1987

Decision Problems Resulting from Grammatical Inference

Sandor Horvath
Eotvos Lorand University of Sciences

Efim Kinber
Sacred Heart University

Arto Salomaa
University of Turku, Finland

Sheng Yu

Recommended Citation

Horvath, S., Kinber, E., Salomaa, A., & Yu, S. (1987). Decision problems resulting from grammatical inference. *Annales Academiae Scientiarum Fennicae Series A. I. Mathematica,12,* 287-298.

Annales Academiæ Scientiarum Fennicæ
Series A. I. Mathematica
Volumen 12, 1987, 287—298

DECISION PROBLEMS RESULTING FROM GRAMMATICAL INFERENCE

SÁNDOR HORVÁTH, EFIM KINBER, ARTO SALOMAA and SHENG YU

1. *Introduction.* Grammatical inference is one of the classical areas of language theory. The basic task is to infer the rules of a grammar, or at least some specific subset of them, from some samples. (See, for instance, [1].) Inference problems are particularly natural when associated with devices generating a language as a sequence of words. Then the task is to determine the whole sequence (and possibly the rules of the device) from some part of the sequence. Typical examples deal with L systems, [3].

The particular problem investigated in this contribution has been initiated in connection with studies concerning the inference of programs, see [1] and [2]. The set-up is as follows. We are given a finite set $\mathscr{D}=\{D_1, \dots, D_n\}$ of derivations. We consider grammars G that can be inferred from $\mathscr{D}$ in the following sense. It is possible to "realize" each derivation step in each D_i using productions of G, and all productions of G result from some such derivation step. Attention will be mostly restricted to context-free grammars G. However, our basic definition below is more general.

Let G_1 and G_2 be two grammars inferred from $\mathscr{D}$. Intuitively, G_1 and G_2 are "close to each other" or "similar". Thus it is conceivable that, at least for some specially restricted $\mathscr{D}$'s, the equivalence problem of grammars inferrable from $\mathscr{D}$ is decidable.

However, it turns out that the equivalence problem is undecidable even if attention is restricted to context-free grammars and the $\mathscr{D}$'s considered are in some sense very simple. Results to this effect will be presented in this paper. The results show that some well-known undecidability results for context-free grammars hold true even if some specific additional information is available. We consider only grammars but analogous results hold for L systems as well.

We present various techniques in our undecidability proofs, obtaining a sequence of results of growing strength.

The following very natural set-up leads to decidability results. We are given all words that can be derived by derivation trees of some fixed height i, $i=0, 1, 2, \dots$. From this information we have to infer the grammar. Various versions of this problem are obtained, depending on the additional information available and, for instance, whether we want all possible grammars or just one of them.

A brief description of the contents of this paper follows. The basic definitions are given in Section 2. Section 3 proves the first undecidability result. The result is shown in the next section to hold even in a special case. Sections 5 and 6 establish two further strengthenings of undecidability. The final section discusses the inference problem based on finite sets of words.

Some of the results were presented already in [4].

2. *Definitions.* The reader is referred to [5] for all unexplained notions in language theory. Consider two disjoint alphabets Σ_N *(nonterminals)* and Σ_T *(terminals)* and denote

$$\Sigma = \Sigma_N \dot{\cup} \Sigma_T.$$

Some letter S in Σ_N is specified as the *start symbol.*

A finite sequence of words α_i over Σ

$$D\colon \alpha_1, \dots, \alpha_k \quad (k \geqq 2)$$

is referred to as a *derivation* provided each of the words α_i, $i<k$, contains a letter of Σ_N. The pairs (α_i, α_{i+1}), $i=1, \dots, k-1$, are referred to as *steps* of D. A pair (β, γ) of words over Σ is termed a *rule* or a *production inferred* from the step (α_i, α_{i+1}) if and only if there are words β_1 and β_2 such that

$$\alpha_i = \beta_1 \beta \beta_2, \quad \alpha_{i+1} = \beta_1 \gamma \beta_2$$

and, furthermore, β contains a letter of Σ_N. The same rule may be inferred from several steps.

By customary notations, derivations are often written

$$\alpha_1 \Rightarrow \alpha_2 \Rightarrow \dots \Rightarrow \alpha_k$$

and productions $\beta \to \gamma$. Productions are called *context-free* if and only if β is a letter of Σ_N.

Let now

$$\mathcal{D} = \{D_1, \dots, D_n\}$$

be a finite set of derivations. Then a grammar

$$G = (\Sigma_N, \Sigma_T, S, P)$$

is said to be *inferred* from $\mathcal{D}$ if and only if the production set P is obtained by inferring exactly one production from every step in every derivation D_j. G is termed *context-free* if and only if all productions in P are context-free. The notation $\mathcal{G}(\mathcal{D})$ (respectively $\mathcal{G}_{CF}(\mathcal{D})$) stands for the family of all (respectively all context-free) grammars inferred from $\mathcal{D}$.

Observe that, by our definitions, nonterminals, terminals and the start symbol can be recognized from the derivations D_j.

For instance, assume that $\mathcal{D}$ consists of the following three derivations:

$$S \Rightarrow SS, \quad S \Rightarrow S_1 \Rightarrow \lambda, \quad S_1 S_1 \Rightarrow aS_1 bS_1. \tag{1}$$

(Capital letters are nonterminals. Lower case letters are terminals. The empty word is denoted by λ.) Then the only context-free grammar G inferred from $\mathscr{D}$ is determined by the productions

$$S \to SS \mid S_1, \quad S_1 \to aS_1b \mid \lambda.$$

However, also the grammar G_1 determined by the productions

$$S \to SS \mid S_1, \quad S_1 \to \lambda, \quad S_1S_1 \to aS_1bS_1$$

is inferred from $\mathscr{D}$. No further grammars can be inferred from $\mathscr{D}$. Clearly,

$$L(G) = \{a^ib^i \mid i \geqq 0\}^* \quad \text{and} \quad L(G_1) = \{ab\}^*.$$

Thus, the two grammars inferred from $\mathscr{D}$ are not equivalent.

We want to investigate the existence of algorithms to the following effect. Given $\mathscr{D}$ and two grammars inferred from $\mathscr{D}$, the algorithm decides whether or not the grammars are equivalent. Hereby the selection of $\mathscr{D}$'s may be somehow restricted. Attention may also be restricted to the families $\mathscr{G}_{CF}(\mathscr{D})$.

3. *Basic undecidability*. Consider the grammars G_1 and G_2 defined on p. 90 of [5]. (The grammars are based on a given instance of the Post correspondence problem.) We now define the set $\mathscr{D}$ as follows. $\mathscr{D}$ contains all one-step derivations obtained from productions common to both G_1 and G_2. Thus, the production $S_0 \to A$ common to both G_1 and G_2 yields the derivation $S_0 \Rightarrow A$ in $\mathscr{D}$. Furthermore, $\mathscr{D}$ contains the three derivations

$$S_iS_i \Rightarrow \sharp S_i, \quad i = 1, 3, 4. \tag{2}$$

(As defined in [5], G_2 results from G_1 by replacing the productions $S_1 \to \sharp$ and $S_3 \to \sharp$ with the production $S_4 \to \sharp$. In $\mathscr{D}$ these three "exceptional" productions are represented by (2).)

Consider now the following grammar G_1' in $\mathscr{G}(\mathscr{D})$. The productions

$$S_1 \to \sharp, \quad S_3 \to \sharp, \quad S_4S_4 \to \sharp S_4$$

are inferred from steps (2). From all other steps in $\mathscr{D}$ we of course infer the unique production inferrable from that step. The grammar G_2' is obtained in the same way by inferring the productions

$$S_1S_1 \to \sharp S_1, \quad S_3S_3 \to \sharp S_3, \quad S_4 \to \sharp$$

from steps (2). It is clear that G_i' and G_i are equivalent, for $i = 1, 2$, because it is not possible to derive a word with two nonterminals according to G_i'. Hence, the equivalence problem for grammars in $\mathscr{G}(\mathscr{D})$ is undecidable, even if attention is restricted to context-sensitive grammars.

This result will now be strengthened to concern context-free grammars as well.

Consider two arbitrary context-free grammars G_A and G_B with start symbols A and B and with disjoint nonterminal alphabets not containing any of the letters S listed below. The set $\mathscr{D}$ is now defined as follows.

$\mathcal{D}$ consists of all one-step derivations corresponding to the productions of G_A and G_B and, moreover, of the following derivations:

$$S \Rightarrow S_1S_2, \quad S \Rightarrow S_1S_3,$$

$$S_1 \Rightarrow \lambda, \quad S_2 \Rightarrow \lambda, \quad S_3 \Rightarrow \lambda,$$

$$S_1S_2 \Rightarrow S_1 \sharp A \sharp S_2, \quad S_1S_3 \Rightarrow S_1 \sharp B \sharp S_3. \tag{3}$$

(Here $\sharp$ is considered to be a terminal letter. The start symbol is S.)

Consider the following two grammars G_1 and G_2 inferred from $\mathcal{D}$. We have to list only the productions inferred from (3), since all other productions of G_1 and G_2 are uniquely determined by the derivation steps. The productions inferred from (3) are in G_1

$$S_1 \to S_1 \sharp A \sharp, \quad S_1 \to S_1 \sharp B \sharp$$

and in G_2

$$S_2 \to \sharp A \sharp S_2, \quad S_3 \to \sharp B \sharp S_3.$$

Clearly,

$$L(G_1) = (\sharp L(G_A) \sharp \cup \sharp L(G_B) \sharp)^*$$

and

$$L(G_2) = (\sharp L(G_A) \sharp)^* \cup (\sharp L(G_B) \sharp)^*.$$

This means that G_1 and G_2 are equivalent if and only if one of the languages $L(G_A)$ and $L(G_B)$ is contained in the other. Since the latter condition is obviously undecidable for arbitrary context-free grammars G_A and G_B, we have established the following result.

Theorem 1. *There is no algorithm for deciding, given $\mathcal{D}$ and two context-free grammars G_1 and G_2 inferred from $\mathcal{D}$, whether or not G_1 and G_2 are equivalent.*

4. *Undecidability for restricted $\mathcal{D}$'s.* We shall show in this section that our problem remains undecidable even if $\mathcal{D}$ is "minimal". We consider minimality in two senses.

We have observed that sometimes only one production can be inferred from a derivation step. This is always the case when the left side consists of one nonterminal only, for instance, $S \Rightarrow S_1S_2$, but it is true also in some other cases, for instance, $SS \Rightarrow aSb$. Let us call a derivation step *ambiguous* if and only if more than one production can be inferred from it. The total number of ambiguous steps in the derivations of $\mathcal{D}$ is called the *order* of $\mathcal{D}$.

Thus, the order of $\mathcal{D}$ defined by (1) is one. The order of the set $\mathcal{D}$ used in the proof of Theorem 1 is two. Clearly, if the order of $\mathcal{D}$ is zero then $\mathcal{G}(\mathcal{D})$ contains only one grammar and, hence, the equivalence problem is trivial. On the other hand, the proof of Theorem 1 shows that the equivalence problem of $\mathcal{G}_{CF}(\mathcal{D})$ remains undecidable if $\mathcal{D}$ is restricted to sets with order $\leqq 2$. We now strengthen this result to concern sets with order $\leqq 1$.

Consider the proof of Theorem 1. We modify the set $\mathcal{D}$ as follows. The grammars G_A and G_B are as before. As before, $\mathcal{D}$ also contains all one-step derivations corresponding to the productions of G_A and G_B. But now the additional derivations in $\mathscr{D}$ are

$$S \Rightarrow S_1S_2, \quad S_1 \Rightarrow S_1\#A\#, \quad S_1 \Rightarrow \lambda, \quad S_2 \Rightarrow \lambda,$$

$$S_1S_2 \Rightarrow S_1\#B\#S_2. \tag{4}$$

Observe that (4) is the only ambiguous step and, consequently, the order of $\mathscr{D}$ equals one. To define a grammar inferred from $\mathscr{D}$, it suffices to tell the production inferred from (4).

Consider now the context-free grammars G_1 and G_2 inferred from $\mathscr{D}$ such that the production inferred from (4) is

$$S_1 \rightarrow S_1\#B\# \quad \text{and} \quad S_2 \rightarrow \#B\#S_2,$$

respectively. Clearly,

$$L(G_1) = (\#L(G_A)\# \cup \#L(G_B)\#)^*, \tag{5}$$

$$L(G_2) = (\#L(G_A)\#)^*(\#L(G_B)\#)^*. \tag{6}$$

As before we see that the equivalence of G_1 and G_2 is undecidable.

We finally show the undecidability of our problem when $\mathscr{D}$ is minimized in the sense that it contains only one derivation.

We shall define a set $\mathscr{D}$ consisting of only one derivation D. As before, the grammars G_A and G_B constitute the starting point but now we assume that they are also reduced. Let G_A contain t and G_B u productions. The first $t+u+3$ steps in the derivation D are:

$$\begin{aligned}
&S \Rightarrow S_1S_2 \Rightarrow S_1\#A\#S_2 \Rightarrow S_1\#A\# \#A\# S_2\\
&\Rightarrow \ldots \Rightarrow S_1(\#A\#)^tS_2 \Rightarrow S_1\#B\#(\#A\#)^tS_2\\
&\Rightarrow \ldots \Rightarrow S_1(\#B\#)^u(\#A\#)^tS_2\\
&\Rightarrow S_1(\#B\#)^u(\#A\#)^t \Rightarrow (\#B\#)^u(\#A\#)^t.
\end{aligned}$$

The derivation D now continues in such a way that each of the occurrences of B and A is transformed into a terminal word. The transformation is arbitrary otherwise but the derivation starting from the ith occurrence of B (respectively A) must use the ith production of G_B (respectively G_A), for all i.

Consider now the grammars G_1 and G_2 inferred from $\mathscr{D}$ such that both contain the production $S \rightarrow S_1S_2$, $S_1 \rightarrow \lambda$, $S_2 \rightarrow \lambda$, and all productions of G_A and G_B. Furthermore, G_1 contains the productions

$$S_1 \rightarrow S_1\#A\# \mid S_1\#B\#$$

and G_2 the productions

$$S_1 \rightarrow S_1\#B\#, \quad S_2 \rightarrow \#A\#S_2.$$

$L(G_1)$ and $L(G_2)$ are as in (5) and (6) except that in (6) the order of the two catenation factors has been commuted.

The results of Section 4 are summarized in the following theorem.

Theorem 2. *The undecidability result of Theorem* 1 *holds true even if* $\mathcal{D}$ *is restricted either to sets of order one or else to sets consisting of one proper derivation (i.e., a derivation beginning with the start symbol and ending with a terminal word).*

5. *Further strengthening.* A very natural next step is to combine the two conditions of Theorem 2. This will be done in our next theorem.

Theorem 3. *The undecidability result of Theorem* 1 *holds true even if* $\mathcal{D}$ *is restricted to sets of order one consisting of one proper derivation.*

Proof. We first observe that the following problem is easily shown to be undecidable. We are given two reduced λ-free linear grammars G_A and G_B, with start symbols A and B, and with no common nonterminals. We have to decide whether or not one of the languages $L(G_A)$ and $L(G_B)$ is contained in the other.

The following argument applies reduction to this undecidable problem.

Given G_A and G_B, we consider the following proper derivation of order one. It is understood that all capital letters appearing in the derivation (apart from A and B) are nonterminals not appearing in G_A and G_B. As before, $\sharp$ is a terminal letter. We now indicate the derivation; additional explanations are given later on.

$$\begin{aligned}
&S \Rightarrow S_1S_2 \Rightarrow S_1\sharp B\sharp S_2 \Rightarrow S_1KA\sharp\sharp B\sharp S_2\\
&\Rightarrow S_1\sharp A\sharp\ \sharp B\sharp S_2 \Rightarrow Q_1\sharp A\sharp\ \sharp B\sharp S_2\\
&\Rightarrow Q_2A\sharp\ \sharp A\sharp\ \sharp B\sharp S_2 \Rightarrow Q_1(A\sharp)^2\sharp A\sharp\ \sharp B\sharp S_2\\
&\Rightarrow Q_2(A\sharp)^3\sharp A\sharp\ \sharp B\sharp S_2 \Rightarrow \dots\\
&\Rightarrow Q_1(A\sharp)^{2k}\sharp A\sharp\ \sharp B\sharp S_2\\
&\Rightarrow (A\sharp)^{2k}\sharp A\sharp\ \sharp B\sharp S_2\\
&\Rightarrow (A\sharp)^{2k}\sharp A\sharp\ \sharp B\sharp R_1\\
&\Rightarrow (A\sharp)^{2k}\sharp A\sharp\ \sharp B\sharp\ \sharp BR_2\\
&\Rightarrow (A\sharp)^{2k}\sharp A\sharp\ \sharp B\sharp(\sharp B)^2R_1\\
&\Rightarrow (A\sharp)^{2k}\sharp A\sharp\ \sharp B\sharp(\sharp B)^3R_2\\
&\Rightarrow \dots \Rightarrow (A\sharp)^{2k}\sharp A\sharp\ \sharp B\sharp(\sharp B)^{2m}R_1\\
&\Rightarrow (A\sharp)^{2k}\sharp A\sharp\ \sharp B\sharp(\sharp B)^{2m} \Rightarrow \dots,
\end{aligned}$$

where the final part consists of deriving a terminal word from each of the A's and B's in such a way that every rule of G_A and G_B is applied at least once. The constants k and m are chosen to be large enough for this purpose. This is certainly possible. Observe, in particular, that there will be no difficulties as regards the parity of the number of A's and B's: it does not matter if we have to take some derivation twice.

The only ambiguous step in the above derivation is the second one: the derivation is of order one. Let us denote by G_1 the grammar having the rule

$$S_1 \to S_1 \sharp B \sharp,$$

and by G_2 the grammar having the rule

$$S_2 \to \sharp B \sharp S_2.$$

G_1 and G_2 are the only grammars inferred from the above derivation. They have as common rules all the rules of G_A and G_B and, moreover, the following rules:

$$S \to S_1 S_2, \quad S_1 \to S_1 K A \sharp, \quad K \to \sharp,$$
$$S_1 \to Q_1, \quad Q_1 \to Q_2 A \sharp, \quad Q_1 \to \lambda,$$
$$Q_2 \to Q_1 A \sharp, \quad S_2 \to R_1, \quad R_1 \to \sharp B R_2,$$
$$R_1 \to \lambda, \quad R_2 \to \sharp B R_1.$$

Denoting $L(G_1)$, $L(G_2)$, $L(G_A)$ and $L(G_B)$ shortly by L_1, L_2, L_A and L_B, respectively, we see that

$$L_1 = ((L_A \sharp)^2)^* (\sharp (L_A \cup L_B) \sharp)^* ((\sharp L_B)^2)^*$$

and

$$L_2 = ((L_A \sharp)^2)^* (\sharp L_A \sharp)^* (\sharp L_B \sharp)^* ((\sharp L_B)^2).$$

This means that $L_1 = L_2$ exactly in case one of the languages L_A and L_B is contained in the other. Since the latter condition is undecidable, we have completed the proof.

The above derivation is somewhat more complicated than actually necessary because we have taken care of the additional condition: whenever G_A and G_B are unambiguous, so are G_1 and G_2.

Observe, secondly, that the only ambiguous step in the derivation given in the proof of Theorem 3, in fact, increases the number of nonterminals. (The step was $S_1 S_2 \Rightarrow S_1 \sharp B \sharp S_2$.) This increase is not necessary, as seen from the following modification:

$$S \Rightarrow S_1 S_2 \Rightarrow S_1 \sharp S_2 \Rightarrow S_1 K A \sharp \sharp S_2$$
$$\Rightarrow S_1 \sharp A \sharp \sharp S_2 \Rightarrow (\sharp A \sharp)^2 \sharp S_2$$
$$\Rightarrow (\sharp A \sharp)^2 \sharp \sharp B \sharp S_2 \Rightarrow (\sharp A \sharp)^2 \sharp (\sharp B \sharp)^2$$
$$\Rightarrow (\sharp A \sharp)^3 \sharp (\sharp B \sharp)^2$$
$$\Rightarrow \ldots \Rightarrow (\sharp A \sharp)^k \sharp (\sharp B \sharp)^2$$
$$\Rightarrow (\sharp A \sharp)^k \sharp (\sharp B \sharp)^3$$
$$\Rightarrow \ldots \Rightarrow (\sharp A \sharp)^k \sharp (\sharp B \sharp)^m \Rightarrow \ldots$$

where the continuation is as before. Again the only ambiguous step is the second one. It gives the production $S_1 \to S_1 \sharp$ to G_1 and the production $S_2 \to \sharp S_2$ to G_2.

The productions common to G_1 and G_2 are now, apart from productions of G_A and G_B, simpler than before:

$$S \to S_1S_2, \quad S_1 \to S_1KA\sharp, \quad K \to \sharp,$$
$$S_1 \to \sharp A\sharp, \quad S_2 \to \sharp B\sharp S_2, \quad S_2 \to \sharp B\sharp,$$
$$A \to A\sharp\sharp A, \quad B \to B\sharp\sharp B.$$

Using the same notations as before, we now conclude that

$$L_1 = (\sharp L_A\sharp)^+((\sharp L_A\sharp)\cup\{\sharp\})^*(\sharp L_B\sharp)^+$$

and

$$L_2 = (\sharp L_A\sharp)^+((\sharp L_B\sharp)\cup\{\sharp\})^*(\sharp L_B\sharp)^+.$$

This implies that $L_1=L_2$ if and only if L_A and L_B coincide, so the reduction is again completed. We have, thus, established the following result.

Theorem 4. *The undecidability result of Theorem* 1 *holds true even in* $\mathcal{D}$ *is restricted to sets of order one consisting of one proper derivation, in which the only ambiguous step is nonterminal-nonincreasing.*

We remark that the grammars G_1 and G_2 inferred in the above construction are always ambiguous (independently of the ambiguity or nonambiguity of G_A and G_B). We do not know whether this is indeed unavoidable.

6. *Not even recursively enumerable.* Two further strengthenings of the undecidability results obtained so far will now be mentioned.

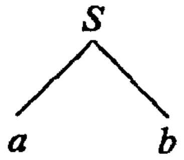
(i) In all results we may restrict ourselves to a two-letter alphabet. The construction needed is the standard one: the letter a_i, $i=1, \dots, n$, is encoded as ab^ia.

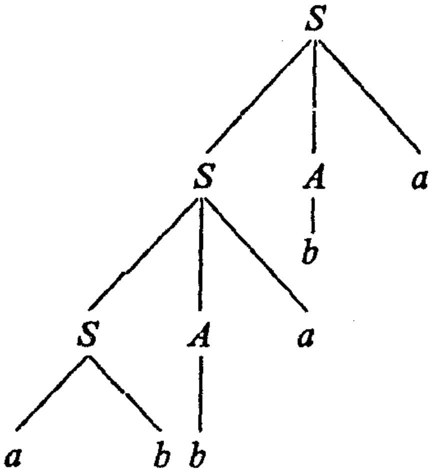
(ii) The problems P we have considered have the property that the set $\{X|P$ holds$\}$ is not recursively enumerable. This follows because, for instance, the set of pairs of equivalent linear grammars (of the type we considered) is not recursively enumerable. To put it differently, our problems are not even "partially decidable", as this phenomenon is sometimes called.

7. *Grammatical inference from finite sets.* This final section of our paper deals with a broad area yielding, contrary to our previous theorems, rather strong decidability results. The area concerns the very basics of language theory. The subsequent considerations are to be understood as an initial contribution only: we hope to return to the topic in another paper.

Essentially, we have the following problem. An oracle gives us, one after the other, *finite subsets* F_0, F_1, F_2, ... of a language $L(G)$. We know the rule by which the sets F_i are obtained from $L(G)$. Based on this information, we have to determine G (or a grammar equivalent to it). We are also given some a priori information, for instance, an upper bound on the number of nonterminals of G.

We consider only the following rule of forming the sets F_i. Given a context-free grammar G, the set $F_i(G)$, $i=0, 1, 2, \dots$, consists of those words of $L(G)$ that can be derived by a tree of *height i*. Thereby, height is determined by the nonterminal part of the tree only. For instance, the trees





for the terminal words *ab* and *abbaba* are of height 0 and 2, respectively.

We can now formulate our problem as follows.

Algorithmic problem. Given a sequence H_i, $i=0, 1, 2, \ldots,$ of finite sets of words, construct (if possible) a grammar G with the property $H_i = F_i(G)$, for all i.

The sequence H_i is "given" (as already indicated above) by listing the sets one after the other and not, for instance, by a formula in predicate calculus. Sometimes this is quite essential for decidability.

Let us consider an example. We have the information that G has only one nonterminal and only one nonterminating production, that is, production with nonterminals on the right side. We start to unfold the list of the F-sets:

$$F_0 = \{a^2\}, \quad F_1 = \{a^7\}, \quad F_2 = \{a^{12}, a^{17}\}.$$

From the additional information and F_0 we conclude that the terminal alphabet consists of a alone. Let us denote the only nonterminal by S. Then F_0 tells us that $S \to a^2$ is the only terminating production.

What could the nonterminating production be? By F_1, there are three possibilities:

$$S \to a^5S, \quad S \to a^3SS, \quad S \to aSSS.$$

(We assume without loss of generality that the a's precede the S's on the right side: the order does not affect the language.)

The first possibility would produce only one word to F_2, and the third possibility three words. We immediately check that the second possibility is in harmony with F_2. Hence, the only possible grammar is

$$S \to a^2, \quad S \to a^3SS.$$

The following result is easy to establish along the lines of this example.

Theorem 5. *Assume that the following additional information is given: G has only one nonterminal and only one nonterminating production. If $F_0=\{a^k\}$, $F_1=\{a^l\}$ and $m=k+l\geqq 1$, then at most one grammar can be inferred from the F_i-sequence. It is uniquely determined by the sets F_i with $i\leqq 2$.*

Theorem 5 does not hold for $m=0$. In fact, in case $F_0=F_1=F_2=\{\lambda\}$ all of the productions $S\to S^n$ $(n\geqq 1)$ serve the purpose. We want to emphasize that the uniqueness in Theorem 5 refers to the above convention: all a's precede all S's.

In the following example we have the same additional information as above. Now the list begins:

$$F_0 = \{a, a^3\}, \quad F_1 = \{a^{10}, a^{12}, a^{14}\}.$$

This is enough in this case. $S\to a$ and $S\to a^3$ are the only terminating productions. Since the cardinality of F_1 is three, the nonterminating production has to have two S's on its right side. Hence, it must be $S\to a^8SS$.

Again, the result holds also in general.

Theorem 6. *With the same additional information as in Theorem 5, if $F_0=\{a^m, a^n\}$, $m\neq n$, then at most one grammar can be inferred from the F_i-sequence. It is uniquely determined by the sets F_0 and F_1.*

The additional information of Theorems 5 and 6 can be defined as follows, without introducing the further simplification of a one-letter terminal alphabet.

Let G be a context-free grammar with the productions

$$S \to \alpha_1 S\alpha_2 S \dots \alpha_{k-1} S\alpha_k,$$
$$S \to \beta_i, \quad i = 1, \dots, m,$$

where the α's and β's are terminal words (maybe empty). We know that the grammar to be inferred has such a form.

The sets F_i can now be defined in a concise way:

$$F_0(G) = \{\beta_1, \dots, \beta_m\},$$
$$F_i(G) = \{\alpha_1\gamma_1\alpha_2\gamma_2\dots a_{k-1}\gamma_{k-1}\alpha_k \mid \text{ each } \gamma \text{ in } F_j(G)$$
$$\text{with } j < i, \text{ at least one } \gamma \text{ in } F_{i-1}(G)\},$$

where $i\geqq 1$.

If there are two nonempty β's in $F_0(G)$ containing no common letter, we call G *boundary-marked.*

Theorem 7. *If G is boundary-marked, there is an algorithm that constructs from $F_1(G)$ and $F_2(G)$ a grammar equivalent to G.*

The proof of Theorem 7 is based on the idea that the condition of boundary-markedness can be applied to deduce the nonterminating production. $F_0(G)$ is not needed because, in the first approximation, the terminating productions and the nonterminating one can be guessed simultaneously.

The following uninferability result shows that the grammars must be inferred from a class that is somehow bounded. Consider the following very simple sequence:

$$F_i = \{a^i\}, \quad i = 0, 1, \dots .$$

Assume that the grammar to be inferred is right-linear.

Take now any algorithm working on the sequence F_i. It has to make its decision after some initial segment of the sequence, say, $F_0, F_1, \dots, F_k$. Then the choice may be the simple grammar

$$G\colon S \to \lambda, \quad S \to aS.$$

However, the true sequence might have been

$$H_i = \begin{cases} \{a^i\} & \text{for} \quad i \leqq k, \\ \{a^k b^{i-k}\} & \text{for} \quad i > k. \end{cases}$$

Hence, our algorithm inferred a wrong grammar, a correct one being:

$$\begin{aligned} &S_j \to \lambda, && j = 0, \dots, k, \\ &S_j \to aS_{j+1}, && j = 0, \dots, k-1, \\ &S_k \to bS_k, \end{aligned}$$

where S_0 is the start symbol (and the second line is empty for $k=0$).

Thus, either the number of nonterminals or the number of productions has to be a priori bounded. It does not suffice to bound the length of the right sides of the productions.

Once an a priori bound has been fixed, a general method for our inference problem would proceed as follows.

The set F_0 gives all productions $S \to \alpha$, where α is a terminal word. Then the set F_1 gives from our bounded set finitely many candidates $G_1, \dots, G_m$. Each of the candidates G_j satisfies the conditions

$$F_0(G_j) = F_0, \quad F_1(G_j) = F_1.$$

Considering F_2, a subset of G_j's is obtained: grammars in this subset must satisfy also the equation for F_2. And so forth. At each step, the set of candidates becomes smaller or remains unchanged. Such a "stable" step must occur since we are beginning with a finite set. It seems likely that every grammar remaining after this stable step can be inferred from the F_i-sequence, or else there are no inferrable grammars.

Acknowledgement. This work was done in 1986 when the three non-Finnish authors were visiting the University of Turku.

References

[1] ANGLUIN, D. and C. H. SMITH: Inductive inference: theory and methods. - Comput. Surveys, 15, no. 3, 1983.

[2] BARZDIN, J. M.: Inductive inference of automata, functions and programs. - Proceedings of the International Congress of Mathematians, (Vancouver, B. C., 1974), Vol. 2, 1975, 455—460 (Russian).

[3] HERMAN, G. and G. ROZENBERG: Developmental systems and languages. - North-Holland Publishing Co., Amsterdam—Oxford, 1975.

[4] KINBER, E., A. SALOMAA and S. YU: On the equivalence of grammars inferred from derivations. - EATCS Bulletin 29, 1986, 39—46.

[5] SALOMAA, A.: Jewels of formal language theory. - Computer Science Press, Rockville, Md., 1981.

Sándor Horváth

Department of Computer Science
Eötvös Loránd University
Budapest
Hungary

Efim Kinber

Computing Center
Latvian State University
Riga
USSR

Arto Salomaa

Mathematics Department
University of Turku
SF-20500 Turku
Finland

Sheng Yu

105-22 Friendship Bldg
Qianjin Road
Tianjin
China

Received 20 March, 1987