Identifying Left-Right Deterministic Linear Languages

Jorge Calera-Rubio* and Jose Oncina**

D. Lenguajes y Sistemas Informaticos, Univ. de Alicante, E-03080 Alicante, Spain
{calera,oncina}@dlsi.ua.es

Abstract. Recently an algorithm to identify in the limit with polynomial time and data Left Deterministic Linear Languages (Left DLL) and, consequently Right DLL, was proposed. In this paper we show that the class of the Left-Right DLLformed by the union of both classes is also identifiable. To do that, we introduce the notion of n-negative characteristic sample, that is a sample that forces an inference algorithm to output an hypothesis of size bigger than n when strings from a non identifiable language are provided.

An important subclass of the linear languages is that of the *Left Deterministic Linear Languages* (Left DLL or LDLL) [1]. They are generated by *Left Deterministic Linear Grammars* such that the next rule to use in the parsing of a string is determined by observing the leftmost terminal in the unparsed part of the string. Regular languages, $\{a^n b^n \mid n \geq 0\}$ and $\{a^m b^n c^n \mid n, m \geq 0\}$ are some examples of languages in the class, while $\{a^n b^n c^m \mid n, m \geq 0\}$ is not.

The class formed by the reversals of the languages in LDLL is called the *Right* DLL (RDLL) and can be defined in a symmetrical way in which LDLL are defined. The regular languages, $\{a^n b^n \mid n \geq 0\}$ and $\{a^n b^n c^m \mid n, m \geq 0\}$ are examples of languages in the class.

Recently [1] it was showed that the class of the LDLL can be identified in the limit from polynomial time and data. Obviously, the class of the RDLL can be identified just by reversing the strings before to introduce them on the LDLL inference algorithm.

In this paper we propose a new class of languages, the Left-Right DLL (LRDLL), formed by the union of LDLL and RDLL. We show that this class can be identified in the limit from polynomial time and data.

In ref. [1] was defined an algorithm LDLGA$(\cdot)$ that identifies the LDLG. Reminding that RDLL $=$ LDLL^R. Then, it is easy to build an algorithm RDLGA$(\cdot)$ for RDLG such that RDLGA$(S) = ($LDLGA$(S^R))^R$.

Now, for LRDLL let us define an algorithm (LRDLGA) that given a sample S, computes LDLGA(S) and RDLGA(S), and returns the hypothesis grammar with a lower number of non terminals.

* This work was supported by the Spanish CICyT trough project TIC2003-08496-C04.

** This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

If the target language is in LDLL − RDLL and the sample is big enough, LDLGA will provide the canonical LDLG for the language, but the RDLGA is going to produce bigger and bigger grammars as the sample grows. Then there has to exist a point when the correct hypothesis will be output.

The case when the language is in RDLL − LDLL is similar. When the target is in both classes, LRDLGA outputs the smaller of both representations.

Now, in order to show the identification in the limit from polynomial time and data, we have to show the existence of a polynomial characteristic set. The idea is to find a sample such that, if the language is not in the class it forces the algorithm to output an hypothesis with size bigger that a given parameter. Let us formalize this idea:

Definition 1 (n-negative characteristic sample). *Let $\phi(\cdot)$ an inference algorithm that identifies in the limit the class of languages $\mathcal{L}$ in terms of the class of grammars $\mathcal{G}$, let a language $L \notin \mathcal{L}$, $C = (C_+, C_-)$ is a n-*negative characteristic sample* of L (n-NCS) for ϕ if $\forall I = (I_+, I_-) : C_+ \subseteq I_+, C_- \subseteq I_-, \|\phi(I)\| \geq n$.*

In our case, we are going to use the number of non terminals as the size of a grammar. Then, if we can show that for every language in LDLL − RDLL (or RDLL − LDLL) with n non terminals we can find a polynomial size $(n+1)$-NCS for RDLGA (LDLGA), the union of the characteristic sample for LDLGA (RDLGA) of the language with the $(n+1)$-NCS will be a polynomial size characteristic sample for the LRDLGA.

Let us show that this polynomial size n-NCS exists. Following the notation in [1] it is not hard to prove the next proposition and theorem:

Proposition 1. *Let $L \notin$ LDLL and let $n \in \mathbb{N}$. As $L \notin$ LDLL we know that $|\operatorname{Sp}_L|$ is infinite, let Sp_{L_n} be the n smaller $x \in \operatorname{Sp}_L$. Let $S = (S_+, S_-)$ be a sample of L, S is an n-negative characteristic sample (n-NCS) for LRDLGA if:*

1. $\forall x \in \operatorname{Sp}_{L_n} \forall a \in \Sigma : xa \in \operatorname{Pr}(L) \Rightarrow \exists xaw \in S_+$
2. $\forall x \in \operatorname{Sp}_{L_n} \forall a \in \Sigma : \operatorname{CSF}_L(xa) \neq \emptyset \Rightarrow \operatorname{tail}_{S_+}(xa) = \operatorname{tail}_L(xa)$
3. $\forall x, y \in \operatorname{Sp}_{L_n} \forall a \in \Sigma : \operatorname{CSF}_L(xa) \neq \emptyset \wedge xa \not\equiv_L y \Rightarrow$
   $\exists v : xav \operatorname{tail}_L(xa) \in S_+ \wedge yv \operatorname{tail}_L(y) \in S_- \ \vee$
   $\exists v : yv \operatorname{tail}_L(y) \in S_+ \wedge xav \operatorname{tail}_L(xa) \in S_-$
4. $\forall x \in \operatorname{Sp}_{L_n} : x \operatorname{tail}_L(x) \in L \Rightarrow x \operatorname{tail}_L(x) \in S_+$

Theorem 1. *For any $L \notin$ LDLL there is an n-negative characteristic sample of polynomial size.*

Acknowledgement: The authors thank Colin de la Higuera for fruitful discussions on the subject.

References

1. de la Higuera, C., Oncina, J.: Learning deterministic linear languages. In: Computational Learning Theory, COLT 02. Number 2375 in Lecture Notes in Artificial Intelligence, Springer Verlag (2002) 185–200