ON INFERRING LINEAR SINGLE-TREE LANGUAGES

ERKKI MÄKINEN

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF TAMPERE

REPORT A-1998-11

UNIVERSITY OF TAMPERE
DEPARTMENT OF COMPUTER SCIENCE
SERIES OF PUBLICATIONS A
A-1998-11, NOVEMBER 1998

ON INFERRING LINEAR SINGLE-TREE LANGUAGES

ERKKI MÄKINEN

University of Tampere
Department of Computer Science
P.O.Box 607
FIN-33101 Tampere, Finland

ISBN 951-44-4463-9
ISSN 0783-6910

On inferring linear single-tree languages

Erkki Mäkinen [1]

Department of Computer Science, University of Tampere, P.O. Box 607,
FIN-33101 Tampere, Finland

Abstract

We consider the inferability of linear single-tree languages. We are able to show that if the terminal productions of the corresponding grammars obey a simple restriction, then the languages generated are inferable from positive samples only. Moreover, we solve an open problem posed by Greibach, Shi, and Simonson concerning the unambiguity of certain ultralinear STG's.

Keywords: context-free language, linear single-tree language, grammatical inference, identification in the limit.

1 Introduction

A context-free grammar is a *single-tree grammar* (STG) if every nonterminal symbol has at most one production whose right hand side contains nonterminal symbols [3]. In a *linear grammar* the right hand side of every production contains at most one nonterminal symbol. In this paper we consider linear STG's and the languages generated by these grammars, i.e. *linear single tree languages* (LSTL's). Greibach et al. [3] have shown that LSTL's are always deterministic and bounded. We show here that if the terminal productions of a linear STG obey a simple additional restriction then the LSTL generated is inferable from positive samples only. Moreover, we solve an open problem posed by Greibach, Shi, and Simonson concerning the unambiguity of certain ultralinear STG's.

We assume a familiarity with the basics of formal language theory and grammatical inference as given e.g. in [4] and [1], respectively. As inference criterion we use "identification in the limit" [2,1]. If not otherwise stated, we follow the

[1] E-mail: em@cs.uta.fi. Work supported by the Academy of Finland (Project 35025).

notations and definitions of [4]. The empty word is denoted by λ, and the length of a word α by $lg(\alpha)$.

A production of the form $A \rightarrow w$, where w is a terminal string, is said to be *terminating.* A *continuing* production has the form $A \rightarrow vBw$, where v and w are terminal strings and B is a nonterminal. A production with A on the left hand side is said to an $A - production$.

2 The inference algorithm

Regular languages cannot be inferred from positive samples only [2]. This negative result has initiated a search for subclasses of regular languages having the desired inference property. Moreover, several non-regular classes of languages inferable from positive samples only have been found, see e.g. [5,6].

We make an additional restriction to the form of the terminal productions in linear STG's: we insist that in each terminal production $A \rightarrow w$, the right hand side w consists of a unique terminal symbol. Hence, if $A \rightarrow a$ and $B \rightarrow a$ are productions, then $A = B$. This restriction does not affect to linear STGs' capacity to produce non-regular languages.

Consider a linear STG G and two words w_1 and w_2 in $L(G)$ with $lg(w_1) \geq lg(w_2)$. The derivations producing w_1 and w_2 have the forms

$$S \Rightarrow \alpha_1 \Rightarrow \alpha_2 \ldots \Rightarrow \alpha_m \Rightarrow \alpha_{m+1} \Rightarrow \ldots \Rightarrow \alpha_n = w_1$$

$$S \Rightarrow \beta_1 \Rightarrow \beta_2 \ldots \Rightarrow \beta_m = w_2.$$

Because G is a STG, we have $\alpha_i = \beta_i$, for $i = 1, \ldots, m-1$.

Suppose that the nonterminal in α_{m-1} and β_{m-1} is A. Then we have

$$w_1 = w'v_1w''$$

and

$$w_2 = w'v_2w'',$$

where $A \Rightarrow^+ v_1$ is a derivation and $A \rightarrow v_2$ is a production. We say that (w', w'') is the *longest common pair* of w_1 and w_2.

Consider a set Q of words of words of length two or more from a known LSTL. The longest common pair of Q is determined by the two shortest words (ties are broken arbitrarily) in Q as above. The longest common pair of a set Q is denoted by $lcp(Q)$. Since the words considered are from a LSTL, $lcp(Q)$ consists of a common prefix and a common suffix of all the words in Q.

Given a set Q of sample words from an unknown LSTL, we must have a method for "parsing" the sample words, i.e. for finding out the longest common pairs.

Suppose all the words in Q are of length two or more, and let w_1 and w_2 be the two shortest words in Q with lengths p and q, $p \leq q$, respectively.

In the worst case, there are p possibilities to locate the symbol produced by a terminal production in the shortest word w_1. In the second step, the found longest common pair is erased from the words in Q, and there are at most $q-p-2$ possibilities to locate the symbol produced by a terminal production, and so on. It follows that it is always possible to find the longest common pairs for all sets of words obtained by repeatedly erasing the longest common pairs found. The longest common pairs so found for an unknown language are not necessarily unique. (Consider e.g. a set of words over an unary alphabet.) However, any parsing of the set of samples makes it possible to construct a linear STG as in our algorithm below. In what follows, we use the notation $lcp(Q)$ also for longest common pairs obtained by the above "parsing" method for a set Q from an unknown language.

Each word in Q is produced by a derivation starting with the only S-production. (S is the start symbol.) In our inference algorithm we conjecture that this production has the form $S \rightarrow x_1A_1y_1$, where $lcp(Q) = (x_1, y_1)$. Next, we erase the prefix x_1 and the suffix y_1 from the words in Q and obtain a new set Q' of words (of length two or more). The only continuing A_1-production has the form $A_1 \rightarrow x_2A_2y_2$, where $lcp(Q') = (x_2, y_2)$, and so on.

If the original set of samples contains words of length one, say the word a, we take the terminating production $S \rightarrow a$ to the resulting grammar. Similarly, in any step of the algorithm, a word of length one, say b, in a Q-set implies a terminating production of the form $A_i \rightarrow b$.

Since we suppose that terminating productions are unique, finding a terminating production may cause a merging process. As an example, consider a sample $\{aacb, aadcdb, aadefgdb\}$. We first obtain the productions $S \rightarrow aaA_1b$ and $A_1 \rightarrow c$. In the next step, we obtain the productions $A_1 \rightarrow dA_2d$ and $A_2 \rightarrow c$. Since we have terminating productions $A_1 \rightarrow c$ and $A_2 \rightarrow c$, we must have $A_1 = A_2$.

Erasing the prefixes and suffixes from the words in Q-sets may eventually lead to a situation where there is exactly one word (of length two or more) left in the set. In the above example, this final word (after erasing the pairs (aa, b) and (d, d)) is efg. In order to avoid erroneous merging, our algorithm conjectures the production $A_1 \rightarrow efg$. It follows that some of the conjectures outputted by our algorithm are not necessarily linear STG's in the strict sense defined in this paper. This, however, does not effect to the correctness of the algorithm. As soon as there are enough "short" samples, the algorithm finds out the correct structure of the grammar, and it does not matter what we do to the longest sample. Notice that no problem appears when the longest sample is not unique. Two samples of equal length always determines the correct ending of the corresponding derivations in linear STG's, since they differ on a single symbol only.

Consider now the situation where a new sample word w is received. If the length of w is "new" in Q (w is the first word in the sample of length $lg(w)$ and $lg(w) > 1$), it will refine the conjecture, since one of the longest common pairs is changed.

The algorithm can now be given as follows.

Algorithm 1 (LST)

```
Input: A set $Q$ of sample words, $| Q |> 1$.
Output: A linear STG $G = (V, \Sigma, P, S)$.
begin
  $i := 0$;
  $P := \emptyset$;
  parse $Q$ to a obtain a sequence of longest common pairs;
  while $| Q |> 1$ begin
    take $A_i \rightarrow x_i A_{i+1} y_i$, where $lcp(Q) = (x_i, y_i)$, to $P$;
    erase the prefix $x_i$ and the suffix $y_i$ from the words in $Q$;
    remove the words of length one from $Q$;
    if $a$ is removed from $Q$
      then take $A_i \rightarrow a$ to $P$;
    if $A_i \rightarrow a$ and $A_j \rightarrow a$, $j < i$, are in $P$
      then merge $A_i$ and $A_j$;
    $i := i + 1$;
  end; { while }
  if $w$ is the only word in $Q$
    then take $A_{i-1} \rightarrow w$ to $P$;
  the set of terminals $V \setminus \Sigma$ and the set of terminals $\Sigma$ consist of
  those symbols appearing in $P$;
  $S = A_0$;
end; { LST }
```

Let $G = (V, \Sigma, P, S)$ be a linear SGT having n nonterminals. If the input sample Q contains the subset

$$Q_s = \{w \mid S \Rightarrow^k w, k < n^2\}$$

(where k stands for the number of derivation steps applied), then different sequences of longest common pairs are possible only when they imply the same conjecture. Moreover, $Q_s \subseteq Q$ implies that all the recursive subderivations are found, and avoiding merges with the longest samples does not effect to the outputted conjecture. Hence, we have the following theorem.

Theorem 1 *LSTL's (as defined in this paper) are inferable from positive samples only.*

3 On a problem posed by Greibach, Shi, and Simonson

A *k-ultralinear grammar* is one where every sentential form contains at most k nonterminals. A *k-linear languages* are those generated by k-linear grammars. Greibach et al. [3] have proved that there is a 4-ultralinear STG that generates an inherently ambiguous language. Moreover, they conjectured that all 3-ultralinear STG's generate unambiguous languages. We are able to show that this conjecture does not hold true. Namely, consider the following 3-ultralinear STG:

$$S \rightarrow ABC$$
$$A \rightarrow aAb \mid \lambda$$
$$B \rightarrow aB \mid \lambda$$
$$C \rightarrow bCa \mid \lambda.$$

The language generated is $\{a^k b^k a^m b^n a^n \mid k, m, n \geq 0\}$. The words of the form $a^p b^p a^p$, $p > 0$, have more than one derivation, i.e. the grammar is ambiguous. This example refutes the conjecture of Greibach et al. [3].

References

[1] D. Angluin and C.H. Smith, Inductive inference: theory and methods. *ACM Comput. Surv.* **15** (1983), 237–269.

[2] E.M. Gold, Language identification in the limit. *Inform. Contr.* **10** (1967), 447–474.

[3] S. Greibach, W. Shi, and S. Simonson, Single tree grammars. In: J. Ullman (ed.), *Theoretical Studies in Computer Science*, Academic Press, 1992, 73–99.

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

[5] T. Koshiba, E. Mäkinen, and Y. Takada, Learning deterministic even linear languages from positive examples. *Theoret. Comput. Sci.* **185** (1997), 63–79.

[6] T. Yokomori, Polynomial time learning of very simple grammars from positive data. *Proc. 4th Workshop on Computational Learning Theory* (1991), 213–227.