REDUCTION OF ENUMERATION IN GRAMMAR ACQUISITION (*) (**)

STEFANO CRESPI-REGHIZZI
Istituto di Elettrotecnica ed Elettronica
Politecnico di Milano
Milan, Italy

1. INTRODUCTION

An acquisition theory for language must contain according to Chomsky (1):

(i) an enumeration of the class $\{y_i\}$ of possible sentences;
(ii) an enumeration of the class $\{s_i\}$ of possible structural descriptions;
(iii) an enumeration of the class $\{G_j\}$ of possible grammars;
(iv) specification of function $(i,j) \to f$ which specifies the structure s assigned by any grammar G. to any sentences

(v) a way of evaluating and selecting possible alternative grammars.

A model for language acquisition attempts to model language learning, or more precisely the learning of linguistic "competence". In a very abstract version, the model is an algorithm which works on certain primary linguistic data, and produces a grammar.

This paper presents a model for language acquisition which operates on the following primary data:

a) a finite sample of sentences, and
b) their structural descriptions;
c) non-sentences identified as such.

We note that the availability of structure (item b)) differentiates this model from other current studies on the acquisition of grammar (2) (3).

The knowledge of structure, prior to the acquisition of a grammar, can be defended on three grounds. First a child learning a language has stress and intonation information available to him, and this could be interpreted as a type of structural information. Second, if our grammars describe the base component(deep structure) of a language, then there is an intimate relation between structure and meaning, the structure being a prerequisite for understanding a sentence. The widespread belief the there must be a partially semantic basis for the acquisition of syntax then implies the availability of some structural information to the learner of the language. Third, availability of structure greally reduces the number of alternative possible grammars, and insures that the acquired grammar generates sentences with structures consistent with their meanings.

Given the data, a model could consider the enumeration (iii) of attainable grammars, and test their compatibility with the data a), b) and c). The test is possible because of (iv).

The model then selects one of the compatible grammars by means of the evaluation measure (v). A more accurate model should also explain the gradual development of an appropriate hypothesis, and the continual accretion of linguistic competence, rather than just considering the idealized instantaneous moment of acquisition of the correct grammar.

We note with Chomsky that different evaluation measures will assign different ranks to alternative hypotheses regarding the language of which the primary data

* Work supported by Consiglio Nazionale delle Ricerche.
** The first part of this study was done in the Dept.of Computer Science, University of California, Los Angeles.

are a sample. Hence choice of an evaluation measure for grammars amounts to deciding which generalizations about language are significant.

The class of possible hypotheses must be limited, if a realistic theory of syntax acquisition is to be developed. However, the limitation must still yield a class of grammars that is adequate in strong (and a_ fortiori weak) generative capacity. But beyond this, the requirement of feasibility is the major constraint of the model.

I shall have very few words to add to Chomsky's formulation of the problem, mainly for carrying further the formalization. My concern was for (1) delimiting a class of grammars which proved adequate in strong generative capacity, and defining a strategy for (2) reduction of possible hypotheses, and for (3) selecting a unique grammar. There is some similarity between the grammars that we are going to describe, and Bar-Hillel's (4) categorial grammars.

A model similar to the one that we shall describe - although of reduced scope - proved its feasibility on the computer (5)(6).

Following Gold (7), Feldman (2), Crespi-Reghizzi (5)(6) and Biermann (3), consider a context-free source grammar G_S and a source language $L_S = L(G_S)$.

The parenthesis grammar $[G_S]$ is derived from $G = (V_N, V_T, P, S)$ by replacement of every production $A \to u$, u in $V - V_N$, where, $V = V_N \cup V_T$, with the parenthesized production $A \to [u]$, where "[" and "]" are not in V. Note that "renaming productions" i.e. productions such as $A \to B$, where B is a nonterminal, are not parenthesized

Example

G1: $S \to HS \mid Ha$ $\quad$ [G1] : $S \to [HS] \mid [Ha]$

$H \to bH \mid a$ $\quad$ $H \to [bH] \quad [a]$

A structural information sequence I_S of language L_S is the ordered set:

$I_S = \{s_1, s_2, \ldots\}$ where s_i is in $L([G_S])$, and every sentence of the latter language occurs sooner or later, perhaps with repetition, in L_S.

We note that I_S provides items a) and b) of the primary data. The information sequence could start with the strings:

I1 = {[[a] a], [[b[b[a]]] a] ,

[[a[[a] [[a]a]]] , ...}

Alternatively, we shall write [[a] a] as $\underline{a}$ a .

Similarly, we introduce a negative information sequence N_S defined as an enumeration of all the non-sentences:

$N_S = \{n_1, n_2, \ldots\}$ where n_i is in $V_P^* - L_S$, $V_P = V_T \cup \{[,]\}$.

A positive sample $S_t = \{s_1, s_2, \ldots, s_t\}$ and a negative sample $M_t = \{n_1, n_2, \ldots, n_t\}$ compose the primary data available to the model at the discrete time t. At each time t we are interested in observing the grammar G_t which is output by the grammar acquisition device to account for the sample S_t and M_T. G_t must meet two requirements for compatibility with the data:

1) $L([G_t]) \supseteq S_t$; 2) $L([G_t]) \cap M_t = \emptyset$

If the model has to account for the increasing linguistic skill of the learner we have to consider the grammars G_{t_1}, G_{t_2}, ..., at successive time instants $t_1, t_2, \ldots$ corresponding to samples $S_{t_1}, S_{t_2}, \ldots$. We shall expect the guesses of the algorithm to become closer to the source language, as the sample is enlarged.

The model is said to identify the source grammar G_S in the limit if there is a time t' such that for t >t', and for any information sequences I_S, N_S :

3) $G_t = G_{t'}$; 4) $L([G_t]) = L([G_S])$.

In other words, identification in the limit implies that there are primary data S_T, M_T, t>t', which cause the model to select a grammar which is not later changed and is strongly equivalent to the source grammar. This must be true for any information sequences for that source grammar.

Two results, due to Gold (7), should be mentioned.

Theorem 1 - Let C be a class of decidable grammars (i.e. grammars for which it is decidable whether a string is generated by the grammar). Then, if G_S is in C there is an algorithm which identifies G_S in the limit.

Theorem 2 - Let C be a class of grammars which generam all the finite languages and any one infinite language. Assume the primary data consist only of a positive information sequence. Then, if G_S is in C there exists no algorithm for the class C which is able to identify G_S in the limit,

As a special case of Theorem 1, context free languages are identifiable in the limit, if sentences and non-sentences (identified as such) are given. If the latter are not available, Theorem ? implies that not even finite state languages can be identified in the limit.

We note that the proof of the first theorem is based on the fact that grammars are enumerable, and on the possibility of testing 1) and 2) for a decidable grammar. Such a proof does not lend itself to a feasible grammar acquisition method for context-free grammars, because of the astronomical number of grammars to be te sted.

2. A CHARACTERIZATION OF POSSIBLE GRAMMARS

In a previous work (5)(6) we have introduced a subclass of context-free grammars, termed free operator precedence grammars, for which identification in the limit is possible when only positive information is available. In this sequel we discuss an extension of these concepts,to include more general classes of languages.

We shall first define a family of classes of grammars(K-dist inct grammars)which cover the entire spectrum of context-free languages.

When a further restriction on the form of productions is imposed we obtain the family of k-distinct and k-homogeneous grammars, whose strong generative capacity does not cover the full spectrum of context-free languages,although our new grammars are able to account for self-embedding, nesting and other features required for natural languages.

For each class of grammars in the family,identification in the limit - without knowledge of non-sentences - is theoretically possible and practically feasible.

We now introduce some new definitions. Let G = (V V , P , S) be a context-free grammar, and let $V_p = V_T$ U{(,)}.

Define the left profile of order k of a string x in V^* as:

5) $L_k(x) = \{u \mid y = uv,\ x \overset{*}{\underset{[G]}{\Rightarrow}} y,\ u \text{ in } V_P^k,\ v \text{ in } V_P^*$ **or** $u = y\{\$\}^{k-|y|},\ x \overset{*}{\underset{[G]}{\Rightarrow}} y,\ y \text{ in } V_P^+,\ \$ \text{ is not in } V_P\}$

Similarly, the right profile of order k of a string x is:

6) $R_k(x) = \{u \mid y = vu,\ x \overset{*}{\underset{[G]}{\Rightarrow}} y,\ u \text{ in } V_P^k,\ v \text{ in } V_P^*$ **or** $u = \{\$\}^{k-|y|}y,\ x \overset{*}{\underset{[G]}{\Rightarrow}} y,\ y \text{ in } V_P^+,\ \$ \text{ is not in } V_P\}$

Example:

For the previous G1 we have :

x	$L_1(x)$	$L_2(x)$	$R_1(x)$	$R_2(x)$
HS	[	[a,[b	]	a],]]
Ha	[	[b,[a	a	]a
bH	b	b[	]	a],]]
a	a	a$	a	$a

Next, we define the k-profile $p_k(x)$ of a string x in V^+ as:

7) $_k(x) = (L_k(x) ; R_p(x))$

The k-profile of a nonterminal A is defined as:

8) $P_k(A) = (\bigcup_i L_k(x_i) ; \bigcup_i R_k(x_i))$,where the union is performed for every production $A \rightarrow x_i$ in the production set.

A grammar is k-distinct iff, for any two nonterminals A, B: $P_k(A) = P_k(B)$ implies A = B.

Example:

G1 is 1-distinct since:

$P_1(S) = (\{[\} ; \{a,]\})$

$P_1(H) = (\{a,b\} ; \{a,]\})$

It is obvious that if a grammar is k-distinct, it is also m-distinct, for any m>k.

Let $G = (V_N, V_T, P, S)$, and suppose that for A, B in V_N:

$T_A = \{x \mid A \overset{*}{\underset{G}{\Rightarrow}} x,\ x \text{ in } V_P^+\} \neq \{x \mid B \overset{*}{\underset{G}{\Rightarrow}} x,\ x \text{ in } V_P\} = T_B$. In this case we say that G has no duplicated productions. It would be easy to prove that every c.f. grammar has a strongly equivalent grammar with no duplicated productions and we can restrict our attention to non duplicated grammars without loss of generality.

Next we prove that any context-free grammar with no duplicated productions is k-distinct for some k>0.

Theorem 3 - G is k-distinct for some k>0.

Proof - Let u be the shortest string which is in T but not in T , and let j = u +1. Then u$ is in L.(x.), for some A->x. in P1! Since u$ is not in L.(y.) for any other production B->y , , it follows that P.(A) ≠ P.(B). In the same way determine j for any pair A, B in V , and let k be the largest of the integers thus determined. Then G is k-distinct.

Non-terminals of a k-distinct grammar are uniquely characterized by their k-profiles, which can be used as standard names for nonterminal characters.

Example:

Gl is l-distinct and can be rewritten as follows

G1 : <[; a,]> → <a,b ; a,]><[; a,]>|
<a,b ; a,]> a
<a,b ; a,]>→ b<a,b ; a,]>|a

* For the sake of readibility we write $P_k(A)$ as <...;...> .

It is obvious that the set of nontermi nal names for any k-distinct grammar of finite terminal vocabulary is finite. It follows that :

Theorem 4 - The class of k-distinct grammars with standard nonterminal names and with right parts of bounded length, over a finite terminal alphabet V_T, is finite.

Proof - For any production $A \to x$ we have $x \leq n$, forsome $n>0$. Bur there are finitely many strings of length n+1 over the finite s $eV_p U$ net of nontermi nals of a k-distinct grammar } .

3. A GRAMMAR ACQUISITION ALGORITHM

The algorithm to be described is able to identify in the limit a subclass of k-distinct grammars, which we will define after presenting the algorithm.

Consider a sample S of L , and let $s=a_1 a_2 \ldots a_n$ be a string of S . Enclose s between special delimiters a_o = -L and a = X . The algorithm is somehow similar to a syntax analyzer, and makes use of a pushdown stack where all scanned symbols are copied and renamed $Q_o Q_1 \ldots Q_i$ Provisions for detecting errors in the strings (e.e. non-balancing brackets)c#uld be easily added.

Comment algorithm for the construction of a k-distinct grammar from a string s;

```
Qo: = ao; i := 0; h := 1 ;
L1: if  ah ≠ "⊥" then
begin  i := j := i + 1 ; Qi := ah; h :=h+1;
  if  Qi = "]" then
L2:begin  j := j - 1;
     if  Qj ≠ "["  then go to L2;
     Qj := pk (Qj+1 ... Qi-1);
     Create the production "<pk(Qj+1...
     ..Qi-1)> →Qj+1...Qi-1";
     i := j ; go to L1
  end else go to L1;
end
create the production  'S→Qi';
```

It is best seen from an example that the grammar G. constructed by the algorithm from the stris, is coitible with s., i.e., s. is in L([Gi]).

Example

Applying the algorithm to s_1 = a_ a, we derive the 1-distinct grammar G_1 :——

Intermediate steps	Grammar G_1
[[a] a]	<a ; a> → a
[<a ; a> a]	<[; a> → <a ; a>a
S	S → <[; a>

or in a more readable form :

G_1 : A → a
B → Aa
S → B

In order to complete the grammar acquisition process, the algorithm is then applied to the sentences $s_1, s_2, \ldots, s_\tau$ of the sample S , yielding the grammars $G_1, G_2, \ldots, G_\tau$. Next the union is performed:

$$G'_\tau = G_1 \cup G_2 \cup \cdots \cup G_\tau$$

Note that the grammars $G_1, G_2, \ldots, G_\tau$ have non-disjoint nonterminal vocabularies.

Consider the sample S_3 ={ a a, b b a a, a a a a} from the grammar G_1.

Carrying on the example, from the second sentence S_2 = b b a a we derive G_2:

G_2: <a ; a>→ a	A → a
<b ;]>→b<a ; a>	C → bA
<b ;]>→b<b ;]>	C → bC
<[; a>→<b ;]> a	B → Ca
S → <[; a>	S → B

and, finally, from s_3 = a a a a we derive G_3:

G_3: <a ; a> → a	A → a
<[; a>→ <a ; a>a	B → Aa
<[;]>→<a ; a><[; a>	D → AB
<[;]>→<a ; a><[;]>	D → AD
S → <[;]>	S → D

The union $G'_3 = G_1 \cup G_2 \cup G_3$ is:

G'_3 : A → a
B → Aa|Ca
C → bA|bC
D → AB|AD
S → B | D

Examining G'_3 we see that G'_3 is compatible with S_3, but that L([G'_3]) ⊂ L([G1]) since for instance b a a a a is not generated by G'_3. Considering now a larger sample

$S_5 = S_3 \cup$ {b a a a a , b a a a} , we derive the grammar G'_5 as $G'_5 = G'_3 \cup G_4 \cup G_5$.

G'_5 A → a
B → Aa|Ca
C → bA|bC
D → AB|AD|CB|CD
S → B|D

This grammar is strongly equivalent to G1, and the grammar acquisition procedure is thus complexed.

Any grammar produced by the preceding algorithm is k-distinct, and, in addition has the following properties:

(1) There are no renaming rules, except for $S \to A$, A in V_N;

(2) for any two alternatives $A \to x$, $A \to y$, where A is in $V_N - S$, $p_k(x) = p_k(y)$.

Definition : a grammar having properties (1) and (2) is called k - homogeneous .

Consequently, the hypotheses space for this grammar acquisition model consists of the class of k-distinct and k-homogeneous grammars (shortened to k-d.h.). We already know from Theorem 3 that any c.f. grammar has a strongly equivalent k-distinct form.

It would be easy to prove that any context-free grammar admits a k-homogeneous strongly equivalent form. What is more relevant to our discussion is that there are context-free grammars which do not admit a strongly equivalent k-d.h. grammar, i.e. a grammar that is simultaneously k-distinct and k-homogeneous. An example is provided by the grammar G2 :

G2 : $S \to Ha$
$H \to Sa|a$

which generates the set $\{a^{2n} | n = 1,2,...\}$ with a leftlinear structure. G2 is 1-distinct but not 1-homogeneous, because $p_1(Sa) = (\{ [\}; \{a\}) \neq p_1(a) = (\{a\};\{a\})$.

Any attempt to render G2 1-homogeneous (by splitting $H \to Sa|a$ into $H_1 \to Sa$, $H_2 \to a$) will alter the profiles in such a way that the new grammar is no longer 1-distinct. The same can be shwon to be true for any order k of profiles.

On the other hand we notice that the same language is generated with a different structure by the 1-d.h. grammar G3:

G3 : $S \to aH|aK$
$H \to a$
$K \to Sa$

We do not know how to characterize in general the languages which cannot be strongly generated by k-d.h. grammars. From preliminary observations it seems that these languages are not too relevant to natural (or programming)languages, as is the case with $\{a^{2n} | n = 1,2,...\}$, and that restri£tion of the hypothesis space to the class of k-d.h. languages is hopefully not too severe. An accurate examination of this crucial point is the main target of our future investigations.

Assume now that the source language L_S admits a k-d.h. gram G_S.rIf the order k of the source grammar is known, the algorithm will select at any instant t just one grammar out of the k-d.h. grammars which are compatible with S . The selection of the grammar is performed by the algorithm with a criterion which leads to identification in the limit from a positive sample. The selected grammar generates the smallest language among the k-d.h. grammars which are compatible with the sample.

Lemma 4 - Let G =G(S)be the k-d.h.gramma? derived By the algorithm from the sample S .Then there is no other k-d.h. grammar G' such that: S $L([G'])$ $L([G_t])$.

Proof : The productions of G are clearly necessary for any k-d.h. grammar in order to generate S . Therefore G' may differ from G only by some additional productions, 5nd it follows that $L([G']) \supseteq L([G_t])$.

Our main result can now be proved.

Theorem 5 - Let S be a sample from a positive information sequence I_S, where L_S is generated by a k-d.h. grammar G . Let G_I be the k-d.h. grammar constructed by the algorithm from S . Then

$$\lim_{t \to \infty} L([G_t]) = L([G_S]),$$

that is the algorithm identifies G_S in the limit .

Proof : Let n be an upper bound on the length of any production in G . rrom the facts that 1) the class of k-d. h. grammars with productions of bounded length has finite cardinality, and 2) G generates the smallest language among the grammars of the previous class which are compatible with S , it follows that there is a time t' and a sample S_t' which causes the algorithm to derive a grammar which is not later changed and is strongly equi valent to G .

We note t'hat we are able to identify a language in the limit without knowledge of a negative information sequence. The paradox, with respect to Gold's Theorem 2, comes from the fact that the class of k-d.h. grammars with productions of bounded length does not generate all finite languages.

We now discuss the case that the source language is k-d.h. , but the order k is unknown. In this case we make use of nonsentences, provided by a negative sample M , in order to discard the grammars produced by the previous algorithm which are not compatible with the negative sample. The algorithm is the following:

comment acquisition of a k-d.h. grammar for a source language of order k unknown. A positive sample S_t and a negative sample M_t are given;
L: k := 1 ;
G_t := k-d.h. grammar derived from S_t by previous algorithm;
if $L([G_t]) \cap M_t \neq \phi$ then

```
begin
  k := k + 1;
  go to L;
end;
```

In this case identification in the limit is still possible but a negative information N_S is needed in order to determine the value of k which a priori is unknown.

Example

The source language is defined by the grammar G4 :

```
G4:  S → HS|Hb
     H → HH|a
```

It can be computed that L4 is a 2-d.h. language. Consider the positive sample S.

S_4 = {a b , a a b , a a b , a a a a b}

Assume k = 1, and derive a 1-d.h. grammar from the four sentences of S_4:

```
G4(1) :  <a ; a> → a                    A→a
         <[ ; b>→<a ; a>b               B→Ab|Cb
         <[ ;] >→<a ; a><a ; a>         C→AA|AB|C
         <[ ; b>→<[ ;] >b               S→B|C
         <[ ;] >→<a ; a> <[; b>
         <[ ;] >→<[ ;] > <[;] >
              S  →<[ ;b>
              S  →<[ ;_| >
```

Suppose now that a negative sample is given:

M_1 = {a a} = {n_1}

Since n_1 is in L([$G_4^{(1)}$]), this implies that a 1-d.h. hypothesis is not adequate. In other words G , which defines the smallest language in the 1-d.h. class, over-generalizes the sample S .

Consequently, we consider the 2-d.h. grammar derived from S :

```
G4(2) :  <a$ ; $a>→a                          A→a
         <[a ;]b>→<a$ ; $a>b                  B→Ab
         <[a ;a]>→<a$ ;$a><a$ ; $a>           C→AA
         <[ [;]b>→<[a ; a]> b                 D→Cb|Fb
         <[a ; b]>→<a$ ; $a><[a ;]b>          E→AB
         <[ [;]]>→<[a ; a]><[a ; a]>          F→CC|FC
         <[ [;]b>→<[ [;] ]> b                 S→B|D|E
         <[ [;]]>→<[ [;] ]><[a ; a]>
              S → <[a ;] b>
              S → <[ [;] b>
              S → <[a ; b]>
```

It can be verified that n_1 = a a is not generated by G , which can be assumed as the current hypothesis. Actually, identification of G_4 is not yet complete. If the positive sample were enlarged, the algorithm would eventually derive a 2-d.h. grammar which is strongly equivalent to G_4.

The last case to be discussed occurs when there is no integer k such that G_S is k-d.h.Then any attempt of identifying $G_S$by means of our algorithm will fail.The current hypothesis of order k derived by the algorithm from any positive sample can always be falsified by suitable enlargement of the negative sample against which the grammar is tested.

3. CONCLUSION

If we compare our results with Chomsky's formulation of the problem of grammar acquisition, we see that we are far from a fully satisfactory solution, since the class of k-distinct, k-homogeneous grammars that we have considered does not cover the full spectrum of context-free languages, much less that of Chomsky's transformational grammars.

On the positive side, the class of grammars for which we have provided a solution does not seem less adequate than context-free grammars for modeling linguistic competence, since self-embedding and nesting can be accounted for.

Our approach aimed at studying a learjning situation where the language acquisition device uniquely determines a grammar from given primary data, which consist of a sample of sentences and structure descriptions. Following this step, the grar^mar can be tested for over-generalization against a set of non-sentences. If any non-sentence is accepted by the grammar, the learning device steps up a parameter which enlarges the class of attainable grammars, and determines a grammar in the enlarged class. The procedure always converges to the source grammar, if the latter falls in the hypotheses space that we have already mentioned.

Perhaps the most significant result of this study is the definition of a sufficiently limited - and still interesting - class of grammars, and the use of primary data which include structure descri£tions in order to reduce the number of grammars which, at a certain moment, are compatible with the available data.

A simple strategy selects an appropriate "minimal" grammar which can be extended, if necessary ,as additional input sentences are added to the sample.

4. AKNOWLEDGEMENT

I am gratefull to Dr.ing. P. Delia Vigna and F. Schreiber for their valuable discussion of the problem. Many thanks are due to the reviewers of the manuscript for their many criticisms and sug, gestions.

5. REFERENCES

(1) CHOMSKY, N., "Aspects of the theory of syntax", MIT Press, Cambridge, Mass., 1964.

(2) FELDMAN, J.A., "Some decidability re suits on grammatical inference and complexity", AI MEMO n. 89, Stanford University, Stanford, 1969.

(3) BIERMANN, A.W. and FELDMAN, J.A., "A survey of results in grammatical inference", Proc. Int. Conf. on Frontiers in pattern recognition, Honolulu, Jan. 18-20, 1971.

(4) BAR-HILLEL, Y., "Language and information", Addison-Wesley, Reading, Mass., 1964.

(5) CRESPI-REGHIZZI, S., "The mechanical acquisition of precedence grammars", Thesis, University of California, School of Engineering, Los Angeles, 1970.

(6) CRESPI-REGHIZZI, S., "An effective model for grammar inference", Proc. IPIP Congress, Ljubljana, 1971.

(7) GOLD, M., "Language identification in the limit", Infor. and control, 10, 447-47H, 1967.