y_1ay_2$ and $a \in \Sigma \cup \{\lambda\}$

$$x \sim_k y \iff v_k(x_1) = v_k(y_1) \land v_k(x_2) = v_k(y_2)$$

Definition 6. *We will say that $L \subseteq \Sigma^*$ is k-testable synchronized half-split iff it is the union of some equivalence classes of $\sim_k$. L is locally testable synchronized half-split if it is k-testable synchronized half-split for any $k > 0$.*

We will denote the language class of *locally testable synchronized half-split* even linear languages by $\mathcal{LTSHS}_{ELL}$. Then, we can infer k *locally testable synchronized half-split* even linear languages from constructors by applying the script showed in Figure 4.

```
Input: A finite sample $S$ and constructors $\mathcal{H}_L, \mathcal{H}_M$ and $\mathcal{H}_R$
Method:
        Enumerate $\mathcal{H}_L$ and $\mathcal{H}_R$
        /* $card(\mathcal{H}_L) = n$ and $card(\mathcal{H}_R) = m$ */
        $N = \{A_{ij} : 0 \leq i \leq n, 0 \leq j \leq m\}$
        $S = A_{00}$
        $\forall x \in S$
            $\forall u, v, w \in \Sigma^*\ a, b \in \Sigma : x = uawbv$ with $|u| = |v|$
            /* $[u]_{\equiv_k}$ is the $i$th equivalence class in $\mathcal{H}_L$ */
            /* $[v^r]_{\equiv_k}$ is the $j$th equivalence class in $\mathcal{H}_R$ */
            /* $[ua]_{\equiv_k}$ is the $k$th equivalence class in $\mathcal{H}_L$ */
            /* $[v^r b]_{\equiv_k}$ is the $l$th equivalence class in $\mathcal{H}_R$ */
              $A_{ij} \rightarrow aA_{kl}b \in P$
              if $w \in \mathcal{H}_M$ then $A_{kl} \rightarrow w \in P$
Output: $G = (N, \Sigma, P, S)$
```

Figure 4. *A method to infer languages from $\mathcal{LTSHS}_{ELL}$.*

The identification in the limit of the class $\mathcal{LTSHS}_{ELL}$ can be easily deduced from the convergence of hypotheses $\mathcal{H}_L$ and $\mathcal{H}_R$ by using the algorithm $A_{\mathcal{LT}}$. Observe that the method proposed in Figure 4 is *conservative*, that is, all the classes of relation $\equiv_k$ are combined iff they have any representant in the input sample.

Example 3. Let us take the input sample and constructors of Example 2. Then the hypothesis $\mathcal{H}$ obtained by the method of Figure 4 is the following grammar

$$\begin{array}{ll} A_{00} \rightarrow aA_{11}b & A_{44} \rightarrow bA_{53}b \mid \lambda \\ A_{11} \rightarrow bA_{22}a & A_{53} \rightarrow aA_{44}a \mid bA_{54}a \\ A_{22} \rightarrow bA_{33}b & A_{54} \rightarrow bA_{53}b \mid aA_{43}b \\ A_{33} \rightarrow aA_{44}a & A_{43} \rightarrow bA_{54}a \end{array}$$

The following result characterizes the grammars that generate the languages of $\mathcal{LTSHS}_{ELL}$.

Property 2. Let $k > 0$ and let $G = (N, \Sigma, P, S)$ be an even linear grammar such that the following condition holds: $(\forall A \in N)$ if $S \underset{G}{\overset{*}{\Rightarrow}} x_1Ay_1$ then $S \underset{G}{\overset{*}{\Rightarrow}} x_2Ay_2$ iff $x_1 \equiv_k x_2$ and $y_1 \equiv_k y_2$. Then, $L(G) \in k - \mathcal{LTSHS}_{\mathcal{ELL}}$

Proof.

Let us suppose that the grammar G holds the condition that the property states. Then, obviously, if we take two different strings of $L(G)$ which can be obtained during a derivation process in the grammar as follows

$$S \underset{G}{\overset{*}{\Rightarrow}} x_1Ay_1 \underset{G}{\Rightarrow} x_1ay_1$$
$$S \underset{G}{\overset{*}{\Rightarrow}} x_2Ay_2 \underset{G}{\Rightarrow} x_2ay_2$$

it can be easily proved that, according to the last derivations, $x_1 \equiv_k x_2$ and $y_1 \equiv_k y_2$. The strings of the language $L(G)$ can be obtained by at most $\mathcal{O}(card(N))$ equivalence classes according to $\sim_k$. So, $L(G) \in k - \mathcal{LTSHS}_{\mathcal{ELL}}$ □

The definition of *non synchronized half-split* even linear languages is quite simple. Here, we do not force every equivalence class to be linked with its correspondent (right or left) equivalence class. We will define the relation $\approx_k$ between strings as follows: $(\forall x, y \in \Sigma^*)$ with $|x_1| = |x_2|, |y_1| = |y_2|$, $x = x_1 a x_2$, $y = y_1 a y_2$ and $a \in \Sigma \cup \{\lambda\}$

$$x \approx_k y \Longleftrightarrow v_k(x_1) = v_k(y_1) \vee v_k(x_2) = v_k(y_2)$$

Definition 7. *We will say that $L \subseteq \Sigma^*$ is k-testable non synchronized half-split iff it is the union of some equivalence classes of $\approx_k$. L is locally testable non synchronized half-split if it is k-testable non synchronized half-split for any $k > 0$.*

The class of *locally testable non synchronized half-split* even linear languages will be denoted by $\mathcal{LTNSHS}_{ELL}$. Then we can infer *k locally testable non synchronized half-split* even linear languages from constructors by applying the script showed in Figure 5.

```
Input: A finite sample $S$ and constructors $\mathcal{H}_L, \mathcal{H}_M$ and $\mathcal{H}_R$
Method:
          Enumerate $\mathcal{H}_L$ and $\mathcal{H}_R$
          /* $card(\mathcal{H}_L) = n$ and $card(\mathcal{H}_R) = m$ */
          $N = \{A_{iL} : 0 \leq i \leq n\} \cup \{A_{jR} : 0 \leq j \leq m\} \cup \{S\}$
          $\forall x \in S$
              $\forall u \in \Sigma^*\ a \in \Sigma : x = uawv$ with $|ua| \leq |v|$
              /* $[u]_{\equiv_k}$ is the $i$th equivalence class in $\mathcal{H}_L$ */
              /* $[ua]_{\equiv_k}$ is the $k$th equivalence class in $\mathcal{H}_L$ */
                  $\forall b \in \Sigma$
                      $A_{iL} \to aA_{kL}b \in P$
                      if $w \in \mathcal{H}_M$ then $A_{kL} \to w \in P$
          $\forall x \in S$
              $\forall v \in \Sigma^*\ b \in \Sigma : x = uwbv$ with $|bv| \leq |u|$
              /* $[v^r]_{\equiv_k}$ is the $i$th equivalence class in $\mathcal{H}_R$ */
              /* $[v^r b]_{\equiv_k}$ is the $k$th equivalence class in $\mathcal{H}_R$ */
                  $\forall a \in \Sigma$
                      $A_{iR} \to aA_{kR}b \in P$
                      if $w \in \mathcal{H}_M$ then $A_{kR} \to w \in P$
          if $A_{0L} \to w$ then $S \to w \in P$
          if $A_{0R} \to w$ then $S \to w \in P$
Output: $G = (N, \Sigma, P, S)$
```

Figure 5. *A method to infer languages from $\mathcal{LTNSHS}_{ELL}$.*

Again, the identification in the limit of the class $\mathcal{LTNSHS}_{ELL}$ can be easily deduced from the convergence of hypotheses $\mathcal{H}_L$ and $\mathcal{H}_R$ by using the algorithm $A_{\mathcal{LT}}$.

Example 4. Let us take the input sample and constructors of Example 2. Then the hypothesis $\mathcal{H}$ obtained by the method of Figure 5 is the following grammar

$$\begin{array}{ll}
S \to aA_{1L}a \mid aA_{1L}b \mid aA_{1R}b \mid bA_{1R}b & \\
A_{0L} \to aA_{1L}a \mid aA_{1L}b & A_{0R} \to aA_{1R}b \mid bA_{1R}b \\
A_{1L} \to bA_{2L}a \mid bA_{2L}b & A_{1R} \to aA_{2R}a \mid bA_{2R}a \\
A_{2L} \to bA_{3L}a \mid bA_{3L}b & A_{2R} \to aA_{3R}b \mid bA_{3R}b \\
A_{3L} \to aA_{4L}a \mid aA_{4L}b & A_{3R} \to aA_{4R}a \mid bA_{4R}a \\
A_{4L} \to \lambda \mid bA_{5L}a \mid bA_{5L}b & A_{4R} \to \lambda \mid aA_{3R}b \mid bA_{3R}b \\
A_{5L} \to aA_{4L}a \mid aA_{4L}b \mid bA_{5L}a \mid bA_{5L}b &
\end{array}$$

The following result characterizes the grammars that generate languages of the class $\mathcal{LTNSHS}_{ELL}$.

Property 3. Let $k > 0$ and let $G = (N, \Sigma, P, S)$ be an even linear grammar such that the following condition holds: $(\forall A \in N)$ if $S \overset{*}{\underset{G}{\Rightarrow}} x_1Ay_1$ then $S \overset{*}{\underset{G}{\Rightarrow}} x_2Ay_2$ iff $x_1 \equiv_k x_2$ or $y_1 \equiv_k y_2$. Then, $L(G) \in k - \mathcal{LTNSHS}_{\mathcal{ELL}}$

Proof. Let us suppose that the grammar G holds the condition that the property states. Then, obviously, if we take two different strings of $L(G)$ which can be obtained during a derivation process in the grammar as follows

$$S \overset{*}{\underset{G}{\Rightarrow}} x_1Ay_1 \underset{G}{\Rightarrow} x_1ay_1$$
$$S \overset{*}{\underset{G}{\Rightarrow}} x_2Ay_2 \underset{G}{\Rightarrow} x_2ay_2$$

then, it can be easily proved that, according to the last derivations, $x_1 \equiv_k x_2$ or $y_1 \equiv_k y_2$. The strings of the language $L(G)$ can be obtained by $\mathcal{O}(card(N))$ equivalence classes according to $\approx_k$. So, $L(G) \in k - \mathcal{LTNSHS}_{\mathcal{ELL}}$ □

From properties 2 and 3, we can obtain the following result

Corollary 1. $\mathcal{LTNSHS}_{ELL} \subseteq \mathcal{LTSHS}_{ELL}$

5 Conclusions and Future Works

We have introduced different transformations over input strings of even linear languages in order to characterize several language subclasses that can be inferred from positive sample. The learning algorithms that we have used are those concerning locally testable languages. Obviously, by selecting different learning algorithms for regular languages we can define different subclasses of even linear languages. Under this approach, there have been such reductions as those showed in [Fe99, KMT97, RN88]. We will explore the relationship between the classes of languages defined in such works and our approach by using σ and g.

On the other hand, the reductions that we have used goes in the same direction as the definition of *function distinguishable languages* [Fe00b]. We will explore the relationship between function distinguishability and the reductions that we have proposed.

Finally, we have selected only two learning algorithm concerning local testability. The selection of different learning algorithms for local testability as those showed in [Ru97] (i.e right and left locally testable, piecewise locally testable, etc.) will define new inferrable language classes inside even linear language class. We will explore such direction in order to obtain a complete catalog of inferrable locally testable even linear languages.

Acknowledgements

José M. Sempere is grateful to Erkki Mäkinen for nice comments and corrections made to this work.

References

[AP64] V. Amar and G. Putzolu. *On a Family of Linear Grammars.* Information and Control 7, pp 283-291. 1964.

[An80] D. Angluin. *Inductive Inference of Formal Languages from Positive Data.* Information and Control 45, pp 117-135. 1980.

[An82] D. Angluin. *Inference of Reversible Languages.* Journal of the Association for Computing Machinery. Vol 29 No 3, pp 741-765. July 1982.

[AS83] D. Angluin and C. Smith. *Inductive Inference : Theory and Methods.* Computing Surveys, vol. 15. No. 3, pp 237-269. 1983.

[EST96] J.D. Emerald, K.G. Subramanian and D.G. Thomas. *Learning code regular and code linear languages.* Proceedings of the Third International Colloquium on Grammatical Inference ICGI-96. (L. Miclet, C. de la Higuera, eds). LNAI Vol. 1147, pp 211-221. Springer. 1996.

[EST98] J.D. Emerald, K.G. Subramanian and D.G. Thomas. *Learning a subclass of context-free Languages.* Proceedings of the 4th International Colloquium ICGI-98. (V. Honavar, G. Slutzki, eds). LNAI Vol. 1433, pp 223-243. 1998.

[EST00] J.D. Emerald, K.G. Subramanian and D.G. Thomas. *Inferring Subclasses of contextual languages.* Proceedings of the 5th International Colloquium ICGI 2000. (A. Oliveira, ed). LNAI Vol. 1891, pp 65-74. 2000.

[Fe99] H. Fernau. *Learning of terminal distinguishable languages.* Technical Report WSI–99–23. Universität Tübingen (Germany), Wilhelm-Schickard-Institut für Informatik, 1999.

[Fe00a] H. Fernau. *k-gram extensions of terminal distinguishable languages.* Proceedings of the International Conference on Pattern Recognition ICPR 2000, Vol. 2 pp 125-128. IEEE Press. 2000.

[Fe00b] H. Fernau. *Identification of function distinguishable languages.* Proceedings of the 11th International Conference on Algorithmic Learning Theory ALT 2000. (H. Arimura, S. Jain, A. Sharma, eds.). LNCS Vol. 1968 pp 116-130. Springer-Verlag 2000.

[Ga88] P. García. *Explorabilidad Local en Inferencia Inductiva de Lenguajes Regulares y Aplicaciones.* Ph.D. Thesis. Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia. 1988.

[GVO90] P. García, E. Vidal and J. Oncina. *Learning Locally Testable Languages in the Strict Sense.* Proceedings of the First International Workshop on Algorithmic Learning Theory. pp 325-338. 1990.

[Go67] M. Gold. *Language Identification in the Limit.* Information and Control 10, pp 447-474. 1967.

[HU79] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley Publishing Co. 1979.

[KMT97] T. Koshiba, E. Mäkinen, Y. Takada. *Learning deterministic even linear languages from positive examples.* Theoretical Computer Science 185, pp 63-97. 1997.

[Mä00] E. Mäkinen. *On inferring zero-reversible languages.* Acta Cybernetica 14, pp 479-484. 2000.

[McP71] R. McNaughton and S. Papert. *Counter-free automata.* MIT Press. 1971.

[RN87a] V. Radhakrishnan. *Grammatical Inference from Positive Data: An Efective Integrated Approach.* PhD. Thesis. Department of Computer Science and Engineering. IIT, Bombay. 1987.

[RN87b] V. Radhakrishnan and G. Nagaraja. *Inference of Regular Grammars via Skeletons.* IEEE Trans. on Systems, Man and Cybernetics, 17, No. 6 pp 982-992. 1987.

[RN88] V. Radhakrishnan and G. Nagaraja. *Inference of Even Linear Languages and Its Application to Picture Description Languages.* Pattern Recognition, 21, No. 1. pp 55-62. 1988.

[Ru97] J. Ruiz. *Familias de Lenguajes Explorables : Inferencia Inductiva y Caracterización Algebraica.* Ph.D. Thesis. Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia. 1997.

[Sa97] Y. Sakakibara. *Recent advances of grammatical inference.* Theoretical Computer Science 185, pp 15-45. 1997.

[SG94] J.M. Sempere and P. García. *A Characterization of Even Linear Languages ans its Application to the Learning Problem.* Proceedings of ICGI'94 (R. Carrasco and J. Oncina, eds.). LNAI Vol. 862, pp 38-44. Springer-Verlag. 1994.

[Ta88] Y. Takada. *Grammatical Inference of Even Linear Languages based on Control Sets.* Information Processing Letters 28, No. 4, pp 193-199. 1988.