INFERRING FINITE TRANSDUCERS

ERKKI MÄKINEN

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF TAMPERE

REPORT A-1999-3

UNIVERSITY OF TAMPERE
DEPARTMENT OF COMPUTER SCIENCE
SERIES OF PUBLICATIONS A
A-1999-3, FEBRUARY 1999

INFERRING FINITE TRANSDUCERS

ERKKI MÄKINEN

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

ISBN 951-44-4518-X
ISSN 0783-6910

Inferring finite transducers

Erkki Mäkinen [1]

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

Abstract

We consider the inference problem for finite transducers using different kinds of samples (positive and negative samples, positive samples only, and structural samples). Given pairs of input and output words, our task is to infer the finite transducer consistent with the given pairs. We show that this problem can be solved in certain special cases by using known results on the inference problem for linear languages.

Keywords: formal languages, inductive inference, finite transducers, linear languages.

1 Introduction

A finite transducer is a finite automaton which emits an output string during each move made. It defines a translation, i.e. a set of pairs of strings. In each pair (α, β), α is a word over the input alphabet and β is a word over the output alphabet. A classical work considering the use of translations on compilers is [1].

This note deals with the inductive inference properties of finite transducers and the translations realized by them. Given a set of pairs of input and output strings, we consider the problem of inferring a transducer consistent with the pairs. Transducers are earlier studied in the context of inductive inference by Oncina *et al.* in [9].

We assume a familiarity with the basics of formal language theory and grammatical inference as given e.g. in [5] and [2], respectively. As inference criterion we use "identification in the limit" [4,2]. 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 [5]. The empty word is denoted by λ, the mirror image of a word w by w^R, and the length of a word α by $lg(\alpha)$.

2 Preliminaries

A finite transducer is a 6-tuple $M = (Q, \Sigma, \Delta, \delta, q_0, F)$, where Q is a finite set of *states*, Σ is a finite *input alphabet*, Δ is a finite *output alphabet*, δ is a mapping from $Q \times (\Sigma \cup \{\lambda\})$ to finite subsets of $Q \times \Delta^*$, q_0 is the *initial state*, and $F \subseteq Q$ is the set of *final states*. The translation realized by M is $T = \{(x, y) \mid (q_0, x, \lambda) \vdash^* (q, \lambda, y), x \in \Sigma^*, y \in \Delta^*, q \in F\}$, where the relation $\vdash^*$ is defined as usual. The mapping δ can be given also as a set of *moves* (q, u, p, v), where $(p, v) \in \delta(q, u)$, $p, q \in Q$, $u \in \Sigma^*$, $v \in \Delta^*$. Translations realized by finite transducers are called *regular translations*. Regular translations are also known as rational translations [3].

A finite transducer $M = (Q, \Sigma, \Delta, \delta, q_0, F)$ is *deterministic* if the following conditions hold for each state q in Q:

- either $\delta(q, a)$ contains at most one element for each $a \in \Sigma$, and $\delta(q, \lambda)$ is empty, or
- $\delta(q, \lambda)$ contains one element, and for all $a \in \Sigma$, $\delta(q, a)$ is empty.

Otherwise, a finite transducer is *non-deterministic*.

Recall that in *linear grammars* all productions have either the form $A \to uBv$, where A and B are nonterminals and u and v are (possibly empty) terminal strings, or the form $A \to u$, where u is a (possibly empty) terminal string. The former productions are called *continuing* and the latter ones are *terminating*. A language L is linear if there exists a linear grammar generating L. We suppose that all grammars are reduced, i.e. each nonterminal and terminal symbol appears in some derivation from the start symbol to a terminal string.

The following well-known fact establishes a relationship between regular translations and linear languages.

Theorem 1 *[10] T is a regular translation if and only if there exists a linear language L such that $T = \{(x, y) \mid x\#y^R \in L\}$, where $\#$ is a new symbol.*

In what follows, it is essential that from a given linear grammar, it is possible to uniquely construct the corresponding finite transducer. The left hand sides of the productions correspond to the states of the transducer, and the transition leaving from the states are obtained from the corresponding right hand sides. If $A \to uBv$ is a production, then the corresponding move is (q_A, u, q_B, v). Terminating productions are of the form $A \to \#$, where $\#$ is the separator between the two parts of the words. The corresponding move is $(q_A, \lambda, q_f, \lambda)$, where q_f is a final state of the finite transducer.

Any linear language can be generated by a linear grammar with productions of the form $A \rightarrow \lambda$, $A \rightarrow aB$, and $A \rightarrow Ba$ [11]. If we suppose that a linear grammar is in this normal form, we obtain a finite transducer where $\delta \subseteq Q \times (\Sigma \cup \{\lambda\}) \times (\Delta \cup \{\lambda\}) \times Q$. These transducer are called *1-bounded.* In a 1-bounded regular transducer each input and output string related to a transtion is a single terminal (from Σ or Δ, respectively) or the empty string λ.

The companion grammar of 1-bounded finite transducer $M = (Q, \Sigma, \Delta, \delta, q_0, F)$ has continuing productions of the form $A \rightarrow \alpha B \beta$, where $\alpha \in \Sigma \cup \{\lambda\}$ and $\beta \in \Sigma \cup \{\lambda\}$, and terminating productions of the form $A \rightarrow \#$, where $\#$ is the new symbol.

The purpose of this note is to apply the results obtained for inferring linear languages when inferring finite transducers from pairs of input and output strings.

3 Inferring linear languages

Takada [14] has introduced an inference algorithm for linear grammars with all continuing productions of the form $A \rightarrow aBb$, where a and b are single terminals, and all terminating productions of the form $A \rightarrow ab$, $A \rightarrow a$, or $S \rightarrow \lambda$, where S is the start symbol. We call these grammars *even linear.* A language L is an *even linear language* if there exists an even linear grammar generating L.

Let $G = (N, \Sigma, P, S)$ be an even linear grammar whose productions are uniquely labeled by the symbols of an alphabet Π. If a sequence ϕ of labeled productions is applied in a derivation $\beta \Rightarrow^* \gamma$, we write $\beta \Rightarrow^\phi \gamma$. If C is subset of Π^*, then the *language generated by G with control set C* is $L_C(G) = \{w \in \Sigma^* \mid S \Rightarrow^\phi w, \phi \in C\}$.

Takada [14] showed that a grammar scheme with productions of the form $S \rightarrow \lambda$, $S \rightarrow a$, $S \rightarrow ab$, and $S \rightarrow aSb$ is sufficient for all even linear grammars if regular control sets are used. The use of the grammar scheme with control sets reduces the inference problem for even linear languages to the inference problem for regular languages. As a consequnce, we have the following

Theorem 2 *[14] Even linear languages are inferable in the limit from positive and negative samples.*

For many practical purposes it is more natural to consider inference algorithms using positive samples only. For subclasses of even linear languages such inference algorithms are given in [6,8].

We say that an even linear grammar is *terminal-fixed* if $A \to aBb$ and $C \to aDb$ implies $A = C$ and $B = D$. If $A \to aBb$ and $C \to aDb$ implies $B = D$, we say that the grammar in question is *almost terminal-fixed*. An even linear language is terminal-fixed (resp. almost terminal-fixed) if there is a terminal-fixed (resp. almost terminal-fixed) even linear grammar generating it.

Theorem 3 *[8] Terminal-fixed even linear languages can be inferred from positive samples in linear time.*

Theorem 4 *[8] Almost terminal-fixed even linear languages can be inferred from positive samples.*

Additional conditions for the inferability of certain subclasses of even linear languages from positive samples are given in [6]. However, these conditions are not directly characterized by the form of single productions and we omit them here.

Sempere and Nagaraja [13] have considered the inferability of a subclass of linear languages from positive structural samples. In addition to an input string, the corresponding parsing tree without labels of internal nodes is given. As in the case of [6], these results characterize language and grammar classes by conditions which are not "local" to the productions. Hence, instead of the Sempere-Nagaraja results on structural inference for linear grammars, we use more general results by Sakakibara [12].

A context-free grammar $G = (N, \Sigma, P, S)$ is *reversible* if (1) $A \to \alpha$ and $B \to \alpha$ in P implies $A = B$ and (2) $A \to \alpha B \beta$ and $A \to \alpha C \beta$, where α and β are arbitrary strings over $N \cup \Sigma^*$, in P implies $B = C$. Hence, a context-free grammar is reversible if and only if it is (1) *invertible* and (2) *reset-free*. All context-free languages can be generated by reversible context-free grammars.

Theorem 5 *[12] The structural grammatical inference problem for reversible context-free grammars can be solved in polynomial time.*

Instead of general context-free grammars, we need here only linear ones. Hence, in the definition of reset-freeness, we have α and β in Σ^*.

4 The results

A 1-bounded finite transducer $M = (Q, \Sigma, \Delta, \delta, q_0, F)$ is *length-preserving* if $(p, \beta) \in \delta(q, a)$ implies $lg(\beta) = 1$, for all $q \in Q$ and $a \in \Sigma$, and $\delta(q, \lambda)$ is empty for all q. Length-preserving finite transducers correspond to even linear languages.

Given a pair of input and output strings from a translation realized by a length-preserving finite transducer, we can always combine the input and output terminals related to the same transition of the transducer. If $(a_1 a_2 \ldots a_n, b_1 b_2 \ldots b_n)$ is a pair of input and output strings, then the corresponding word produced by the companion even linear grammar is $a_1 a_2 \ldots a_n \# b_n \ldots b_2 b_1$, where $\#$ is the sepator. We have moves $(q_0, a_1, q_1, b_1), (q_1, a_2, q_2, b_2), \ldots, (q_{n-1}, a_n, q_n, b_n)$, for some states q_i, $i = 1, 2, \ldots, n-1$, in M, with $q_n \in F$.

By using Takada's algorithm [14] we can infer the finite transducer in the limit. We state the result in terms of translations as follows.

Theorem 6 *Regular translations realized by length-preserving finite transducers are inferable from positive and negative samples.*

If inference from positive samples only is prefered, then further restrictions to the form of transition functions of finite transducers must be set. When comparing the concepts of deterministic finite transducers and almost terminal-fixed even linear grammars, we notice that although the underlying ideas are quite the same, the concepts do not match. Hence, in order to apply Theorems 3 and 4, we must modify the concept of deterministic finite transducers.

We say that a length-preserving finite transducer $M = (Q, \Sigma, \Delta, \delta, q_0, F)$ is *state-deterministic*, if $(p_1, b) \in \delta(q_1, a)$ and $(p_2, b) \in \delta(q_2, a)$ implies $p_1 = p_2$ and $q_1 = q_2$. Similarly, M is *almost state-deterministic* if $(p_1, b) \in \delta(q_1, a)$ and $(p_2, b) \in \delta(q_2, a)$ implies $p_1 = p_2$. Now we clearly have a one-to-one correspondence between (almost) state-deterministic finite transducers and (almost) terminal-fixed even linear languages. Hence, we can write

Theorem 7 *Regular translations realized by state-deterministic finite transducers are inferable in linear time from positive samples only.*

Almost state-deterministic finite transducers are also inferable from positive samples only, but no linear time algorithm is known.

In the rest of this section we give up the assumption that finite transducers are length-preserving. We can relax the assumptions concerning transducers if we simultaneously strengthen the form of inference used. For now on, we suppose that structural samples are available. In the case of translations this means that we know how input and output strings are combined from the substrings related to the transitions of the transducer. A sample pair $(\alpha_1\alpha_2\ldots\alpha_n, \beta_1\beta_2\ldots\beta_n)$, is now given in the form $((\alpha_1,\beta_1),(\alpha_2,\beta_2)\ldots(\alpha_n,\beta_n))$, where each α_i in Σ^* and each β_i in Δ^*, corresponding a sequence of moves $(q_0,\alpha_1,q_1,\beta_1),(q_1,\alpha_2,q_2,\beta_2),\ldots,(q_{n-1},\alpha_n,q_n,\beta_n)$. Since we deal with linear grammars, this is the same information as used in the structural grammatical inference problem [12,13].

In order to apply Theorem 5, we need a restriction on the form of transition functions of finite transducers. We say that a finite transducer $M = (Q,\Sigma,\Delta,\delta,q_0,F)$ is *reversible* if (1) moves (q_1,u,p,v) and (q_2,u,p,v) implies $q_1 = q_2$, for all $p \in Q$, $u \in \Sigma^*$, and $v \in \Delta^*$, and (2) moves (q,u,p_1,v) and (q,u,p_2,v) implies $p_1 = p_2$, for all $q \in Q$, $u \in \Sigma^*$, and $v \in \Delta^*$.

By theorem 5 we now have

Theorem 8 *Regular translations realized by reversible finite transducers are inferable from positive structural samples.*

5 Conclusions

We have been able to characterize finite transducers realizing inferable regular translations. Depending on the form of samples available (positive and negative sample, positive samples only, or structural samples), we have different restrictions on the form of the transducers considered.

References

[1] A.V. Aho, and J.D. Ullman, *The Theory of Parsing, Translation, and Compiling. Volume I: Parsing*. Prentice-Hall, 1972.

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

[3] J. Berstel, *Transductions and Context-Free Languages*. B.G. Teubner, 1979.

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

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

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

[7] E. Mäkinen, On the structural grammatical inference problem for some classes of context-free grammars. *Inform. Process. Lett.* **42** (1992), 1–5.

[8] E. Mäkinen, A note on the grammatical inference problem for even linear languages. *Fundam. Inf.* **25** (1996), 175–181.

[9] J. Oncina, P. Garcia, and E. Vidal, Learning subsequential transducers for pattern recognition interpretation tasks. *IEEE Trans. Pattern. Anal. Machine Intelligence* **PAMI-15** (1993), 448–458.

[10] A.L. Rosenberg, A machine realization of the linear context-free languages. *Inform. Contr.* **10** (1967), 175–188.

[11] A. Salomaa, *Formal Languages*. Academic Press, 1973.

[12] Y. Sakakibara, Efficient learning of context-free grammars from positive structural examples. *Inform. Comput.* **97** (1992), 23–60.

[13] J.M. Sempere, and G. Nagaraja, Learning a subclass of linear languages from positive structural information. *Lecture Notes in Computer Science* **1433** (1998), 162–174.

[14] Y. Takada, Grammatical inference for even linear languages based on control sets. *Inform. Process. Lett.* **28** (1988), 193–199.