Contextual Grammars as Generative Models of Natural Languages

Solomon Marcus*
University of Bucharest

Carlos Martín-Vide†
Rovira i Virgili University

Gheorghe Păun‡
Institute of Mathematics of the Romanian Academy

The paper discusses some classes of contextual grammars—mainly those with "maximal use of selectors"—giving some arguments that these grammars can be considered a good model for natural language syntax.

A contextual grammar produces a language starting from a finite set of words and iteratively adding contexts to the currently generated words, according to a selection procedure: each context has associated with it a selector, a set of words; the context is adjoined to any occurrence of such a selector in the word to be derived. In grammars with maximal use of selectors, a context is adjoined only to selectors for which no superword is a selector. Maximality can be defined either locally or globally (with respect to all selectors in the grammar). The obtained families of languages are incomparable with that of Chomsky context-free languages (and with other families of languages that contain linear languages and that are not "too large"; see Section 5) and have a series of properties supporting the assertion that these grammars are a possible adequate model for the syntax of natural languages. They are able to straightforwardly describe all the usual restrictions appearing in natural (and artificial) languages, which lead to the non-context-freeness of these languages: reduplication, crossed dependencies, and multiple agreements; however, there are center-embedded constructions that cannot be covered by these grammars.

While these assertions concern only the weak generative capacity of contextual grammars, some ideas are also proposed for associating a structure to the generated words, in the form of a tree, or of a dependence relation (as considered in descriptive linguistics and also similar to that in link grammars).

1. Introduction

Contextual grammars were introduced by Marcus (1969), as "intrinsic grammars," without auxiliary symbols, based only on the fundamental linguistic operation of inserting words in given phrases, according to certain contextual dependencies. More precisely, contextual grammars include **contexts** (pairs of words), associated with

* University of Bucharest, Faculty of Mathematics, Str. Academiei 14, 70109 Bucharest, Romania. E-mail: solomon@imar.ro

† Research Group on Mathematical Linguistics and Language Engineering (GRLMC), Rovira i Virgili University, Pl. Imperial Tàrraco 1, 43005 Tarragona, Spain. E-mail: cmv@astor.urv.es

‡ Institute of Mathematics of the Romanian Academy, P. O. Box 1-764, 70700 Bucharest, Romania. E-mail: gpaun@imar.ro. Research supported by the Academy of Finland, project 11281, and Spanish Secretaría de Estado de Universidades e Investigación, grant SAB 95-0357.

selectors (sets of words); a context can be adjoined to any associated word-selector. In this way, starting from a finite set of words, we can generate a language.

This operation of iterated selective insertion of words is related to the basic combinatorics on words, as well as to the basic operations in rewriting systems of any type. Indeed, contextual grammars, in the many variants considered in the literature, were investigated mainly from a mathematical point of view; see Păun (1982, 1985, 1994), Păun, Rozenberg and Salomaa (1994), and their references. A complete source of information is the monograph Păun (1997). A few applications of contextual grammars were developed in connection with action theory (Păun 1979), with the study of theatrical works (Păun 1976), and with computer program evolution (Bălănescu and Gheorghe 1987), but up to now no attempt has been made to check the relevance of contextual grammars in the very field where they were motivated: linguistics, the study of natural languages. A sort of a posteriori explanation is given: the variants of contextual grammars investigated so far are not powerful enough, hence they are not interesting enough; what they can do, a regular or a context-free grammar can do as well. However, a recently introduced class of contextual grammars seems to be quite appealing from this point of view: the grammars with a maximal use of selectors (Martín-Vide et al. 1995). In these grammars, a context is adjoined to a word-selector if this selector is the largest on that place (no other word containing it as a proper subword can be a selector). Speaking strictly from a formal language theory point of view, the behavior of these grammars is not spectacular: the family of generated languages is incomparable with the family of context-free languages, incomparable with many other families of contextual languages, and (strictly) included in the family of context-sensitive languages, properties rather common in the area of contextual grammars.

This type of grammar has a surprising property, however, important from a linguistic point of view: all of the three basic features of natural (and artificial) languages that lead to their non-context-freeness (reduplication, crossed dependencies, and multiple agreements) can be covered by such grammars (and no other class of contextual grammars can do the same). Technically, the above mentioned non-context-free features lead to formal languages of the forms $\{xcx \mid x \in \{a,b\}^*\}$ (duplicated words of arbitrary length), $\{a^n b^m c^n d^m \mid n,m \geq 1\}$ (two crossed dependencies), and $\{a^n b^n c^n \mid n \geq 1\}$ ([at least] three correlated positions). All of them are non-context-free languages and all of them can be generated in a surprisingly simple way by contextual grammars with selectors used in the maximal mode.

Examples of natural language constructions based on reduplication were found, for instance, by Culy (1985), and Radzinski (1990), whereas crossed dependencies were demonstrated for Swiss German by Shieber (1985); see also Partee, ter Meulen and Wall (1990) or a number of contributions to Savitch et al. (1987). Multiple agreements were identified early on in programming languages (see, for example, Floyd [1962]), and certain constructions having such characteristics can also be found in natural languages. We shall give some arguments in Section 4.

Some remarks are in order here. Although we mainly deal with the syntax of natural languages, we sometimes also mention artificial languages, mainly programming languages. Without entering into details outside the scope of our paper,[1] we adopt the standpoint that natural and artificial languages have many common features (Man-

1 A word of warning: When we invoke statements concerning various topics, some of which have been debated for a long time, we do not necessarily argue for these statements and we do not consider the adequacy of contextual grammars as either proved or disproved by them. We simply mention a connection between a linguistic fact and a feature of our grammars.

aster Ramer 1993). For instance, we consider these languages infinite and organized on successive levels of grammaticality, whose number is unlimited in principle, although practically only a finite number of such levels can be approached. In Marcus (1981–83), where an effective analysis of contextual ambiguity in English, French, Romanian, and Hungarian is proposed, practical difficulties imposed a limitation to two levels of grammaticality for English (one level excluding compound words, the other level allowing the building of compound words) and Hungarian, but six levels for the analysis of French verbs. The reason for this situation is the "open" character of natural languages, making it impossible to formulate a necessary and sufficient condition for a sentence to be well-formed. As is pointed out by Hockett (1970), the set of well-formed strings in a natural language should be both finite and infinite, a requirement that is impossible to fulfill in the framework of classical set theory; for a related discussion, see Savitch (1991, 1993). This can also be related to Chomsky's claim that a basic problem in linguistics is to find a grammar able to generate all and only the well-formed strings in a natural language. Chomsky's claim presupposes that natural languages have the status of formal languages, but not everyone agrees with this notion. Even for programming languages, many authors reject the idea that well-formed strings constitute a formal language; see, for instance, the various articles in the collective volume Olivetti (1970), as well as Marcus (1979).

Returning to constructions specific to natural languages, we have found the surprising fact that the language $\{a^n cb^m cb^m ca^n \mid n, m \geq 1\}$ cannot be generated by contextual grammars with a maximal global use of selectors. Observe the center-embedded structure of this language and the fact that it is an "easy" linear language. As Manaster Ramer (1994, 4) points out, "the Chomsky hierarchy is in fact highly misleading..., suggesting as it does, for example, that center-embedded structures (including mirror-images) are simpler (since they are context-free) than cross-serial structures (including reduplications). Yet we know that natural languages abound in reduplications but abhor mirror-images (Rounds, Manaster Ramer, and Friedman 1987) and it also appears that, other things being equal, cross-serial structures are easier to process than center-embedded ones."

This point brings to mind Chomsky's arguments (1964, 120–25) that center-embedded constructions can be handled by the grammar (the description of competence), but not by the performance system. Here competence itself is not able to cover the center-embedded construction. However, we have to mention the fact that other similar constructions can be covered by contextual grammars (with or without maximal use of selectors). This is the case with $\{wc\ mi(w) \mid w \in \{a,b\}^*\}$, where $mi(w)$ is the mirror image of w. Also the language $\{w\ mi(w) \mid w \in \{a,b\}^*\}$ can be generated when the maximal use of selectors is considered, but not without involving this feature.

The difference between these last two languages suggests another point supporting the adequacy of contextual grammars: from the Chomsky hierarchy point of view, there is no difference between these languages; rather, their grammars are similar. This is not the case in the contextual grammar framework, and this also corresponds to our intuition: having a marker (the central c here) is helpful, it is significantly easier to process a language when certain positions of its sentences are specified. (Further illustrations of this point can be found in Section 4.) We conclude that contextual grammars with a maximal use of selectors seem adequate from these points of view for modeling natural languages.[2]

2 We do not claim and we do not intend to prove (because we cannot) that a contextual grammar with

In the architecture and the functioning of a contextual grammar one can note two contradictory basic ingredients. On the one hand, because we use **adjoining**, not **rewriting** (moreover, we do not use nonterminal symbols), the strings are always increased. At every step, we preserve all previously introduced symbols and we add new ones. This looks quite limiting for the power of these grammars. On the other hand, in contextual grammars there is a clear **context-sensing** capability, the contexts are adjoined to their selectors and depend on them. Context-sensitivity is in general a powerful property. Context-sensitivity plus **erasing** produces everything. In many cases in formal language theory, this combination leads to characterizations of recursively enumerable languages. Such a result has been proved by Ehrenfeucht, Păun, and Rozenberg (1997) for contextual grammars with unrestricted use of selectors. In the last section of this paper, we prove that this is also true for the case of maximal use of selectors. Specifically, we prove that every recursively enumerable language, L, can be written in the form $L = h_1(h_2^{-1}(L'))$, where h_1, h_2 are morphisms and L' is a language generated by a contextual grammar with maximal use of selectors. The proof uses the same construction as in Ehrenfeucht, Păun, and Rozenberg (1997), adapted to our class of grammars. The effect of h_1, h_2^{-1} can also be achieved by a sequential transducer (with finite memory), hence we may state the theorem in the form: every recursively enumerable language is a sequential translation of a contextual language (generated with maximal use of selectors). As a consequence, we find that our grammars can generate languages outside any family of languages that is strictly included in the family of recursively enumerable languages and is closed under direct and inverse morphisms or under finite sequential transducers. Important families in formal language theory have these properties: the family of context-free languages, several families in the regulated rewriting area (see Dassow and Păun [1989]), including indexed languages and programmed languages. Together with the fact that the language $\{a^n cb^m cb^m ca^n \mid n, m \geq 1\}$ mentioned above is linear, we get the incomparability of our families with many families in the Chomsky hierarchy or in its refinements.

This relates to another statement of Manaster Ramer's (1994, 4): "The question as posed by Chomsky [about the place of natural languages in a hierarchy of generative devices] seems to suggest that the class of natural languages will be found somewhere in the Chomsky hierarchy. Yet this need not be the case, and probably is not. It is entirely possible, for example, that a realistic theory of natural languages would define a class of languages which is incommensurate with the Chomsky types, e.g., a few regular languages, a few non-regular context-free languages, a few non-context-free context-sensitive languages, and so on. Indeed, it has been pointed out ... that, if finite languages are to be excluded from linguistic theory as Chomsky himself has always contended, then the class of natural languages will necessarily be a non-Chomsky class, since all the Chomsky classes do contain finite languages." Maybe contextual grammars (with maximal use of selectors) are one example of such a realistic possibility.

The discussion above has concerned the weak generative capacity of contextual

maximal use of selectors is the best model for natural language syntax, that these grammars can describe all types of constructions in natural languages or in other languages, or that, for instance, we can describe in a satisfactory manner the syntax of English. Maybe even other classes of contextual grammars have to be imagined, which will be better than the existing ones. Further efforts should be made to clarify the relevance of contextual grammars of various types for the study of natural languages. For instance, we can report no practical experience in writing a contextual grammar for a fragment of a natural language. In short, our goal is to acquaint the reader with contextual grammars and to convince him or her that these grammars deserve further investigation—of a mathematical and, more importantly, of a linguistic type.

grammars (with maximal use of selectors). Recently (see Martín-Vide and Păun [1998]), some attempts were made to introduce a structure into the strings generated by contextual grammars. An easy way to do so is to associate a tree to a derivation (just add a pair of parentheses to each context, then build a tree in the usual way: when reading a left parenthesis add a new edge, when reading a right parenthesis go back along the current edge, etc.) or a graph describing a dependence relation similar to those discussed in descriptive linguistics (see Chapter VI of Marcus [1967]) or in link grammars (Sleator and Temperley 1991; Grinberg, Lafferty, and Sleator 1995). We briefly present these possibilities here, although the linguistic relevance of the obtained structures is still being researched.

Let us also mention that, by definition, contextual grammars are (fully) lexicalized (in accordance with many current trends in formal syntax) and that their languages have the bounded growth property.

In view of all these results and properties, we believe that contextual grammars are an attractive model for natural language syntax, completing (but not necessarily competing with) the existing models, and that they deserve further investigation.

2. Definitions

In this section, we introduce the classes of grammars we shall investigate in this paper. As usual, given an **alphabet** V (which we also call **vocabulary**), we denote by V^* the set of all **words** (equivalently: **strings**) over V, including the empty one, which is denoted by λ. The set of all nonempty words over V, hence $V^* - \{\lambda\}$, is denoted by V^+. The *length* of $x \in V^*$ is denoted by $|x|$ and its **mirror image** (also called the **reversal**) by $mi(x)$. The families of finite, regular, linear, context-free, context-sensitive, and recursively enumerable languages are denoted by *FIN, REG, LIN, CF, CS, RE,* respectively. For the elements of formal language theory we use, we refer to Harrison (1978), Rozenberg and Salomaa (1997), and Salomaa (1973).[3]

A contextual grammar (with choice) is a construct:

$$G = (V, A, (S_1, C_1), \ldots, (S_n, C_n)), \ n \geq 1,$$

where V is an alphabet, A is a finite language over V, $S_1, \ldots, S_n$ are languages over V, and $C_1, \ldots, C_n$ are finite subsets of $V^* \times V^*$.

The elements of A are called **axioms** (starting words), the sets S_i are called selectors, and the elements of sets C_i, written in the form (u, v), are called contexts. The pairs (S_i, C_i) are also called **productions**. The intuition behind this construction is that the contexts in C_i may be adjoined to words in the associated set S_i. Formally, we define the direct derivation relation on V^* as follows:

$$x \Longrightarrow_{in} y$$

iff $x = x_1x_2x_3$, $y = x_1ux_2vx_3$, where $x_2 \in S_i, (u, v) \in C_i$, for some $i, 1 \leq i \leq n$.

Denoting by $\Longrightarrow_{in}^*$ the reflexive and transitive closure of the relation $\Longrightarrow_{in}$, the language generated by G is:

$$L_{in}(G) = \{z \in V^* \mid w \Longrightarrow_{in}^* z, \text{ for some } w \in A\}.$$

3 As general mathematical notations, we use: $\subseteq$ (inclusion, not necessarily proper), $\subset$ (proper inclusion), $\in$ ("is an element of"), $\emptyset$ (the empty set), 2^X (the family of all subsets of the set X).

Consequently, $L_{in}(G)$ contains all words of A, as well as all words that can be obtained from them by adjoining finitely many contexts, according to the selection imposed by the pairing (S_i, C_i).

Remark 1
The previous definition of a contextual grammar is called **modular**. Sometimes, it is useful to present a contextual grammar in the so-called **functional form**, that is, as a construct $G = (V, A, C, \varphi)$, where V and A are as above, C is a finite set of contexts over V, and φ: $V^* \longrightarrow 2^C$ associates sets of contexts from C to strings in V^*. Then we write $x \Longrightarrow_{in} y$ iff $x = x_1x_2x_3$, $y = x_1ux_2vx_3$, for some $(u, v) \in \varphi(x_2)$, $x_1, x_2, x_3 \in V^*$.

It is easy to see that starting from a contextual grammar in the modular presentation, $G = (V, A, (S_1, C_1), \ldots, (S_n, C_n))$, we can consider its functional counterpart $G' = (V, A, C, \varphi)$, with:

$$\begin{aligned} C &= \bigcup_{i=1}^{n} C_i, \\ \varphi(x) &= \{(u, v) \mid (u, v) \in C_i, x \in S_i, 1 \leq i \leq n\}, \ x \in V^*. \end{aligned}$$

Conversely, from a grammar given as $G = (V, A, C, \varphi)$ with:

$$C = \{(u_1, v_1), \ldots, (u_n, v_n)\},$$

we can pass, for instance, to $G' = (V, A, (S_1, C_1), \ldots, (S_n, C_n))$, taking, for each $i, 1 \leq i \leq n$:

$$C_i = \{(u_i, v_i)\},$$

and S_i the set of strings in V^* to which the context (u_i, v_i) can be adjoined, that is:

$$S_i = \{x \in V^* \mid (u_i, v_i) \in \varphi(x)\}.$$

The two grammars G and G' are clearly equivalent in both cases.

Thus, in the proofs below we shall use that presentation of a contextual grammar which is more appropriate (economical) for that case.

Remark 2
The derivation relation defined above has been denoted by $\Longrightarrow_{in}$ in order to distinguish it from the **external** derivation defined for G, where the context is adjoined at the ends of the derived word: $x \Longrightarrow_{ex} y$ iff $y = uxv$ for $(u, v) \in C_i$, $x \in S_i$, for some $i, 1 \leq i \leq n$. In Marcus (1969), only the external derivation is considered, for grammars presented in the functional form, without restrictions on the selection mapping. Contextual grammars with internal derivation were introduced in Păun and Nguyen (1980).

We do not investigate the external derivation here.

Two natural variants of the relation $\Longrightarrow_{in}$ defined above were considered by Martín-Vide et al. (1995):

$$x \Longrightarrow_{Ml} y$$

iff $x = x_1x_2x_3$, $y = x_1ux_2vx_3$, for $x_2 \in S_i$, $(u, v) \in C_i$, for some $1 \leq i \leq n$, and there are no $x_1', x_2', x_3' \in V^*$ such that $x = x_1'x_2'x_3'$, $x_2' \in S_i$, and $|x_1'| \leq |x_1|$, $|x_3'| \leq |x_3|$, $|x_2'| > |x_2|$;

$$x \Longrightarrow_{Mg} y$$

iff $x = x_1x_2x_3$, $y = x_1ux_2vx_3$, for $x_2 \in S_i$, $(u,v) \in C_i$, for some $1 \leq i \leq n$, and there are no $x_1', x_2', x_3' \in V^*$ such that $x = x_1'x_2'x_3'$, $x_2' \in S_j$, for some $1 \leq j \leq n$, and $|x_1'| \leq |x_1|$, $|x_3'| \leq |x_3|$, $|x_2'| > |x_2|$.

We say that $\Longrightarrow_{Ml}$ is a derivation in the **maximal local** mode (the word-selector x_2 is maximal in S_i) and $\Longrightarrow_{Mg}$ is a derivation in the **maximal global** mode (the word-selector x_2 is maximal with respect to all selectors $S_1, \ldots, S_n$).

For $\alpha \in \{Ml, Mg\}$, we denote:

$$L_\alpha(G) = \{z \in V^* \mid w \Longrightarrow_\alpha^* z, \text{ for some } w \in A\}.$$

If in a grammar $G = (V, A, (S_1, C_1), \ldots, (S_n, C_n))$, all selectors $S_1, \ldots, S_n$ are languages in a given family F, then we say that G is a contextual grammar with F choice (or with F selection). The families of languages $L_\alpha(G)$, for G a contextual grammar with F choice, are denoted by $CL_\alpha(F)$, where $\alpha \in \{in, Ml, Mg\}$. Here we consider F one of the families FIN, REG only. (It is natural to deal with selectors that are as simple as possible, otherwise the grammar is no longer of "practical" interest. Still, for the case of regular selectors we have here a sort of two-level grammar, because in order to completely describe a contextual grammar, we also need a grammatical description for the selector languages. However, using a selector S_i means deciding the membership of a substring of the current string with respect to S_i; when S_i is a regular language, this question can be solved in real time, using the simplest type of recognizers: a deterministic finite automaton. Derivations where the selectors are used in the minimal mode (no subword of a word-selector can be a selector) are introduced by Martín-Vide et al. (1995); we do not discuss this variant here.

3. Generative Capacity

First, we recall some results from previous papers devoted to contextual grammars of the basic type or with maximal use of selectors, then we prove new results about the power of the latter classes of grammars.

The relations between families of contextual languages, defined above, and between these families and families in the Chomsky hierarchy, pictured in the diagram in Figure 1, were proved by Martín-Vide et al. (1995). An arrow from a family F_1 to a family F_2 indicates the strict inclusion $F_1 \subset F_2$; the dotted arrow indicates an inclusion not known to be proper. Families not related by a path in this diagram are not necessarily incomparable. The families $CL_{Mg}(REG)$ and CF are incomparable with all families $CL_\alpha(F)$, $\alpha \in \{in, Ml\}, F \in \{FIN, REG\}$; CF is incomparable with $CL_{Mg}(REG)$, too.

Here are three languages used by Martín-Vide et al. (1995) in order to prove some of these strict inclusions and incomparabilities (we will need these languages later):

$$L_1 = \{a^nb^ma^nb^m \mid n, m \geq 1\} \in CL_\alpha(REG) - CL_\beta(FIN), \quad \alpha, \beta \in \{in, Ml, Mg\},$$
$$L_2 = \{a^nb \mid n \geq 1\} \cup \{a^nb^n \mid n \geq 1\} \in CL_{Mg}(FIN) - CL_\alpha(REG), \quad \alpha \in \{in, Ml\},$$
$$L_3 = \{x\, mi(x) \mid x \in \{a,b\}^*\} \in CL_\alpha(REG) - CL_{in}(REG), \quad \alpha \in \{Ml, Mg\}.$$

Note that languages L_2 and L_3 are linear, but L_1 is not context-free. In Păun (1985), it is proved that $CL_{in}(FIN) - CF \neq \emptyset$.

Here is a grammar generating the language L_2 in the Mg mode:

$$G = (\{a,b\}, \{ab, a^2b^2\}, (\{ab\}, \{(a,\lambda)\}), (\{a^2b^2 \mid n \geq 1\}, \{(a,b)\})).$$

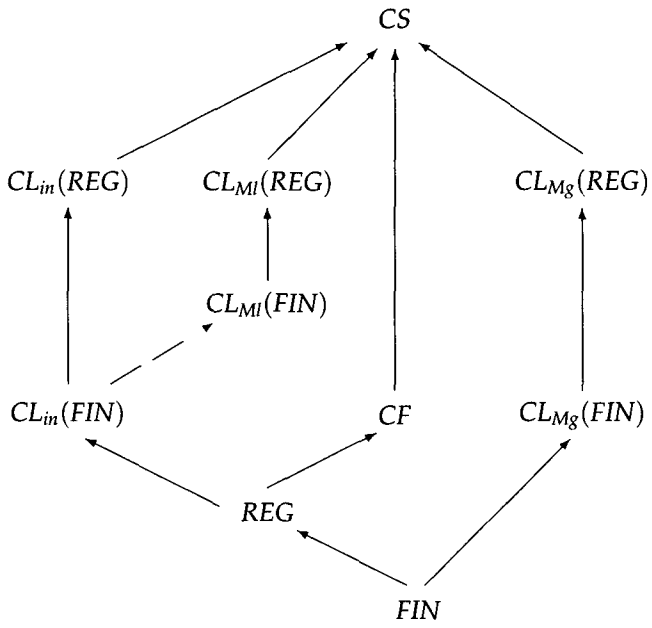

Figure 1
Relations between families of contextual languages and families in the Chomsky hierarchy.

Indeed, for any word a^nb with $n \geq 1$ only the first context can be used, and for any word a^nb^n with $n \geq 2$ only the second context can be used (here, the use of selectors in the maximal global mode is essential, in order to prevent the adjoining of the first context to words of the form $a^nb^n, n \geq 2$, in such a way as to destroy the equality of the number of a and of b occurrences). However, $L_2 \notin CL_{Ml}(REG) \cup CL_{in}(REG)$. Assume the contrary and take $G' = (\{a,b\}, A, (S_1, C_1), \ldots, (S_n, C_n))$ such that $L_\alpha(G') = L_2, \alpha \in \{in, Ml\}$.

In order to generate all strings $a^nb, n \geq 1$, we need a context (a^i, a^j), with $i + j > 1$, either associated with a^sb for $s \geq 0$ (then $j = 0$), or with $a^k, k \geq 0$. For $\alpha = in$, the contradiction is clear: strings $a^{n'}b^n$ with $n' > n$ can be produced in both cases.

Assume that G' is used in the maximal local mode. In order to generate the strings $a^nb^n, n \geq 1$, we also need a context $(a^k, b^k), k \geq 1$. This context cannot be applied to a string to which (a^i, a^j) above can be applied (from a^mb we get $a^{m+k}b^{k+1}$, which is not in L_2). Therefore (a^k, b^k) and (a^i, a^j) can be used independently (these contexts belong to sets C_s and C_t in G', respectively, with $1 \leq s, t \leq n, s \neq t$). This implies that (a^i, a^j) can be applied to a string a^qb^q with large enough q, again producing strings that are not in L_2.

The relationships between the family $CL_{Mg}(FIN)$ and other families $CL_\alpha(F)$, $\alpha \in \{in, Ml\}, F \in \{FIN, REG\}$, as well as between $CL_{Mg}(FIN)$ and CF, are not settled by Martín-Vide et al. (1995). We solve most of these problems here.

We start with two results having a linguistic relevance. The first one points out a surprising limitation of contextual grammars with global maximal use of selectors: there are center-embedded structures that cannot be generated by such grammars even

when regular selectors are used. Specifically, let us consider the language:

$$L_4 = \{a^n cb^m cb^m ca^n \mid n, m \geq 1\}.$$

Note that this is a linear language in Chomsky's sense and that it belongs to the families $CL_{in}(FIN)$ and $CL_{Ml}(FIN)$. For the grammar:

$$G = (\{a, b, c\}, \{acbcbca\}, (\{bcb\}, \{(b, b)\}), (\{acbcbca\}, \{(a, a)\})),$$

we have $L_{in}(G) = L$; the context (a, a) cannot be used after using the context (b, b), hence the grammar G can first generate any word of the form $a^n cbcbca^n, n \geq 1$, then any word $a^n cb^m cb^m ca^n, n, m \geq 1$; each selector consists of one word only, hence the local maximal use of selectors imposes no restriction, $L_{in}(G) = L_{Ml}(G)$.

In contrast to these observations, we have the following result:

Theorem 1
The language L_4 is not in the family $CL_{Mg}(REG)$.

Proof
Assume that $L_4 = L_{Mg}(G)$ for some grammar $G = (\{a, b, c\}, A, (S_1, C_1), \ldots, (S_k, C_k))$. In order to generate strings $a^n cb^m cb^m ca^n$ with arbitrarily large n and m we need:

- contexts (a^i, a^i) associated with selectors of the form $a^p cb^r cb^r ca^q$, for some $p, q \geq 0$, $r \geq 1$,
- contexts (b^j, b^j) associated with selectors of the form $b^s cb^t$, for some $s, t \geq 1$ (if one of s, t is zero, then we can introduce occurrences of b in front of the first occurrence of c or after the third occurrence of c in strings $a^n cb^m cb^m ca^n$ with large enough m).

If in a derivation we use an a-context, then no b-context can be used at a subsequent step: either the a-context is still applicable or an a-context with a larger selector is applicable, while the central subword $cb^m cb^m c$ has not been changed; the b-contexts use proper subwords of $cb^m cb^m c$, hence they are not allowed in the Mg mode.

Therefore, the derivations in G start by a phase:

$$w \Longrightarrow_{Mg}^{*} a^{n_1} cb^m cb^m ca^{n_1},$$

where only b-contexts are used, then (possibly) continue by a phase:

$$a^{n_1} cb^m cb^m ca^{n_1} \Longrightarrow_{Mg}^{*} a^n cb^m cb^m ca^n,$$

where only a-contexts are used (and the subword $cb^m cb^m c$ is not modified).

For a given $n \geq 1$, denote:

$$M(n) = \{x \in L_{Mg}(G) \mid w \Longrightarrow_{Mg}^{*} x \text{ by using only } b\text{-contexts}, \; w \in A, w = a^n cb^m cb^m ca^n, \text{ for some } m \geq 1\}.$$

Let:

$$n_0 = \max\{n \mid M(n) \text{ is infinite}\}.$$

All strings in $M(n_0)$ are of the form $a^{n_0}cb^mcb^mca^{n_0}, m \geq 1$. Denote:

$$M'(n_0) = \{w \in L_{Mg}(G) \mid w \Longrightarrow_{Mg} x \text{ by using a } b\text{-context}, \ x \in M(n_0)\}.$$

Because $M(n_0)$ is infinite and we use a finite set of contexts, the set $M'(n_0)$ is also infinite. Because each $w \in M(n_0)$ is derived using a b-context, it follows that no a-context can be applied to w, otherwise the derivation is not done in the Mg mode. However, for each m such that $a^{n_0}cb^mcb^mca^{n_0} \in M'(n_0)$, all strings $z = a^ncb^mcb^mca^n, n \geq 1$, are in L_4. Let us denote their set with $M''(n_0)$. In order to generate such strings with arbitrarily large n, we have to use a-contexts. Because such a context $(a^i, a^i) \in \varphi(a^pcb^mcb^mca^q)$ cannot be applied to $a^{n_0}cb^mcb^mca^{n_0}$, it follows that at least one of the relations $p > n_0$, $q > n_0$ holds. We have seen that a-contexts can be used only after b-contexts. Therefore, the strings in $M''(n_0)$ must be generated starting from axioms $a^{n_1}cb^{m_1}cb^{m_1}ca^{n_1}$ with $n_1 > n_0$. By the choice of n_0, such axioms are able to generate only finitely many strings of the form $a^{n_1}cb^mcb^mca^{n_1}$. The set $M''(n_0)$ is infinite, the set of axioms is finite, hence $M''(n_0)$ cannot be covered by strings generated in this way, a contradiction. The equality $L_4 = L_{Mg}(G)$ is not possible, $L_4 \notin CL_{Mg}(REG)$.

Note that the type of selectors plays no role in the previous argument, hence $L_4 \notin CL_{Mg}(F)$, for any family F of languages.

The fact that $L_4 \in CL_\alpha(FIN) - CL_{Mg}(REG)$, for $\alpha \in \{in, Ml\}$, should be contrasted with the fact that $L_3 \in CL_\alpha(REG) - CL_{in}(REG)$, for $\alpha \in \{Ml, Mg\}$: there are center-embedded constructions that cannot be handled by grammars with global maximal use of selectors, but the "total mirror language" can be generated when using a maximal restriction and not in the free case.

On the other hand, the family $CL_{Mg}(FIN)$ goes surprisingly far in the Chomsky hierarchy. The result will be stressed, indirectly, in Section 6, but we prefer to also give an example of a language that belongs to the family $CL_{Mg}(FIN)$ and looks quite complex. Together with the previous theorem, this example settles the relationships between the family CF and families $CL_\alpha(F), \alpha \in \{Mg, Ml\}, F \in \{FIN, REG\}$.

Theorem 2
The family $CL_{Mg}(FIN)$ contains non-context-free languages.

Proof
Consider the grammar:

$$G = (\{a, b\}, \{aab\}, (\{b\}, \{(a, \lambda), (b, a)\}), (\{abb\}, \{(bb, a)\}), (\{ba, babb\}, \{(\lambda, \lambda)\})).$$

Let us examine the intersection of the language $L_{Mg}(G)$ with the regular language:

$$R = (bba)^+a^+.$$

The family CF is closed under intersection with regular languages; if $L_{Mg}(G) \in CF$, then $L_{Mg}(G) \cap R \in CF$. However, this does not hold, because, as we shall prove below, we obtain:

$$L_{Mg}(G) \cap R = \{(bba)^{2^n}a^n \mid n \geq 2\}.$$

This is not a context-free language.

Indeed, examine the derivations in G, with a global maximal use of selectors, starting from aab and leading to words of the form $(bba)^na^m, n, m \geq 1$ (such words are elements of R).

As an illustration of the arguments that follow, let us consider a short example: take $n = 3$. We have to proceed as follows:

$$\begin{aligned} aa\underline{abb}a &\Longrightarrow_{Mg} aa\underline{bb}abbaa \Longrightarrow_{Mg} \underline{abb}abbaabbaa \\ &\Longrightarrow_{Mg} bbabba\underline{abb}aabbaa \Longrightarrow_{Mg} bbabbabbabbaa\underline{abb}aa \\ &\Longrightarrow_{Mg} bbabbabbabbaa\underline{abb}abbaaa \Longrightarrow_{Mg} bbabbabbabbabbabbaa\underline{abb}aaa \\ &\Longrightarrow_{Mg} bbabbabbabbabbabbabbaaaa = (bba)^{2^3}a^3. \end{aligned}$$

(The selector used at each step is underlined.)

Let us now examine a derivation of a general form. The first context of the first production, (a, λ), can be adjoined to occurrences of the symbol b only when these occurrences do not have a symbol a to their right-hand side; in such a case, the selector ba is present, which is larger than b, thus preventing the use of the first production. Thus, from aab we can produce any word of the form $a^n b, n \geq 2$.

To such a word we can adjoin the context (b, a), thus obtaining $a^n bba$. From now on, the selector b can never be used: the symbols b in pairs of b occurrences will not be separated, and there can never be four adjacent occurrences of b. (This could happen only when two symbols b are already present and two further ones are introduced by the context (bb, a); this means that we would have started from a word $xbbabbx'$, but with such a word we are not allowed to use the selector abb, because the longer selector $babb$ is present.) Therefore, the right occurrence of b in each pair bb is followed by an occurrence of a, and thus the use of the selector b is forbidden by the selector ba in the last production, whereas for the left b in a pair bb we cannot use the selector b, because the selector abb is present. Thus, from a word of the form $a^n bba$ we have to obtain a word in R using only the second production of the grammar. This means that every occurrence of a will go to the right, using this production. Crossing a pair bb, each occurrence of a introduces one more pair bb, as well as one more a. Hence, each use of the production doubles the number of occurrences of the pair bb. Since we eventually get a word starting with bb, this means that one pair bb has crossed all occurrences of a; at every step, one further a is introduced. One copy of a remains in triples bba, the other must migrate to the suffix of the word. Consequently, the obtained string is of the form $(bba)^m a^p$, where m is the number of times of using the production $(\{abb\}, \{(bb, a)\})$ minus one, and p is the number of initial occurrences of a, that is $p = n$. This implies that the occurrence of a immediately to the left-hand side of the initial pair bb has crossed one pair bb (doubling it), the next one has crossed two pairs bb (doubling them), and so on until the leftmost occurrence of a, the n-th one. In total, we have n doublings, because we started from $a^n bba$. This means that m above is equal to 2^n, that is the obtained word is of the form $(bba)^{2^n} a^n$. This completes the proof.

Corollary 1
The families CF, $CL_{Mg}(FIN)$ are incomparable.

Proof
Theorem 1 shows that $CF - CL_{Mg}(FIN) \neq \emptyset$, whereas from Theorem 2 we know that $CL_{Mg}(FIN) - CF \neq \emptyset$.

Returning to the diagram in Figure 1, we now know that any two families not linked by a path in this diagram are incomparable, except the pairs $(CL_{in}(REG), CL_{Ml}(FIN))$, $(CL_{in}(REG), CL_{Ml}(REG))$, and $(REG, CL_{Mg}(FIN))$, $(REG, CL_{Mg}(REG))$.

For the convenience of the reader, we list all pairs (F_1, F_2) of families of contextual languages from this diagram that are not known in be included in each other, specifying in each case the known incomparability arguments in the form $L, (F_1, F_2), L'$, with the following meaning: $L \in F_1 - F_2, L' \in F_2 - F_1$. When L or L' is not specified, it means that no such language is known. The languages L_1, L_2, L_3, L_4 used are those mentioned above:

$$\begin{array}{lll}
L_4, & (CL_{in}(FIN), CL_{Mg}(FIN)), & L_2 \\
L_4, & (CL_{in}(FIN), CL_{Mg}(REG)), & L_2 \\
L_1, & (CL_{in}(REG), CL_{Ml}(FIN)) & \\
 & (CL_{in}(REG), CL_{Ml}(REG)), & L_3 \\
L_4, & (CL_{in}(REG), CL_{Mg}(FIN)), & L_2 \\
L_4, & (CL_{in}(REG), CL_{Mg}(REG)), & L_2 \\
L_4, & (CL_{Ml}(FIN), CL_{Mg}(FIN)), & L_2 \\
L_4, & (CL_{Ml}(FIN), CL_{Mg}(REG)), & L_2 \\
L_4, & (CL_{Ml}(REG), CL_{Mg}(FIN)), & L_2 \\
L_4, & (CL_{Ml}(REG), CL_{Mg}(REG)), & L_2.
\end{array}$$

Both families $CL_{in}(FIN)$ and $CL_{Mg}(FIN)$ contain non-context-free languages, but there are linear languages not in $CL_{in}(REG), CL_{Ml}(REG)$ (for example: L_2) or in $CL_{Mg}(REG)$ (for example: L_4). We conjecture that $REG \subseteq CL_{Mg}(FIN)$, however, the construction from the proof of the inclusion $REG \subseteq CL_{in}(FIN)$ from Ehrenfeucht, Păun, and Rozenberg (1997) cannot be directly modified for the Mg case.

4. On the Linguistic Relevance of Contextual Grammars with Maximal Use of Selectors

With regard to their linguistic foundations, contextual grammars are closely related to American distributional linguistics, the potential of which they try to exploit. Let us quote some words of Manaster Ramer (1994, 4): "It is my contention that, until the early 1960's, the situation, as revealed by a close mathematical analysis of the underlying issues, was this: (a) there was no basis for concluding that 'in principle' natural languages were anything but context-sensitive (and it should have been clear that nothing was likely to change that result), (b) it was clear that phrase structure was inadequate in terms of its descriptive devices, and (c) it should have been clear (since it had been admitted in print) that phrase structure left out some of the descriptive devices of immediate constituent analysis. The right thing to have done would have been to pursue a more accurate formalization of immediate constituent analysis, and a more detailed analysis of just how much context-sensitivity was really required for natural languages."

The generative process in a contextual grammar is based on two dual linguistic operations, which are among the most important in both natural and artificial languages: insertion of a string in a given context and adding a context to a given string. Descriptive distributional linguistics developed in the U.S.A. in the 1940s and 1950s is entirely based on these ideas. To some extent, a similar idea is behind some aspects of Chomsky grammars; for instance, the difference between a context-free and a context-sensitive rule is that a certain substitution, generally valid in a context-free grammar, becomes possible only in a given context as soon as the grammar is no longer context-free, but context-sensitive.

Any derivation in a contextual grammar is a finite sequence of such operations, starting from an initial finite stock of strings, simple enough to be considered primitive well-formed strings (axioms).

Given a language L over the alphabet V, each context (u, v) over V selects a set of

strings x such that $uxv \in L$. We say in this case that x is accepted by (u, v) in L or that (u, v) accepts x in L. Any set C of contexts over V selects the set X of those strings that are accepted in L by any context in C. Obviously, X is here maximal, because it is the set of *all* strings with the relevant property. The dual phenomenon is the following: each string x over V **selects** the set $C(x)$ of those contexts that accept x in L. To any set E of strings over V we associate the set of contexts accepting in L any string in E. In short, given a language L, each set of contexts (strings) selects, with respect to L, a set of strings (contexts); in other words, each language over V determines a precise interplay of strings and contexts over V.

A natural question can be raised: could we now follow an inverse itinerary, by starting from a finite stock A of strings (over V) simple enough to be considered primitive well-formed strings (axioms), and by considering a finite set of couples $(S_i, C_i), 1 \leq i \leq n$, where S_i is a set of strings, while C_i is a finite set of contexts, to ask what is (are) the language(s) with respect to which C_i selects $S_i, 1 \leq i \leq n$? The idea of a contextual grammar, in its various forms, is born from the attempt to answer this question. A series of details about this topic can be found in Marcus (1997).

Let us consider again the three non-context-free constructions in natural languages mentioned in the introduction. The (non-)context-freeness of natural and programming languages has been investigated since the early sixties (Bar-Hillel and Shamir [1964]; Floyd [1962], among others). While for Algol 60 and for all advanced programming languages, the question has been settled from the very beginning—these languages are not context-free—a long debate was necessary concerning natural languages. We shall use information about this question from Gazdar and Pullum (1985); the reader might also consult Pullum (1985, 1986, 1987) and Pullum and Gazdar (1982).

The general technique in approaching this problem is the same for both programming and natural languages. Look for special constructions that seem, intuitively, to require a non-context-free competence. In order to extract them from the studied language, use an intersection with a regular language. Because CF is closed under intersection with regular sets, if the result is not context-free, then we have a proof that the initial language is not context-free.

The basic constructions of this type are duplication of arbitrarily long subwords, dependencies (agreements) between crossed pairs of subwords, and dependencies acting on (at least) three correlated subwords. The basic features of programming languages requiring dependencies are the necessity of declaring identifiers and names of procedures, and of defining labels.

In natural languages, such replications and dependencies can appear either at the level of the vocabulary or at the level of the sentences in a given language. The question is not simple, because it might not be clear what is grammatical and what is not grammatical with respect to a natural language. However, there are now convincing examples of non-context-free constructions in many languages. At the level of the vocabulary, the case of Bambara, a language from the Mande family in Africa (Culy 1985) is illustrative: compound words of the form *string-of-words-o-string-of-words* are possible in this language. The corresponding formal language consists of words of the form xcx, for x an arbitrarily long word over an alphabet not containing the symbol c (this symbol corresponds to the separator o in the Bambara construction). Because we can always codify words using two symbols, we work here with the language:

$$M_1 = \{xcx \mid x \in \{a, b\}^*\}.$$

Another non-context-free construction has been found in a dialect of German spoken around Zurich, Switzerland (Shieber 1985; Pullum 1985), which allows construc-

tions of the form $NP_a^m\ NP_d^n\ V_a^{m'}\ V_d^{n'}$, where NP_a are accusative noun phrases, NP_d are dative noun phrases, V_a are accusative-demanding verbs, V_d are dative-demanding verbs, and the numbers match up, that is $m = m', n = n'$. This leads to languages of the form:

$$M_2 = \{a^m b^n c^m d^n \mid n, m \geq 1\}.$$

Both of these constructions can be easily found in programming languages, too. The proof of Floyd (1962) that Algol 60 is not context-free leads to a language of M_1 type. Intersecting any Algol-like language with a regular language consisting of strings of the form:

$$\text{begin; real } x; \ldots, \text{ go to } label1; \ldots, y := 1; \ldots label2 : \ldots; \text{ end}$$

we force the equalities $x = y, label1 = label2$, hence a language like M_2 is obtained.

If, however, we intersect an Algol-like language with the regular set of strings of the form:

$$\text{begin; real } x; y := z; \text{ end}$$

then we force the equalities $x = y = z$, which can be translated into a language of the form:

$$M_3 = \{a^n b^n c^n \mid n \geq 1\}.$$

Concerning a natural language version of this form, Manaster Ramer (1993, 12) says: "The interaction of two different constructions (coordination and serial-verb formation) gives rise to patterns essentially of the form $a^n b^n c^n$ (and, more generally, $(a^n b)^m$) in Dutch and German, but there is no indication that any one construction in any language has this property." Also according to Manaster Ramer (1994, 21), "Columnar structures like $a^n b^n c^n, a^n b^n c^n d^n$, etc. (for all positive n) seem not to exist by themselves as constructions but do appear as compositions of two constructions (in particular, the serial verb construction of German or the cross-serial construction of Dutch together with coordination of the verb clusters) ... in these terms, natural languages possess an important property different from the usual formal languages. Namely, in natural languages individual constructions often have forms which no natural language, taken as a whole, can have. Thus, reduplication is common (probably universal), but there is no natural language which is made up, in its entirety, of reduplications."

Counterparts of these much-used examples of non-context-free languages can be identified in other areas, such as the semiotics of folklore (Marcus 1978).

None of the languages M_1, M_2, M_3 is context-free, and this is an easy exercise in any formal language textbook. Moreover, M_1 and M_3 belong to no family $CL_{in}(F)$, for arbitrary F (even more general than *FIN* and *REG*). The argument is similar in all cases: in the free mode of using selectors, one cannot sense the place where the context must be added without producing a parasitic word. Take, for instance, the case of M_3. If, in order to introduce arbitrarily many occurrences of a, we use a context $(a^i, b^i c^i)$, $i \geq 1$, associated with words of the form $a^j b^k, j, k \geq 0$, then $a^{j+k+1} b^{j+k+1} c^{j+k+1} \Longrightarrow_{in} a^{k+1} a^i a^j b^k b^i c^i b^{j+1} c^{j+k+1}$ is a correct derivation, but the word produced is not in M_3. A similar parasitic word is obtained if we use a context of the form $(a^i b^i, c^i), i \geq 1$, associated with $b^j c^k, j, k \geq 1$, and for contexts of the form $(a^i b^j, b^k c^l), i = j + k = l \geq 1$, associated with words $b^p, p \geq 0$. At least one such context is necessary, hence no grammar can generate M_3 in the free mode without producing parasitic strings.

However, *all three languages mentioned above can be generated by using the selectors in*

Table 1
Languages generated by various contextual grammars.

	$CL_{in}(FIN)$	$CL_{in}(REG)$	$CL_{Ml}(FIN)$	$CL_{Ml}(REG)$	$CL_{Mg}(FIN)$	$CL_{Mg}(REG)$
M_1	No	No	No	Yes	No	Yes
M_1'	No	No	No	No	No	?
M_2	No	Yes	No	Yes	No	Yes
M_3	No	No	No	Yes	No	Yes
M_4	Yes	Yes	Yes	Yes	Yes	Yes
M_4'	No	No	No	Yes	No	Yes

the maximal mode, both in the local and the global way. Here are grammars proving this assertion:

$$G_1 = (\{a,b,c\}, \{c\}, (\{c\}\{a,b\}^*, \{(a,a),(b,b)\})),$$
$$G_2 = (\{a,b,c,d\}, \{abcd\}, (ab^+c, \{(a,c)\}), (bc^+d, \{(b,d)\})),$$
$$G_3 = (\{a,b,c\}, \{abc\}, (b^+, \{(a,bc)\})).$$

The reader can easily check that $L_{Ml}(G_i) = L_{Mg}(G_i) = M_i, i = 1,2,3$. Notice how simple these grammars are, even compared with regulated context-free grammars (Dassow and Păun 1989), which, in some sense, are specially designed for handling such languages.

What is significant here is that *all* of these languages, hence *all* of the subjacent syntactic restrictions, can be handled by contextual grammars with both a local and a global maximal use of selectors, although—as we have seen—the overall generative power of such grammars is not "too large": there are context-free languages (even linear ones: remember the language in Theorem 1) that they cannot generate. On the other hand, the power of these grammars is not "too small." Theorem 2 from Section 3 and Theorems 3, and 4 from Section 6 explain the meaning of this statement.

At the beginning of Section 3, we mentioned that $M_2 \in CL_{in}(REG)$. This also follows from grammar G_2, for which we have $L_{in}(G_2) = L_{Mg}(G_2) = L_{Ml}(G_2)$: the two selectors are disjoint and their elements are "marked strings," bounded by fixed symbols, hence no selector string is the subword of another selector string. The maximality feature is, however, essential for G_1 and G_3, because, as we have mentioned before, the languages M_1 and M_3 cannot be generated by contextual grammars working in the *in* mode.

Consider now the "unmarked" variant of the language M_1 above, that is:

$$M_1' = \{xx \mid x \in \{a,b\}^*\},$$

as well as the marked and unmarked mirror image languages:

$$M_4 = \{xc\ mi(x) \mid x \in \{a,b\}^*\},$$
$$M_4' = \{x\ mi(x) \mid x \in \{a,b\}^*\}.$$

For reference, we indicate the possibility of generating these languages by contextual grammars of various types in Table 1.

Proofs of the assertions represented in Table 1 can be found in Martín-Vide et al. (1995), some of them were mentioned above, or can be easily found by the reader. For

the sake of the completeness, some hints for the proofs not discussed here are given in the appendix.

It is worth emphasizing the clear difference between marked and unmarked languages: the former are easier to handle than the latter. There is also a clear difference between contextual grammars and Chomsky grammars, with respect to the languages listed above. For instance, M_4 and M_4' are of the same complexity when they are generated by Chomsky grammars (both of them are linear and can be generated by almost identical grammars); in the framework of contextual grammars, M_4 and M_4' are significantly different. This also holds for M_1 and M_1'. The case of contextual grammars is closer to our intuition, because the existence of a marker makes it very easy to check the property defining the strings in our languages (knowing the "center," we can directly check the relation between the two halves of the strings).

It is known that the language M_3 mentioned above cannot be generated by a tree adjoining grammar (TAG) in the pure form introduced by Joshi, Levy, and Takahashi (1975), but $CF \subset TAL$, where TAL is the family of languages generated by TAGs without additional features (see also Section 21.2 in Partee, ter Meulen, and Wall [1990]). In view of the languages L_2 and L_3 in Section 3, which are context-free but not in $CL_{Mg}(REG)$, or $CL_{Ml}(REG)$, respectively, it follows that TAL is incomparable with each of the families $CL_{Ml}(REG)$, and $CL_{Mg}(REG)$. However, TAGs with constraints (for instance, with null adjoining contraints; see, for example, Joshi [1987] and references therein) can generate all languages M_1, M_2, and M_3; hence, a proper superfamily of TAL is obtained. The relationships between such enlarged TAL families and families of contextual languages are not settled yet.

An important question in this framework is whether or not the languages in the families $CL_\alpha(REG)$, $\alpha \in \{Ml, Mg\}$, are *mildly* context-sensitive. It is obvious that, by definition, contextual languages have the bounded growth property: the set of contexts is finite, passing from one string to another means adjoining of a context from a finite set, and all generated strings belong to the language. However, we do not know whether or not the languages in families $CL_\alpha(REG), \alpha \in \{Ml, Mg\}$ are parsable in polynomial time.

In general, the parsing of languages generated by contextual grammars (of any type, not only with maximal use of selectors) is a research area still open. There are several attempts to define contextual automata (see, for example, Păun [1982], Jančar et al. [1996], and Miquel-Vergés [1997]). Some of them characterize a number of families of contextual languages, and some of them recognize families that do not correspond to classes of contextual grammars. However, no systematic study of parsing complexity has been done, even for basic classes of contextual grammars. (Of course, because in contextual grammars we do not have erasing operations but only adjoining, we always generate context-sensitive languages, hence membership is decidable.)

The only complexity results known at the moment concern external contextual grammars with regular (even context-free) selectors, and a variant of internal contextual grammars with regular selectors used in a "localized" manner: the selector used at any derivation step should "touch" the context used at the previous step. Ilie (1997a, 1997b) proved that the parsing of the languages generated by such grammars can be done in polynomial time.

Let us close this section with the observation that contextual grammars have another property much discussed recently: they are *lexicalized* (we might say "fully lexicalized"), as each of their productions (pair selector-context) consists of terminal symbols only.

5. Attempts to Associate a Structure to Contextual Languages

In this section, we investigate further the adequacy of contextual grammars for describing the syntax of natural languages.

One of the features of context-free grammars and of other grammars based on context-free core rules (TAGs included) most useful for linguistics is the fact that a derivation can be described by a tree defining a structure of the generated sentence. On this basis, the difference between weak generative capacity and strong generative capacity was introduced: the former refers to the set of sentences that a grammar produces, while the latter refers to the set of pairs composed by a sentence and its phrase-structure tree.

Only very recently (see Martín-Vide and Păun [1998]) some possibilities for introducing a structure to the words generated by contextual grammars were considered. We present here some ideas from Martín-Vide and Păun (1998), without entering into details; research is still in progress. We only want to show that various natural solutions exist for structuring contextual languages. For instance, a tree can be associated to a derivation in a contextual grammar, as we describe below.

Consider the parentheses [and] and denote by B their set. A string $w \in (V \cup B)^*$, where V is an alphabet, is said to be minimally Dyck covered if:

1. w can be reduced to λ by using reduction rules of the form $[x] \to \lambda$, for $x \in V^+$;
2. if $w = w_1]w_2[w_3$, with $w_1, w_3 \in (V \cup B)^*$ and $w_2 \in V^*$, then $w_2 = \lambda$.

We denote by $MDC(V)$ the language of all minimally Dyck covered strings over the alphabet V.

To any string $x \in MDC(V)$ we can associate a tree $\tau(x)$ with labeled edges in the following way:

- draw a dot representing the root of the tree; the tree will be represented with the root up and the leaves down;
- scan x from the left to the right and grow $\tau(x)$ according to the following two rules:
- for each maximal substring $[w$ of x, for $w \in V^*$ (hence after w we find either [or]), we draw a new edge, starting from the current point of the partially constructed $\tau(x)$, marked with w on its left side, and placed to the right of the currently constructed tree;
- for each maximal $w], w \in V^*$, not scanned yet (hence, either before w we find], or $w = \lambda$ and to the left of] we have a substring $[z$ for some $z \in V^*$ already scanned), we climb the current edge, writing w on its right side.

Here is a simple example. The tree corresponding to the string:

$$x = [a[ab][ab[ab[c]b]b]a][a]$$

(which is clearly in $MDC(\{a, b, c\})$) is presented in Figure 2. The nodes are numbered in the order of producing them (1 is the root).

A **bracketed contextual grammar** is a construct:

$$G = (V, A, (S_1, C_1), \ldots, (S_n, C_n)),$$

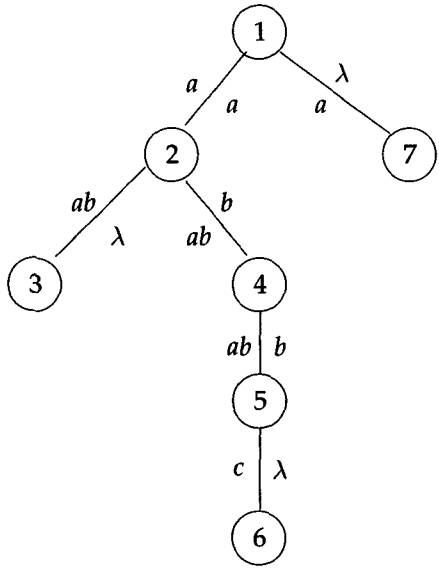

Figure 2
Tree corresponding to the string $x = [a[ab][ab[ab[c]b]b]a][a]$.

where V is an alphabet, A is a finite subset of $MDC(V)$, $S_i \subseteq MDC(V)$, and C_i are finite subsets of $V^* \times V^* - (\lambda, \lambda)$, for all $1 \leq i \leq n$; in turn, $n \geq 1$.

For $x, y \in MDC(V)$ we define:

$$x \Longrightarrow_{in} y$$

iff $x = x_1x_2x_3$, $y = x_1[ux_2v]x_3$, where $x_1, x_3 \in (V \cup B)^*, x_2 \in MDC(V)$, and $x_2 \in S_i, (u, v) \in C_i$, for some $1 \leq i \leq n$. (Clearly, if $x \in MDC(V)$ and $x \Longrightarrow_{in} y$, then $y \in MDC(V)$, hence the definition above is consistent.)

The **string language** generated by a bracketed contextual grammar $G = (V, A, (S_1, C_1), \ldots, (S_n, C_n))$ is defined by:

$$L(G) = \{pr_V(z) \mid w \Longrightarrow_G^* z, \text{ for some } w \in A\},$$

where $pr_V(z)$ denotes the projection of $z \in (V \cup B)^*$ on V, that is the string in V^* obtained by removing [and] from z.

We can also associate to G the **bracketed language** $BL(G)$ defined by:

$$BL(G) = \{(pr_V(z), \tau(z)) \mid w \Longrightarrow_G^* z, \text{ for some } w \in A\}.$$

Note the fact that each string in $L(G)$ is paired with a tree in $BL(G)$; however, the string should be read on the edges of this tree, not on leaf nodes as in the case of derivation trees of context-free grammars. The linguistic significance of such a tree is not yet clear to us, hence we do not insist on this idea (the ambiguity of contextual grammars and languages can be defined in this framework, but how the tree illuminates the grammatical structure of a sentence remains to be clarified).

Another idea considered by Martín-Vide and Păun (1998), closer to linguistics, is to introduce a **dependence relation** on the set of symbols appearing in axioms,

selectors, and contexts of a contextual grammar. We present some of the details of this idea informally below.

Consider an alphabet V and a string $x \in V^*$. We denote:

$$M(x) = \{1, 2, \ldots, |x|\}$$

and we write $x = x(1)x(2)\ldots x(n)$, for $n = |x|, x(i) \in V, 1 \leq i \leq n$. Any antireflexive relation on $M(x)$ is called a dependence relation on x. Let ρ_x be such a relation (antireflexivity means $i\ \rho_x\ i$ for no value of i). The pair (x, ρ_x) is called a **structured string**. If $i\ \rho_x\ j$, then we say that $x(j)$ **depends** on $x(i)$. Let us denote by ρ_x^+ the transitive closure of ρ_x. If $i\ \rho_x^+\ j$, then we say that $x(j)$ is **subordinate** to $x(i)$. A structured string (x, ρ_x) can be represented in a graphical form by writing the elements $x(1), \ldots, x(n)$ of x in a row and drawing above them arcs $(x(i), x(j))$ for $i\ \rho_x\ j$. A structured string (x, ρ_x) is called a **simple string of center** $x(i_0)$ if the graph associated to it as described above is a tree with the root marked with $x(i_0)$ (the center corresponds to the predicative element of a sentence).

The notion of a structured string is well-known in linguistics: see, for example, Chapter VI of Marcus (1967). A related notion has been recently considered, that of a link grammar: see Sleator and Temperley (1991), or Grinberg, Lafferty, and Sleator (1995). In a link grammar, the elements of a sentence are correctly related in a **linkage**, according to a pairing of left and right **connectors** given for each word in the dictionary, providing that the obtained dependence relation has several properties: the associated graph is connected, planar, etc. Because we do not investigate here the possibility of producing correct linkages, in the sense of Sleator and Temperley (1991), by using contextual grammars (such results appear in Martín-Vide and Păun [1998]), we do not formally define the notion of a link grammar.

For a structured string (x, ρ_x), $x \in V^+$, and a substring y of x, we denote by $\rho_x|_y$ the *restriction* of ρ_x to y, defined in the natural way (we remove the symbols of x not appearing in y and we collect the remaining pairs of ρ_x).

Now, a **structured contextual grammar** is a construct:

$$G = (V, A, P),$$

where V is an alphabet, A is a finite set of structured strings over V, and P is a finite set of triples of the form $\langle x, (u, v); \rho_{uxv} \rangle$, with $x \in V^+, (u, v) \in V^* \times V^*$, and ρ_{uxv} a dependence relation over uxv such that $\rho_{uxv}|_x = \emptyset$.

The elements of A are called axioms, the triples in P are called **productions**; in a production $\langle x, (u, v); \rho_{uxv} \rangle$, the string x is the selector, (u, v) is the context and ρ_{uxv} is a relation defining the structure of uxv; note that no dependence is considered between the elements of x. (Thus, we consider here only grammars with finite selectors.)

The derivation relation is defined (only for structured strings) as follows: for $(x, \rho_x), (y, \rho_y), x, y \in V^+$, we write:

$$(x, \rho_x) \Longrightarrow_G (y, \rho_y)$$

iff $x = x_1x_2x_3$, $y = x_1ux_2vx_3$, for $x_1, x_3 \in V^*$ and $\langle x_2, (u, v); \rho_{ux_2v} \rangle \in P$, such that $\rho_y|_{x_1x_2x_3} = \rho_x$, and $\rho_y|_{ux_2v} = \rho_{ux_2v}$. In words, the string x is enlarged with the context (u, v) and the structure of x is extended according to the dependencies imposed by ρ_{ux_2v}; due to the restriction $\rho_{ux_2v}|_{x_2} = \emptyset$, the dependencies in x are not modified when adjoining u, v. The elements of x_2 can be linked to elements of x_1, x_3, but the elements of u, v participate only in dependencies with elements of the selector string x_2.

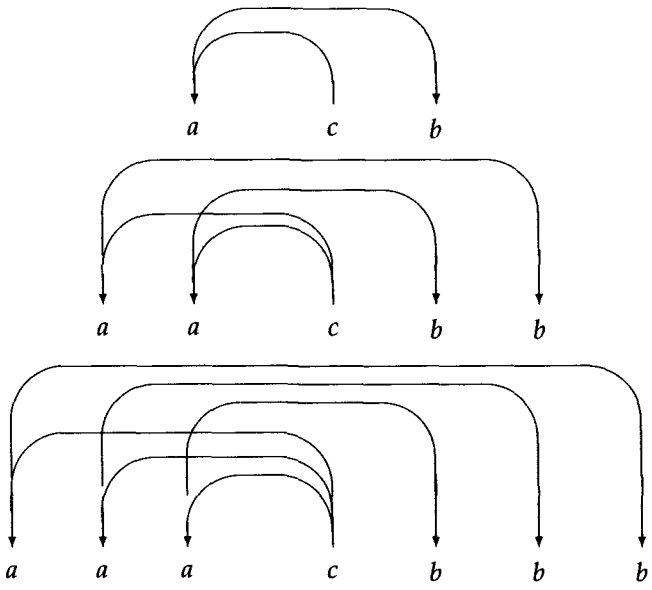

Figure 3
First three strings generated by G_1.

The string language generated by G is:

$$L(G) = \{w \in V^* \mid (x, \rho_x) \Longrightarrow_G^* (w, \rho_w), \text{ for some } (x, \rho_x) \in A\}.$$

The language of structured strings generated by a grammar G as above is:

$$SL(G) = \{(w, \rho_w) \mid (x, \rho_x) \Longrightarrow_G^* (w, \rho_w), \text{ for some } (x, \rho_x) \in A\}.$$

Let us examine two examples. For the grammar:

$$G_1 = (\{a, b, c\}, \{(acb, \{(2,1), (1,3)\})\}, \{\langle c, (a, b); \{(2,1), (1,3)\}\rangle\}),$$

we obtain:

$$\begin{aligned} L(G_1) &= \{a^n c b^n \mid n \geq 1\}, \\ SL(G_1) &= \{(a^n c b^n, \{(n+1, i), (i, 2n+2-i) \mid 1 \leq i \leq n\}) \mid n \geq 1\}. \end{aligned}$$

The first three strings generated by G_1 are represented in Figure 3.

The structured strings generated by G_1 are simple strings with center c; the structure graph is not planar if we preserve the order of elements of strings when writing them in a row as above.

For the grammar:

$$G_2 = (\{a, b, c\}, \{(acb, \{(2,1), (2,3)\})\}, \{\langle c, (a, b); \{(2,1), (2,3)\}\rangle\}),$$

we obtain:

$$\begin{aligned} &L(G_2) = \{a^n c b^n \mid n \geq 1\}, \\ &SL(G_2) = \{(a^n c b^n, \{(n+1, i), (n+1, 2n+2-i) \mid 1 \leq i \leq n\}) \mid n \geq 1\}. \end{aligned}$$

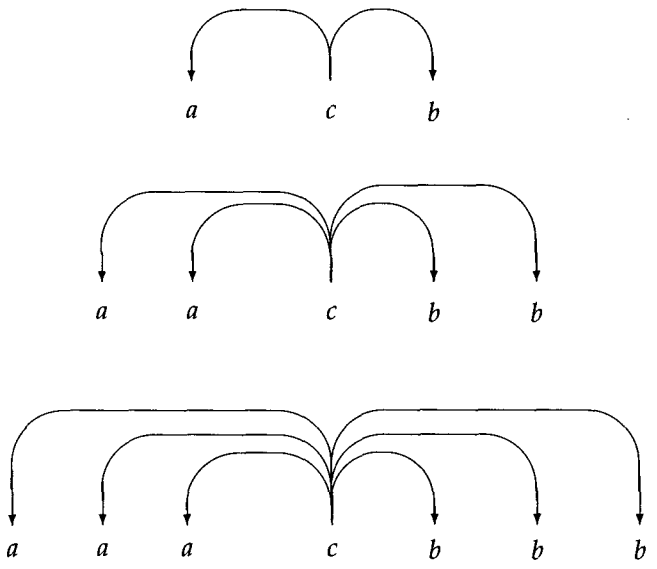

Figure 4
First three strings generated by G_2.

One sees that G_1 and G_2 are weakly equivalent, they generate the same string language, but they are not strongly equivalent, the structures of the same strings generated by G_1 and G_2 are not identical. For instance, the first three strings generated by G_2 are as shown in Figure 4.

We again obtain simple strings with center c, but the graphs describing the structure of the strings in the representation above are planar.

These examples suggest that classes of structured contextual grammars should be considered on the basis of a classification of the graphs associated to their generated strings. Thus, a grammar $G = (V, A, P)$ is said to be connected, simple, or planar if the graphs associated to the relation describing the structure of the strings generated by G is connected, a tree, or planar (when the string is written on a horizontal line, as before), respectively. Moreover, we can use these properties as restrictions on the grammar, selecting from the languages $L(G)$, and $SL(G)$ only the (structured) strings whose structure graph has the properties mentioned above. Of course, many other variants can be defined; for instance, we can consider the various types of projectivity (progressive, regressive, strong, and so on), as investigated in Chapter VI of Marcus (1967).

The above definitions of bracketed and structured contextual grammars can be extended in an obvious way to grammars with maximal use of selectors. Some results in this area can be found in Martín-Vide and Păun (1998), but a lot of questions remain to be clarified. The main problem is to find the most useful and natural type of structured contextual grammars for describing the structure of natural language syntactic constructions.

6. Representations of Recursively Enumerable Languages

Completing the study on (weak) generative power of contextual grammars from Section 3, we now give a result proving the eccentric position of families of contextual languages with regard to the Chomsky hierarchy.

Ehrenfeucht, Păun, and Rozenberg (1997) prove that each recursively enumerable language L can be written in the form $L = h_1(h_2^{-1}(L'))$, for $L' \in CL_{in}(FIN)$ and h_1, h_2 two morphisms. In view of the inclusion $CL_{in}(FIN) \subseteq CL_{Ml}(FIN)$, this result is valid also for $L' \in CL_{Ml}(FIN)$ and $L' \in CL_{Ml}(REG)$. Because $CL_{in}(FIN) - CL_{Mg}(REG) \neq \emptyset$, the result of Ehrenfeucht, Păun, and Rozenberg (1997) is not directly valid for $L' \in CL_{Mg}(REG)$ or $L' \in CL_{Mg}(FIN)$. However, the result of Ehrenfeucht, Păun, and Rozenberg (1997) can also be extended to these cases. Because we shall use it below, we outline here the construction of Ehrenfeucht, Păun, and Rozenberg (1997).

Take $L \subseteq T^*, L \in RE$, and a type-0 Chomsky grammar $G_0 = (N, T, S, P)$ for L. Consider the new symbols $[,], \vdash$, and construct the contextual grammar G with the alphabet:

$$V = N \cup T \cup \{[,], \vdash\},$$

the starting string S, and the following productions:

1. $(\{u\}, \{([,]v)\})$, for $u \to v \in P$,
2. $(\{\alpha[u]\}, \{(\vdash, \alpha)\})$, for $\alpha \in N \cup T$, $u \to v \in P$,
3. $(\{\alpha \vdash \beta\}, \{(\vdash, \alpha)\})$, for $\alpha, \beta \in N \cup T$.

Consider also the set:

$$R = \{[u] \mid u \to v \in P\} \cup \{\vdash \alpha \mid \alpha \in N \cup T\}.$$

For each string $w \in R$, consider a new symbol, b_w; denote by $D = \{b_w \mid w \in R\}$ their set. We define the coding $h_1 : (D \cup T)^* \longrightarrow T^*$ by:

$$h_1(b_w) = \lambda, w \in R, \quad h_1(a) = a, a \in T,$$

as well as the morphism $h_2 : (D \cup T)^* \longrightarrow V^*$ by:

$$h_2(b_w) = w, w \in R, \quad h_2(a) = a, a \in T.$$

One obtains the equality $L = h_1(h_2^{-1}(L_{in}(G)))$.

The idea is the following: h_2^{-1} is defined on $(R \cup T)^*$, hence all derivations in G that do not produce words in $(R \cup T)^*$ will be "lost"; thus, h_2^{-1} acts like an intersection with the regular language $(R \cup T)^*$, plus the conversion of each string $w \in R$ into the associated symbol b_w. In order to obtain a string in $(R \cup T)^*$, a derivation in G must follow a derivation in G_0, in the sense that each rule $u \to v \in P$ is simulated by a production of type 1, $(\{u\}, (\{([,]v)\})$, thus replacing u with $[u]v$. The parentheses [,] "kill" the word u. Productions of types 2 and 3 allow "living" symbols α to go to the right, across "dead" symbols; also $\vdash$ is a "killer," specifically, of the symbol placed immediately to its right. The requirement that a word in $(R \cup T)^*$ must eventually be reached imposes the use of productions of type 1 for living u only, and the use of productions of types 2 and 3 for living α and dead u and β, respectively. After using these rules, u is dead, v is living (type 1), the first α is dead, the new one is living

(types 2 and 3). This ensures that the obtained word contains only dead symbols and killers in words of R and living terminal symbols. By means of h_1, h_2^{-1}, only the living terminals remain.

Note that the construction of Ehrenfeucht, Păun, and Rozenberg (1997) does not work directly for the global maximal case: the grammar G_0 can contain, for instance, two rules $u \to v$, $u' \to v'$ with u a proper subword of u'; the first rule cannot be simulated in G when u' is present, because we are forced to use the maximal selector, u' in this case. However, the proof can be modified to cover the case of global maximal selectors as well.

Theorem 3
Every language $L \in RE$ can be written in the form $L = h_1(h_2^{-1}(L'))$, where $L' \in CL_{Mg}(FIN)$ and h_1, h_2 are two morphisms.

Proof
Take $L \subseteq T^*, L \in RE$, and take a type-0 Chomsky grammar $G_0 = (N, T, S, P)$ for L in the Kuroda normal form, that is containing rules of the forms:

1. $X \to YZ$, $X \to a$, $X \to \lambda$, for $X, Y, Z \in N, a \in T$,
2. $XY \to ZU$, for $X, Y, Z, U \in N$.

(Context-free rules and non-context-free rules, respectively, all of them with left-hand and right-hand members of length at most two.)

Take a new symbol, $c \notin T$, and construct the Chomsky grammar $G_1 = (N \cup \{S'\}, T \cup \{c\}, S', P')$, where:

$$\begin{aligned} P' = & \{S' \to Sc\} \cup \\ & \cup \{XY \to ZU \mid XY \to ZU \in P,\ X, Y, Z, U \in N\} \cup \\ & \cup \{X\alpha \to x\alpha \mid X \to x \text{ is a rule of type 1 in } P \text{ and } \alpha \in N \cup T \cup \{c\}\}. \end{aligned}$$

It is easy to see that $L(G_1) = L(G_0)\{c\}$.

Now start the procedure of Ehrenfeucht, Păun, and Rozenberg (1997) from the grammar G_1, constructing the contextual grammar G exactly as in Ehrenfeucht, Păun, and Rozenberg (1997) and extending the morphisms h_1, h_2 by:

$$h_1(c) = \lambda,$$

$$h_2(c) = c.$$

Because all rules in P', excepting $S' \to Sc$, which is used only once, have left-hand members of the same length, the maximal restriction of using the associated selectors has no effect. Concerning selectors u and $\alpha[u]$, appearing in productions of type 1 and type 2, respectively, the first selector for u is already dead (as is the case of the second selector), so its use is illegal; it leads to nonsuccessful derivations. The symbol c is preserved by h_2^{-1} and it is erased by h_1. Consequently, with the details of the proof in Ehrenfeucht, Păun, and Rozenberg (1997), we obtain $L = h_1(h_2^{-1}(L_{Mg}(G)))$, which completes the proof.

Corollary 2
Every $L \in RE$ can be written in the form $L = g(L')$, where $L' \in CL_{Mg}(FIN)$ and g is a generalized sequential transducer.

Proof
A sequential transducer can simulate at the same time both the work of h_1 and of h_2^{-1}. □

These results have a rather interesting consequence.

Theorem 4
Every family F of languages such that $LIN \subseteq F \subset RE$ that is closed under direct and inverse morphisms is incomparable with each family $CL_\alpha(F')$, for $\alpha \in \{Ml, Mg\}$ and $F' \in \{FIN, REG\}$.

Proof
Consider a family F with the properties mentioned above. Because $LIN \subseteq F$ and $LIN - CL_\alpha(REG) \neq \emptyset$ for both $\alpha \in \{Ml, Mg\}$, we get $F - CL_\alpha(F') \neq \emptyset$ for α, F' as above. Let us now prove that also the assertion $CL_\alpha(F') - F \neq \emptyset$ holds, for α and F' as above. Assume the contrary, that is, $CL_\alpha(F') \subseteq F$. The closure of $CL_\alpha(F')$ under direct and inverse morphisms should be included in the closure of F under direct and inverse morphisms. From Theorem 3 we know that the closure of $CL_\alpha(F')$ under direct and inverse morphisms is equal to RE. From the closure properties of F, the closure of F under direct and inverse morphisms is equal to F. This implies $RE \subseteq F$, contradicting the strictness of the inclusion $F \subset RE$ from the theorem statement. □

Important families in formal language theory that fulfill the conditions in Theorem 4 are: (1) languages generated by programmed grammars without appearance checking but possibly using λ-rules introduced by Rosenkrantz (1969) (they are equivalent to many other grammars with context-free core rules applied in a regulated manner: see Dassow and Păun [1989]; (2) indexed languages (Aho 1968); (3) ET0L languages (generated by extended tabled interactionless Lindenmayer systems; ET0L is the largest family in this area—see Rozenberg and Salomaa [1980]; and (4) other subfamilies of ET0L (for instance, E0L). Each of the families $CL_\alpha(FIN), \alpha \in \{in, Ml, Mg\}$, contains (context-sensitive) languages outside these families. Therefore, the families $CL_\alpha(FIN)$ occupy a quite eccentric position in the Chomsky hierarchy (Figure 5).

7. Summary and Final Remarks

In this paper, we have continued the investigation of contextual grammars with (global or local) maximal use of selectors, recently introduced by Martín-Vide et al. (1995). We have mainly borne in mind issues concerning the adequacy of these grammars as an alternative model (with respect to Chomsky grammars) for the syntax of natural languages, because "the arguments against the adequacy of phrase structure grammar (as defined by Chomsky) are absolutely incontrovertible (although they also apply to full context-sensitive grammars and to unrestricted grammars), that is, the constructions of natural languages cannot be described in an adequate way using the descriptive mechanisms of such grammars.... Bizarre though it may sound, ... Bloomfield's theory of constructions is probably the best point of departure for future work on the subject" (Manaster Ramer 1994, 20). We need to keep in mind, as Manaster Ramer (1994) points out, that "the kinds of mathematical models we are used to are, of course, largely derived from Chomsky's early work on phrase structure, and this in turn represents ... the formalization of a terribly diminished, impoverished, and even caricatured idea of immediate constituent analysis, created by Leonard Bloomfield" (p. 22).

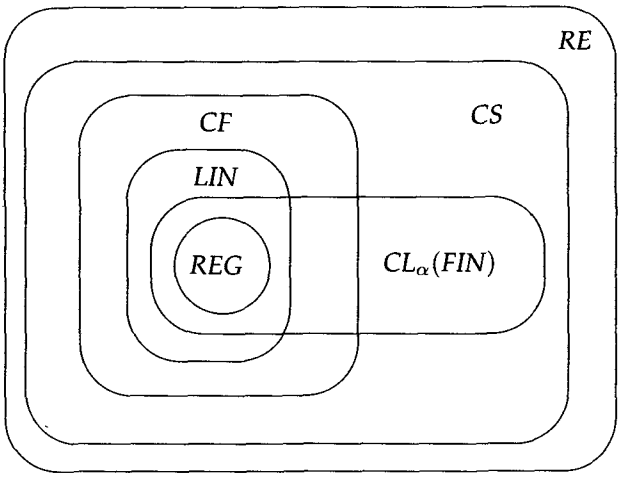

Figure 5
Position of the families $CL_\alpha(FIN)$ in the Chomsky hierarchy.

Our essential arguments have been the following:

1. The families of contextual languages are incomparable with some basic families in the Chomsky hierarchy (with *LIN* and *CF*) or in refinements of this hierarchy (programmed languages, indexed languages, languages generated by various classes of Lindenmayer systems). Some pieces of evidence indicate that perhaps natural languages occupy a similar incommensurate position with regard to Chomsky's classification.
2. Contextual grammars with global maximal use of selectors cannot generate all languages based on center-embedded constructions, as Chomsky linear grammars (and TAGs) do. Such constructions seem not to be very frequent in natural languages.
3. Contextual grammars with (global or local) maximal use of selectors can generate, in a very easy way, the three basic non-context-free constructions in natural languages: reduplication, crossed dependencies, multiple agreements.
4. Contextual grammars are sensitive to using markers, languages of the form $\{wcw \mid w \in \{a,b\}^*\}$ and $\{wc\ mi(w) \mid w \in \{a,b\}^*\}$ are handled more easily (i.e., by classes of grammars with simpler features) than $\{ww \mid w \in \{a,b\}^*\}$ and $\{w\ mi(w) \mid w \in \{a,b\}^*\}$. This again corresponds to our intuition, but it does not fit the Chomsky hierarchy.
5. By definition, contextual grammars are "fully" lexicalized (they use only terminal symbols), and their languages have the bounded growth property, which is specific to natural languages (and one of the main

ideas behind the notion of mild context-sensitivity, see Joshi [1985]): each word generated by a contextual grammar—excepting the axioms—is obtained by adjoining a context from a finite set.

6. If we intend our model to convey some cognitive meaning, we must say that the simple operation of adjoining might be closer than rewriting to the way our brain may work when building a sentence. It is hard to imagine our brain using auxiliary intermediate sentences of a nonterminal type. Instead, it looks more *natural*, in the proper sense of the word, to start with a collection of well-formed sentences, maybe acquired from experience, and to produce new well-formed ones by adding further words, in pairs that can observe dependencies and agreements, and in accordance with specified selectors, which can ensure the preservation of grammaticality. Of course, this is only a speculation, but it also fits with the general idea of "natural computation": for example, nature seems not to use the rewriting operation in the area of genetics, where recombination (crossing over) of chromosomes is the basic evolutionary operation (together with nondeterministic insertion and deletion operations, which, again, are not rewriting) and where no "nonterminal symbol" is used. Further discussion of this topic can be found in Martín-Vide (1997).

7. A structure for the words generated by a contextual grammar can be introduced in various ways. By parenthesizing the contexts, we get a tree. Considering dependence relations on symbols appearing in axioms, contexts, and selectors, we can obtain structured strings of a type well investigated in descriptive linguistics and very similar to the phrase-linkage structures produced by a link grammar.

A number of the previous points need further investigation. There are also several topics that are important from a linguistic point of view and that are still poorly investigated for contextual grammars. The main one concerns the parsing algorithms and their complexity. Polynomial parsing algorithms were found for a few variants of contextual grammars, which is encouraging, but the problem is still open for the variants discussed in this paper.

The main aim of this paper was to call the reader's attention to contextual grammars, to prove that they deserve further research efforts, especially in terms of their linguistic adequacy and relevance. It is our (optimistic) belief that such efforts will be rewarded.

Appendix: Proofs of Some Assertions Represented in Table 1

Proof

Assume that $M_1 \in CL_\alpha(FIN)$, $\alpha \in \{Ml, Mg\}$, take a grammar G for M_1, and consider a string of the form:

$$z_i = aba^2b^2 \ldots a^ib^icaba^2b^2 \ldots a^ib^i,$$

for a large-enough integer i. In order to produce such a string, we need a derivation:

$$w = w_1w_2w_3 \Longrightarrow_\alpha w_1uw_2vw_3 = z_i.$$

It is obvious that $|w_2|$ depends on i, and so cannot be bounded; therefore G cannot have finite selectors only. □

Proof
Assume that $M_1' \in CL_{in}(F)$, for $F \in \{FIN, REG\}$. Take $G = (\{a,b\}, A, (S_1, C_1), \ldots, (S_n, C_n))$ such that $L_{in}(G) = M_1'$. There is at least one context (u,v) in G with $uv \neq \lambda$; all the strings in M_1' are of even length, so $|uv|$ must be even. Take x in the selector of (u,v) and consider the strings xa^ixa^i, $i \geq |uv|$. Then $uxva^ixa^i \in L_{in}(G)$, so $uxva^ixa^i = yy$, for $y \in \{a,b\}^*$. This implies:

$$uxva^{i-|uv|/2} = a^{|uv|/2}xa^i,$$

that is, $uxv = za^j, j \geq 1$. In the same way, starting from xb^ixb^i we get that $uxv = z'b^k, k \geq 1$, a contradiction. As in point 1, we obtain that $M_1' \notin CL_\alpha(FIN), \alpha \in \{Ml, Mg\}$. □

Proof
Assume that $M_1' = L_{Ml}(G)$, for any $G = (\{a,b\}, A, (S_1, C_1), \ldots, (S_n, C_n))$ with regular selectors. Let us denote:

$$\begin{aligned} \varphi(x) &= \{(u,v) \mid (u,v) \in C_i, x \in S_i, 1 \leq i \leq n\}, \ x \in \{a,b\}^*, \\ \varphi^{-1}((u,v)) &= \{x \in \{a,b\}^* \mid (u,v) \in \varphi(x)\}, \ (u,v) \in C_i, 1 \leq i \leq n. \end{aligned}$$

All strings $a^mba^mb, m \geq 1$, are in M_1'. Take such a string with arbitrarily large m. If there is a derivation step $a^q \Longrightarrow_{Ml} a^mba^mb$, then there is a context $(u,v) = (a^{i_1}ba^{i_2}, a^{i_3}b) \in \varphi(a^p)$, for $p \leq q$. As $m = i_2 + p + i_3$, it follows that p is arbitrarily large. The set $\varphi^{-1}((u,v))$ is regular (it is the union of a finite number of regular sets), so it contains an infinite number of strings of the form a^s (we apply a pumping lemma to a^p in $\varphi^{-1}((u,v))$). Therefore, (u,v) must be used for a maximal selector of the form a^t. In this way, a string $a^{j_1}ba^{j_2}ba^{j_3}$ can be produced, with bounded j_1, j_3 and arbitrarily large j_2. Such a string is not in M_1', a contradiction. Therefore, in the derivation of a^mba^mb there exists an arbitrary number of derivation steps of the form:

$$a^sba^sb \Longrightarrow_{Ml} a^{s+k}ba^{s+k}b,$$

with $k \geq 1$ and $a^pba^q \in \varphi^{-1}((a^k, a^k))$. Consider now a string:

$$w = a^{i_1}ba^{i_2}ba^{i_1}ba^{i_2}b,$$

with arbitrarily large i_1, i_2. Each such string is in M_1'. If a^pba^q above or any other selector of the form $a^rba^{r'}$ from $\varphi^{-1}((a^k, a^k))$ is maximal in w, then we shall produce a string which is not in M_1'. On the other hand, a^pba^q is a subword of w, so the selectors included in $\varphi^{-1}((a^k, a^k))$ must contain a string that is a strict superword of a^pba^q, in order to prevent the generation of a parasitic word. Such a superword can only be of the forms $a^{j_1}ba^{i_2}ba^{j_2}$ or $a^{j_1}ba^{i_1}ba^{j_2}$. In both cases, the middle subword, $ba^{i_2}b$ or $ba^{i_1}b$, respectively, is arbitrarily long. As elements of a regular language, such strings have pumping properties. Let us consider the case of $ba^{i_2}b$ (the second one is similar). This means that all the strings of the form:

$$z = a^{j_1}ba^{i_2+rh}ba^{j_2},$$

for $r \geq 1$ and all $h \geq 0$, are in $\varphi^{-1}((u,v))$. Take such a string z with h being large enough to have:

$$i_2 + rh > j_1 + j_2.$$

Consider the string:

$$w = a^{i_2+rh-j_2}ba^{i_2+rh}ba^{j_2}.$$

Because:

$$i_2 + rh = (i_2 + rh - j_2) + j_2,$$

we have $w \in M_1'$. Because:

$$i_2 + rh - j_2 > j_1,$$

the context (a^k, a^k) is applicable to w. That is:

$$w \Longrightarrow_{Ml} a^{i_2+rh-j_2+k} b a^{i_2+rh} b a^{j_2+k}.$$

The string obtained is not in M_1', a contradiction. In conclusion, M_1' cannot be in $CL_{Ml}(REG)$. The previous argument does not hold for the global maximal derivation, so the relation $M_1' \in CL_{Mg}(REG)$ remains open. □

Proof
For the grammar $G = (\{a, b, c\}, \{c\}, (\{c\}, \{(a, a), (b, b)\}))$, we have $L_\alpha(G) = M_4$ for all α. □

Proof
The fact that $M_4' \notin CL_{in}(REG)$ is already proved in Păun (1982). As for M_1, one can easily prove that $M_4' \notin CL_\alpha(FIN), \alpha \in \{Ml, Mg\}$. On the other hand, for the grammar

$$G = (\{a, b\}, \{\lambda\}, (\{a, b\}^*, \{(a, a), (b, b)\})),$$

we have $L_{Ml}(G) = L_{Mg}(G) = M_4'$. Hence $M_4' \in CL_\alpha(REG), \alpha \in \{Ml, Mg\}$. □

Acknowledgments
The authors are much indebted to three anonymous reviewers, who very carefully read a previous draft of the manuscript, and who proposed a number of modifications that have improved both the readability and the content of this paper.

References

Aho, Alfred V. 1968. Indexed grammars. An Extension of context-free grammars. *Journal of the ACM*, 15:647–671.

Bar-Hillel, Yehoshua and Eliyahu Shamir. 1964. Finite state languages: Formal representations and adequacy problems. In Yehoshua Bar-Hillel, editor, *Language and Information*. Addison-Wesley, Reading, MA, pages 87–98.

Bălănescu, Tudor and Marian Gheorghe. 1987. Program tracing and languages of action. *Revue roumaine de linguistique—Cahiers de linguistique théorique et appliquée*, 32:167–170.

Chomsky, Noam. 1964. On the Notion "Rule of Grammar". In Janet A. Fodor and Jerrold J. Katz, editors, *The Structure of Language: Readings in the Philosophy of Language*. Prentice Hall, Englewood Cliffs, N.J., pages 50–118.

Culy, Christopher. 1985. The complexity of the vocabulary of Bambara. *Linguistics and Philosophy*, 8:345–351.

Dassow, Jürgen and Gheorghe Păun. 1989. *Regulated Rewriting in Formal Language Theory*. Springer, Berlin.

Ehrenfeucht, Andrzej, Gheorghe Păun, and Grzegorz Rozenberg. 1997. On representing recursively enumerable languages by internal contextual languages. *Theoretical Computer Science*. To appear.

Floyd, R. W. 1962. On the non-existence of a phrase-structure grammar for Algol-60. *Communications of the ACM*, 5:483–484.

Gazdar, Gerald and Geoffrey K. Pullum. 1985. Computationally relevant properties of natural languages and their grammars. *New Generation Computing*, 3:273–306.

Grinberg, Dennis, John Lafferty, and Daniel Sleator. 1995. A robust parsing algorithm for link grammars. In *Proceedings of the Fourth International Workshop on Parsing Technologies*, pages 111–125, Prague/Karlovy Vary.

Harrison, Michael. 1978. *Introduction to Formal Language Theory*. Addison-Wesley, Reading, MA.

Hockett, Charles. 1970. *The State of the Art*. Mouton, The Hague.

Ilie, Lucian. 1997a. On computational complexity of contextual languages. *Theoretical Computer Science*. To appear.

Ilie, Lucian. 1997b. The computational complexity of Marcus contextual languages. Submitted.

Jančar, Petr, František Mráz, Martin Plátek, Martin Procházka, and Jörg Vogel. 1996. Restarting automata, Marcus grammars and context-free languages. In Jürgen Dassow, Grzegorz Rozenberg, and Arto Salomaa, editors, *Developments in Language Theory II*. World Scientific, Singapore, pages 102–111.

Joshi, Aravind K. 1985. How much context-sensitivity is required to provide structural descriptions: Tree adjoining grammars. In David Dowty, Lauri Karttunen, and Arnold Zwicky, editors, *Natural Language Processing: Psycholinguistic, Computational, and Theoretical Perspectives*. Cambridge University Press, New York, pages 206–250.

Joshi, Aravind K. 1987. An introduction to tree adjoining grammars. In Alexis Manaster Ramer, editor, *Mathematics of Language*. John Benjamins, Amsterdam, pages 87–114.

Joshi, Aravind K., Leon S. Levy, and M. Takahashi. 1975. Tree adjunct grammars. *Journal of Computer and Systems Sciences*, 10:136–163.

Manaster Ramer, Alexis. 1993. Capacity, complexity, construction. *Annals of Mathematics and Artificial Intelligence*, 8(1-2): Mathematics of language, pages 1–16.

Manaster Ramer, Alexis. 1994. Uses and misuses of mathematics in linguistics. X Congreso de Lenguajes Naturales y Lenguajes Formales, Sevilla.

Marcus, Solomon. 1967. *Algebraic Linguistics. Analytical Models*. Academic Press, New York.

Marcus, Solomon. 1969. Contextual grammars. *Revue roumaine des mathématiques pures et appliquées*, 14:1525–1534.

Marcus, Solomon, editor. 1978. *La sémiotique formelle du folklore*. Klincksieck, Paris/Publishing House of the Romanian Academy, Bucharest.

Marcus, Solomon. 1979. Linguistics for programming languages. *Revue roumaine de linguistique—Cahiers de linguistique théorique et appliquée*, 16(1):29–38.

Marcus, Solomon, editor. 1981–83. *Contextual Ambiguities in Natural and Artificial Languages*. Communication and Cognition Monographs, 2 volumes. Ghent.

Marcus, Solomon. 1997. Contextual grammars and natural languages. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages*, volume 2, pages 215–235.

Martín-Vide, Carlos. 1997. Natural computation for natural language. *Fundamenta Informaticae*, 31.2:117–124.

Martín-Vide, Carlos, Alexandru Mateescu, Joan Miquel-Vergés, and Gheorghe Păun. 1995. Internal contextual grammars: Minimal, maximal and scattered use of selectors. In M. Koppel and Eliyahu Shamir, editors, *Proceedings of The Fourth Bar-Ilan Symposium on Foundations of Artificial Intelligence (BISFAI'95)*, pages 132–142. Ramat Gan/Jerusalem. Also in M. Koppel and Eliyahu Shamir, editors, *Proceedings of the fourth Bar-Ilan Symposium on Foundations of Artificial Intelligence. Focusing on Natural Languages and Artificial Intelligence: Philosophical and Computational Aspects*. AAAI Press, Menlo Park, CA, 1997, pages 159–168.

Martín-Vide, Carlos and Gheorghe Păun. 1998. Structured contextual grammars. *Grammars*: To appear.

Miquel-Vergés, Joan. 1997. *Models Algebraics Analítics del Llenguatge. Un Exemple: les Gramàtiques Contextuals; Formalització i Estudi de les Seves Aplicacions*. Ph.D. dissertation, Rovira i Virgili University, Tarragona.

Olivetti, Convegno. 1970. *Linguaggi nella Società e nella Tecnica*. Edizzioni di Comunità, Milano.

Partee, Barbara H., Alice ter Meulen, and Robert E. Wall. 1990. *Mathematical Methods in Linguistics*. Kluwer, Dordrecht.

Păun, Gheorghe. 1976. Languages associated to a dramatic work. *Revue roumaine de linguistique: Cahiers de linguistique théorique et appliquée*, 13:605–611.

Păun, Gheorghe. 1979. A formal linguistic model of action systems. *Ars Semeiotica*, 2:33–47.

Păun, Gheorghe. 1982. *Contextual Grammars*. The Publishing House of the Romanian Academy, Bucharest, in Romanian.

Păun, Gheorghe. 1985. On some open problems about Marcus contextual grammars. *International Journal of Computer Mathematics*, 17:9–23.

Păun, Gheorghe. 1994. Marcus contextual grammars. After 25 years. *Bulletin of the EATCS*, 52:263–273.

Păun, Gheorghe. 1997. *Marcus Contextual Grammars*. Kluwer, Boston, Dordrecht.

Păun, Gheorghe and Xuan My Nguyen. 1980. On the inner contextual grammars. *Revue roumaine des mathématiques pures et*

appliquées, 25:641–651.
Păun, Gheorghe, Grzegorz Rozenberg, and Arto Salomaa. 1994. Marcus contextual grammars: Modularity and leftmost derivation. In Gheorghe Păun, editor, *Mathematical Aspects of Natural and Formal Languages*. World Scientific, Singapore, pages 375–392.
Pullum, Geoffrey K. 1985. On two recent attempts to show that English is not a CFL. *Computational Linguistics*, 10:182–186.
Pullum, Geoffrey K. 1986. Footloose and context-free. *Natural Language and Linguistic Theory*, 4(3):409–414.
Pullum, Geoffrey K. 1987. Nobody goes around at LSA meetings offering odds. *Natural Language and Linguistic Theory*, 5(2):303–309.
Pullum, Geoffrey K. and Gerald Gazdar. 1982. Natural languages and context-free languages. *Linguistics and Philosophy*, 4:471–504.
Radzinski, Daniel. 1990. Unbounded syntactic copying in Mandarin Chinese. *Linguistics and Philosophy*, 13:113–127.
Rosenkrantz, Daniel J. 1969. Programmed grammars and classes of formal languages. *Journal of the ACM*, 16:107–131.
Rounds, William C., Alexis Manaster Ramer, and Joyce Friedman. 1987. Finding natural languages a home in formal language theory. In Alexis Manaster Ramer, editor, *Mathematics of Language*. John Benjamins, Amsterdam, pages 375–392.
Rozenberg, Grzegorz and Arto Salomaa. 1980. *The Mathematical Theory of L Systems*. Academic Press, New York.
Rozenberg, Grzegorz and Arto Salomaa, editors. 1997. *Handbook of Formal Languages*, 3 volumes. Springer, Berlin.
Salomaa, Arto. 1973. *Formal Languages*. Academic Press, New York.
Savitch, Walter J. 1991. Infinity is in the eyes of the beholder. In C. Georgopoulos and R. Ishihara, editors, *Interdisciplinary Approaches to Language: Essays in Honor of S.-Y. Kuroda*. Kluwer, Dordrecht, pages 487–500.
Savitch, Walter J. 1993. Why it might pay to assume that languages are infinite. *Annals of Mathematics and Artificial Intelligence*, 8(1-2): Mathematics of language, pages 17–25.
Savitch, Walter J., Emmon Bach, William Marsh, and Gila Safran-Naveh, editors. 1987. *The Formal Complexity of Natural Language*. Kluwer, Dordrecht.
Shieber, Stuart M. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8:333–343.
Sleator, Daniel and D. Temperley. 1991. Parsing English with a link grammar. Technical Report CMU-CS-91-196, School of Computer Science, Carnegie Mellon University, Pittsburgh.